



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Σχεδιασμός και κατασκευή ολοκληρωμένου
συστήματος καταγραφής και παρακολούθησης
περιβαλλοντολογικών συνθηκών σε θερμοκήπιο**

**Design and implementation of an environmental
monitoring and auditing system for a greenhouse**

Γιαννόπουλος Νικόλαος

Επιβλέπων Καθηγητής: Δρ. Μηνάς Δασυγένης

Κοζάνη, Φεβρουάριος 2013

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	2
Κατάλογος πινάκων.....	4
Κατάλογος εικόνων	5
Πρόλογος.....	8
Περίληψη.....	9
Λέξεις κλειδιά.....	10
Abstract	11
Keywords.....	12
Ευχαριστίες.....	13
Διάρθρωση κειμένου	14
Κεφάλαιο 1: Θεωρητικό υπόβαθρο	15
1.1 Προγραμματισμός διαδικτύου	15
1.1.1 HTML.....	15
1.1.2 CSS.....	17
1.1.3 Javascript	17
1.1.4 PHP.....	18
1.1.5 GET και POST	19
1.1.6 AJAX.....	19
1.1.7 MySQL	20
1.2 Αρχιτεκτονική arduino	20
1.2.1 Γενικές πληροφορίες σχετικά με το Arduino	20
1.2.2 Μοντέλα μικροελεγκτών Arduino	20
1.2.3 Arduino Shields	21
1.2.4 Τεχνικά χαρακτηριστικά Arduino UNO.....	22
1.2.5 Τροφοδοσία Arduino UNO	22
1.2.6 Μνήμη Arduino UNO	23
1.2.7 Ακροδέκτες Arduino UNO	23
1.3 Πλατφόρμες ανάπτυξης λογισμικού.....	24
1.3.1 Notepad ++	24
1.3.2 Xampp	24
1.3.3 Arduino IDE	25
Κεφάλαιο 2: Υλικό μέρος συστήματος	28
2.1 Υλικό υλοποίησης επικοινωνίας με το διαδίκτυο	30
2.2 Περιβαλλοντικοί αναλογικοί αισθητήρες	31
2.2.1 Αισθητήρας Θερμοκρασίας LM 335.....	31
2.2.2 Αισθητήρας Υγρασίας 808h5V5	32
2.2.3 Αισθητήρας έντασης φωτός Photocell	33
2.2.4 Αισθητήρας υγρασίας χρώματος.....	34
2.3 Συσκευές δράσης ψηφιακά ρυθμιζόμενες	35
2.3.1 N-MOSFET	36
2.3.2 Ανάλυση μεθόδου συνδεσμολογίας των συσκευών δράσης	37
2.3.3 Χαρακτηριστικά των συσκευών δράσης και η συνδεσμολογία τους	38

2.4	Μετατροπή κυκλώματος σε ηλεκτρονική πλακέτα	40
2.5	Κατασκευή Μακέτας Θερμοκηπίου	43
2.6	Χαρακτηριστικά συσκευής.....	45
Κεφάλαιο 3: Λογισμικό μέρος συστήματος		46
3.1	Περιγραφή απαιτήσεων συστήματος.....	47
3.2	Περιγραφή βάσης δεδομένων.....	48
3.3	Λειτουργίες ιστοχώρου για διαχειριστή	57
3.3.1	Είσοδος διαχειριστή στον ιστοχώρο.....	57
3.3.2	Έξοδος διαχειριστή από τον ιστοχώρο.....	58
3.3.3	Στατιστικά ιστοχώρου	59
3.3.4	Χρήστες ιστοχώρου	59
3.3.5	Αναφορά σφαλμάτων	61
3.3.6	Μηνύματα χρηστών.....	61
3.4	Λειτουργίες ιστοχώρου για απλό χρήστη	62
3.4.1	Λειτουργία Εισόδου - Εξόδου	63
3.4.2	Λειτουργία ενημέρωσης συνθηκών σε πραγματικό χρόνο.....	63
3.4.3	Λειτουργία ενεργοποίησης manual σεναρίου.....	65
3.4.4	Λειτουργία ελέγχου σεναρίων ημέρας	66
3.4.5	Λειτουργία δημιουργίας σεναρίου.....	67
3.4.6	Λειτουργία προβολής ημερολογίου γεγονότων.....	70
3.4.7	Λειτουργία προβολής ημερήσιας λίστας γεγονότων	71
3.4.8	Λειτουργία προσθήκης σεναρίου στον προγραμματισμό.....	72
3.4.9	Λειτουργία προβολής – επεξεργασίας σεναρίου	73
3.4.10	Λειτουργία διαγραμμάτων χρήσης.....	75
3.4.11	Λειτουργία προβολής και επεξεργασίας στοιχείων χρήστη	77
3.4.12	Λειτουργία αυτόματης επικοινωνίας συστήματος με χρήστη	78
3.4.13	Λειτουργία επικοινωνίας με το διαχειριστή	80
3.4.14	Λειτουργία προβολής σφαλμάτων.....	80
3.5	Προγραμματισμός αισθητήρων, συσκευών δράσης και Arduino	81
3.5.1	Βασικές συναρτήσεις, βιβλιοθήκες και εργαλεία.....	81
3.5.2	Προγραμματισμός αισθητήρων	84
3.5.3	Προγραμματισμός συσκευών δράσης	87
3.5.4	Επικοινωνία arduino και ιστοχώρου.....	87
Κεφάλαιο 4: Μελλοντικές επεκτάσεις συστήματος		93
4.1	Μελλοντικές επεκτάσεις σε θέματα τροφοδοσίας.....	94
4.1.1	Χρήση μπαταρίας	94
4.1.2	Χρήση ανανεώσιμων πηγών ενέργειας	94
4.2	Μελλοντικές επεκτάσεις σε θέματα επικοινωνίας.....	95
4.2.1	Χρήση Wifi.....	95
4.2.2	Χρήση GPRS.....	95
4.3	Μελλοντικές επεκτάσεις ως προς τη χρήση αισθητήρων	96
4.4	Μελλοντικές επεκτάσεις γενικής χρήσης συστήματος.....	96
Κεφάλαιο 5: Συμπεράσματα.....		98
Βιβλιογραφία.....		100

Κατάλογος πινάκων

Πίνακας 1: HTML ετικέτες	17
Πίνακας 2: Μοντέλα μικροελεγκτών Arduino	21
Πίνακας 3: Μοντέλα Arduino Shields.....	21
Πίνακας 4: Τεχνικά χαρακτηριστικά Arduino UNO	22
Πίνακας 5: Εργαλεία Ανάπτυξης Arduino IDE.....	25
Πίνακας 6: Περιγραφή λειτουργίας φωτεινών ενδείξεων του Ethernet Shield	31
Πίνακας 7: Χαρακτηριστικά συσκευών δράσης	38
Πίνακας 8: Κόστος κατασκευής συσκευής	45
Πίνακας 9: Κατανάλωση ενέργειας συσκευής	45
Πίνακας 10: Πίνακας devices	49
Πίνακας 11: Πίνακας page_statistics.....	50
Πίνακας 12: Πίνακας users.....	50
Πίνακας 13: Πίνακας data	52
Πίνακας 14: Πίνακας scenarios - new_scenarios	53
Πίνακας 15: Πίνακας schedule_scenarios	55
Πίνακας 16: Πίνακας charts	55
Πίνακας 17: Πίνακας messages.....	56
Πίνακας 18: Πίνακας troubleshoots	56
Πίνακας 19: Συμβολοσειρά σεναρίου	89

Κατάλογος εικόνων

Εικόνα 1: Κανόνας σύνταξης CSS	17
Εικόνα 2: Παράδειγμα κώδικα javascript.....	18
Εικόνα 3: Αρχιτεκτονική AJAX.....	19
Εικόνα 4: Arduino UNO.....	20
Εικόνα 5: Arduino Leonardo	20
Εικόνα 6: Arduino Mega 2560	20
Εικόνα 7: Arduino LilyPad.....	21
Εικόνα 8: Arduino Mega ADK	21
Εικόνα 9: Arduino Fio	21
Εικόνα 10: Arduino Pro.....	21
Εικόνα 11: Arduino BT	21
Εικόνα 12: Arduino Nano.....	21
Εικόνα 13: Arduino Wifi shield	21
Εικόνα 14: Arduino Ethernet shield	21
Εικόνα 15: Wireless SD shield.....	21
Εικόνα 16: Arduino GPRS shield.....	21
Εικόνα 17: Είσοδοι/Έξοδοι τροφοδοσίας Arduino UNO.....	22
Εικόνα 18: Ψηφιακά pins του Arduino UNO.....	23
Εικόνα 19: Αναλογικά pins του Arduino UNO.....	24
Εικόνα 20: Notepad++	24
Εικόνα 21: Verify.....	25
Εικόνα 22: Upload.....	25
Εικόνα 23: New	25
Εικόνα 24: Open.....	25
Εικόνα 25: Save.....	25
Εικόνα 26: Serial Monitor	25
Εικόνα 27: Απεικόνιση Arduino IDE.....	26
Εικόνα 28: Βασική αρχιτεκτονική Κώδικα Arduino.....	27
Εικόνα 29: Η έννοια του πράκτορα λογισμικού.....	29
Εικόνα 30: Κύκλωμα συστήματος	30
Εικόνα 31: Arduino UNO και Arduino Ethernet Shield	30
Εικόνα 32: Αισθητήρας θερμοκρασίας LM 335	31
Εικόνα 33: Συνδεσμολογία αισθητήρα θερμοκρασίας LM 335	32
Εικόνα 34: Αισθητήρας Υγρασίας 808h5V5.....	32
Εικόνα 35: Συνδεσμολογία αισθητήρα υγρασίας 808h5V5	33
Εικόνα 36: Αισθητήρας έντασης φωτός Photocell	33
Εικόνα 37: Συνδεσμολογία αισθητήρα έντασης φωτός Photocell.....	34
Εικόνα 38: Συνδεσμολογία αισθητήρα υγρασίας εδάφους	35
Εικόνα 39: Ανεμιστήρας	35
Εικόνα 40: UV Led.....	36
Εικόνα 41: Ηλεκτροβάννα	36
Εικόνα 42: N-MOSFET.....	37
Εικόνα 43: IRF 520	37
Εικόνα 44: Παράδειγμα συνδεσμολογία με N-MOSFET	38

Εικόνα 45: Συνδεσμολογία ηλεκτροβάνας.....	39
Εικόνα 46: Συνδεσμολογία ανεμιστήρα.....	39
Εικόνα 47: Συνδεσμολογία UV LED	40
Εικόνα 48: Φωτοευαίσθητη πλακέτα μονής όψης.....	40
Εικόνα 49: Αρχική μορφή κυκλώματος στο breadboard.....	41
Εικόνα 50: Κύκλωμα αισθητήρων	41
Εικόνα 51: Κύκλωμα συσκευών δράσης.....	42
Εικόνα 52: Αποτέλεσμα εκτύπωσης.....	42
Εικόνα 53: Διαδικασία αποχάλκωσης	43
Εικόνα 54: Τελικό αποτέλεσμα	43
Εικόνα 55: Μακέτα Θερμοκηπίου.....	44
Εικόνα 56: Μακέτα θερμοκηπίου εσωτερικό.....	44
Εικόνα 57: Βάση δεδομένων – phpmyadmin	49
Εικόνα 58: Κώδικας php για device password	49
Εικόνα 59: Είσοδος χρηστών	57
Εικόνα 60: Υπενθύμιση κωδικού	58
Εικόνα 61: Έξοδος χρηστών	58
Εικόνα 62: Στατιστικά ιστοχώρου.....	59
Εικόνα 63: Χρήστες επεξεργασία - προβολή	60
Εικόνα 64: Χρήστες - δημιουργία χρήστη	60
Εικόνα 65: Αναφορά ασφαμάτων.....	61
Εικόνα 66: Μηνύματα χρηστών	62
Εικόνα 67: Ιστοχώρος απλού χρήστη.....	63
Εικόνα 68: Ενημέρωση κατάστασης θερμοκηπίου	64
Εικόνα 69: Ενημέρωση κατάστασης θερμοκηπίου εκτός τομέα Ελέγχου	65
Εικόνα 70: Βήματα manual.....	65
Εικόνα 71: Εκκίνηση manual	66
Εικόνα 72: Προγραμματισμός απενεργοποίηση.....	67
Εικόνα 73: Περιορισμός ενεργοποίησης σεναρίων	67
Εικόνα 74: Προγραμματισμός - πληροφορίες σεναρίου	67
Εικόνα 75: Δημιουργία σεναρίου - Βήμα 1	68
Εικόνα 76: Έλεγχος δεδομένων.....	69
Εικόνα 77: Δημιουργία Σεναρίου - Βήμα 2.....	70
Εικόνα 78: Ημερολόγιο γεγονότων	70
Εικόνα 79: Λίστα προηγούμενης ημερομηνίας.....	71
Εικόνα 80: Λίστα τρέχουσας - επόμενη ημερομηνίας.....	72
Εικόνα 81: Προσθήκη σεναρίου.....	73
Εικόνα 82: Προβολή - επεξεργασία - διαγραφή σεναρίου	74
Εικόνα 83: Επεξεργασία αισθητήρων	74
Εικόνα 84: Διαγράμματα διαδικασία	76
Εικόνα 85: Διαγράμματα προβολή.....	76
Εικόνα 86: Προβολή επεξεργασία στοιχείων χρήστη	78
Εικόνα 87: Ρυθμίσεις συσκευής	78
Εικόνα 88: Email συστήματος.....	79
Εικόνα 89: Επικοινωνία με το διαχειριστή.....	80
Εικόνα 90: Προβολή ασφαμάτων χρήστη	81

Εικόνα 91: Κώδικας αισθητήρα φωτός.....	84
Εικόνα 92: Κώδικας αισθητήρα υγρασίας.....	85
Εικόνα 93: Κώδικας αισθητήρα θερμοκρασίας.....	85
Εικόνα 94: Κώδικας αισθητήρα εδάφους.....	86
Εικόνα 95: Κώδικας συσκευών δράση.....	87
Εικόνα 96: Επικοινωνία arduino και βάσης	88
Εικόνα 97: Κώδικας αποκωδικοποίησης.....	90
Εικόνα 98: Δεδομένα αποστολής arduino	91
Εικόνα 99: Λειτουργία κώδικα Arduino	92

Πρόλογος

Η διπλωματική εργασία που περιγράφεται στο παρόν κείμενο, ασχολείται με τη δημιουργία ενός ενσωματωμένου συστήματος το οποίο συνδέεται μέσω διαδικτύου με τον αντίστοιχο ιστοχώρο, με σκοπό την παρακολούθηση και την καταγραφή των περιβαλλοντικών συνθηκών ενός θερμοκηπίου, καθώς και τον έλεγχο συσκευών δράσης (ανεμιστήρα, ποτίσματος κτλ). Η υλοποίηση του συστήματος έγινε με βάση την αρχιτεκτονική Arduino. Για την καλύτερη αντίληψη των συνθηκών και των αναγκών ενός θερμοκηπίου, αλλά και βελτιστοποίηση συνολικά του συστήματος, κατασκευάστηκε μακέτα θερμοκηπίου, στην οποία εκπονήθηκε η διπλωματική εργασία και έγιναν οι διάφορες δοκιμές που χρειάστηκαν. Το σύστημα στο σύνολο του ονομάστηκε "Smart Greenhouse" ("Έξυπνο" θερμοκήπιο) λόγω των χαρακτηριστικών και ιδιοτήτων του, που το κατατάσσουν στις "έξυπνες" συσκευές.

Υπεύθυνος και εμπνευστής της παραπάνω εργασίας ήταν ο Δρ. Δασυγένης Μηνάς, ενώ η υλοποίηση ανατέθηκε στο φοιτητή Γιαννόπουλο Νικόλαο.

Περίληψη

Η τεχνολογική εξέλιξη των τελευταίων ετών, σε συνδυασμό με την ανάπτυξη του διαδικτύου και της δικτύωσης και αλληλεπίδρασης των διαφόρων συσκευών κι συστημάτων, έχει επηρεάσει όλους τους τομείς της οικονομικής και κοινωνικής ζωής. Στην Ελλάδα η γεωργία αποτελεί έναν παραδοσιακά ισχυρό κλάδο ο οποίος την τελευταία διετία αποτελεί πόλο έλξης για νέους επιχειρηματίες αλλά και για ερασιτέχνες για λόγους υγείας αλλά και οικονομίας. Μία πλέον αποδοτική λύση σε θέματα παραγωγής είναι η χρήση θερμοκηπίων. Όμως πέραν του αρχικού κόστους κατασκευής απαιτούν και δαπάνη προσωπικού χρόνου και ενέργειας για την συνεχή παρακολούθηση τους.

Το ολοκληρωμένο σύστημα της εργασίας δίνει λύσεις στον γεωργό σε θέματα παρακολούθησης, αυτοματισμού και ιχνηλασιμότητας καθώς από τον υπολογιστή του ή το κινητό του μπορεί να έχει πρόσβαση μέσω διαδικτύου σε λειτουργίες και στοιχεία που σχετίζονται με το θερμοκήπιο του. Το σύστημα αποτελείται:

Ιστοχώρος: Αποτελεί το γραφικό περιβάλλον του συστήματος όπου ο χρήστης αλληλεπιδρά με το σύστημα.

Αισθητήρες: Αισθητήρες θερμοκρασίας, υγρασίας, φωτός, υγρασίας χώματος (αισθητήρες προτύπου). Διαμορφώνονται ανάλογα με τις ανάγκες του χρήστη.

Συσκευές Δράσης: Ανεμιστήρας, UV-λαμπτήρας, Ηλεκτροβάννα (συσκευές προτύπου - μακέτας). Διαμορφώνονται ανάλογα με τις ανάγκες του χρήστη. Διαμορφώνονται ανάλογα με τις ανάγκες του χρήστη.

Μικροελεγκτής Arduino: Συνδέονται οι αισθητήρες και οι συσκευές δράσης καθώς και η βάση δεδομένων. Χαμηλού κόστους και χαμηλής κατανάλωσης ισχύος.

Λέξεις κλειδιά

Ενσωματωμένα συστήματα, Arduino, Προγραμματισμός διαδικτύου, Php, MySQL, Γλώσσα C, Θερμοκήπιο, "έξυπνες" συσκευές, Πρακτορικό σύστημα, Αυτοματισμός, Ενσωμάτωση ανθρώπινης ευφυΐας.

Abstract

Technology, nowadays, change the world rapidly. Most of all, internet and smart devices, made a breakthrough in everyday's life. In modern societies, almost everything and everyone is now online. The evolution of those two fields affect a lot of other and not so technological familiar filed, such as agriculture.

Especially in Greece, agriculture is one of most important fields for the economical balance of the country for centuries. Nowadays, because of the economical crisis and the lack of jobs, new entrepreneurs want to invest money in this filed. What is more, is that not only people who want to make profit from agriculture, but also ordinary people with regular jobs want to cultivate their own products for a better life and for saving money. In order to achieve these goals, both categories are in favor of using greenhouses because of the safety that they provide in rapid weather changes. But except from the initial cost of the greenhouse, human energy and most important significant amount of time must be spent.

This thesis, provide a solution to the above problem. "Smart Greenhouse" is used in order the user to keep track on the environmental conditions and to control devices that are significant for the progress of the plants. The embedded system that has been made has the following parts:

Website: The UI of the system. User interacts with the system that is applied to the greenhouse via internet.

Arduino controller: Sensors and devices are attached to the arduino controller. Exchange of data and action are made through the website.

Sensors: Temperature sensor, humidity sensor, soil moisture sensor, light sensor (these are witch are used in this thesis statement). Sensors can be varied, according to the needs of the user.

Devices: Fan, UV-Led, Solenoid (these are witch are used in this thesis statement). Devices can be varied, according to the needs of the user.

Keywords

Embedded systems, arduino, web programming, php, MySQL, C, greenhouse, smart devices, Human Intelligence Computing, Ambient Intelligence, Agents.

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον Δρ.Δασυγένη Μηνά, για την επιλογή και την εμπιστοσύνη στο πρόσωπό μου με την ανάθεση της παρούσας διπλωματικής, για την ελευθερία κατά τον σχεδιασμό και την υλοποίηση, αλλά και την συμβολή και καθοδήγηση για βελτιστοποίηση της ποιότητας του συστήματος. Επίσης, θα ήθελα να ευχαριστήσω τους γονείς μου και την αδερφή μου, για τις θυσίες τους και την υποστήριξή τους όλα αυτά τα χρόνια, αλλά και την ιδιαίτερη συμβολή τους κατά την σχεδίαση του συστήματος. Τέλος θα ήθελα να ευχαριστήσω φίλες, φίλους και συμφοιτητές για την οποιαδήποτε βοήθειά τους κατά την διάρκεια εκπόνησης της εργασίας.

Διάρθρωση κειμένου

Στο πρώτο κεφάλαιο, αναλύεται το θεωρητικό υπόβαθρο από το οποίο αντλήθηκαν οι γνώσεις και τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του έργου. Αναλύονται οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν, τα εργαλεία και ο μικροελεγκτής Arduino που ήταν η βάση της υλοποίησης.

Στο δεύτερο κεφάλαιο, παρουσιάζεται η σχεδίαση του κυκλώματος απεικόνισης βήμα προς βήμα. Δίνονται τα τεχνικά χαρακτηριστικά κάθε εξαρτήματος και ο τρόπος σύνδεσής τους στο μικροελεγκτή.

Στο τρίτο κεφάλαιο, παρουσιάζεται το λογισμικό μέρος του συστήματος. Ο σχεδιασμός της βάσης δεδομένων, η ανάπτυξη του ιστοχώρου διαχείρισης με παραδείγματα και εικόνες από την κάθε σελίδα και ο προγραμματισμός του μικροελεγκτή, αναλύονται σε αυτό το κεφάλαιο. Αναλύονται επίσης, οι βιβλιοθήκες που χρησιμοποιήθηκαν και ο τρόπος σύνδεσης του μικροελεγκτή με το διαδίκτυο και έπειτα με τον ιστοχώρο.

Το τέταρτο κεφάλαιο ασχολείται με τις μελλοντικές επεκτάσεις του συστήματος. Με την εφαρμογή τους στην σύγχρονη εκδοχή του συστήματος θα βελτιστοποιηθούν οι υπηρεσίες που παρέχει.

Στο πέμπτο κεφάλαιο, συνοψίζεται το έργο της διπλωματικής εργασίας. Αποτελεί και το τελευταίο κεφάλαιο.

Κεφάλαιο 1

Θεωρητικό υπόβαθρο

Η θεωρητική προσέγγιση των συστατικών στοιχείων που χρησιμοποιήθηκαν κατά την εκπόνηση της διπλωματικής εργασίας, αποτελούν το αντικείμενο μελέτης του κεφαλαίου αυτού. Τα στοιχεία αυτά χωρίζονται στον προγραμματισμό διαδικτύου, στην αρχιτεκτονική arduino που χρησιμοποιήθηκε, καθώς και των πλατφορμών ανάπτυξης λογισμικού.

1.1 Προγραμματισμός διαδικτύου

Ο προγραμματισμός διαδικτύου αφορά τη γνώση σχετικά με τις διαφορετικές τεχνολογίες που χρησιμοποιήθηκαν κατά την ανάπτυξη του ιστοχώρου του ολοκληρωμένου συστήματος καταγραφής και παρακολούθησης περιβαλλοντικών συνθηκών ενός θερμοκηπίου (Smart Greenhouse). Παρακάτω ακολουθεί ανάλυση των τεχνολογιών αυτών.

1.1.1 HTML

Η HTML (ακρωνύμιο για HyperText Markup Language ή Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων [1].

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες, οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός web browser είναι να διαβάζει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης, να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Για την κατασκευή του ιστοχώρου χρησιμοποιήθηκαν οι παρακάτω HTML ετικέτες:

Στοιχείο	Λειτουργία
<code><!DOCTYPE></code>	Η δήλωση του DOCTYPE πρέπει να είναι η πρώτη που γίνεται σε ένα έγγραφο HTML. Είναι μια εντολή στο πρόγραμμα περιήγησης για το ποια είναι η έκδοση της HTML στο έγγραφο.
<code><head></code>	Η ετικέτα <code><head></code> είναι βασική, καθώς στο πεδίο της ορίζονται δεδομένα όπως το είδος των χαρακτήρων που χρησιμοποιούνται (UTF – 8, ANSI κτλ), εξωτερικές πηγές αρχείων κτλ.
<code><title></code>	Η ετικέτα <code><title></code> καθορίζει τον τίτλο της ιστοσελίδας στο internet.
<code><body></code>	Η ετικέτα <code><body></code> ορίζει το σώμα του εγγράφου. Όλα τα περιεχόμενα ενός εγγράφου HTML, όπως κείμενο, υπερ-σύνδεσμοι, εικόνες, πίνακες, λίστες, κλπ. Βρίσκονται μέσα σε αυτό.
<code><div></code>	Η ετικέτα <code><div></code> καθορίζει ένα τμήμα σε ένα έγγραφο HTML. Χρησιμοποιείται για την ομαδοποίηση των στοιχείων, ώστε να δοθούν κοινές μορφοποιήσεις, με τη χρήση του id.
<code><form></code>	Η ετικέτα <code><form></code> χρησιμοποιείται για να δημιουργηθεί μια φόρμα HTML, με τις οποίες αλληλεπιδρούν οι χρήστες. Χρησιμοποιείται επίσης για να περνάει δεδομένα σε ένα διακομιστή.
<code><input></code>	Η ετικέτα <code><input></code> χρησιμοποιείται για να εισάγει πληροφορίες ο χρήστης. Περιλαμβάνονται σε μια ετικέτα <code><form></code> . Το χαρακτηριστικό type δηλώνει τη μορφή της ετικέτας input (type=text/ password/ button/ radio / checkbox/ number/ email/ date).
<code><table></code>	Ορίζει έναν πίνακα στην HTML. Ο πίνακας αποτελείται από ένα ή περισσότερα <code><tr></code> , <code><td></code> ή <code><th></code> .

<class>	Η ετικέτα <class> χρησιμοποιείται για την ομαδοποίηση χαρακτηριστικών μορφοποίησης που δίνονται σε αντικείμενα της HTML.
<h1> έως <h6>	Οι ετικέτες <h1> έως <h6> χρησιμοποιούνται για να καθορίσουν το μέγεθος των γραμμμάτων για τις επικεφαλίδες.

Πίνακας 1: HTML ετικέτες

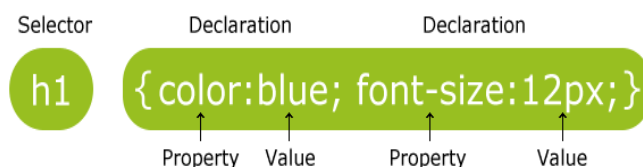
1.1.2 CSS

Η CSS (Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ) ή (αλληλουχία φύλλων στυλ) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης [2]. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει το παρουσιαστικό μιας ιστοσελίδας δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

Ένας κανόνας css αποτελείται από δύο κύρια μέρη:

- 1) **Επιλογέας (selector)** : Στοιχείο προς μορφοποίηση. Μπορεί να είναι είτε μία ετικέτα html ή κάποιο συγκεκριμένο αντικείμενο html.
- 2) **Δηλώσεις (declaration)**: Αποτελούνται από τις ιδιότητες που χαρακτηρίζουν τον επιλογέα και τις τιμές που παίρνουν αυτές.

Παρακάτω απεικονίζεται ένα πρότυπο σύνταξης κανόνα css:



Εικόνα 1: Κανόνας σύνταξης CSS

1.1.3 Javascript

Η JavaScript [3] είναι η πιο δημοφιλής scripting γλώσσα στον κόσμο. Είναι η τυπική γλώσσα που χρησιμοποιείται σε ιστοσελίδες, αλλά χρησιμοποιείται ευρέως και από desktop εφαρμογές, εφαρμογές κινητής τηλεφωνίας, καθώς και διακομιστές διαδικτύου.

Μια γλώσσα scripting είναι μια ελαφριά γλώσσα προγραμματισμού που υποστηρίζει τη συγγραφή σεναρίων. Σεναρία είναι γραμμές κώδικα που μπορούν να ερμηνεύονται και να εκτελούνται χωρίς μεταγλώττιση.

Βασικές λειτουργίες της javascript είναι οι εξής:

- 1) Χειρισμός CSS.
- 2) Δυναμική αλλαγή ετικετών και περιεχομένου HTML.
- 3) Αποθήκευση και ανάκτηση πληροφοριών στον υπολογιστή του χρήστη.
- 4) Εκτέλεση μετά από συμβάν (όπως ενεργοποίηση επιλογής μενού).
- 5) Επικοινωνία με php, xml, json αρχεία, αλλά και με άλλες ιστοσελίδες.

Η υλοποίηση της javascript είναι client – side (υπολογιστής πελάτη). Το χαρακτηριστικό αυτό προσδίδει ασφάλεια σε δεδομένα που εισάγονται, καθώς πριν αυτά αποσταλούν στο διακομιστή (server) μπορούν να ελεγχθούν για την εγκυρότητα τους. Με αυτόν τον τρόπο επιτυγχάνεται η αποφυγή κακόβουλων επιθέσεων από τους χρήστες.

```
if(s<10)
{
s="0"+s;
}
if (s==60)
{
s="00";
m++;
if(m<10)
{
m="0"+m;
}
document.getElementById("m").innerHTML=m;
reload();
program(1,1,1);
}
```

Εικόνα 2: Παράδειγμα κώδικα javascript

1.1.4 PHP

Η PHP [4] είναι μια γλώσσα προγραμματισμού βασισμένη σε σενάρια. Είναι κατάλληλη για την ανάπτυξη διαδικτυακών εφαρμογών και προγραμμάτων συστήματος. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Η PHP περιέχει κάποιες βασικές μεταβλητές οι οποίες βοηθούν στην ανάπτυξη διαδικτυακών δυναμικών εφαρμογών, όπως αναφέρθηκε παραπάνω. Παρακάτω αναλύονται οι μεταβλητές αυτές και τι αντικατοπτρίζουν:

- 1) **\$_GET**: Η προκαθορισμένη μεταβλητή \$ _GET χρησιμοποιείται για τη συλλογή τιμών σε μια φόρμα (method= "GET"). Οι πληροφορίες που αποστέλλονται από μια φόρμα με τη μέθοδο GET είναι ορατές σε όλους, εμφανίζονται στη γραμμή διευθύνσεων του browser και επίσης υπάρχει όριο στην ποσότητα των πληροφοριών προς αποστολή.
- 2) **\$_POST**: Η προκαθορισμένη μεταβλητή \$ _POST χρησιμοποιείται για τη συλλογή τιμών σε μια φόρμα (method= "POST"). Οι πληροφορίες που αποστέλλονται από μια φόρμα με τη μέθοδο POST είναι αόρατες στους άλλους και δεν έχει όρια για τον όγκο των πληροφοριών κατά την αποστολή.
- 3) **\$_REQUEST**: Πρόκειται για μία καθολική μεταβλητή η οποία περιέχει τις τιμές της \$_POST, \$_GET και \$_COOKIE. Ο τρόπος με τον οποίο χρησιμοποιείται χρειάζεται ιδιαίτερη προσοχή λόγω της ευαισθησίας των δεδομένων.
- 4) **\$_COOKIE**: Ένα cookie είναι ένα μικρό αρχείο που ενσωματώνει ο server στον υπολογιστή του χρήστη. Η συνάρτηση setcookie() χρησιμοποιείται για να ορίσει ένα cookie.
- 5) **\$_SESSION**: Μια μεταβλητή συνεδρίας χρησιμοποιείται για την αποθήκευση πληροφοριών, ή για την αλλαγή των ρυθμίσεων για μια

συνεδρία χρήστη. Οι μεταβλητές συνεδρίας είναι διαθέσιμες σε όλες τις σελίδες σε μια εφαρμογή. Κατά την ενασχόληση με μια εφαρμογή μπορούν να εκτελεστούν ενέργειες και αλλαγές δεδομένων. Η κατάσταση αυτή είναι μια συνεδρία. Μια συνεδρία PHP, επιτρέπει την αποθήκευση πληροφοριών του χρήστη στο διακομιστή για μελλοντική χρήση (δηλαδή όνομα, στοιχεία αγορών, κλπ). Ωστόσο, οι πληροφορίες συνεδρίας είναι προσωρινές και θα διαγράφονται αφού ο χρήστης έχει αποχωρήσει από το δικτυακό τόπο. Για μόνιμη αποθήκευση μπορεί να χρησιμοποιηθεί μια βάση δεδομένων.

1.1.5 GET και POST

Η συσκευή που έχει αναπτυχθεί σύμφωνα με τα εργαλεία που παρουσιάζονται στις παρακάτω ενότητες, επικοινωνεί με το server που φιλοξενείται η βάση δεδομένων. Τα δεδομένα αυτά μπορούν να σταλούν χρησιμοποιώντας κάποιον από τις μεθόδους που παρέχει το http πρωτόκολλο, GET ή POST.

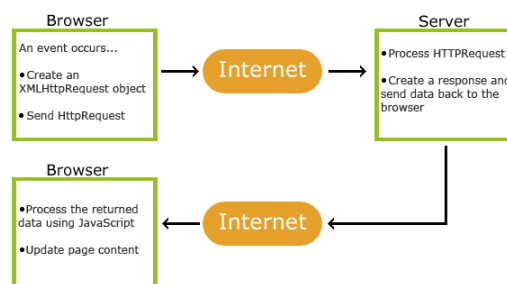
Το σύστημα “Smart Greenhouse” χρησιμοποιεί τη μέθοδο POST για ανταλλαγή πληροφοριών μεταξύ της συσκευής που έχει αναπτυχθεί και του ιστοχώρου. Οι λόγοι που επιλέχθηκε η συγκεκριμένη μέθοδο, είναι οι εξής:

1. Τα δεδομένα που στέλνονται με τη μέθοδο POST δε μπορούν να εμφανισθούν ούτε να αποθηκευθούν στη μνήμη κάποιου browser (φυλλομετρητή) ή κάποιου προγράμματος που επιβλέπει τη βάση, σε αντίθεση με τη μέθοδο GET.
2. Δεν υπάρχει όριο δεδομένων με τη μέθοδο POST, σε αντίθεση με τη GET.
3. Κατά την αποστολή κωδικών, η χρήση GET αμβλύνει σημαντικά την ασφάλεια του συστήματος, καθώς τα δεδομένα είναι φανερά σε εξωτερικούς παράγοντες, σε αντίθεση με τη μέθοδο POST.

Βάση των παραπάνω πλεονεκτημάτων, ουσιαστικά, της μεθόδου POST, έγινε και η επιλογή της. Με αυτόν τον τρόπο τα δεδομένα που ανταλλάσσονται μεταξύ της συσκευής και της βάσης, έχουν μεγαλύτερη ασφάλεια, από ότι θα είχαν αν χρησιμοποιούνταν η μέθοδος GET.

1.1.6 AJAX

Η αρχιτεκτονική AJAX (Asynchronous JavaScript And XML) [5] δεν είναι μια νέα γλώσσα προγραμματισμού, αλλά ένα νέος τρόπος χρησιμοποίησης των υπάρχοντων προτύπων. Επιτρέπει της ανταλλαγή δεδομένων με έναν server, και την ενημέρωση τμημάτων μιας ιστοσελίδας χωρίς να φορτώνεται εκ νέου ολόκληρη η σελίδα, επιτυγχάνοντας με αυτόν τον τρόπο πιο γρήγορη απεικόνιση των δεδομένων. Παρακάτω απεικονίζεται ο τρόπος με τον οποίο λειτουργεί η μέθοδος AJAX:



Εικόνα 3: Αρχιτεκτονική AJAX

1.1.7 MySQL

Η MySQL είναι μια πολύ γρήγορη σε απόδοση και ισχυρή σε δυνατότητες υλοποίηση ενός συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων. Τα δεδομένα στη MySQL αποθηκεύονται σε αντικείμενα βάσης δεδομένων τα οποία ονομάζονται πίνακες. Ένας πίνακας είναι μια συλλογή από σχετικές καταχωρήσεις δεδομένων και αποτελείται από στήλες και γραμμές. Οι βάσεις δεδομένων είναι χρήσιμες για την αποθήκευση πληροφοριών σε κατηγορίες.

Η διαχείριση των δεδομένων στις βάσεις γίνονται μέσω των SQL εντολών, στη MySQL. Η πιο σημαντική ίσως κατηγορία εντολών που χρησιμοποιείται είναι τα queries. Ένα query είναι ένα ερώτημα ή ένα αίτημα. Με τη MySQL, επιτρέπεται η διερεύνηση σε μια βάση δεδομένων και η επιστροφή ζητηθέντων πληροφοριών.

1.2 Αρχιτεκτονική arduino

Η αρχιτεκτονική arduino, αποτελείται από ένα σύνολο μικροελεγκτών υλικών συστημάτων που με την κατάλληλη παροχή λογισμικού, αυτοματοποιούν δραστηριότητες που επιθυμούν οι χρήστες. Στην ενότητα αυτή, παρουσιάζεται ο μικροελεγκτής Arduino που χρησιμοποιήθηκε, τα χαρακτηριστικά του, διαθέσιμα εξαρτήματα που μπορούν να συνδεθούν απ' ευθείας, αλλά και αυτά ανάλυση εκείνων που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος.

1.2.1 Γενικές πληροφορίες σχετικά με το Arduino

Το σύστημα στήθηκε πάνω σε μικροελεγκτή της αρχιτεκτονικής Arduino [7]. Πρόκειται για ανοιχτού λογισμικού πλατφόρμα πρωτοτύπων ηλεκτρονικών συσκευών που βασίζονται στην ευελιξία και στην ευκολία χρήσης υλικού και λογισμικού. Οι arduino συσκευές μπορούν να αλληλεπιδρούν με το περιβάλλον κάνοντας λήψη σημάτων μέσα από μια ποικιλία αισθητήρων. Τα έργα που βασίζονται σε αυτούς τους μικροελεγκτές, μπορούν να είναι αυτόνομα ή μπορούν να επικοινωνούν με το λογισμικό που τρέχει σε έναν υπολογιστή (π.χ. A, b, c, Processing, Bs).

1.2.2 Μοντέλα μικροελεγκτών Arduino

Στον πίνακα 2 παρουσιάζονται τα διάφορα μοντέλα μικροελεγκτών arduino. Οι διαφορές τους συνήθως είναι στους ακροδέκτες (pins) που έχουν, στην τάση εισόδου και εξόδου, καθώς και στα χαρακτηριστικά των συστημάτων που υλοποιούν (arduino Mega συνήθως για βιομηχανικά συστήματα – arduino Uno συνήθως για συστήματα έρευνας ή ερασιτεχνικά). Για την ανάπτυξη του συστήματος Smart Greenhouse χρησιμοποιήθηκε το arduino Uno.

Μοντέλα μικροελεγκτών Arduino		
		
Εικόνα 4: Arduino UNO	Εικόνα 5: Arduino	Εικόνα 6: Arduino Mega 2560

	Leonardo	
 Εικόνα 7: Arduino LilyPad	 Εικόνα 8: Arduino Mega ADK	 Εικόνα 9: Arduino Fio
 Εικόνα 10: Arduino Pro	 Εικόνα 11: Arduino BT	 Εικόνα 12: Arduino Nano

Πίνακας 2: Μοντέλα μικροελεγκτών Arduino

1.2.3 Arduino Shields

Η έννοια του shield (ασπίδα) στην αρχιτεκτονική arduino, είναι η ενσωμάτωση επιπλέον υλικού (hardware) στο μικροελεγκτή που του προσδίδει μια νέα ιδιότητα, κυρίως σε θέματα επικοινωνίας. Με την προσαρμογή του υλικού αυτού η επικοινωνία από σειριακή (μέσω usb) μετατρέπεται στην αντίστοιχη που διακρίνει το shield. Για την ανάπτυξη του συστήματος της διπλωματικής αυτής χρησιμοποιήθηκε το arduino Ethernet shield. Στον πίνακα 3 αναφέρονται διάφορα μοντέλα arduino shields.

Μοντέλα Arduino Shields	
Μοντέλο:	Περιγραφή:
 Εικόνα 13: Arduino Wifi shield	Επιτυγχάνει ασύρματη επικοινωνία μέσω wifi με το διαδίκτυο. Περιλαμβάνεται θύρα υποδοχής SD.
 Εικόνα 14: Arduino Ethernet shield	Επιτυγχάνει ενσύρματη επικοινωνία με το διαδίκτυο με τη χρήση καλωδίου Ethernet. Απαιτείται η χρήση router (δρομολογητή) για την εισαγωγή του καλωδίου. Περιλαμβάνεται θύρα υποδοχής SD.
 Εικόνα 15: Wireless SD shield	Επιτυγχάνει ασύρματη επικοινωνία με μία προκαθορισμένη ασύρματη μονάδα. Η εμβέλεια της επικοινωνίας ορίζεται έως και 100 πόδια σε εσωτερικούς χώρους ή έως και 300 πόδια σε εξωτερικού. Περιλαμβάνεται θύρα υποδοχής SD.
 Εικόνα 16: Arduino GPRS shield	Επιτυγχάνει επικοινωνία με τη χρήση GSM δικτύου. Απαιτείται η χρήση GSM κάρτας από κάποιον τηλεφωνικό πάροχο κινητής τηλεφωνίας.[8]

Πίνακας 3: Μοντέλα Arduino Shields

1.2.4 Τεχνικά χαρακτηριστικά Arduino UNO

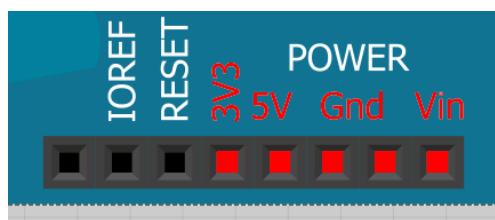
Όπως αναφέρθηκε και παραπάνω, στο σύστημα που αναπτύχθηκε χρησιμοποιήθηκε η έκδοση του Arduino UNO. Ο λόγος που χρησιμοποιήθηκε η έκδοση αυτή είναι το χαμηλό κόστος αγοράς (ελάχιστη τιμή 21 ευρώ – μέγιστη 26 ευρώ, στην ελληνική αγορά) και η χαμηλή κατανάλωση ρεύματος. Εκτός αυτού ήταν το πιο χρήσιμο για την έρευνα που υλοποιήθηκε για την πρώτη έκδοση του συστήματος, καθώς ανάλογα με τις απαιτήσεις του χρήστη μπορούν να προσθαφαιρεθούν αισθητήρια όργανα και συσκευές δράσης. Στον πίνακα 4 απεικονίζονται τα τεχνικά χαρακτηριστικά του Arduino UNO.

Τεχνικά χαρακτηριστικά Arduino UNO	
Μικροελεγκτής:	ATMEGA328
Τάση λειτουργίας:	5V
Τάση εισόδου:	7-12V
Όρια τάσης εισόδου:	6-20V
Ψηφιακοί ακροδέκτες I/O:	14, (6 PWM έξοδοι)
Αναλογικοί ακροδέκτες εισόδου:	6

Πίνακας 4: Τεχνικά χαρακτηριστικά Arduino UNO

1.2.5 Τροφοδοσία Arduino UNO

Το Arduino UNO μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φιν 2.1mm που βρίσκεται στην κάτω αριστερή γωνία. Για την αποφυγή προβλημάτων, η εξωτερική τροφοδοσία θα πρέπει να είναι από 7 ως 12V. Η εικόνα 16 παρουσιάζει τις εισόδους και εξόδους τροφοδοσίας του Arduino UNO.



Εικόνα 17: Είσοδοι/Έξοδοι τροφοδοσίας Arduino UNO

Οι ακροδέκτες τροφοδοσίας είναι οι ακόλουθοι:

Vin: Η τάση εισόδου της πλακέτας όταν χρησιμοποιεί εξωτερική πηγή ενέργειας. Η τροφοδοσία τάσης γίνεται μέσω αυτού του ακροδέκτη.

5V: Η τάση που χρησιμοποιείται από τα διάφορα μέρη της πλακέτας και το μικροελεγκτή είναι 5V. Η τάση αυτή, την οποία δίνει αυτός ο ακροδέκτης, είναι είτε η τάση 5V που δίνει η σύνδεση με USB, είτε η ρυθμισμένη τάση που δίνεται μέσω του Vin.

3.3V: Η τάση αυτή παράγεται από το ολοκληρωμένο FTDI. Το όριο άντλησης ρεύματος είναι 50mA.

GND: Είσοδοι γείωσης.

1.2.6 Μνήμη Arduino UNO

Ο μικροεπεξεργαστής ATmega328, ο οποίος είναι αντίστοιχος με το μικροεπεξεργαστή UNO έχει τρεις ομάδες μνήμης. Διαθέτει flash memory, στην οποία αποθηκεύονται τα Arduino sketch, SRAM (static random access memory), στην οποία δημιουργείται το sketch και χρησιμοποιεί τις μεταβλητές όταν τρέχει, και EEPROM, η οποία χρησιμοποιείται από τους προγραμματιστές για την αποθήκευση μακροχρόνιων πληροφοριών.

2KB μνήμης SRAM: Η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματα για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. Η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή πατηθεί το κουμπί επανεκκίνησης.

1KB μνήμης EEPROM: Μπορεί να χρησιμοποιηθεί για εγγραφή ή ανάγνωση δεδομένων από τα προγράμματα. Σε αντίθεση με την SRAM, δε χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή επανεκκίνησης.

32KB μνήμης Flash: 2 KB χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware είναι αναγκαίο για την εγκατάσταση προγραμμάτων στο μικροελεγκτή μέσω της θύρας USB. Τα υπόλοιπα 30KB της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή. Η μνήμη Flash, δε χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή επανεκκίνησης.

1.2.7 Ακροδέκτες Arduino UNO

Ο μικροελεγκτής Arduino UNO έχει ακροδέκτες (pins) που χωρίζονται σε αναλογικούς και ψηφιακούς. Οι εικόνες 17 και 18 απεικονίζουν τα pins και γίνεται επεξήγηση της λειτουργίας τους.



Εικόνα 18: Ψηφιακά pins του Arduino UNO

Ακροδέκτες 0 και 1: λειτουργούν ως RX και TX της σειριακής θύρας όταν το πρόγραμμά ενεργοποιεί τη σειριακή θύρα. Έτσι, όταν το πρόγραμμά στέλνει δεδομένα στη σειριακή θύρα, αυτά προωθούνται και στη θύρα USB μέσω του ελεγκτή Serial-Over-USB, αλλά και στον ακροδέκτη 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή. Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμά ενεργοποιήσει το σειριακό interface, χάνει 2 ψηφιακές εισόδους/εξόδους η πλατφόρμα.

Ακροδέκτες 2 και 3: λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Ρυθμίζονται μέσα από το πρόγραμμά, ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει άμεσα και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

Ακροδέκτες 3, 5, 6, 9, 10 και 11: μπορούν να λειτουργήσουν και ως ψευδό-αναλογικές εξοδοι με το σύστημα PWM (Pulse Width Modulation).



Εικόνα 19: Αναλογικά pins του Arduino UNO

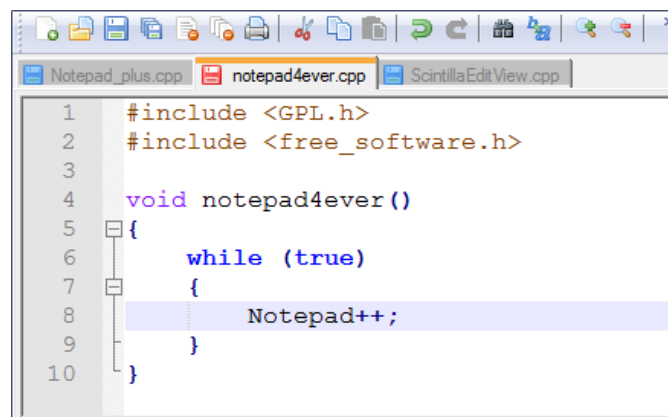
Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN όπως φαίνεται και στην εικόνα 18, υπάρχει μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με τη σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Με αυτόν τον τρόπο, αν τροφοδοτηθεί ο ακροδέκτης AREF με 3.3V και στη συνέχεια διαβάσει κάποιον ακροδέκτη αναλογικής εισόδου στο οποίο εφαρμόζεται τάση 1.65V, το Arduino θα επιστρέψει την τιμή 512.

1.3 Πλατφόρμες ανάπτυξης λογισμικού

Στην ενότητα αυτή γίνεται αναφορά σχετικά με τις πλατφόρμες ανάπτυξης λογισμικού που χρησιμοποιήθηκαν. Για το προγραμματισμό και την κατασκευή του ιστοχώρου χρησιμοποιήθηκε το Notepad++ [9] με την ταυτόχρονη χρήση του τοπικού διακομιστή Xampp [10], ανοιχτού λογισμικού και τα δύο. Για το προγραμματισμό του arduino χρησιμοποιήθηκε το Arduino IDE [11] που επίσης είναι ανοιχτό λογισμικού. Για την επεξεργασία εικόνων χρησιμοποιήθηκε το Photoshop [12] ενώ για την κατασκευή σε εικόνες των κυκλωμάτων χρησιμοποιήθηκε το Fritzing [13] . Επίσης χρησιμοποιήθηκε το πρόγραμμα EAGLE [26] για τη σχεδίαση της πλακέτας του κυκλώματος.

1.3.1 Notepad ++

Το Notepad++ [9] χρησιμοποιήθηκε για την ανάπτυξη όλων των script σε php, javascript, html, mysql, ajax, css. Πρόκειται για ανοιχτού λογισμικού πρόγραμμα, το οποίο εκτός από τις παραπάνω μπορεί να χρησιμοποιηθεί για τη συγγραφή και άλλων γλωσσών προγραμματισμού.



Εικόνα 20: Notepad++

1.3.2 Xampp

Ο τοπικός διακομιστής Xampp [10], χρησιμοποιήθηκε για την ανάπτυξη του ιστοχώρου με την απεικόνιση που προσέφερε. Πρόκειται για ανοιχτού λογισμικού πρόγραμμα και προσομοιώνει τη φιλοξενία ενός ιστοχώρου σε web server (διακομιστής διαδικτύου)

1.3.3 Arduino IDE

Το περιβάλλον ανάπτυξης Arduino [11] περιέχει ένα πρόγραμμα επεξεργασίας κειμένου, για τη σύνταξη του κώδικα, μια περιοχή στην οποία εμφανίζονται μηνύματα, μία κονσόλα κειμένου και μια γραμμή εργαλείων υπό μορφή κουμπιών. Συνδέεται με το hardware μέρος του arduino για να φορτώσει προγράμματα και να επικοινωνεί μαζί τους. Ο κώδικας που έχει γραφεί για το Arduino ονομάζεται sketch. Στον πίνακα 5 παρουσιάζονται τα εργαλεία του περιβάλλοντος ανάπτυξης, υπό μορφή κουμπιών και στην εικόνα 20 το ίδιο το περιβάλλον.

Εργαλεία ανάπτυξης Arduino IDE	
Εργαλείο:	Περιγραφή:
 Εικόνα 21: Verify	Ελέγχει για συντακτικά λάθη στον κώδικα.
 Εικόνα 22: Upload	Μεταγλωττίζει τον κώδικα και το φορτώνει στο Arduino. Αν δεν είναι συντακτικά σωστός δε μπορεί να γίνει η φόρτωση.
 Εικόνα 23: New	Δημιουργεί ένα νέο sketch.
 Εικόνα 24: Open	Παραθέτει ένα μενού με όλα τα sketch. Ενεργοποιώντας ένα από αυτά, θα ανοίξει αυτόματα στο τρέχον παράθυρο.
 Εικόνα 25: Save	Αποθηκεύει ένα sketch.
 Εικόνα 26: Serial Monitor	Ανοίγει τη σειριακή οθόνη. Μέσω αυτής παρακολουθείται η ανταλλαγή δεδομένων που γίνεται στη σειριακή θύρα.

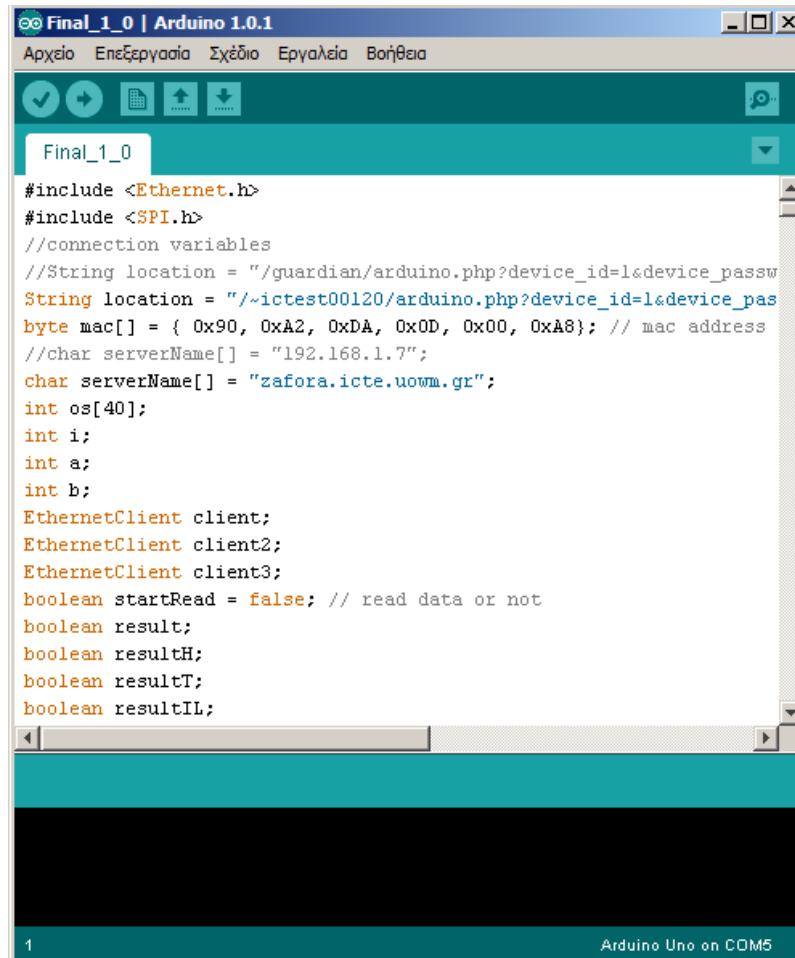
Πίνακας 5: Εργαλεία Ανάπτυξης Arduino IDE

Το Arduino IDE είναι βασισμένο σε Java [16] και συγκεκριμένα παρέχει:

- Ένα πρακτικό περιβάλλον για τη συγγραφή των προγραμμάτων, με συντακτική χρωματική σήμανση.
- Βιβλιοθήκες για προέκταση της, εύκολες στην πρόσβαση μέσω διαδικτύου.
- Compiler (μεταγλωττιστής) για τη μεταγλώττιση των sketch.
- Μία σειριακή οθόνη (serial monitor) που παρακολουθεί τις επικοινωνίες της

σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για τον προσδιορισμό τυχόν λαθών των sketch (σεναρίων).

- Την επιλογή για μεταφορά των μεταγλωττισμένων sketch στο Arduino.



```
#include <Ethernet.h>
#include <SPI.h>
//connection variables
//String location = "/guardian/arduino.php?device_id=1&device_passw
String location = "/~ictest00120/arduino.php?device_id=1&device_pas
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0x00, 0xA8}; // mac address
//char serverName[] = "192.168.1.7";
char serverName[] = "zafora.ict.e.uowm.gr";
int os[40];
int i;
int a;
int b;
EthernetClient client;
EthernetClient client2;
EthernetClient client3;
boolean startRead = false; // read data or not
boolean result;
boolean resultH;
boolean resultT;
boolean resultIL;
```

Εικόνα 27: Απεικόνιση Arduino IDE

Η γλώσσα συγγραφής κώδικα του Arduino βασίζεται στη γλώσσα Wiring [14] μια παραλλαγή C/C++ [15] για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C, καθώς και μερικά χαρακτηριστικά της C++. Για compiler χρησιμοποιείται ο AVR gcc [17] και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc.

Λόγω της καταγωγής της από τη C, στη γλώσσα συγγραφής κώδικα του Arduino, μπορούν να χρησιμοποιηθούν ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στη C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για τη διαχείριση του ειδικού hardware του Arduino.

Τα προγράμματα του Arduino διαιρούνται σε τρία μέρη:

- 1) Δομή (structure)
- 2) Τιμές (values)
- 3) Συναρτήσεις (functions)

Στην παρακάτω εικόνα, εμφανίζεται η βασική δομή της αρχιτεκτονικής του κώδικα μιας arduino συσκευής.



Εικόνα 28: Βασική αρχιτεκτονική Κώδικα Arduino

Κεφάλαιο 2

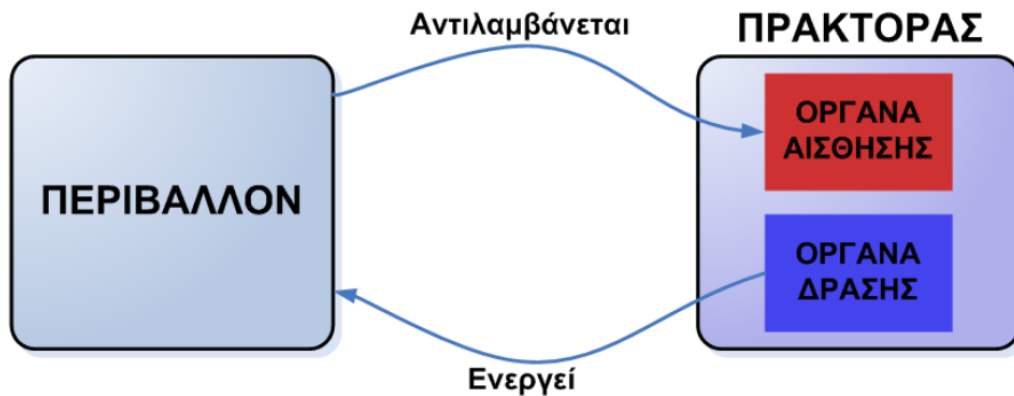
Υλικό μέρος συστήματος

Η αρχιτεκτονική Arduino επιτρέπει τη σύνδεση διαφόρων εξαρτημάτων (αισθητήρων, συσκευών κτλ) τα οποία με τον κατάλληλο προγραμματισμό επιτελούν έναν στόχο, όπως αναφέρθηκε και στο κεφάλαιο 1. Βασισμένο σε αυτές τις αρχές είναι το υλικό μέρος του συστήματος χρησιμοποιείται από το σύστημα Smart Greenhouse.

Το υλικό (hardware) μέρος του συστήματος χωρίζεται σε 6 επιμέρους τμήματα:

1. Υλικό υλοποίησης επικοινωνίας με το διαδίκτυο
2. Περιβαλλοντικοί αναλογικοί αισθητήρες
3. Συσκευές δράσης ψηφιακά ρυθμιζόμενες
4. Μετατροπή κυκλώματος σε ηλεκτρονική πλακέτα
5. Κατασκευή μακέτας θερμοκηπίου
6. Χαρακτηριστικά συσκευής

Ουσιαστικά, το σύστημα ακολουθεί εν μέρει, τη δομή ενός πράκτορα λογισμικού, η οποία απεικονίζεται στην εικόνα 29. Υπάρχει δηλαδή ένα περιβάλλον (θερμοκήπιο) και το σύστημα (πράκτορας), το οποίο μέσω των οργάνων αίσθησης (περιβαλλοντικοί αναλογικοί αισθητήρες) αντιλαμβάνεται το τι συμβαίνει και στη συνέχεια, με τον κατάλληλο προγραμματισμό (αναλύεται στο 3^ο κεφάλαιο), παίρνει τις αποφάσεις για να ενεργοποιήσει ή όχι τα όργανα δράσης (συσκευές δράσης ψηφιακά ρυθμιζόμενες).



Εικόνα 29: Η έννοια του πράκτορα λογισμικού

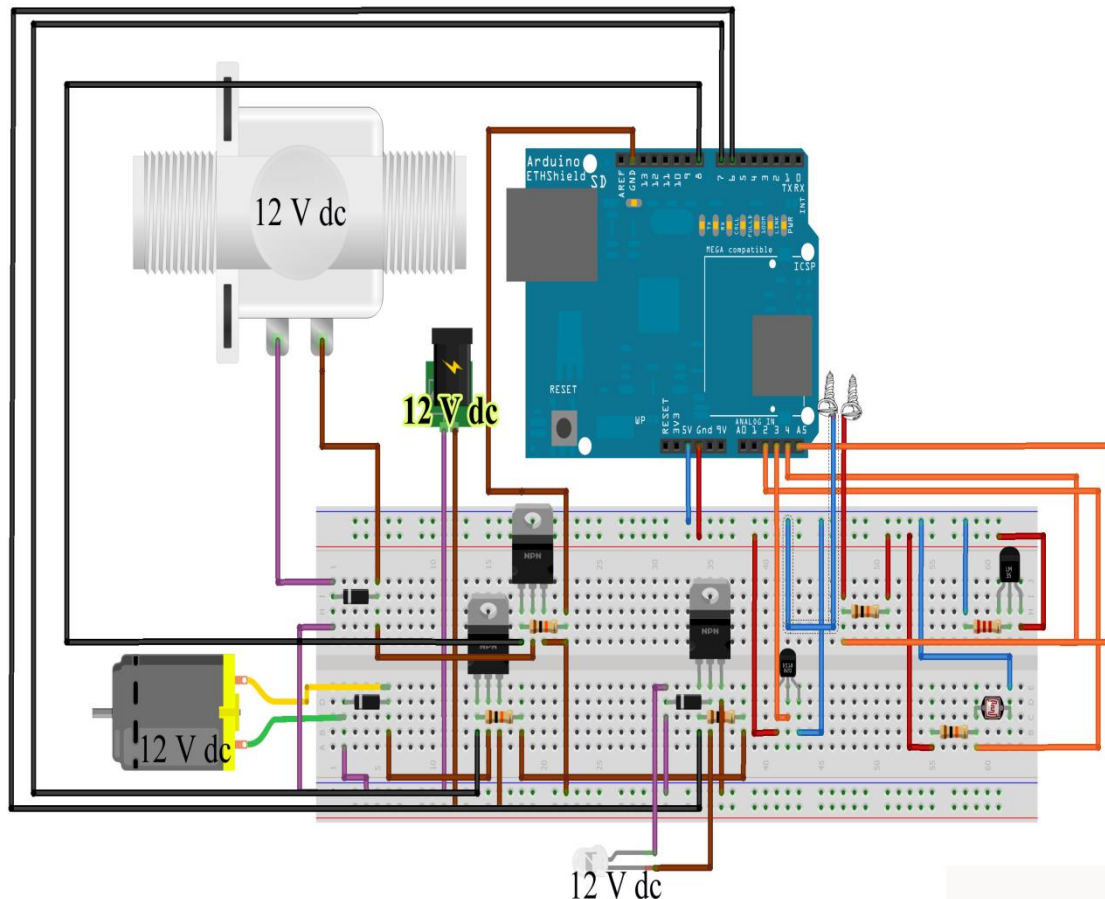
Η τελική μορφή του κυκλώματος το οποίο κατασκευάστηκε απεικονίζεται στην εικόνα 30. Το σύστημα αποτελείται από:

1. **Αισθητήρια όργανα:**
 - i. Αισθητήρας θερμοκρασίας LM 335
 - ii. Αισθητήρας υγρασίας 808h5V5
 - iii. Αισθητήρας έντασης φωτός Photocell
 - iv. Αισθητήρας υγρασίας χώματος
2. **Όργανα δράσης:**
 - i. Ανεμιστήρας
 - ii. Ηλεκτροβάννα
 - iii. UV-led

Για την καλύτερη κατανόηση του κυκλώματος χρησιμοποιείται ο παρακάτω χρωματικός κώδικας όσον αφορά τα καλώδια:

- Καλώδια μαύρα: Σύνδεση ψηφιακών ακροδεκτών Arduino.
- Καλώδια πορτοκαλί: Σύνδεση αναλογικών ακροδεκτών Arduino.
- Καλώδια καφέ: Σύνδεση γείωσης και αρνητικών πόλων τάσης 12V.
- Καλώδια μωβ: Σύνδεση θετικών πόλων τάσης 12V.
- Καλώδια μπλε: Σύνδεση θετικών πόλων τάσης 5V.
- Καλώδια κόκκινα: Σύνδεση γείωσης και αρνητικών πόλων τάσης 5V.

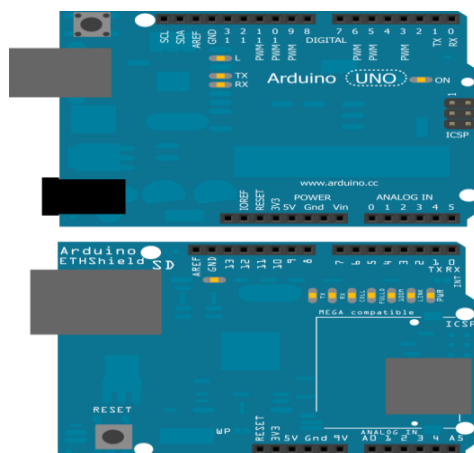
Λεπτομέρειες σχετικά με τις επιμέρους συνδέσεις αναλύονται περαιτέρω, στις επόμενες ενότητες.



Εικόνα 30: Κύκλωμα συστήματος

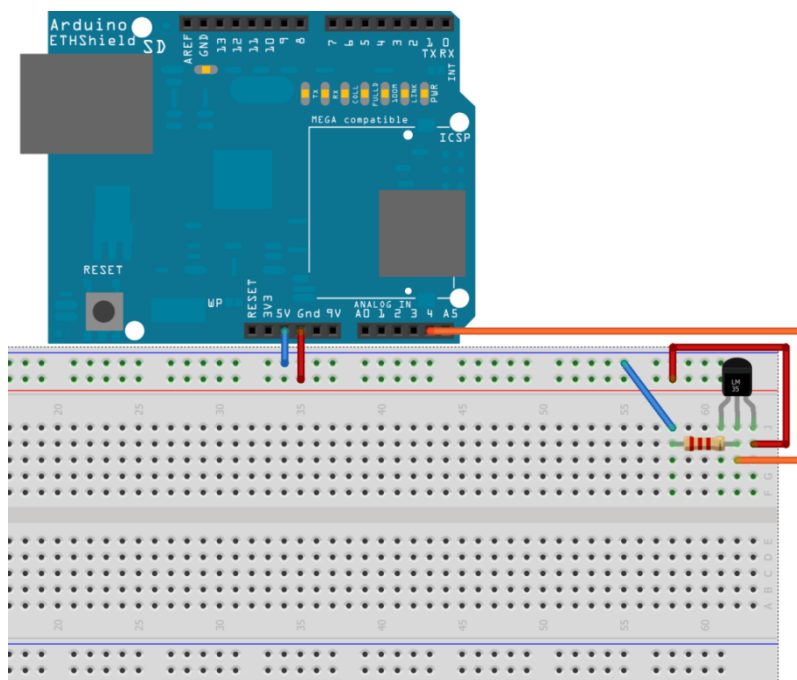
2.1 Υλικό υλοποίησης επικοινωνίας με το διαδίκτυο

Η επικοινωνία του υλικού μέρους του συστήματος Smart Greenhouse, επετεύχθη με τη χρήση του Arduino Ethernet Shield [18], όπως αναφέρθηκε και παραπάνω. Οι ακροδέκτες των δύο συνεργαζόμενων συσκευών, οι οποίες είναι κατασκευασμένες και από την ίδια εταιρεία, έχουν ίδια τοποθέτηση στις πλακέτες, ώστε εκείνες του Ethernet Shield να εισάγονται στις αντίστοιχες υποδοχές του Arduino Uno. Στην εικόνα 31 που ακολουθεί απεικονίζεται αυτή τους η ομοιότητα.



Εικόνα 31: Arduino UNO και Arduino Ethernet Shield

Κατά τη σύνδεση του αισθητήρα με το μικροελεγκτή Arduino, παρατηρήθηκε η υπερθέρμανσή του, μετά από το χρονικό διάστημα της μίας ώρας. Το γεγονός αυτό οφειλόταν στην τάση λειτουργίας του αισθητήρα που είναι από 2V έως 3V, ενώ δεχόταν 5V. Γι' αυτό λοιπόν χρησιμοποιήθηκε σε σειρά μία αντίσταση των 2,2kΩ, ώστε να μειώσει τα 5V σταθερής τάσης που δίνει το Arduino, επιτρέποντας την ομαλή λειτουργία του αισθητήρα. Παρακάτω απεικονίζεται η συνδεσμολογία που χρησιμοποιήθηκε (εικόνα 33).



Εικόνα 33: Συνδεσμολογία αισθητήρα θερμοκρασίας LM 335

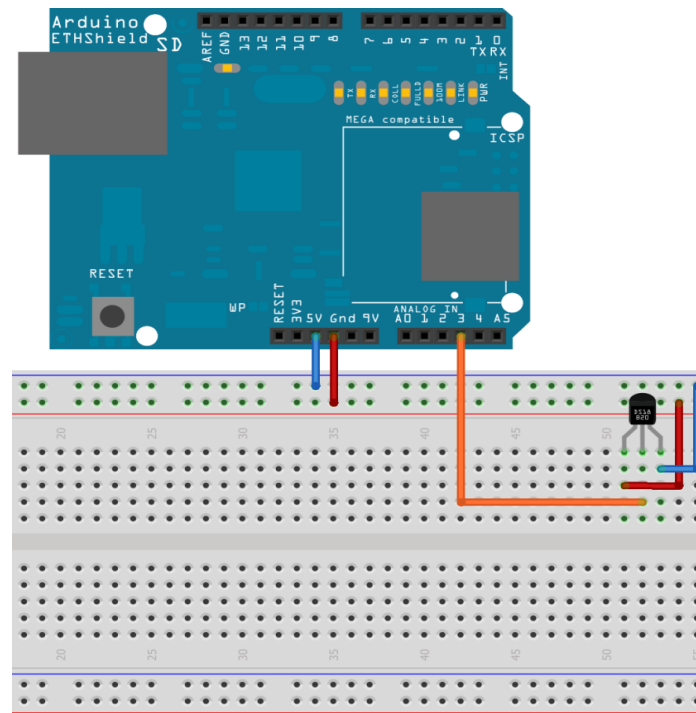
2.2.2 Αισθητήρας Υγρασίας 808h5V5

Για τη μέτρηση της υγρασίας του περιβάλλοντος του θερμοκηπίου, χρησιμοποιήθηκε ο αισθητήρας υγρασίας 808h5V5 [20] (εικόνα 34). Ο αισθητήρας αυτός έχει το πλεονέκτημα, συγκριτικά με αντίστοιχα μοντέλα, της άμεσης έναρξης λειτουργίας χωρίς κάποιο χρόνο προθέρμανσης του αισθητήρα. Επίσης, μπορεί να δουλέψει και σε περιβάλλοντα που διέπονται από ακραία καιρικά φαινόμενα (-40 C έως 120 C).



Εικόνα 34: Αισθητήρας Υγρασίας 808h5V5

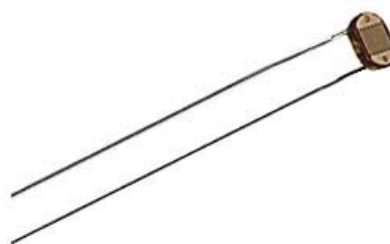
Ο αισθητήρας συνδέεται απευθείας με την αναλογική είσοδο του μικροελεγκτή και τροφοδοτείται και με τα 5V της εξόδου του Arduino. Δε χρειάστηκε κάποιο επιπλέον εξάρτημα για την ομαλή λειτουργία του, γεγονός που είχε συμβάλει και στην απόκτηση του για τη σύσταση του συστήματος. Στην εικόνα 35 απεικονίζεται η συνδεσμολογία που χρησιμοποιήθηκε.



Εικόνα 35: Συνδεσμολογία αισθητήρα υγρασίας 808h5V5

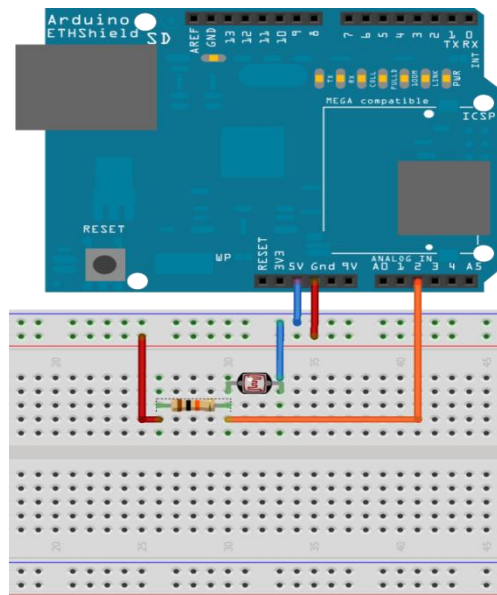
2.2.3 Αισθητήρας έντασης φωτός Photocell

Για την ανίχνευση της έντασης του φωτός στο περιβάλλον του θερμοκηπίου, χρησιμοποιήθηκε ένα Photocell [21] (εικόνα 36). Ο λόγος για τον οποίο επιλέχθηκε ο αισθητήρας αυτός, ήταν το πολύ χαμηλό του κόστος, η άμεση ενεργοποίηση του κατά την εκκίνηση του συστήματος, καθώς και η χαμηλή κατανάλωση ρεύματος (λιγότερο από 1mA κατά μέσο όρο ανεξάρτητα με την τάση εισόδου).



Εικόνα 36: Αισθητήρας έντασης φωτός Photocell

Το Photocell λειτουργεί σαν μία αντίσταση. Ανάλογα με την ποσότητα του φωτός που ανιχνεύει μεταβάλλει σε Ω hm μια μεταβλητή. Η συνδεσμολογία του συγκεκριμένου αισθητήρα απαιτούσε τη χρήση μίας αντίστασης είτε 1k Ω είτε 10 k Ω , σε σειρά. Επιλέχθηκε η δεύτερη, καθώς η πρώτη είναι για υπαίθρια περιβάλλοντα και θα είχε μεγαλύτερη ακρίβεια σε ανίχνευση πολύ φωτεινών εντάσεων και όχι των πιο σκοτεινών, γεγονός που δε συνάδει με τις προδιαγραφές του συστήματος. Η συνδεσμολογία της εικόνας 37, απεικονίζει τη συνδεσμολογία που χρησιμοποιήθηκε.



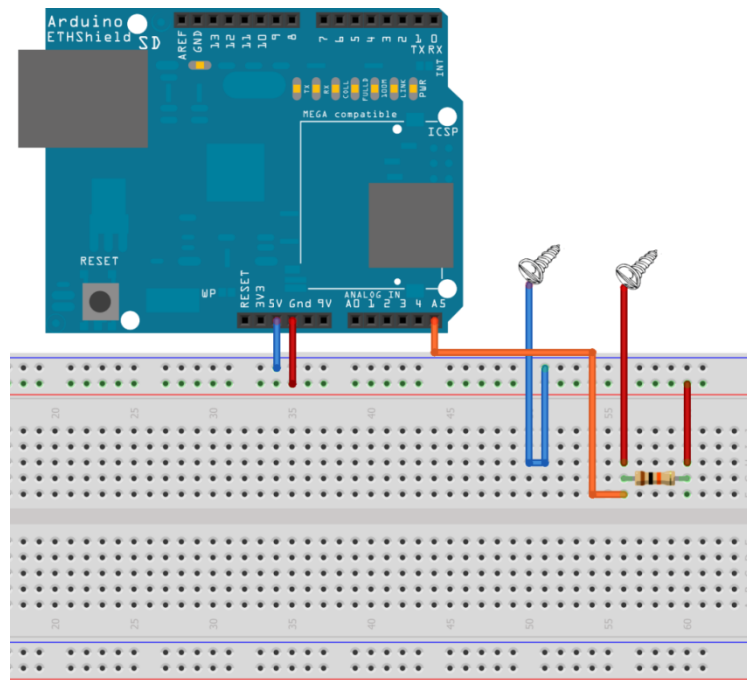
Εικόνα 37: Συνδεσμολογία αισθητήρα έντασης φωτός Photocell

Το Photocell ανιχνεύει την ένταση του φωτός και ανάλογα αυξομειώνει την αντίσταση του. Εφόσον υπάρχει και η αντίσταση των 10 kΩ ανάλογα συμπεριφέρεται και εκείνη. Με αυτόν τον τρόπο αυξάνεται το ρεύμα που διατρέχει τις δύο αυτές αντιστάσεις και αλλάζει εν τέλει την τάση στα άκρα της αντίστασης όπου και λαμβάνεται μέτρηση και μεταφέρεται στην αναλογική είσοδο 2 του μικροελεγκτή.

2.2.4 Αισθητήρας υγρασίας χώματος

Για τη μέτρηση της υγρασίας χώματος, δε χρησιμοποιήθηκε κάποιος αισθητήρας του εμπορίου. Αντιθέτως κατασκευάστηκε με τη χρήση δυο απλών «γαλβανιζέ» καρφιών και μία αντίστασης 10 kΩ (εικόνα 38). Ο τρόπος με τον οποίο λειτουργεί είναι ο εξής:

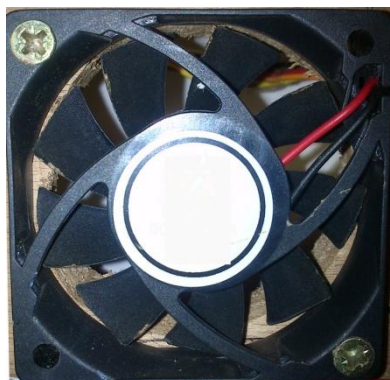
Τα δύο καρφιά ενσωματώνονται μέσα στο έδαφος της περιοχής που χρειάζεται η μέτρηση (στη συγκεκριμένη περίπτωση στη μακέτα που κατασκευάστηκε). Το ένα καρφί έχει συνδεθεί με καλώδιο που φέρει φορτίο τάσης 5V. Το δεύτερο καρφί έχει συνδεθεί με καλώδιο το οποίο συνδέεται με την αντίσταση και τη γείωση. Μεταξύ της αντίστασης και της γείωσης λαμβάνεται η μέτρηση του φορτίου που υπάρχει μεταξύ των δύο καρφιών, καθώς όσο πιο υγρό είναι το έδαφος τόσο περισσότερο φορτίο θα ανιχνευθεί λόγω της ιδιότητας του νερού να συμπεριφέρεται ως καλός αγωγός του ρεύματος. Να σημειωθεί, ότι διαφορετικά είδη εδαφών έχουν και διαφορετική τιμή για την υγρή ή την ξηρή τους κατάσταση η οποία προκύπτει μετά από πειραματικές διαδικασίες.



Εικόνα 38: Συνδεσμολογία αισθητήρα υγρασίας εδάφους

2.3 Συσκευές δράσης ψηφιακά ρυθμιζόμενες

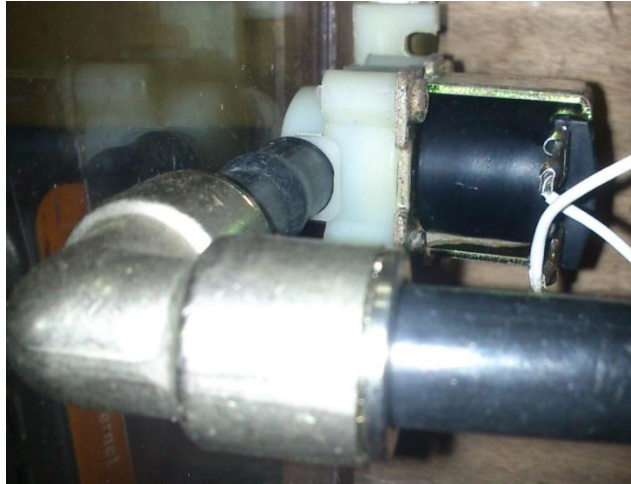
Το σύστημα Smart Greenhouse μπορεί να αντιλαμβάνεται τι συμβαίνει στο περιβάλλον του θερμοκηπίου και με βάση τη νοημοσύνη, που ο χρήστης του έχει εισαγάγει μέσω του προγραμματισμού του, μπορεί να δρα ανάλογα. Η δράση αυτή επιτυγχάνεται μέσω των συσκευών δράσης. Στη μακέτα χρησιμοποιήθηκαν ένας ανεμιστήρας (εικόνα 39), ένα UV led (εικόνα 40) και μια ηλεκτροβάννα (εικόνα 41). Σε πραγματικές συνθήκες, οι ανάγκες ενός θερμοκηπίου είναι περισσότερες, αλλά σε πρώτο στάδιο οι βασικές ανάγκες (πότισμα, μείωση υγρασίας / θερμοκρασίας, υπεριώδης ακτινοβολία) καλύπτονταν με τις συσκευές αυτές.



Εικόνα 39: Ανεμιστήρας



Εικόνα 40: UV Led



Εικόνα 41: Ηλεκτροβάνα

Οι συσκευές δράσης που χρησιμοποιήθηκαν έχουν όλες το εξής κοινό χαρακτηριστικό: Απαιτούν 12 V dc τάση. Όμως όπως έχει αναφερθεί παραπάνω η τάση εισόδου που χρησιμοποιήθηκε ήταν 5V. Για τον έλεγχο λοιπόν των συσκευών μέσω του Arduino χρησιμοποιήθηκε συνδεσμολογία με τη χρήση N-MOSFET και εξωτερικής τροφοδοσίας 12V. Λεπτομερή ανάλυση των επιμέρους τμημάτων και συσκευών δίνεται στη συνέχεια.

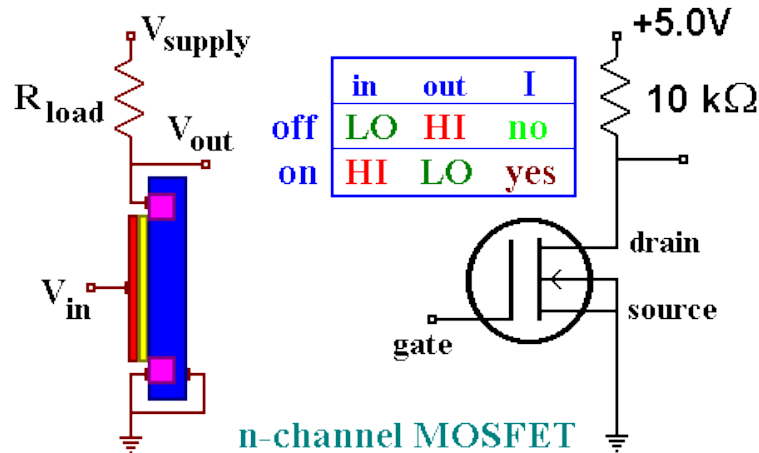
2.3.1 N-MOSFET

Τα τρανζίστορ MOSFET (Metal Oxide Semiconductor Field Effect Transistor, εικόνες 42,43) [22] είναι διατάξεις ελεγχόμενες από τάση οι οποίες δεν απαιτούν μεγάλα ρεύματα οδήγησης όπως οι αντίστοιχες διπολικές διατάξεις. Πρόκειται για τρανζίστορ επίδρασης πεδίου (FET) που λειτουργούν με τρόπο παρόμοιο με τα τρανζίστορ επίδρασης πεδίου επαφής (JFET). Η βασική τους διαφορά έναντι των JFET είναι ότι το δυναμικό που ελέγχει τη λειτουργία τους (δυναμικό πύλης) εφαρμόζεται στην ενεργό περιοχή (κανάλι) διαμέσου ενός μονωτικού στρώματος από κατάλληλο οξείδιο αντί να εφαρμόζεται μέσω μιας p-n επαφής.

Τα MOSFET μπορούν να κατασκευαστούν τόσο μεμονωμένα (διακριτά τρανζίστορ) όσο και ως μέρη ενός ευρύτερου ολοκληρωμένου κυκλώματος. Η ενεργή περιοχή τους μπορεί να είναι ένα κανάλι p-τύπου οπότε το τρανζίστορ χαρακτηρίζεται και ως PMOS ή ένα κανάλι n-τύπου οπότε το τρανζίστορ χαρακτηρίζεται και ως NMOS. Το μεγάλο πλεονέκτημα των δοτών MOSFET είναι η πολύ μικρή ισχύς τους λόγω του μονωτικού στρώματος που παρεμβάλλεται μεταξύ πύλης και καναλιού.

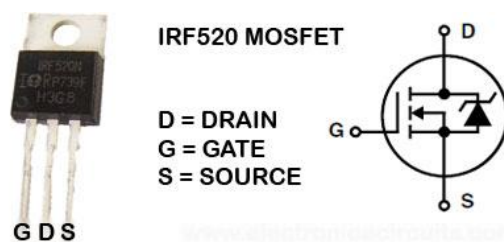
Η κεντρική ιδέα της λειτουργίας των MOSFET είναι η ακόλουθη: Η εφαρμογή μιας διαφοράς δυναμικού μεταξύ πύλης και πηγής δημιουργεί ένα

ηλεκτρικό πεδίο το οποίο διαπερνά το διηλεκτρικό της πύλης και δημιουργεί (MOSFET προσάυξης) ή τροποποιεί (MOSFET διακένωσης) ένα στρώμα αναστροφής. Το στρώμα αναστροφής έχει αντίθετο τύπο αγωγιμότητας από αυτόν του υποβάθρου και ταυτόσημο με αυτόν των διαχύσεων σε πηγή και απαγωγό.



Εικόνα 42: N-MOSFET

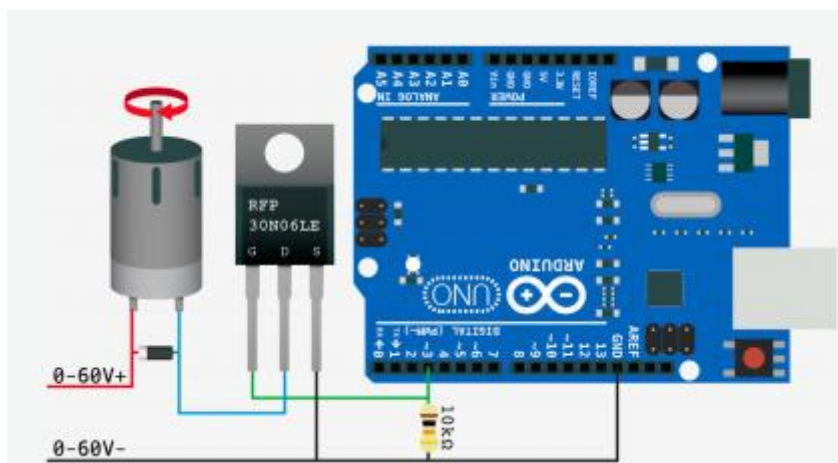
Βάση της κεντρικής ιδέας που αναφέρθηκε στην παραπάνω παράγραφο, χρησιμοποιήθηκαν 3 N-MOSFET, ένα για την κάθε συσκευή δράσης που χρησιμοποιείται στην έκδοση του συστήματος που κατασκευάστηκε. Συγκεκριμένα χρησιμοποιήθηκε το μοντέλο IRF 520 [23] (εικόνα 43). Με τον τρόπο αυτό επετεύχθη η σύνδεση συσκευών υψηλότερου φορτίου τάσης με το μικροελεγκτή που δουλεύει στα 5V. Ουσιαστικά, ο μικροελεγκτής ανοίγει ή κλείνει τις συσκευές οι οποίες τροφοδοτούνται μέσω εξωτερικής πηγής, ανεξάρτητα με την αντίστοιχη του Arduino. Φυσικά, ανάλογα μπορούν να συνδεθούν και συσκευές ακόμα υψηλότερη τάσης, με τη δυνατότητα εγκατάστασης σε ήδη εξοπλισμένα, με αντίστοιχης χρήσης συσκευές, θερμοκήπια.



Εικόνα 43: IRF 520

2.3.2 Ανάλυση μεθόδου συνδεσμολογίας των συσκευών δράσης

Για τη σύνδεση των συσκευών στο κύκλωμα που αλληλεπιδρά με το μικροελεγκτή, εφαρμόστηκε και για τις τρεις συσκευές ο ίδιος τρόπος [25]. Εκτός από N-MOSFET που αναλύθηκαν παραπάνω χρησιμοποιήθηκε και δίοδοι και αντιστάσεις των 10 kΩ. Η εικόνα 44 απεικονίζει τη μέθοδο που χρησιμοποιήθηκε.



Εικόνα 44: Παράδειγμα συνδεσμολογία με N-MOSFET

Στο παράδειγμα της εικόνας χρησιμοποιείται ένα σωληνοειδές το οποίο απαιτεί 60V τάση για να δουλέψει. Τα pin του N-MOSFET G (gate) είναι υπεύθυνο για να επιτρέψει τη ροή του ρεύματος μεταξύ του D (drain) και του S (source), ώστε να δουλέψει η συσκευή. Με αυτόν τον τρόπο το arduino ελέγχει τη συσκευή. Εκτός όμως των παραπάνω εξαρτημάτων, υπάρχει μία διόδος ανάμεσα στο θετικό και αρνητικό πόλο της συσκευής ο οποίος καταλήγει στο D. Η χρησιμότητα της είναι η εξής:

Όταν το σωληνοειδές σταματήσει να λειτουργεί είναι πολύ πιθανό μέσα στα επόμενα 5-10 ms να επιστρέψει μια τάση σαν και αυτήν με την οποία τροφοδοτήθηκε όσον αφορά το μέγεθος. Η διόδος είναι τοποθετημένη με αυτόν τρόπο, ώστε αυτή η τάση να μην εισέρχεται στο D του N-MOSFET, γιατί αν συνέβαινε το γεγονός αυτό θα καίγταν και εκείνο, αλλά και ο μικροελεγκτής. Για την κατασκευή του κυκλώματος του συστήματος χρησιμοποιήθηκε η διόδος IN4007 [24].

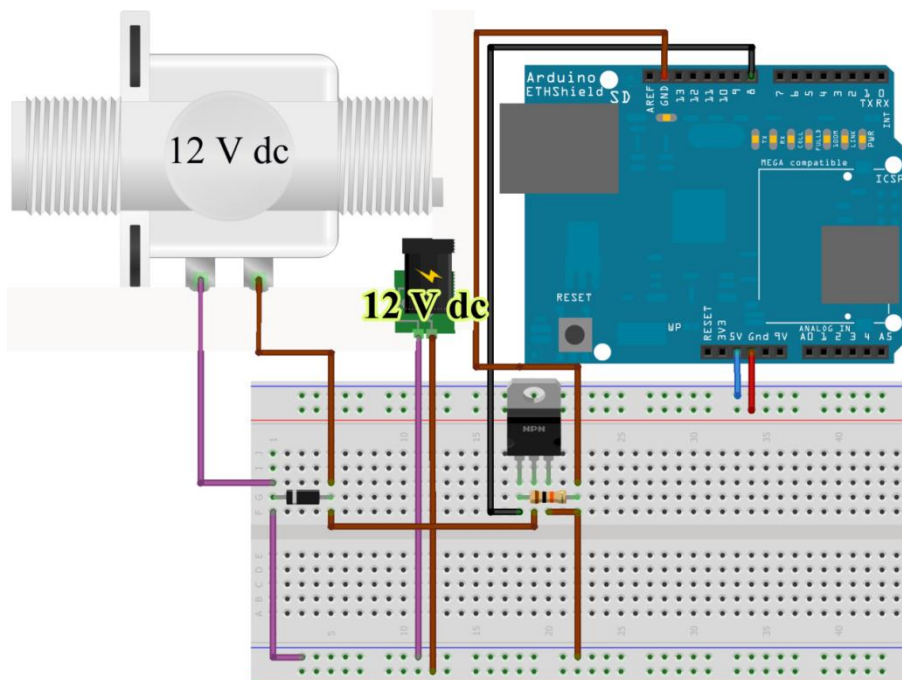
Εκτός των παραπάνω υπάρχει και μια αντίσταση στη συνδεσμολογία των 10 kΩ. Ο ρόλος της είναι πολύ σημαντικός, καθώς όταν το συγκεκριμένο digital pin του Arduino δε στέλνει σήμα, κρατάει σταθερή και χαμηλή το G. Με αυτόν τον τρόπο αποφεύγεται τυχόν επαφή του μικροελεγκτή με απαγορευτικά για τη λειτουργία του μεγέθη τάσης και ρεύματος.

2.3.3 Χαρακτηριστικά των συσκευών δράσης και η συνδεσμολογία τους

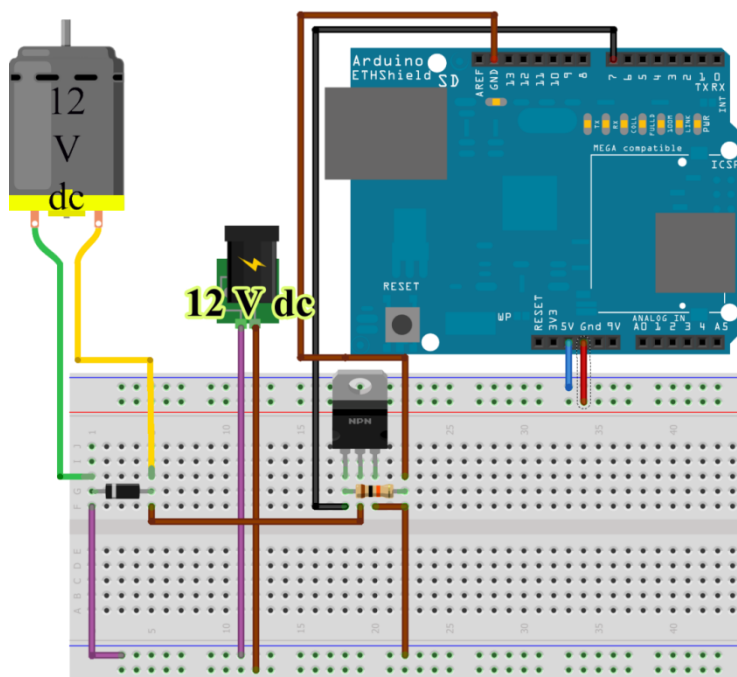
Στον πίνακα 7, αναφέρονται τα χαρακτηριστικά των συσκευών δράσης που χρησιμοποιήθηκαν στο πρότυπο που κατασκευάστηκε. Παρακάτω απεικονίζεται για τη κάθε μία ξεχωριστά, η συνδεσμολογία της.

Χαρακτηριστικά συσκευών δράσης		
Συσκευή:	Τάση Εισόδου:	Ρεύμα Λειτουργίας:
Ανεμιστήρας	12 V dc	0,110 A
Ηλεκτροβάνα	12 V dc	0,180 A
UV LED	12 V dc	0,120 A

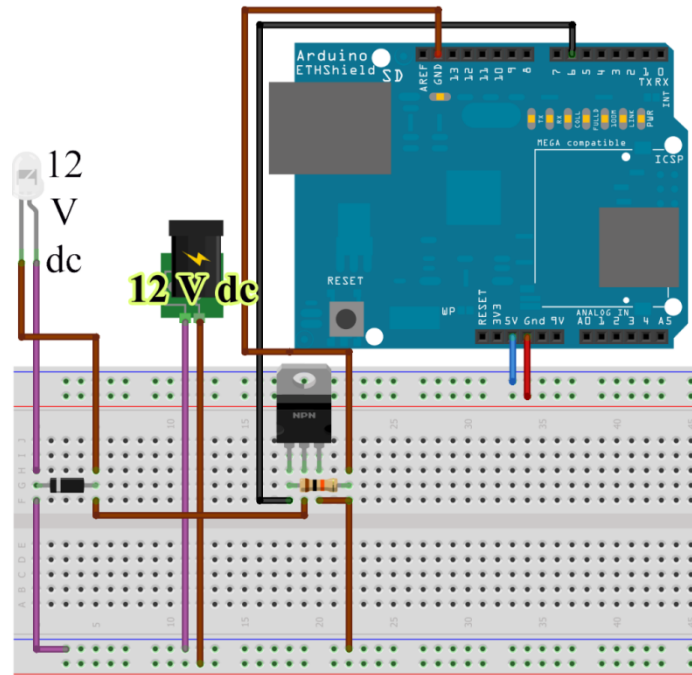
Πίνακας 7: Χαρακτηριστικά συσκευών δράσης



Εικόνα 45: Συνδεσμολογία ηλεκτροβάνας



Εικόνα 46: Συνδεσμολογία ανεμιστήρα



Εικόνα 47: Συνδεσμολογία UV LED

2.4 Μετατροπή κυκλώματος σε ηλεκτρονική πλακέτα

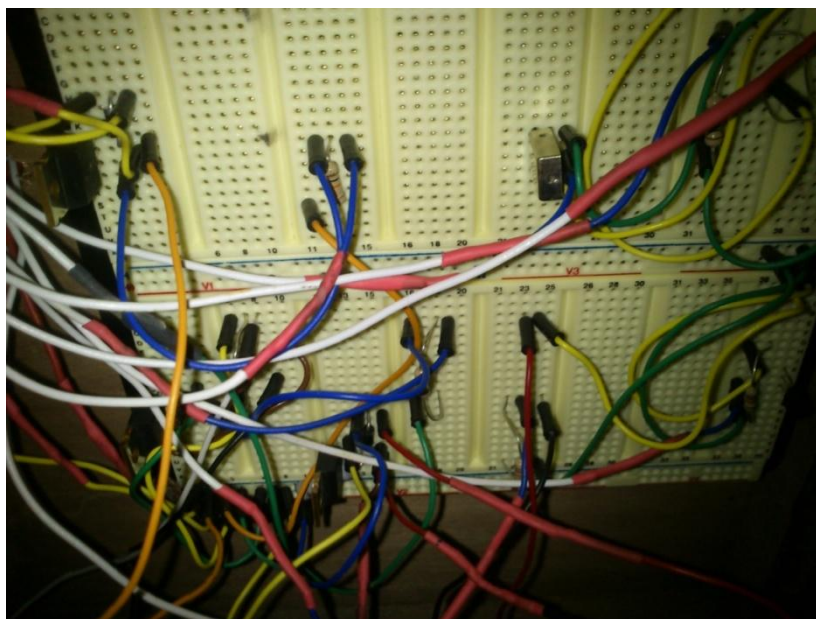
Το κύκλωμα που υλοποιήθηκε για το σύστημα Smart Greenhouse, τελειοποιήθηκε με τη μεταφορά του σε πλακέτα. Τα υλικά τα οποία χρειάστηκαν ήταν τα εξής:

1. Φωτοευαίσθητη πλακέτα μονής όψης. Στην πλακέτα αυτή αποτυπώνεται το κύκλωμα (εικόνα 48).



Εικόνα 48: Φωτοευαίσθητη πλακέτα μονής όψης

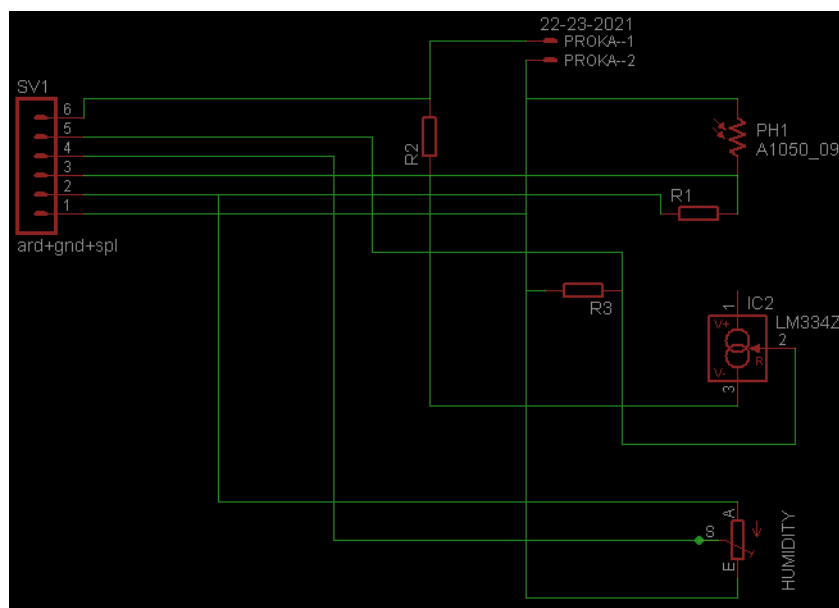
2. Ριζόχαρτο. Το συγκεκριμένο είδος χαρτιού είναι κατάλληλο για την αρχική εκτύπωση του κυκλώματος και τη μεταφορά του στο θάλαμο με τις UV λάμπες.
3. Χημικά:
 - i. Perydrol [27]
 - ii. Tub.O.Flo
 - iii. Υδροχλωρικό οξύ [28]



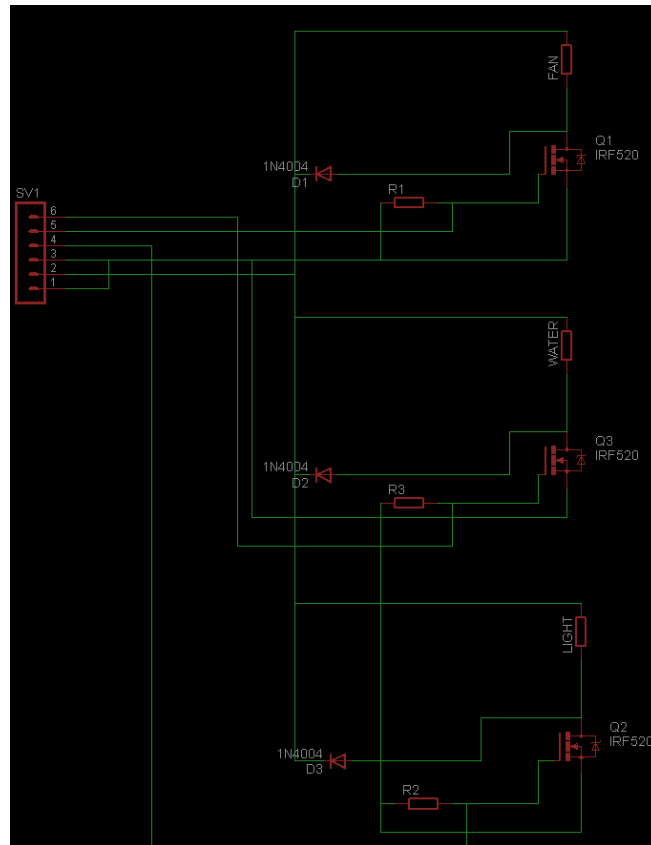
Εικόνα 49: Αρχική μορφή κυκλώματος στο breadboard

Η διαδικασία για την εξαγωγή του κυκλώματος στην πλακέτα είναι η εξής:

1. **Επεξεργασία του κυκλώματος με το πρόγραμμα EAGLE:** Το κύκλωμα χωρίζεται σε δύο μέρη. Το ένα κομμάτι διαχειρίζεται τις συνδέσεις με τους αισθητήρες και τους αναλογικούς ακροδέκτες του Arduino (εικόνα 50), ενώ το δεύτερο κομμάτι διαχειρίζεται τις συσκευές δράσης και αντίστοιχα τους ψηφιακούς ακροδέκτες (εικόνα 51).

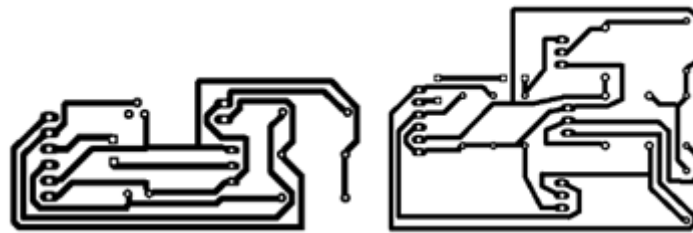


Εικόνα 50: Κύκλωμα αισθητήρων



Εικόνα 51: Κύκλωμα συσκευών δράσης

2. **Εκτύπωση στο ριζόχαρτο:** Η εκτύπωση γίνεται με όσο το δυνατόν περισσότερα dpi (dots per inch). Ο λόγος είναι για να μην υπάρχει κάποιο pixel το οποίο να μην έχει χρωματιστεί καλά γιατί μετά θα δημιουργηθούν προβλήματα στο κύκλωμα (κίνδυνος διαρροών/ βραχυκύκλωμα).



Εικόνα 52: Αποτέλεσμα εκτύπωσης

3. **Τοποθέτηση σε θάλαμο με UV Λάμπες:** Στο θάλαμο εισάγεται το εκτυπωμένο χαρτί τοποθετημένο πάνω από τη φωτοευαίσθητη πλακέτα. Μετά από 45 λεπτά πάνω το κύκλωμα έχει πλέον αποτυπωθεί.
4. **Αποτύπωση σχεδίου στην πλακέτα:** Ακολουθεί η εισαγωγή της φωτοευαίσθητης πλακέτας σε διάλυμα νερού με Tub.O.Flo. Η διάρκεια παραμονής στο μίγμα είναι 5 λεπτά και πλέον αποτυπώνεται καθαρά το κύκλωμα.
5. **Αποχάλκωση:** Τοποθετούμε την πλακέτα μέσα σε μίγμα Pherydrol και υδροχλωρικού οξέως, το οποίο συνεχώς τροφοδοτείται με οξυγόνο. Μετά από 10 λεπτά, η πλακέτα είναι έτοιμη με περιττά κομμάτια χαλκού να έχουν φύγει (εικόνα 53).



Εικόνα 53: Διαδικασία αποχάλκωσης

6. **Κολλήσεις:** Εκτελούνται οι κατάλληλες εργασίες κόλλησης ,με το κολλητήρι, όπως τοποθέτηση αισθητήρων, mosfet, αντιστάσεων, διόδων. Κατόπιν το κύκλωμα είναι έτοιμο και προς λειτουργία (εικόνα 54).



Εικόνα 54: Τελικό αποτέλεσμα

2.5 Κατασκευή Μακέτας Θερμοκηπίου

Κατά την κατασκευή του συστήματος Smart Greenhouse, κατασκευάστηκε και μία μακέτα θερμοκηπίου (εικόνες 55 & 56). Ο λόγος για τον οποίο δημιουργήθηκε ήταν για την πλήρη κατανόηση των συνθηκών και αναγκών που έπρεπε να ληφθούν υπ' όψιν, ώστε η εργασία να ανταποκρίνεται σε πραγματικά δεδομένα.

Το θερμοκήπιο κατασκευάστηκε εξ' ολοκλήρου από ξύλα τα οποία είναι αεροστεγή και αδιάβροχα. Επίσης, χρησιμοποιήθηκε “plexiglass”, ώστε να μπορεί να γίνεται έλεγχος μεταξύ των ενδείξεων του ιστοχώρου και της πραγματικότητας. Οι διαστάσεις του θερμοκηπίου είναι 30 cm μήκος, 20 cm πλάτος, 25 cm ύψος. Το βάρος του ανέρχεται στα 2,5 kg.

Εντός του θερμοκηπίου, εγκαταστάθηκε ένα αποθηκευτικός χώρος νερού για τη λειτουργία της ηλεκτροβάνας, ακριβώς από πάνω της. Η χωρητικότητα του ανέρχεται στα 650mg υγρού. Επειδή η ηλεκτροβάνα δεν έδινε πίεση στο νερό, αλλά ανοιγόκλεινε ένα έμβολο για τη διακοπή ή μη της ροής, έπρεπε η πίεση του νερού να είναι φυσική και αυτό επετεύχθη με τον συγκεκριμένο τρόπο τοποθέτησης της δεξαμενής.



Εικόνα 55: Μακέτα Θερμοκηπίου



Εικόνα 56: Μακέτα θερμοκηπίου εσωτερικό

2.6 Χαρακτηριστικά συσκευής

Στην ενότητα αυτή, αναλύονται τα χαρακτηριστικά της συσκευής. Αυτά διακρίνονται στο συνολικό κόστος κατασκευής (πίνακας 8), αλλά και στην ενεργειακή δαπάνη της συσκευής (πίνακας 9).

Κόστος κατασκευής συσκευής	
Εξαρτήματα :	Τιμή :
Arduino Uno	26,00 €
Arduino ethernet shield	47,00 €
Αισθητήρας Θερμοκρασίας	2,00 €
Αισθητήρας Υγρασίας	25,00 €
Αισθητήρας Φωτισμού	2,00 €
Αισθητήρας Υγρασίας Εδάφους	0,10 €
Ηλεκτρονικός εξοπλισμός	10,00 €
Συνολικό κόστος:	112,10 €

Πίνακας 8: Κόστος κατασκευής συσκευής

Κατανάλωση ενέργειας συσκευής	
Κατανάλωση Συσκευής:	3 Watt
Ημερήσια κατανάλωση συσκευής:	0,072 kWh

Πίνακας 9: Κατανάλωση ενέργειας συσκευής

Κεφάλαιο 3

Λογισμικό μέρος συστήματος

Η λογική με την οποία λειτουργεί το σύστημα “Smart Greenhouse”, αποτελεί το λογισμικό μέρος του συστήματος, που θα αναλυθεί περαιτέρω στο κεφάλαιο αυτό. Ο τρόπος με τον οποίο αλληλεπιδρά το σύστημα με το περιβάλλον στο οποίο είναι εγκαταστημένο μέσω του ιστοχώρου, απαιτεί άμεση και ακριβή εκτέλεση των επιθυμιών του χρήστη. Γι’ αυτό και ο προγραμματισμός των συσκευών και του ιστοχώρου χρειάστηκε περισσότερη χρονική διάρκεια από τη σύνδεση των διαφόρων εξαρτημάτων.

Στο κεφάλαιο αυτό γίνεται αναφορά τόσο στο προγραμματισμό του Arduino όσο και στις λειτουργίες που προσφέρει ο ιστοχώρος. Οι υποενότητες του λογισμικού μέρους του συστήματος είναι οι εξής:

- 1) Περιγραφή απαιτήσεων συστήματος
- 2) Περιγραφή βάσης δεδομένων
- 3) Λειτουργίες ιστοχώρου για διαχειριστή
- 4) Λειτουργίες ιστοχώρου για απλό χρήστη
- 5) Προγραμματισμός αισθητήρων, συσκευών δράσης και Arduino

3.1 Περιγραφή απαιτήσεων συστήματος

Το σύστημα “Smart Greenhouse” αντικαθιστά τη συνεχή παρουσία του κάτοχου του θερμοκηπίου στο χώρο αυτό. Στοχεύει στην παρακολούθηση και καταγραφή των περιβαλλοντικών συνθηκών, καθώς και στον απομακρυσμένο έλεγχο, μέσω του ιστοχώρου που διατίθεται μαζί με τη συσκευή. Για την πλήρη αξιοποίηση του συστήματος έγινε καταγραφή των απαιτήσεων σε ομάδες απλών χρηστών και διαχειριστών.

Ο απλός χρήστης, με το ολοκληρωμένο σύστημα που εκπονήθηκε σε αυτήν τη διπλωματική, έχει την ευκαιρία να παρακολουθεί και να ελέγχει συνθήκες και συσκευές μέσω διαδικτύου. Οι απαιτήσεις οι οποίες έπρεπε να ικανοποιηθούν ήταν οι εξής:

- 1) **Ασφαλής και γρήγορη πρόσβαση:** Λόγω της ευαισθησίας των δεδομένων που διαχειρίζεται το σύστημα, ο χρήστης πρέπει να μπορεί να εισέρχεται στον ιστοχώρο με ασφάλεια, αλλά και γρήγορα. Γι’ αυτό ο κάθε χρήστης έχει τον προσωπικό του κωδικό και η αποθήκευση του κωδικού αυτού στη βάση γίνεται με τη χρήση hash. Με αυτόν τον κωδικό και το email του, εισέρχεται στον ιστοχώρο, όπου έχει πρόσβαση στα δεδομένα που λαμβάνονται από την αντίστοιχη συσκευή.
- 2) **Φιλικό προς το χρήστη περιβάλλον.** Το σύστημα ενδέχεται να χρησιμοποιηθεί και από άτομα με έλλειψη τεχνολογικής εκπαίδευσης. Για αυτόν το λόγο, το σύστημα θα πρέπει να σχεδιασθεί με τέτοιο τρόπο, ώστε να είναι φιλικό προς το χρήστη, να μην του προκαλεί φόβο η χρησιμοποίησή του και να του θυμίζει οικεία εργαλεία και καταστάσεις.
- 3) **Αντικατάσταση ανθρώπινης παρουσίας:** Κύριος στόχος του χρήστη που διαθέτει το σύστημα, είναι η αποφυγή της παρουσίας του συνεχώς στο χώρο του θερμοκηπίου. Γι’ αυτό και δημιουργήθηκαν τα σενάρια. Τα σενάρια ανταποκρίνονται πλήρως στην ανθρώπινη λογική που εφαρμόζεται μέσω των αισθητήρων. Ο χρήστης, λοιπόν, ορίζει τη συμπεριφορά των συσκευών δράσης ανάλογα με το τι συνθήκες επικρατούν στο θερμοκήπιο βάση των δεδομένων των αισθητήρων. Τα σενάρια προγραμματίζονται από το χρήστη να εκτελούνται συγκεκριμένες ώρες και μέρες και με τη μορφή επανάληψης, ανάλογα φυσικά με το τι φυτά έχει σπείρει και διάφορες άλλες γεωπονικής φύσεως παραμέτρους.
- 4) **Άμεσος έλεγχος συσκευών δράσης:** Εκτός από τον προγραμματισμό της συσκευής με τη μορφή σεναρίων, ο χρήστης πρέπει να μπορεί σε πραγματικό χρόνο να ενεργοποιήσει ή να απενεργοποιήσει συσκευές. Γι’ αυτό δημιουργήθηκε το σενάριο manual, το οποίο ελέγχεται μέσω της κεντρικής σελίδας και ανά πάσα στιγμή τίθεται σε ενεργοποίηση από το χρήστη.
- 5) **Καταγραφή συνθηκών και λειτουργίας συσκευών:** Ο χρήστης πρέπει να έχει στη διάθεσή του στατιστικά σχετικά με τις συνθήκες που επικρατούσαν περασμένα χρονικά διαστήματα και το πώς χειρίστηκε τις συσκευές δράσης του θερμοκηπίου. Γι’ αυτό δημιουργήθηκαν τα διαγράμματα, όπου ανάλογα με τον αισθητήρα και τη συσκευή υπάρχουν αντίστοιχα καθημερινά, εβδομαδιαία, μηνιαία και ετήσια.
- 6) **Αλληλεπίδραση με το σύστημα:** Ο χρήστης πρέπει να μπορεί να παρακολουθεί και να ενημερώνεται σε περιπτώσεις μη ομαλής λειτουργίας, χωρίς να εισέρχεται συνέχεια στον ιστοχώρο. Γι’ αυτό ανάλογα με τις ρυθμίσεις που ορίζει, το σύστημα μπορεί να του στέλνει email είτε αναφοράς (με τη χρήση διαγραμμάτων) είτε βλάβης.

- 7) **Επεξεργασία προφίλ χρήστη:** Ο χρήστης πρέπει να μπορεί να αλλάζει email και κωδικό. Η δυνατότητα αυτή υπάρχει στο σύστημα. Επίσης, μπορεί να αλλάζει και γλώσσα (ελληνικά/ αγγλικά).
- 8) **Επικοινωνία με το διαχειριστή:** Ο χρήστης μπορεί να επικοινωνεί με το διαχειριστή του συστήματος για τυχόν απορίες ή προβλήματα. Η δυνατότητα αυτή υπάρχει στον ιστοχώρο.

Για την αντιμετώπιση των απαιτήσεων του χρήστη γίνεται πιο αναλυτικά λόγος στην ενότητα «Λειτουργίες ιστοχώρου για χρήστη». Ακολουθούν οι απαιτήσεις διαχειριστή:

- 1) **Στατιστικά χρήσης:** Ο διαχειριστής χρειάζεται να έχει πρόσβαση σε ποσοστά χρήσης, επίσκεψης και ενεργών χρηστών της σελίδας. Με αυτόν τον τρόπο μπορεί να γνωρίζει για το πόσο δημοφιλής είναι και πόσοι ενεργοί χρήστες υπάρχουν και κατά πόσο χρησιμοποιούν το σύστημα.
- 2) **Δημιουργία, επεξεργασία και διαγραφή χρηστών:** Ο διαχειριστής πρέπει να έχει τη δυνατότητα να δημιουργεί επεξεργάζεται και διαγράφει χρήστες. Επίσης, για ομαλή λειτουργία του ιστοχώρου δε θα πρέπει να έχει γνώση του προσωπικού κωδικού των χρηστών.
- 3) **Καταγραφή σφαλμάτων:** Ο διαχειριστής χρειάζεται να διατηρεί ένα αρχείο με ότι σφάλματα έχουν γίνει στους χρήστες. Επίσης, η διάκριση των σφαλμάτων ανάλογα τους χρήστες, την προέλευση των σφαλμάτων, αλλά και την ημερομηνία, απαιτείται για ταχύτερη ανάγνωση και επεξεργασία.
- 4) **Ανάγνωση μηνυμάτων:** Ο διαχειριστής πρέπει να έχει τη δυνατότητα να διαβάζει τα μηνύματα που του στέλνουν οι χρήστες. Επίσης, θα πρέπει να γνωρίζει ποια έχει διαβάσει και ποια όχι, αλλά φυσικά να μπορεί να τα απαντήσει κιόλας.

Στις παρακάτω ενότητες και πιο συγκεκριμένα στην ενότητα «Λειτουργίες ιστοχώρου για διαχειριστή», γίνεται εκτενή αναφορά στην αντιμετώπιση των απαιτήσεων αυτών. Ακολουθεί η περιγραφή της βάσης δεδομένων του συστήματος.

3.2 Περιγραφή βάσης δεδομένων

Η βάση δεδομένων του ιστοχώρου “Smart Greenhouse”, ονομάζεται “smartgreenhouse” και αποτελείται από 10 πίνακες (εικόνα 57). Η βάση σχεδιάστηκε με τέτοιο τρόπο, ώστε να μπορεί να εξυπηρετήσει ταυτόχρονα πολλούς διαφορετικούς χρήστες. Επίσης, τα δεδομένα που καταγράφονται στη βάση, σαρώνονται διαδοχικά από ελέγχους τόσο μέσω javascript όσο και από αρχεία php. Με αυτόν τον τρόπο, μειώνονται οι πιθανότητες για κακόβουλες επιθέσεις (πχ. sql injection) με σκοπό την αντιγραφή ή αλλοίωση των αποθηκευμένων δεδομένων. Ακολουθεί αναλυτική παρουσίαση κάθε πίνακα ξεχωριστά.

smartgreenhouse.page_statistics visitors : int(11) users : int(11) last_visit : date	smartgreenhouse.devices device_id : int(11) password : varchar(64)	smartgreenhouse.data data_id : int(10) device_id : int(1) dater : date temp : int(2) hum : int(2) inli : int(2) somo : int(1) fan : int(1) watering : int(1) lighting : int(1)	smartgreenhouse.charts chart_id : int(10) device_id : int(5) dater : date filename : varchar(60) folder : int(1)
smartgreenhouse.schedule_scenarios schedule_id : int(1) day : date start_time : time end_time : time scenario_id : int(2) status : int(1) device_id : int(1)	smartgreenhouse.users user_id : int(11) device_id : int(11) email : text password : text userlevel : int(1) last_login : timestamp messages : int(1) live_login : int(1) troubleshoot : int(1) changes : int(1) device_overview : int(1) sensor_overview : int(1) lost : int(1) visits : int(11) active : int(1) private_id : int(5) language : int(1)	smartgreenhouse.troubleshoots troubleshoot_id : int(11) user_id : int(11) dater : date timer : time problem : int(5)	smartgreenhouse.messages message_id : int(3) device_id : int(3) user_id : int(3) subject : varchar(40) message : text dater : datetime r : int(1)
smartgreenhouse.scenarios scenario_id : int(3) scenario_name : varchar(20) device_id : int(3) temp_status : int(1) temp_prior : int(1) temp1 : int(3) temp2 : int(3) hum_status : int(1) hum_prior : int(1) hum1 : int(3) hum2 : int(1) hum_rule : int(2) somo_status : int(1) somo_prior : int(1) somo_limit : int(3) somo_rule : int(2) inli_status : int(1) inli_prior : int(1) inli1 : int(4) inli2 : int(4) inli_rule : int(2) fan : int(1) lighting : int(1) watering : int(1)	smartgreenhouse.new_scenarios scenario_id : int(3) scenario_name : varchar(20) device_id : int(3) temp_status : int(1) temp_prior : int(1) temp1 : int(3) temp2 : int(3) hum_status : int(1) hum_prior : int(1) hum1 : int(3) hum2 : int(1) hum_rule : int(2) somo_status : int(1) somo_prior : int(1) somo_limit : int(3) somo_rule : int(2) inli_status : int(1) inli_prior : int(1) inli1 : int(4) inli2 : int(4) inli_rule : int(2) fan : int(1) lighting : int(1) watering : int(1)		

Εικόνα 57: Βάση δεδομένων – phpmyadmin

1. Πίνακας devices				
Στήλη:	Τύπος:	Σύνθεση:	Κενό:	Προεπιλογή:
Device_id	int(11)	-	Όχι	Καμία
Password	varchar(44)	utf8_general_ci	Όχι	Καμία

Πίνακας 10: Πίνακας devices

Ο παραπάνω πίνακας χρησιμοποιείται για την καταχώρηση των συσκευών και των κωδικών τους στη βάση. Ακολουθεί αναλυτική περιγραφή των πεδίων του:

Device_id: Αποτελεί το πρωτεύον κλειδί του πίνακα. Για κάθε συσκευή που χρησιμοποιείται σε θερμοκήπιο δημιουργείται και ένας ξεχωριστός αριθμός που τη χαρακτηρίζει σε σχέση με τις υπόλοιπες. Αποτελεί πρωτεύον κλειδί του πίνακα.

Password: Για κάθε συσκευή δημιουργείται και το password της που χρησιμοποιείται κατά την επικοινωνία του arduino με τη βάση. Με τον τρόπο αυτό καμία συσκευή που δεν περνάει τον έλεγχο ταυτοποίησης δε μπορεί να επικοινωνήσει με τη βάση. Ο κωδικός αποθηκεύεται με τη μέθοδο κρυπτογράφησης hash [29]. Η παραγωγή αυτού του κωδικού είναι δυναμική και γίνεται με τον παρακάτω κώδικα php:

```
function device_password($device_id)
{
    $device_idd=$device_id;
    $date=strtotime(date("Yfdhms"));
    $date=strrev($date);
    $l=strlen($device_id);
    $l=$device_id*$l;
    $date=$l*$date;
    $a=strlen($date);
    $x="";
    for($i=0;$i<$a;$i++)
    {
        $x=$x+$date[$i];
    }
    $hash=sha1(md5($device_id.$date));
    for($a=0;$a<=$x[5];$a++)
    {
        $hash=sha1($hash);
    }

    $sql="INSERT INTO devices (device_id,password) VALUES ('$device_idd','$hash')";
    @ $result=mysql_query($sql);
    return $result;
}
```

Εικόνα 58: Κώδικας php για device password

Η παραπάνω συνάρτηση παραγωγής κωδικού, προσφέρει μεγάλη ασφάλεια στο σύστημα, καθώς η δημιουργία του, είναι δυναμική. Παράμετροι όπως η ώρα και η ημερομηνία που δημιουργείται, αλλά και το device_id λαμβάνονται υπ' όψιν για τη δημιουργία. Η διαδικασία αυτή εκτελείται μόλις ο διαχειριστής εισάγει για πρώτη φορά ένα χρήστη στο σύστημα.

2. Πίνακας page_statistics				
Στήλη:	Τύπος:	Σύνθεση:	Κενό:	Προεπιλογή:
Visitors	int(11)	-	Όχι	Καμία
Users	int(11)	-	Όχι	Καμία
Last_visit	date	-	Όχι	Καμία

Πίνακας 11: Πίνακας page_statistics

Ο παραπάνω πίνακας χρησιμοποιείται για την αποθήκευση στατιστικών όσον αφορά τη χρήση και τις επισκέψεις του ιστοχώρου. Είναι βασικό εργαλείο για την εξαγωγή συμπερασμάτων από το διαχειριστή. Παρακάτω ακολουθεί αναλυτική περιγραφή:

Visitors: Το σύνολο των επισκέψεων της σελίδας. Κάθε φορά που ένας browser φορτώνει τη σελίδα αυξάνεται κατά 1 το πεδίο των visitors.

Users: Το σύνολο των ενεργών χρηστών, δηλαδή των χρηστών που έχουν συσκευές. Με την δημιουργία ενός νέου χρήστη αυξάνεται και το πεδίο.

Last_visit: Καταγράφει την τελευταία χρονολογικά επίσκεψη.

3. Πίνακας users				
Στήλη:	Τύπος:	Σύνθεση:	Κενό:	Προεπιλογή:
User_id	int(11)	-	Όχι	Καμία
Device_id	int(11)	-	Όχι	Καμία
Email	Text	Utf8_unicode_ci	Όχι	Καμία
Password	Text	Utf8_unicode_ci	Όχι	Καμία
Userlevel	int(1)	-	Όχι	Καμία
Last_login	timestamp	-	Όχι	Καμία
Messages	int(1)	-	Όχι	Καμία
Live_login	int(11)	-	Όχι	Καμία
Troubleshoot	int(1)	-	Όχι	Καμία
Changes	int(1)	-	Όχι	Καμία
Device_overview	int(1)	-	Όχι	Καμία
Sensor_overview	int(1)	-	Όχι	Καμία
Lost	int(1)	-	Όχι	Καμία
Visits	int(11)	-	Όχι	Καμία
Active	int(1)	-	Όχι	Καμία
Private_id	int(5)	-	Όχι	Καμία
Language	int(1)	-	Όχι	Καμία

Πίνακας 12: Πίνακας users

Στον πίνακα users αποθηκεύονται οι χρήστες και τα διάφορα χαρακτηριστικά προφίλ τους. Τα πεδία ερμηνεύονται ως εξής:

User_id: Μοναδικός αριθμός που χαρακτηρίζει το χρήστη σε σχέση με τους υπόλοιπους. Κάθε φορά που εισάγεται ένας χρήστης αυξάνεται ο αριθμός αυτός. Αποτελεί το πρωτεύον κλειδί του πίνακα.

Device_id: Ο κάθε χρήστης συσχετίζεται με τη συσκευή του, μέσω του αναγνωριστικού της που αναλύθηκε παραπάνω.

Email: Για να εισαχθεί ο χρήστης στο σύστημα χρειάζεται η ηλεκτρονική του διεύθυνση. Με αυτήν πραγματοποιεί είσοδο, αλλά και εγγραφή. Μπορεί είτε ο ίδιος είτε ο διαχειριστής να την επεξεργαστούν.

Password: Αποθηκεύεται ο προσωπικός κωδικός του χρήστη με τη μέθοδο κρυπτογράφησης hash. Με αυτόν τον τρόπο και να παραβιαστεί η βάση, κανείς δε θα είναι σε θέση να γνωρίζει τον πραγματικό κωδικό του χρήστη, προσφέροντας ασφάλεια στα δεδομένα του ακόμα και σε καταστάσεις παθογένειας.

Userlevel: Φανερώνει το επίπεδο του χρήστη. Η τιμή 1 είναι για τους χρήστες ενώ η τιμή 0 για τους διαχειριστές.

Last_login: Αποθηκεύει την τελευταία φορά που ο χρήστης του συστήματος πραγματοποίησε είσοδο.

Messages: Αποθηκεύει τον αριθμό των μηνυμάτων που έχει στείλει ο χρήστης στο διαχειριστή. Με την αποστολή ενός μηνύματος, αυξάνεται αυτόματα το πεδίο αυτό.

Live_login: Το πεδίο αυτό χρησιμοποιείται, ώστε κάθε φορά να έχει πρόσβαση μόνο ένας χρήστης στον προσωπικό του λογαριασμό, δηλαδή αν δυο άτομα μοιράζονται τον κωδικό ενός χρήστη μόνο ένας μπορεί να επισκεφθεί το προφίλ του.

Troubleshoot: Το πεδίο αυτό δηλώνει την επιθυμία του χρήστη να λαμβάνει email όταν διαπιστωθεί ότι κάποιος αισθητήρας βρίσκεται εκτός λειτουργίας. Η τιμή 1 δηλώνει αποδοχή αυτής της ρύθμισης ενώ η τιμή 0 το αντίθετο.

Changes: Το πεδίο αυτό δηλώνει την επιθυμία του χρήστη να λαμβάνει email όταν διαπιστωθεί ξαφνική ατμοσφαιρική αλλαγή. Η τιμή 1 δηλώνει αποδοχή αυτής της ρύθμισης ενώ η τιμή 0 το αντίθετο.

Device_overview: Το πεδίο αυτό δηλώνει την επιθυμία του χρήστη να λαμβάνει email αναφοράς για τη χρήση των συσκευών δράσης. Η τιμή 0 δηλώνει την άρνηση αποδοχής αναφοράς, η τιμή 1 την αποστολή αναφοράς κάθε μέρα, η τιμή 2 κάθε εβδομάδα, η τιμή 3 κάθε μήνα και η τιμή 4 κάθε χρόνο. Η αποστολή των email γίνεται με τη χρήση crontabs [30].

Sensor_overview: Το πεδίο αυτό δηλώνει την επιθυμία του χρήστη να λαμβάνει email αναφοράς για τα δεδομένα των αισθητήρων. Η τιμή 0 δηλώνει την άρνηση αποδοχής αναφοράς, η τιμή 1 την αποστολή αναφοράς κάθε μέρα, η τιμή 2 κάθε εβδομάδα, η τιμή 3 κάθε μήνα και η τιμή 4 κάθε χρόνο. Η αποστολή των email γίνεται με την χρήση crontabs [30].

Lost: Το πεδίο αυτό δηλώνει την επιθυμία του χρήστη να λαμβάνει email, όταν διαπιστωθεί ότι δεν υπάρχει επικοινωνία με τη συσκευή. Η τιμή 1 δηλώνει αποδοχή αυτής της ρύθμισης ενώ η τιμή 0 το αντίθετο.

Visits: Αποτελεί το μετρητή των επισκέψεων που έχει πραγματοποιήσει ο χρήστης.

Active: Για να χρησιμοποιήσει ο χρήστης το λογαριασμό του θα πρέπει να έχει ακολουθήσει τη διαδικασία εγγραφής με τη χρήση του πεδίου private_id. Η τιμή 0 δηλώνει τη μη ολοκλήρωση εγγραφής ενώ η τιμή 1 δηλώνει την ενεργοποίηση του χρήστη.

Private_id: Χρησιμοποιείται κατά την εγγραφή ενός χρήστη από το διαχειριστή. Πρόκειται για έναν τυχαίο αριθμό ο οποίος προστίθεται στο τέλος ενός link (διεύθυνση ιστοχώρου) και όταν ο χρήστης το φορτώσει ενεργοποιείται ο λογαριασμός του (πεδίο active: τιμή 1).

Language: Το πεδίο αυτό αποθηκεύει την προτίμηση του χρήστη για τη γλώσσα λειτουργίας του ιστοχώρου. Η τιμή 1 δηλώνει την εμφάνιση στα ελληνικά ενώ η τιμή 2 την εμφάνιση στα αγγλικά.

4. Πίνακας data				
Στήλη:	Τύπος:	Σύνθεση:	Κενό:	Προεπιλογή:
Data_id	int(10)	-	Όχι	Καμία
Device_id	int(11)	-	Όχι	Καμία
Dater	date	-	Όχι	Καμία
Timer	time	-	Όχι	Καμία
Temp	int(2)	-	Όχι	Καμία
Hum	int(2)	-	Όχι	Καμία
Inli	int(3)	-	Όχι	Καμία
Somo	int(1)	-	Όχι	Καμία
Fan	int(1)	-	Όχι	Καμία
Watering	int(1)	-	Όχι	Καμία
Lighting	int(1)	-	Όχι	Καμία

Πίνακας 13: Πίνακας data

Ο πίνακας data χρησιμοποιείται για την αποθήκευση των δεδομένων. Λόγω της σημασίας του πίνακα αυτού τα δεδομένα περνάνε από ενδεδειγμένους ελέγχους ασφάλειας ως προς το περιεχόμενο, ελέγχους ως προς τη χρονική συνέπεια και τις μεταβολές στις ατμοσφαιρικές παραμέτρους. Επίσης, για εξοικονόμηση χώρου και εύρυθμη λειτουργία ο πίνακας εξάγει ωριαίους μέσους όρους. Παρακάτω αναλύονται τα πεδία:

Data_id: Αποτελεί το αναγνωριστικό κάθε εγγραφής. Επίσης, είναι το πρωτεύον κλειδί του πίνακα.

Device_id: Φανερώνει σε ποια συσκευή αντιστοιχεί η εγγραφή.

Dater: Δηλώνει την ημερομηνία που αποθηκεύεται στη βάση η εγγραφή. Στα παρακάτω κεφάλαια αναλύεται πως εκτελείται ο έλεγχος χρονικής συνέπειας κατά την επικοινωνία arduino και βάσης.

Timer: Δηλώνει την ώρα που αποθηκεύεται στη βάση η εγγραφή. Στα παρακάτω κεφάλαια αναλύεται πως εκτελείται ο έλεγχος χρονικής συνέπειας κατά την επικοινωνία arduino και βάσης.

Temp: Αποθηκεύει την τιμή της θερμοκρασίας.

Hum: Αποθηκεύει την τιμή της υγρασίας.

Inli: Αποθηκεύει την τιμή της έντασης του φωτός.

Somo: Αποθηκεύει την κατάσταση του εδάφους. Η τιμή 1 δηλώνει υγρό έδαφος ενώ η τιμή 0 ξηρό έδαφος.

Fan: Αποθηκεύει την κατάσταση λειτουργίας του ανεμιστήρα. Η τιμή 1 δηλώνει ενεργή συσκευή ενώ η τιμή 0 ανενεργή.

Watering: Αποθηκεύει την κατάσταση λειτουργίας της ηλεκτροβάνας. Η τιμή 1 δηλώνει ενεργή συσκευή ενώ η τιμή 0 ανενεργή.

Lighting: Αποθηκεύει την κατάσταση λειτουργίας του UV LED. Η τιμή 1 δηλώνει ενεργή συσκευή ενώ η τιμή 0 ανενεργή.

5. Πίνακας scenarios – new_scenarios				
Στήλη:	Τύπος:	Σύνθεση:	Κενό:	Προεπιλογή:
Scenario_id	int(3)	-	Όχι	Καμία

Scenario_name	varchar(20)	utf8_unicode_ci	Όχι	Καμία
Device_id	int(11)	-	Όχι	Καμία
Temp_status	int(1)	-	Όχι	Καμία
Temp_prior	int(1)	-	Όχι	Καμία
Temp1	int(3)	-	Όχι	Καμία
Temp2	int(3)	-	Όχι	Καμία
Temp_rule	int(2)	-	Όχι	Καμία
Hum_status	int(1)	-	Όχι	Καμία
Hum_prior	int(1)	-	Όχι	Καμία
Hum1	int(3)	-	Όχι	Καμία
Hum2	int(3)	-	Όχι	Καμία
Hum_rule	int(2)	-	Όχι	Καμία
Somo_status	int(1)	-	Όχι	Καμία
Som_prior	int(1)	-	Όχι	Καμία
Somo_limit	int(3)	-	Όχι	Καμία
Somo_rule	int(2)	-	Όχι	Καμία
Inli_status	int(1)	-	Όχι	Καμία
Inli_prior	int(1)	-	Όχι	Καμία
Inli1	int(3)	-	Όχι	Καμία
Inli2	int(3)	-	Όχι	Καμία
Inli_rule	int(2)	-	Όχι	Καμία
Fan	int(1)	-	Όχι	Καμία
Lighting	int(1)	-	Όχι	Καμία
Watering	int(1)	-	Όχι	Καμία

Πίνακας 14: Πίνακας scenarios - new_scenarios

Οι πίνακες scenarios και new_scenarios, οι οποίοι απεικονίζονται παραπάνω, είναι άμεσα συνδεδεμένοι μεταξύ τους. Τα πεδία τους και η δομή τους είναι ακριβώς ίδια γι' αυτό και δε χρησιμοποιήθηκαν δύο ξεχωριστοί πίνακες για την αναπαράσταση τους. Υπάρχει όμως μία διαφορά στη χρήση τους. Ο πίνακας new_scenarios χρησιμοποιείται κατά τη δημιουργία ή επεξεργασία ενός σεναρίου. Μόλις ο χρήστης επιλέξει αποθήκευση, τότε το σενάριο μεταφέρεται στον πίνακα scenarios και εκκαθαρίζεται ο άλλος πίνακας. Αυτό συμβαίνει διότι υπάρχει η περίπτωση ο χρήστης να μην αποθηκεύσει κατά τη διάρκεια επεξεργασίας ή δημιουργίας το σενάριο με απόρροια την εσφαλμένη εγγραφή του και την περαιτέρω σύγχυσή του ιδίου, αλλά και της βάσης εν συνεχεία. Παρακάτω ακολουθεί ανάλυση των πεδίων τους:

Scenario_id: Αναγνωριστικό σεναρίου. Αποτελεί πρωτεύον κλειδί του πίνακα.

Scenario_name: Το όνομα που δίνει ο χρήστης στο σενάριο.

Device_id: Φανερώνει από ποια συσκευή μπορεί να χρησιμοποιηθεί το συγκεκριμένο σενάριο.

Temp_status: Το πεδίο αυτό δηλώνει αν στο σενάριο υπάρχουν προς εξέταση συνθήκες θερμοκρασίας. Η τιμή 0 δηλώνει την απουσία του αισθητήρα ενώ η τιμή 1 δηλώνει την παρουσία του.

Temp_prior: Στην περίπτωση όπου ο αισθητήρας θερμοκρασίας είναι προς εξέταση, τότε δηλώνει τη σειρά με την οποία το συμπεριέλαβε ο χρήστης. Διαφορετικά η τιμή είναι 0.

Temp1: Κάτω όριο θερμοκρασίας.

Temp2: Άνω όριο θερμοκρασίας.

Temp_rule: Ο κανόνας που θα γίνει η λογική πράξη με τον επόμενο αισθητήρα. Η τιμή 0 δηλώνει την απουσία επόμενου αισθητήρα, η τιμή 1 τη λογική πράξη OR και η τιμή 2 τη λογική πράξη AND.

Hum_status: Το πεδίο αυτό δηλώνει αν στο σενάριο υπάρχουν προς εξέταση συνθήκες υγρασίας. Η τιμή 0 δηλώνει την απουσία του αισθητήρα ενώ η τιμή 1 δηλώνει την παρουσία του.

Hum_prior: Στην περίπτωση όπου ο αισθητήρας υγρασίας είναι προς εξέταση, τότε δηλώνει τη σειρά με την οποία το συμπεριέλαβε ο χρήστης. Διαφορετικά η τιμή είναι 0.

Hum1: Κάτω όριο υγρασίας.

Hum2: Άνω όριο υγρασίας.

Hum_rule: Ο κανόνας που θα γίνει η λογική πράξη με τον επόμενο αισθητήρα. Η τιμή 0 δηλώνει την απουσία επόμενου αισθητήρα, η τιμή 1 τη λογική πράξη OR και η τιμή 2 τη λογική πράξη AND.

Somo_status: Το πεδίο αυτό δηλώνει αν στο σενάριο υπάρχουν προς εξέταση συνθήκες υγρασίας εδάφους. Η τιμή 0 δηλώνει την απουσία του αισθητήρα ενώ η τιμή 1 δηλώνει την παρουσία του.

Somo_prior: Στην περίπτωση όπου ο αισθητήρας υγρασίας εδάφους είναι προς εξέταση, τότε δηλώνει τη σειρά με την οποία το συμπεριέλαβε ο χρήστης. Διαφορετικά η τιμή είναι 0.

Somo_limit: Δηλώνει την κατάσταση του εδάφους. Η τιμή 1 αναφέρεται σε υγρό έδαφος ενώ η τιμή 0 σε ξηρό.

Somo_rule: Ο κανόνας που θα γίνει η λογική πράξη με τον επόμενο αισθητήρα. Η τιμή 0 δηλώνει την απουσία επόμενου αισθητήρα, η τιμή 1 τη λογική πράξη OR και η τιμή 2 τη λογική πράξη AND.

Inli_status: Το πεδίο αυτό δηλώνει αν στο σενάριο υπάρχουν προς εξέταση συνθήκες έντασης φωτός. Η τιμή 0 δηλώνει την απουσία του αισθητήρα ενώ η τιμή 1 δηλώνει την παρουσία του.

Inli_prior: Στην περίπτωση όπου ο αισθητήρας φωτισμού είναι προς εξέταση, τότε δηλώνει τη σειρά με την οποία το συμπεριέλαβε ο χρήστης. Διαφορετικά η τιμή είναι 0.

Inli1: Κάτω όριο έντασης φωτός.

Inli2: Άνω όριο έντασης φωτός.

Inli_rule: Ο κανόνας που θα γίνει η λογική πράξη με τον επόμενο αισθητήρα. Η τιμή 0 δηλώνει την απουσία επόμενου αισθητήρα, η τιμή 1 τη λογική πράξη OR και η τιμή 2 τη λογική πράξη AND.

Fan: Κατάσταση λειτουργίας ανεμιστήρα. Η τιμή 0 δηλώνει απενεργοποιημένη συσκευή ενώ η τιμή 1 ενεργοποιημένη.

Lighting: Κατάσταση λειτουργίας UV LED. Η τιμή 0 δηλώνει απενεργοποιημένη συσκευή ενώ η τιμή 1 ενεργοποιημένη.

Watering: Κατάσταση λειτουργίας ηλεκτροβάνας. Η τιμή 0 δηλώνει απενεργοποιημένη συσκευή ενώ η τιμή 1 ενεργοποιημένη.

6. Πίνακας schedule_scenarios

Στήλη:	Τύπος:	Σύνθεση:	Κενό:	Προεπιλογή:
Schedule_id	int(10)	-	Όχι	Καμία
Day	date	-	Όχι	Καμία
Start_time	time	-	Όχι	Καμία
End_time	time	-	Όχι	Καμία
Scenario_id	int(3)	-	Όχι	Καμία

Status	int(1)	-	Όχι	Καμία
Device_id	int(11)	-	Όχι	Καμία

Πίνακας 15: Πίνακας schedule_scenarios

Ο πίνακας schedule_scenarios χρησιμοποιείται για την αποθήκευση του προγραμματισμού των σεναρίων. Καταχωρήσεις που έχουν εκτελεστεί 3 μήνες πριν εκκαθαρίζονται αυτόματα με τη χρήση crontabs [30]. Η λειτουργία των πεδίων του είναι η εξής:

Schedule_id: Αναγνωριστικός αριθμός καταχώρησης προγραμματισμού σεναρίου. Αποτελεί πρωτεύον κλειδί του πίνακα.

Day: Ορίζει την ημερομηνία για την οποία είναι προγραμματισμένο το σενάριο.

Start_time: Ορίζει την ώρα έναρξης εκτέλεσης του σεναρίου.

End_time: Ορίζει την ώρα τερματισμού εκτέλεσης του σεναρίου.

Scenario_id: Ορίζει το αναγνωριστικό του σεναρίου που πρόκειται να εκτελεσθεί.

Status: Ορίζει αν το σενάριο είναι ενεργό (τιμή 1) ή όχι (τιμή 0). Ουσιαστικά, ο χρήστης μπορεί να προγραμματίσει το σενάριο, αλλά να μη θελήσει να εκτελεστεί.

Device_id: Φανερώνει σε ποια συσκευή αναφέρεται ο συγκεκριμένος προγραμματισμός.

7. Πίνακας charts				
Στήλη:	Τύπος:	Σύνθεση:	Κενό:	Προεπιλογή:
Chart_id	int(10)	-	Όχι	Καμία
Device_id	int(3)	-	Όχι	Καμία
Dater	Date	-	Όχι	Καμία
Filename	varchar(60)	utf8_unicode_ci	Όχι	Καμία
Folder	int(1)	-	Όχι	Καμία

Πίνακας 16: Πίνακας charts

Ο πίνακας charts χρησιμοποιείται για καταγραφή των διαγραμμάτων που έχουν δημιουργηθεί. Στην περίπτωση που ο χρήστης επιθυμεί την εμφάνιση ενός διαγράμματος που έχει ήδη δημιουργηθεί μία φορά δε χρειάζεται να επανασχεδιαστεί, αλλά απλά καλείται η διαδρομή αρχείου του. Επίσης, με τη χρήση crontabs [30] ο πίνακας εκκαθαρίζεται από διαγράμματα με ημερομηνία δημιουργίας 3 μηνών πριν. Η περιγραφή των πεδίων του συγκεκριμένου πίνακα έχει ως εξής:

Chart_id: Αναγνωριστικό καταχώρησης στον πίνακα. Αποτελεί πρωτεύον κλειδί του πίνακα.

Device_id: Φανερώνει σε ποια συσκευή αναφέρονται τα διαγράμματα.

Dater: Ημερομηνία δημιουργίας του διαγράμματος.

Filename: Το όνομα του αρχείου που δημιουργήθηκε.

Folder: Ο φάκελος που περιέχει το αρχείο. Η τιμή 1 είναι για το φάκελο sensors (διαγράμματα αισθητήρων) ενώ η τιμή 2 για το φάκελο devices (διαγράμματα συσκευών).

8. Πίνακας messages				
Στήλη:	Τύπος:	Σύνθεση:	Κενό:	Προεπιλογή:
Message_id	int(3)	-	Όχι	Καμία
Device_id	int(3)	-	Όχι	Καμία

User_id	int(3)	-	Όχι	Καμία
Subject	varchar(60)	utf8_unicode_ci	Όχι	Καμία
Message	text	utf8_unicode_ci	Όχι	Καμία
Dater	datetime	-	Όχι	Καμία
r	int(1)	-	Όχι	Καμία

Πίνακας 17: Πίνακας messages

Ο πίνακας messages αποθηκεύει τα μηνύματα που στέλνουν οι απλοί χρήστες στους διαχειριστές. Παρακάτω ακολουθεί η ανάλυση των πεδίων του:

Message_id: Αναγνωριστικό καταχώρησης στον πίνακα. Αποτελεί πρωτεύον κλειδί του πίνακα.

Device_id: Φανερώνει σε ποια συσκευή αναφέρονται τα μηνύματα.

User_id: Αντιστοιχεί σε χρήστη το μήνυμα.

Subject: Καταχωρεί το θέμα του μηνύματος.

Message: Καταχωρεί το κείμενο του μηνύματος.

Dater: Καταχωρεί την ημερομηνία και την ώρα αποστολής.

r: Δείκτης που δηλώνει την ανάγνωση ή μη, του μηνύματος από το διαχειριστή.

9. Πίνακας troubleshoots				
Στήλη:	Τύπος:	Σύνθεση:	Κενό:	Προεπιλογή:
Troubleshoot_id	int(3)	-	Όχι	Καμία
User_id	int(3)	-	Όχι	Καμία
Dater	date	-	Όχι	Καμία
Timer	time	-	Όχι	Καμία
Problem	int(5)	-	Όχι	Καμία

Πίνακας 18: Πίνακας troubleshoots

Ο πίνακας troubleshoots χρησιμοποιείται για την καταχώρηση τυχών σφαλμάτων κατά τη λειτουργία του συστήματος. Παρακάτω γίνεται ανάλυση των πεδίων του:

Troubleshoot_id: Αποτελεί το αναγνωριστικό κάθε εγγραφής σφάλματος. Αποτελεί πρωτεύον κλειδί του πίνακα.

User_id: Φανερώνει σε ποιο χρήστη αντιστοιχεί το σφάλμα.

Dater: Δηλώνει την ημερομηνία που αποθηκεύεται στη βάση η εγγραφή.

Timer: Δηλώνει την ώρα που αποθηκεύεται στη βάση η εγγραφή.

Problem: Δηλώνει τον τύπο του προβλήματος. Ανάλογα με την τιμή που περιέχει, ορίζεται διαφορετικό πρόβλημα ως εξής:

- 1) Τιμή 1: Πρόβλημα λειτουργίας αισθητήρα θερμοκρασίας.
- 2) Τιμή 2: Πρόβλημα λειτουργίας αισθητήρα υγρασίας.
- 3) Τιμή 3: Πρόβλημα λειτουργίας αισθητήρα υγρασίας χώματος.
- 4) Τιμή 7: Ανεπιτυχής αποστολή email σφαλμάτων.
- 5) Τιμή 8: Ανεπιτυχής αποστολή email αναφορών.
- 6) Τιμή 11: Ξαφνική ατμοσφαιρική αλλαγή στον αισθητήρα θερμοκρασίας.
- 7) Τιμή 12: Ξαφνική ατμοσφαιρική αλλαγή στον αισθητήρα υγρασίας.
- 8) Τιμή 13: Ξαφνική ατμοσφαιρική αλλαγή στον αισθητήρα έντασης φωτός.
- 9) Τιμή 15: Διάγνωση προβλήματος από το αρχείο php_report_15_minutes.php (αρχείο που τρέχει κάθε 15 λεπτά και ελέγχει την ομαλή λειτουργία του συστήματος).
- 10) Τιμή x0: Διακοπή επικοινωνίας για x μέρες.

11) Τιμή x9: Διακοπή επικοινωνίας για x λεπτά.

3.3 Λειτουργίες ιστοχώρου για διαχειριστή

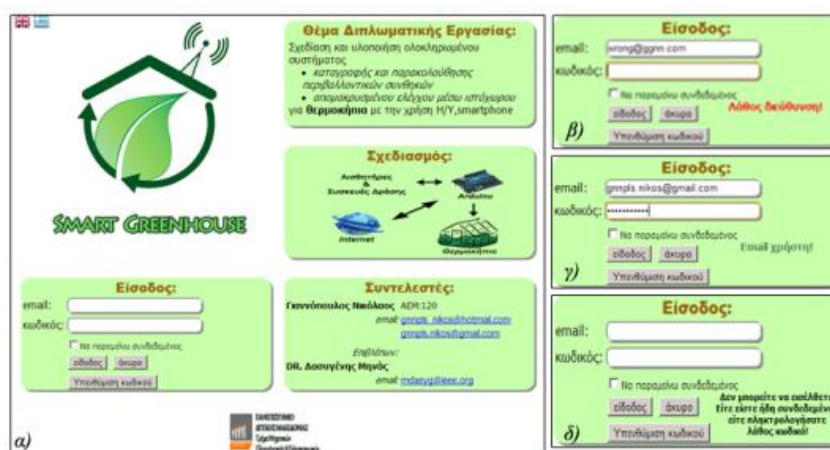
Στην ενότητα αυτή θα γίνει αναφορά στις λειτουργίες που προσφέρει το σύστημα στους διαχειριστές του συστήματος. Η κατασκευή αυτού του τμήματος του ιστοχώρου, έγινε με σκοπό την ικανοποίηση των απαιτήσεων της εν λόγω κατηγορίας χρηστών, που αναλύθηκαν παραπάνω. Ο ρόλος των διαχειριστών είναι βαρύνουσας σημασίας, καθώς είναι υπεύθυνοι για την εύρυθμη λειτουργία του ιστοχώρου, αλλά και την άμεση λύση τυχών προβλημάτων που προκύπτουν στους απλούς χρήστες. Στις παρακάτω υποενότητες γίνεται εκτενή αναφορά στις λειτουργίες που προσφέρονται στο διαχειριστή.

Ο ιστοχώρος του διαχειριστή έχει κατασκευαστεί εξ' ολοκλήρου με τη χρήση AJAX. Με τη μέθοδο αυτή, ο ιστοχώρος φορτώνεται από τον browser μία φορά και η επικοινωνία με τη βάση γίνεται δυναμικά, χωρίς να απαιτείται επανάληψη της φόρτωσης.

3.3.1 Είσοδος διαχειριστή στον ιστοχώρο

Η εισαγωγή στον ιστοχώρο για το διαχειριστή γίνεται μέσω της κεντρικής σελίδας (εικόνα 59α). Η διαδικασία που ακολουθεί είναι κοινή και για τον απλό χρήστη, ενώ μπορεί να γίνει είτε με χρήση της ελληνικής γλώσσας είτε με χρήση της αγγλικής. Ακολουθεί αναλυτική περιγραφή:

Στο τμήμα «Είσοδος» πληκτρολογείται το email και ο προσωπικός κωδικός. Αν δεν υπάρχει το email χρήστη, τότε εμφανίζεται άμεσα μήνυμα λάθους, χωρίς ο χρήστης να στείλει τα δεδομένα του στη βάση (εικόνα 59 β). Σε αντίθετη περίπτωση εμφανίζεται μήνυμα επιβεβαίωσης (εικόνα 59 γ). Η συμπλήρωση του πεδίου αυτού ενέχει ελέγχους περιεχομένου για να αποφευχθούν τυχόν κακόβουλες επιθέσεις. Μόλις ο χρήστης εισάγει τον προσωπικό του κωδικό και επιλέξει «είσοδος», τότε αν ο κωδικός του είναι ο σωστός φορτώνεται η αρχική σελίδα (είτε για το χρήστη είτε για το διαχειριστή). Σε αντίθετη περίπτωση ή σε περίπτωση που κάποιος άλλος είναι συνδεδεμένος με τον ίδιο λογαριασμό, εμφανίζεται μήνυμα λάθους (εικόνα 59 δ). Επίσης, δίνεται η δυνατότητα στο χρήστη να επιλέξει τη χρήση των cookies μέσω της επιλογής «Να παραμείνω συνδεδεμένος», ώστε να μην επαναλαμβάνει την παραπάνω διαδικασία συνεχώς.



The image shows four examples of the login form (Εικόνα 59) for the Smart Greenhouse system. Each example includes a header with the logo and navigation links, and a main content area with a login form and a sidebar with contact information.

α) Main Page: The login form has fields for email (john@xyz.com) and password (12345678). The sidebar lists contact information for the system administrator.

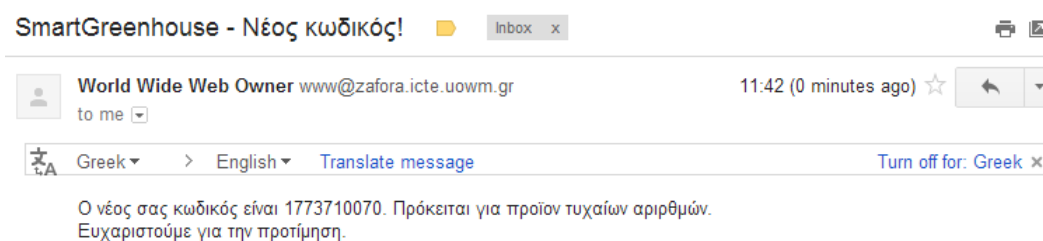
β) Error Message: The login form shows an error message: "Λάθος δειγματοληψία" (Incorrect sampling).

γ) Success Message: The login form shows a success message: "Επιβεβαίωση" (Confirmation).

δ) Error Message: The login form shows an error message: "Λάθος κωδικός" (Incorrect password).

Εικόνα 59: Είσοδος χρηστών

Υπάρχει η πιθανότητα ο χρήστης να ξεχάσει τον κωδικό πρόσβασής του. Στην περίπτωση αυτή με τη χρήση της επιλογής «Υπενθύμιση κωδικού», στέλνεται στο προσωπικό email είτε του χρήστη είτε του διαχειριστή, ένας τυχαία παραγόμενος αριθμός, για να επιτευχθεί η πρόσβαση (εικόνα 60). Ο αριθμός αυτός, όπως και γενικά το περιεχόμενο του κωδικού των χρηστών, δεν είναι εις γνώση του διαχειριστή.

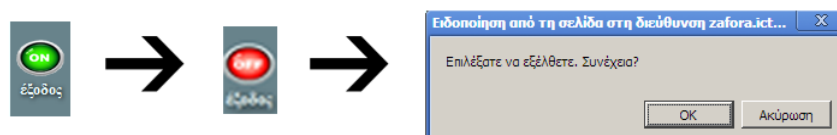


Εικόνα 60: Υπενθύμιση κωδικού

Οι έλεγχοι ως προς περιεχόμενο στο πεδίο «Email», που αναφέρονται παραπάνω γίνονται μέσω του αρχείου “email_validate.php”. Εφόσον η μεταβλητή του πεδίου περάσει από τον έλεγχο ως προς τη μορφή με τη χρήση της συνάρτησης (!preg_match("/^[_\\0-9a-zA-Z-]+@([0-9a-zA-Z][0-9a-zA-Z-]+\\.)+[a-zA-Z]{2,6}\$/i", \$email)), έπειτα ελέγχεται αν υπάρχει στη βάση. Ανάλογα εμφανίζονται τα μηνύματα που απεικονίζονται στις εικόνες 59 β) και 59 γ). Ο έλεγχος γίνεται με τη χρήση AJAX. Επίσης, καθώς ο χρήστης επιλέξει την «Είσοδο» οι έλεγχοι αυτοί επαναλαμβάνονται ξανά στο αρχείο “password_check.php” ώστε να μειωθεί η πιθανότητα ακατάλληλου περιεχομένου σε ερώτημα που θα γίνει στη βάση.

3.3.2 Έξοδος διαχειριστή από τον ιστοχώρο

Η διαδικασία εξόδου από τον ιστόχωρο είναι κοινή τόσο για το διαχειριστή όσο και τον απλό χρήστη. Σε οποιοδήποτε σημείο του ιστοχώρου και να βρίσκονται και οι δύο αυτές κατηγορίες χρηστών, πάντα πάνω δεξιά υπάρχει το εικονίδιο με την ένδειξη “ON” με την υποσημείωση «έξοδος» (ή sign out, αν η γλώσσα εμφάνισης είναι τα αγγλικά), που σημαίνει ότι ο χρήστης είναι συνδεδεμένος και χρησιμοποιεί αυτήν τη στιγμή τον ιστοχώρο. Με την προσπέλαση του εικονιδίου αυτού, αλλάζει η ένδειξη σε “OFF”. Στην περίπτωση που επιλεχθεί η ένδειξη αυτή, τότε εμφανίζεται μήνυμα επιβεβαίωσης εξόδου. Η θετική επιβεβαίωση προκαλεί την έξοδο του χρήστη και την εμφάνιση της κεντρικής σελίδας του ιστοχώρου ενώ η αρνητική επιβεβαίωση αφήνει το χρήστη στην κατάσταση που βρίσκεται. Τα παραπάνω περιγράφονται και από την εικόνα 61.



Εικόνα 61: Έξοδος χρηστών

3.3.3 Στατιστικά ιστοχώρου

Η κεντρική σελίδα του διαχειριστή ονομάζεται statistics (εικόνα 61). Στον τομέα αυτόν, υπάρχουν πληροφορίες σχετικά με το πλήθος των συνολικών επισκεπτών, το πλήθος των ενεργών μελών, καθώς και το πλήθος των ενεργών μελών που χρησιμοποιούν τον ιστοχώρο τη συγκεκριμένη στιγμή.

Statistics	Users	Troubleshoots	Messages
Total Visits		2198	
Members		1	
Members Online		0	

Εικόνα 62: Στατιστικά ιστοχώρου

3.3.4 Χρήστες ιστοχώρου

Η καρτέλα “Users”, δίνει την δυνατότητα στο διαχειριστή να προβάλει, δημιουργήσει, διαγράψει και επεξεργαστεί τους απλούς χρήστες (εικόνα 63 α). Πιο αναλυτικά:

1. **Προβολή – Επεξεργασία:** Ο διαχειριστής μπορεί να προβάλει τους χρήστες. Οι πληροφορίες που συγκεντρώνει είναι οι εξής:
 - i. Device id, Device password: Τις χρησιμοποιεί κατά τον προγραμματισμό της συσκευής, αλλά και του ιστοχώρου. Στην περίπτωση που το μέλος είναι διαχειριστής, τότε το device_id είναι ένας μεγάλος αριθμός (άνω του 999) ενώ δεν υπάρχει device_password (εικόνα 63 β). Σε αντίθετη περίπτωση εμφανίζονται κανονικά τα δεδομένα (εικόνα 63 γ). Παρακάτω (κεφάλαιο 3.5) γίνεται εκτενής αναφορά για τη χρήση αυτών των δύο πεδίων κατά τον προγραμματισμό της συσκευής.
 - ii. Email: Αν το μέλος που προβάλει είναι απλός χρήστης, τότε μπορεί να αλλάξει τη διεύθυνση του, διαφορετικά όχι. Η αλλαγή γίνεται επιλέγοντας τη διεύθυνση αντικαθίσταται από ένα πλαίσιο εισαγωγής. Όταν γίνει η εισαγωγή του νέου email, υπάρχει κουμπί επιβεβαίωσης (εικόνα 63 δ).
 - iii. Last Entry, Total Visits: Ημερομηνία τελευταίας επίσκεψης στον ιστοχώρο και πλήθος επισκέψεων συνολικά.
 - iv. Userlevel: Επίπεδο χρήστη.
 - v. Messages: Το πλήθος των μηνυμάτων που έχει στείλει ο χρήστης. Επιλέγοντάς τον ο χρήστης μεταβαίνει στα μηνύματα του χρήστη (3.3.6 Μηνύματα χρηστών).
 - vi. Action: Δυνατότητα διαγραφής του χρήστη. Σε περίπτωση επιλογής της ρύθμισης αυτή εμφανίζεται μήνυμα επιβεβαίωσης (εικόνα 63 ε).

Statistics	Users	Troubleshoots	Messages
A/A	User_id(device_id)		
-	New Device		
1	12(1000)		
2	40(1)		

α)

Device_id	1000
Device Password	
Email	gnnpls_nikos@hotmail.com
Last Entry	2012-11-29 10:53:38
Total Visits	54
Userlevel	0
Messages	0
Action	

β)

Device_id	1
Device Password	bda5accf070d5938804296ecaab15978d2c45e9c
Email	gnnpls_nikos@gmail.com
Last Entry	2012-11-29 12:06:34
Total Visits	288
Userlevel	1
Messages	11
Action	✗

γ)

Email: Done! Save δ)

Action: Are you sure? ε)

Εικόνα 63: Χρήστες επεξεργασία - προβολή

2. **Δημιουργία:** Ο διαχειριστής μπορεί να δημιουργήσει είτε νέους διαχειριστές είτε απλούς χρήστες (εικόνα 64 α). Στην περίπτωση των νέων χρηστών εισάγει το device_id της συσκευής που τους δόθηκε, το email τους και θέτει το Userlevel στην επιλογή 1 (εικόνα 64 β). Έπειτα παράγεται ο κωδικός συσκευής (device_password) όπως αναφέρθηκε παραπάνω (3.2), χωρίς καμία επίγνωση του κειμένου από το διαχειριστή. Σε αντίθετη περίπτωση εισάγεται σαν device_id ένα νούμερο άνω του 999 και επιλέγεται στο Userlevel η τιμή 2 (εικόνα 64 γ)..

Η ολοκλήρωση της διαδικασίας αυτής γίνεται με την αποστολή ενός email, με τον κωδικό και τη διεύθυνση ενεργοποίησης του λογαριασμού μέσω του πεδίου private_id όπως περιγράφηκε στην ενότητα 3.2 (εικόνα 64 δ).

A/A	User_id(device_id)
-	New Device
1	12(1000)
2	40(1)

α)

Device_id	
Email	
Userlevel	Select Level
Actions	Send Login Info Clear

β)

Device_id	3
Email	new_user@mail.com
Userlevel	User
Actions	Send Login Info Clear

γ)

Device_id	1001
Email	new_admin@mail.com
Userlevel	Admin
Actions	Send Login Info Clear

δ)

SmartGreenhouse - Customer Registration!

smartgreenhouse@zafora.ict.e.uowm.gr 12:57 (3 minutes ago)

Thank you for using our product. SmartGreenhouse welcomes you to our community. Please verify your account by visiting the follow link: [Press Here](#)

After, you will be able to use your device properly by logging in with your email and the random auto-generated password: 1245456250.

Εικόνα 64: Χρήστες - δημιουργία χρήστη

Όταν ζητείται του διαχειριστή να εισάγει ένα email, τότε υπάρχουν αυστηρά κριτήρια ασφάλειας περιεχομένου. Η μορφή του κειμένου μπορεί να είναι μόνο

εκείνη αντίστοιχη ενός email, ενώ δε μπορεί να εισαχθεί κάποιο email από χρήστη που είναι ήδη ενεργός. Ο έλεγχος γίνεται τόσο με τη χρήση AJAX, όσο και με στα αρχεία php, όπως αναφέρθηκε παραπάνω.

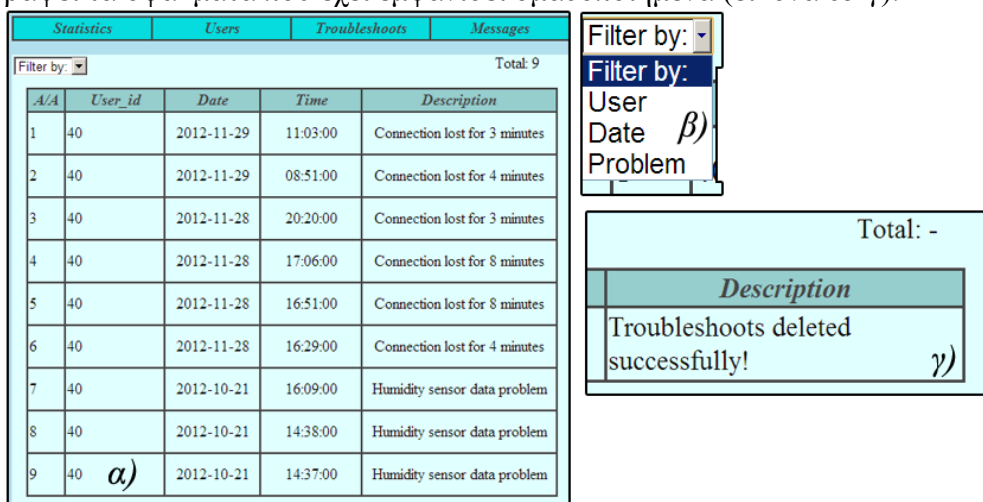
Αντίστοιχα και όταν ζητείται το device_id, μόνο νούμερα μπορούν να εισαχθούν στα αντίστοιχα πεδία. Χρησιμοποιείται η συνάρτηση “is_numeric”, της php. Εφόσον είναι αριθμός, τότε επεξεργάζεται ως προς την ύπαρξη στη βάση. Οι έλεγχοι γίνονται τόσο με AJAX όσο και πριν εκτελεστούν τα ερωτήματα στη βάση. Με τον τρόπο αυτό υπάρχει έλεγχος ακόμα και στους διαχειριστές για τα δεδομένα που εισάγουν, ώστε να αποφεύγεται οποιαδήποτε κακόβουλη επίθεση στη βάση δεδομένων.

3.3.5 Αναφορά σφαλμάτων

Η καρτέλα “Troubleshoots” (εικόνα 65 α) εμφανίζει τα διάφορα σφάλματα που έχουν προκύψει στο σύστημα στο σύνολο του, δηλαδή είτε σφάλματα συσκευής είτε σφάλματα επικοινωνίας ιστοχώρου με το χρήστη. Οι κατηγορίες σφαλμάτων είναι οι εξής:

1. **Αισθητήρας εκτός λειτουργίας (sensor not working):** Αισθητήρας δε λαμβάνει μετρήσεις.
2. **Ξαφνική αλλαγή δεδομένων (sensor data problem):** Ξαφνική μεταβολή από μέτρηση σε μέτρηση σε αισθητήρα, που δηλώνει είτε πρόβλημα στο θερμοκήπιο είτε πρόβλημα στο κύκλωμα της συσκευής.
3. **Απώλεια σύνδεσης (connection problem):** Η συσκευή έχει χάσει την επικοινωνία της με τον ιστόχωρο.
4. **Ανεπιτυχής αποστολή email (mail problem):** Τα email που στέλνει ο ιστοχώρος δε φθάνουν στη διεύθυνση του χρήστη.

Ο διαχειριστής έχει τη δυνατότητα να φιλτράρει τα σφάλματα προς εμφάνιση ανάλογα τον τύπο, το χρήστη και την ημερομηνία (εικόνα 65 β). Επίσης, μπορεί να διαγράψει τα σφάλματα που έχει εμφανίσει ομαδοποιημένα (εικόνα 65 γ).



The screenshot shows the 'Troubleshoots' section of the application. It includes a table with columns: A/A, User_id, Date, Time, and Description. The table lists 9 entries, mostly related to connection loss and humidity sensor data problems. To the right of the table is a 'Filter by:' dropdown menu with options: User, Date, and Problem. Below the table, there is a confirmation message: 'Troubleshoots deleted successfully!' with a green checkmark icon.

A/A	User_id	Date	Time	Description
1	40	2012-11-29	11:03:00	Connection lost for 3 minutes
2	40	2012-11-29	08:51:00	Connection lost for 4 minutes
3	40	2012-11-28	20:20:00	Connection lost for 3 minutes
4	40	2012-11-28	17:06:00	Connection lost for 8 minutes
5	40	2012-11-28	16:51:00	Connection lost for 8 minutes
6	40	2012-11-28	16:29:00	Connection lost for 4 minutes
7	40	2012-10-21	16:09:00	Humidity sensor data problem
8	40	2012-10-21	14:38:00	Humidity sensor data problem
9	40	2012-10-21	14:37:00	Humidity sensor data problem

Εικόνα 65: Αναφορά σφαλμάτων

3.3.6 Μηνύματα χρηστών

Η καρτέλα “Messages” εμφανίζει όλα τα μηνύματα που έχουν στείλει οι απλοί χρήστες στο διαχειριστή (εικόνα 66 α). Εκείνος, γνωρίζει πριν διαβάσει το μήνυμα από ποιο χρήστη προέρχεται, πότε στάλθηκε και ποιο είναι το θέμα του. Επίσης, μπορεί να γνωρίζει αν το έχει διαβάσει ή όχι (εικόνα 66 β). Η λειτουργία του

«φιλτραρίσματος», όπως και παραπάνω στην αναφορά σφαλμάτων (παράγραφο 3.3.5), υπάρχει και σε αυτήν την καρτέλα.

Ο διαχειριστής εμφανίζει το μήνυμα επιλέγοντας το θέμα του. Με την επιλογή αυτή γίνεται μετάβαση σε μια διαφορετική σελίδα όπου εμφανίζεται και το κυρίως σώμα του μηνύματος (εικόνα 66 γ). Ο διαχειριστής μπορεί είτε να διαγράψει το μήνυμα είτε να απαντήσει στο μήνυμα επιλέγοντας την ηλεκτρονική διεύθυνση του αποστολέα. Οι απαντήσεις στέλνονται στο email του απλού χρήστη μέσω επιλεγμένης υπηρεσίας ηλεκτρονικού ταχυδρομείου.

Statistics	Users	Troubleshoots	Messages
Filter by: ▼			Total: 2 α)
A/A	Device_id	Date	Subject
1	1	2012-11-29 12:06:34	Ευχαριστώ (old)
2	1	2012-11-29 12:06:05	Γιαννόπουλος Νικόλαος (new)

Subject
Ευχαριστώ (old)
Γιαννόπουλος Νικόλαος (new) β)

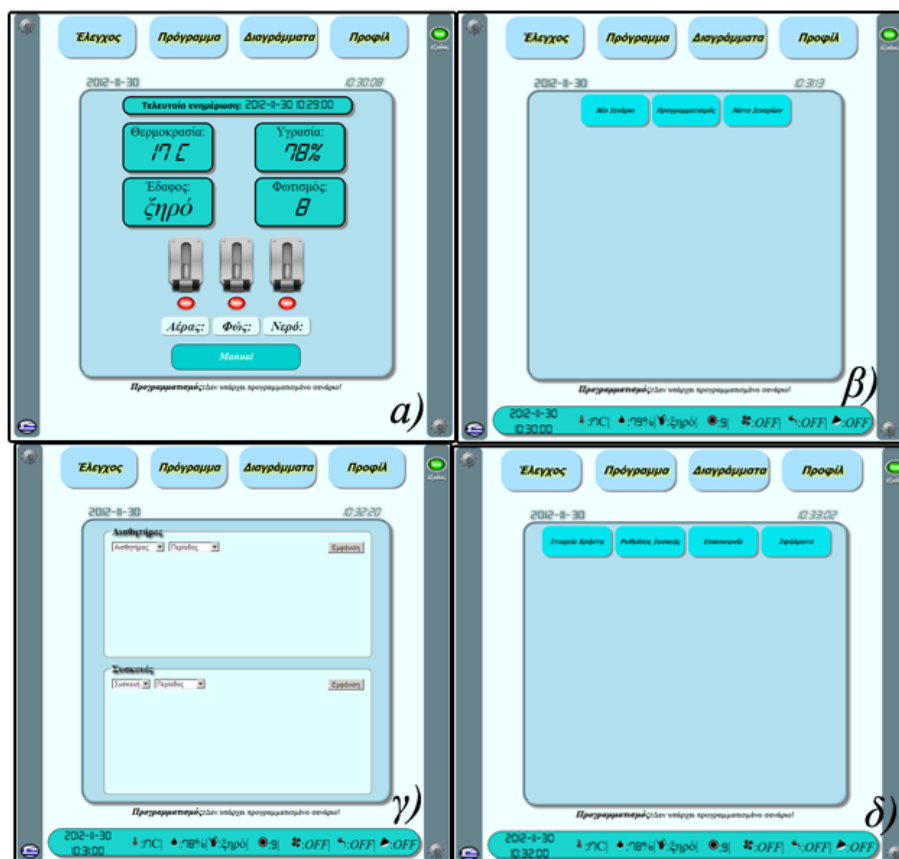
-	Email	Date	Subject
	gmnpls.nikos@gmail.com	2012-11-29 12:06:34	Ευχαριστώ
Ευχαριστώ για την βοήθεια! γ)			

Εικόνα 66: Μηνύματα χρηστών

3.4 Λειτουργίες ιστοχώρου για απλό χρήστη

Στην ενότητα αυτή γίνεται αναφορά στις λειτουργίες που προσφέρει ο ιστοχώρος στους απλούς χρήστες. Σύμφωνα με τις απαιτήσεις απλών χρηστών, που αναφέρθηκαν παραπάνω (παράγραφο 3.1), ο ιστοχώρος σχεδιάστηκε με σκοπό να τις ικανοποιεί και να συνεισφέρει στη χρηστικότητα του συστήματος. Γι' αυτόν το λόγο κατά τα βήματα σχεδίασης του ιστοχώρου, έγιναν δοκιμές με άτομα από διαφορετικές ηλικίες με στόχο να καλυφθούν οι ανάγκες ομάδων που είχαν έλλειψη τεχνογνωσίας, αλλά και ομάδων που η έννοια των ενσωματωμένων συστημάτων, τους είναι γνωστή. Το αποτέλεσμα αυτών των δοκιμών ήταν η δημιουργία του παρακάτω ιστοχώρου (εικόνα 67), ο οποίος αποτελείται από 4 διαφορετικούς τομείς, τον «Έλεγχο» (εικόνα 67 α), το «Πρόγραμμα» (εικόνα 67 β), τα «Διαγράμματα» (εικόνα 67 γ) και το «Προφίλ» (εικόνα 67 δ), που προσφέρουν συγκεκριμένες λειτουργίες και θα αναλυθούν στις παρακάτω υποενότητες. Η εμφάνιση του ιστοχώρου είναι δυνατή σε δύο γλώσσες, ελληνικά και αγγλικά ανάλογα με την προτίμηση του χρήστη.

Προγραμματιστικά, έχει γίνει χρήση της τεχνολογίας AJAX στον τρόπο που εκτελείται η επικοινωνία με τη βάση. Εκτός από τους τέσσερις τομείς που αναφέρθηκαν παραπάνω οι οποίοι όταν επιλεγθούν φορτώνεται εξ' αρχής η σελίδα, οι υπόλοιπες λειτουργίες γίνονται δυναμικά μη επιβαρύνοντας με αυτόν τον τρόπο τον browser, φορτώνοντας ξανά και ξανά τη σελίδα, ενώ γίνονται και οι διαδικασίες ταχύτερες στην προσπέλαση από το χρήστη.



Εικόνα 67: Ιστοχώρος απλού χρήστη

3.4.1 Λειτουργία Εισόδου - Εξόδου

Η λειτουργία εισόδου και εξόδου του χρήστη με τον ιστοχώρο και τα προσωπικά του δεδομένα και ρυθμίσεις είναι κοινή με αυτήν του διαχειριστή. Οι πληροφορίες που χρειάζονται βρίσκονται τοποθετημένες παραπάνω (είσοδος: παράγραφος 3.3.1 / έξοδος: παράγραφος 3.3.2).

3.4.2 Λειτουργία ενημέρωσης συνθηκών σε πραγματικό χρόνο

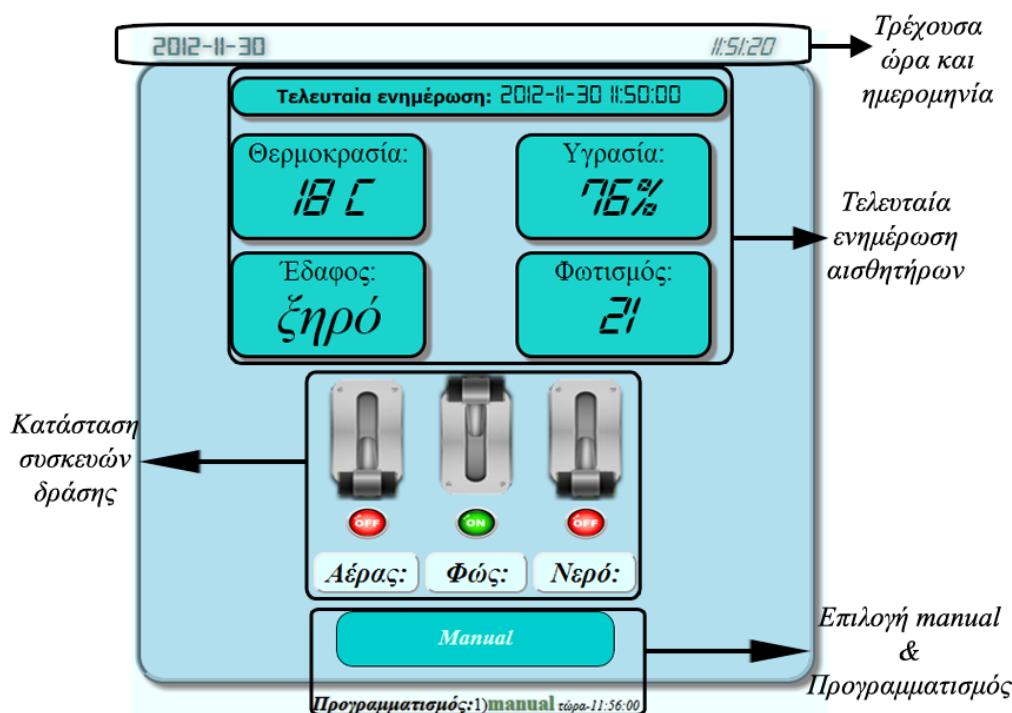
Βασική δυνατότητα που δίνει το σύστημα “Smart Greenhouse” μέσω του ιστοχώρου, είναι η ενημέρωση των συνθηκών που επικρατούν στο θερμοκήπιο σε πραγματικό χρόνο. Στην αρχική σελίδα που φορτώνεται μετά την επιτυχημένη ταυτοποίηση του χρήστη γίνεται η ενημέρωση με τη μορφή ενός πίνακα ελέγχου με χαρακτηριστικές λυχνίες, μοχλούς και οθόνες (εικόνα 67 α). Ο λόγος που επιλέχθηκε αυτή η αναπαράσταση ήταν η εξοικείωση που είχε το μεγαλύτερο ποσοστό των χρηστών που έλαβαν μέρος στις δοκιμές. Με τον τρόπο αυτό, ικανοποιείται η απαίτηση που αναπτύχθηκε παραπάνω (παράγραφος 3.1) για φιλικό προς το χρήστη περιβάλλον. Έτσι το γραφικό περιβάλλον το οποίο χρησιμοποιείται δε δυσκολεύει το χρήστη ενώ παράλληλα είναι οικείο, καθώς πίνακες ελέγχου χρησιμοποιούνται μόνιμα εδώ και δεκαετίες στον τομέα της γεωργίας.

Ο πίνακας ελέγχου που υπάρχει στον τομέα «Έλεγχος», αποτελείται από 5 πλαίσια (οθόνες) που ενημερώνουν για το πότε έγινε η τελευταία ενημέρωση και για κάθε αισθητήρα το αποτέλεσμα της μέτρησης που έλαβε. Επίσης, για να υπάρχει καλύτερη αίσθηση του χρόνου υπάρχει και ένα ρολόι με την τρέχουσα ημερομηνία και ώρα, ώστε να διαπιστώνει ο χρήστης απευθείας τυχόν καθυστερήσεις στην

επικοινωνία του ιστοχώρου (δηλαδή της βάσης) με τη συσκευή που είναι τοποθετημένη στο θερμοκήπιο.

Η κατάσταση των συσκευών αναπαρίσταται με μοχλούς και λυχνίες “ON”/“OFF”. Επομένως αν κάποια συσκευή είναι ενεργοποιημένη η λυχνία είναι “ON” και ο μοχλός είναι σηκωμένος ψηλά, ενώ σε αντίθετη περίπτωση η συσκευή είναι “OFF” και ο μοχλός κατεβασμένος.

Οι πληροφορίες που παρουσιάστηκαν στις παραπάνω δύο παραγράφους απεικονίζονται στην εικόνα 68. Επίσης, υπάρχουν ακόμα δύο στοιχεία στον πίνακα ελέγχου. Το στοιχείο του προγραμματισμού και η επιλογή manual. Τα δύο αυτά στοιχεία θα αναλυθούν περαιτέρω στις παρακάτω υποενότητες.



Εικόνα 68: Ενημέρωση κατάστασης θερμοκηπίου

Η παραπάνω εικόνα (εικόνα 68) απεικονίζει ένα στιγμιότυπο από τον πίνακα ελέγχου. Οι πληροφορίες που εξάγονται είναι οι εξής:

- Τρέχουσα ημερομηνία: 30/11/2012.
- Τρέχουσα ώρα: 11:51:20.
- Τελευταία ενημέρωση: 30/11/2012, 11:50:00. Υπάρχει ένα λεπτό διαφορά μεταξύ της τρέχουσας ώρας και της τελευταίας ενημέρωσης. Το γεγονός αυτό οφείλεται σε θέματα συγχρονισμού της συσκευής με τον ιστόχωρο. Κάθε 15 λεπτά πραγματοποιείται συγχρονισμός ξανά της συσκευής με το σύστημα, ώστε να μην υπάρχει μεγάλη χρονική διαφορά.
- Θερμοκρασία: 18 C.
- Υγρασία: 76 %.
- Κατάσταση Εδάφους: Ξηρό.
- Φωτισμός: 21 (ένταση).
- Ανεμιστήρας: Απενεργοποιημένος.
- Φώς: Ενεργοποιημένο.
- Νερό: Απενεργοποιημένο.
- Προγραμματισμός: Υπάρχει σενάριο που είναι σε λειτουργία και είναι το manual.

Σε περίπτωση που ο χρήστης δε βρίσκεται στον τομέα Ελέγχου, τότε η ενημέρωση σε πραγματικό χρόνο για τις συνθήκες στο θερμοκήπιο γίνεται μέσω μιας οθόνης που τοποθετείται στο τέλος της σελίδας εμφάνισης. Οι αισθητήρες αναπαρίστανται μέσω των αντίστοιχων συμβόλων που τις ερμηνεύουν όπως επίσης και οι συσκευές. Η αναπαράσταση αυτή απεικονίζεται στην εικόνα 69.



Εικόνα 69: Ενημέρωση κατάστασης θερμοκηπίου εκτός τομέα Ελέγχου

3.4.3 Λειτουργία ενεργοποίησης manual σεναρίου

Στην εικόνα 68, υπάρχει η επιλογή manual στο πίνακα ελέγχου της αρχικής σελίδας που φορτώνεται στο χρήστη. Το manual είναι ένα σενάριο χρήσης. Τα σενάρια όπως αναφέρεται και παραπάνω δημιουργήθηκαν με σκοπό την ικανοποίηση της απαίτησης της αντικατάστασης της συνεχούς ανθρώπινης παρουσίας στο χώρο του θερμοκηπίου. Το συγκεκριμένο σενάριο υπάρχει από την εγκατάσταση της συσκευής και ο χρήστης δε μπορεί να το διαγράψει. Υλοποιεί την άμεση ενεργοποίηση ή απενεργοποίηση των συσκευών δράσης. Η διαδικασία είναι η εξής:

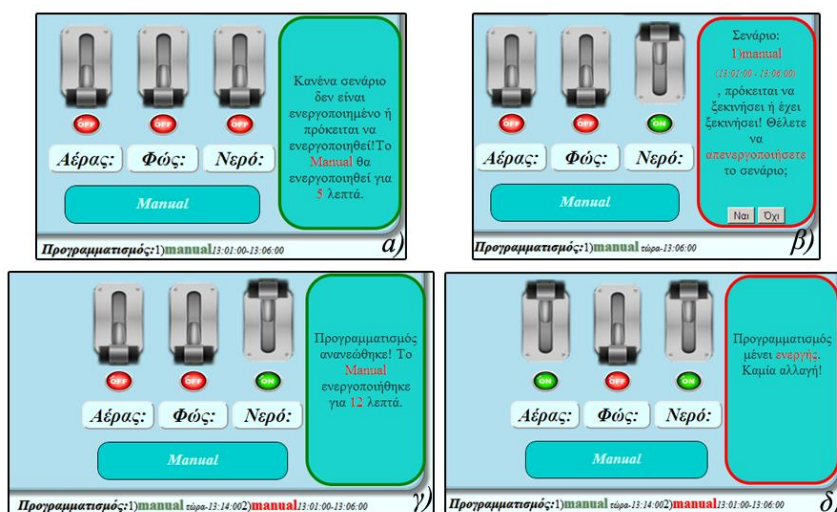
1. Βήμα 1^ο: Ο χρήστης επιλέγει το «Manual».
2. Βήμα 2^ο: Ο χρήστης επιλέγει διάρκεια και την κατάσταση των συσκευών (εικόνα 70 α). Στην εισαγωγή διάρκειας σεναρίου υπάρχουν οι εξής περιορισμοί:
 - Ο χρήστης μπορεί να εισάγει μόνο αριθμούς (εικόνα 70 β). Ο έλεγχος στο πεδίο αυτό γίνεται με τη χρήση της συνάρτησης “is_numeric”, όπως έχει αναφερθεί και παραπάνω.
 - Η διάρκεια δε μπορεί να ξεπερνάει τα 1000 λεπτά (εικόνα 70 γ).
 - Το πεδίο αυτό είναι απαραίτητο και σε περίπτωση που δεν το συμπληρώσει ο χρήστης δεν προχωράει η διαδικασία.
3. Βήμα 3^ο: Ο χρήστης επιλέγει είτε «Εκκίνηση» αν θέλει να ξεκινήσει το σενάριο, είτε «Εξοδος» αν θέλει να ακυρώσει τη διαδικασία.



Εικόνα 70: Βήματα manual

Με τα 3 παραπάνω βήματα ολοκληρώνεται η διαδικασία ενεργοποίησης του manual. Με την επιλογή «Εκκίνηση» μπορούν να συμβούν οι παρακάτω περιπτώσεις:

1. Προγραμματισμός κενός: Στην περίπτωση αυτή δεν υπάρχει κάποιο προγραμματισμένο σενάριο που να επικαλύπτει ή να επικαλύπτεται χρονικά από το manual. Η ενεργοποίηση του σεναρίου είναι άμεση και ενημερώνεται ο χρήστης με αντίστοιχο μήνυμα (εικόνα 71 α). Επίσης, ενημερώνεται και το πεδίο προγραμματισμός που θα αναλυθεί στις παρακάτω υποενότητες.
2. Προγραμματισμός συμπληρωμένος: Στην περίπτωση αυτή υπάρχει κάποιο σενάριο που επικαλύπτει ή επικαλύπτεται χρονικά από τη διάρκεια του manual. Τότε εμφανίζεται αντίστοιχο μήνυμα που περιέχει τα σενάρια που πρέπει να απενεργοποιηθούν για να ενεργοποιηθεί η επιλογή του χρήστη (εικόνα 71 β). Αν ο χρήστης επιβεβαιώσει την απενεργοποίηση, τότε τα σενάρια δε θα εκτελεστούν όπως είναι προγραμματισμένο και θα εκτελεστεί άμεσα το manual (εικόνα 71 γ). Αν ο χρήστης δεν επιβεβαιώσει την απενεργοποίηση, τότε ο προγραμματισμός θα μείνει ως έχει (εικόνα 71 δ). Αντίστοιχα ενημερώνεται και το πεδίο του προγραμματισμού κάτω από την επιλογή manual.



Εικόνα 71: Εκκίνηση manual

Επίσης, υπάρχει και η περίπτωση η διάρκεια του manual σεναρίου να καλύπτει και την επόμενη μέρα από εκείνη της εκκίνησης. Η κατάσταση αυτή αντιμετωπίζεται με τη διάσπαση του manual σε δύο καταχωρίσεις στον προγραμματισμό. Η μία καταχώρηση τελειώνει τις 23:59 της τρέχουσας ημέρας ενώ το δεύτερο ξεκινάει από τις 00:00 της επόμενης και τελειώνει βάση της διάρκειας που έχει οριστεί.

Τέλος, η ώρα εκκίνησης του manual μεταφέρεται πάντα ένα λεπτό αργότερα όπως αντίστοιχα και η ώρα διακοπής. Ο λόγος που συμβαίνει αυτό, είναι για να μην χαθεί κάποιο λεπτό από την επιθυμητή διάρκεια, γιατί αν ενεργοποιούνταν ακριβώς την ώρα έναρξης, τότε θα υπήρχε η περίπτωση το ρολόι της συσκευής να ήταν τέλεια συγχρονισμένο με εκείνο της βάσης και να χανόταν το πρώτο λεπτό λειτουργίας.

3.4.4 Λειτουργία ελέγχου σεναρίων ημέρας

Όπως έχει αναφερθεί παραπάνω και απεικονίζεται στις εικόνες 68,70 και 71 υπάρχει το πεδίο «Προγραμματισμός» κάτω από τον πίνακα ελέγχου. Το πεδίο αυτό φαίνεται και σε όλους τους άλλους τομείς. Υπάρχουν όμως διαφορές και ομοιότητες

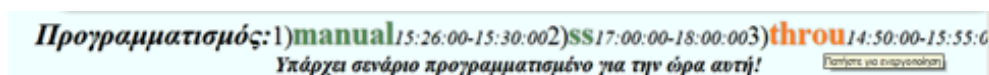
στον τρόπο λειτουργίας του στο τομέα «Έλεγχος» σε σχέση με τους υπόλοιπους, οι οποίες είναι οι εξής:

- **Ομοιότητες:** Σε όλους τους τομείς το πεδίο «Προγραμματισμός» υλοποιεί την ίδια λειτουργία. Ενημερώνει το χρήστη ποια σενάρια έχουν προγραμματιστεί και πότε πρόκειται να εκτελεστούν. Τα σενάρια που είναι προγραμματισμένα και ενεργοποιημένα απεικονίζονται με πράσινο χρώμα ενώ τα σενάρια που είναι απενεργοποιημένα απεικονίζονται με κόκκινο χρώμα (εικόνα 71 δ). Επίσης, δίνει τη δυνατότητα στο χρήστη να ενεργοποιήσει ή να απενεργοποιήσει τα σενάρια ανάλογα με την κατάστασή τους (εικόνα 72).



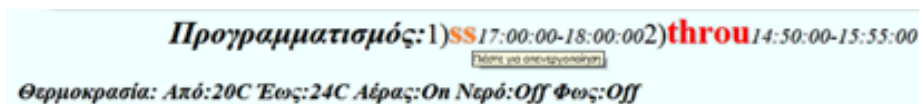
Εικόνα 72: Προγραμματισμός απενεργοποίηση

Σε περίπτωση που ένα σενάριο επικαλύπτει ή επικαλύπτεται δε μπορεί να ενεργοποιηθεί και εμφανίζεται αντίστοιχο μήνυμα. Αυτό συμβαίνει, καθώς κάθε στιγμή μόνο ένα σενάριο μπορεί να είναι ενεργό.



Εικόνα 73: Περιορισμός ενεργοποίησης σεναρίων

- **Διαφορές:** Στον τομέα «Έλεγχος», ο οποίος φορτώνεται αμέσως μετά την επιτυχή ταυτοποίηση του χρήστη, εκτός από τις παραπάνω λειτουργίες υπάρχει και επιπλέον μία. Με το που ο δείκτης του ποντικιού βρεθεί πάνω από τα προγραμματισμένα σενάρια εμφανίζονται οι συνθήκες τους. Με αυτόν τον τρόπο ο χρήστης γνωρίζει με το που ανοίξει την εφαρμογή, τι νοημοσύνη έχει εγκαταστήσει στη συσκευή.



Εικόνα 74: Προγραμματισμός - πληροφορίες σεναρίου

Στην εικόνα 74 πληροφορείται ο χρήστης ότι με το σενάριο “ss” θα ενεργοποιηθεί ο ανεμιστήρας όταν η θερμοκρασία είναι από 20 C έως 24 C.

3.4.5 Λειτουργία δημιουργίας σεναρίου

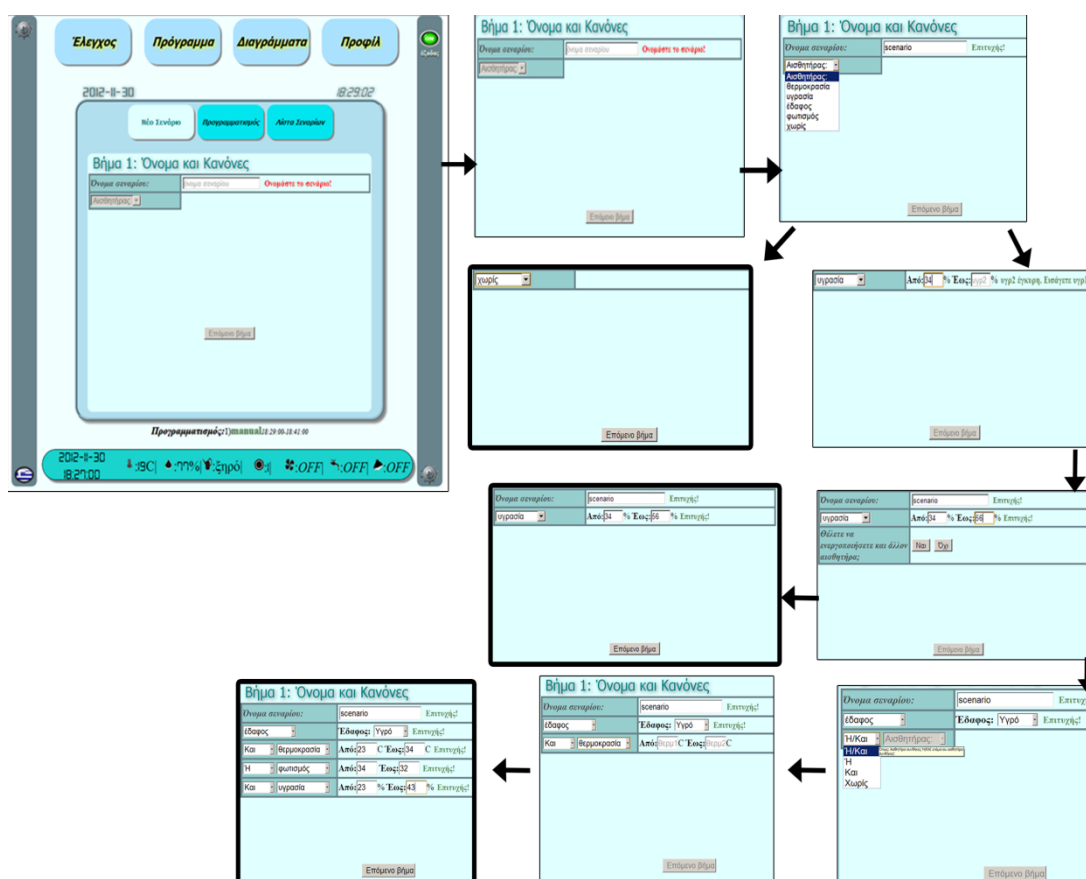
Όπως έχει αναφερθεί και παραπάνω, υπάρχει στο σύστημα η έννοια του σεναρίου. Με τον τρόπο αυτό αναπαρίσταται η ανθρώπινη παρουσία στο χώρο του θερμοκηπίου. Όσους περισσότερους αισθητήρες επιλέξει ένας χρήστης να χρησιμοποιεί η συσκευή, τόσο καλύτερη αντίληψη θα έχει για το χώρο και τόσο περισσότερο θα αυξήσει τη νοημοσύνη της. Επομένως, το σύστημα έχει ενσωματωμένη ευφυΐα, όχι αυτόματη, δηλαδή από κατασκευής του, αλλά βασισμένη στον τρόπο με τον οποίο ο χρήστης αντιλαμβάνεται την καλλιέργεια σε σχέση με τις συνθήκες που επικρατούν στο θερμοκήπιο και τις συσκευές δράσης.

Η δημιουργία του σεναρίου στον ιστόχωρο χωρίζεται σε 2 βήματα:

1. **Βήμα 1^ο – Όνομα και κανόνες:** Η επιλογή «Νέο Σενάριο» μεταφέρει το χρήστη στη δημιουργία σεναρίου. Αρχικά πρέπει να τοποθετήσει ο χρήστης

ένα όνομα σεναρίου. Αν δεν τοποθετηθεί όνομα σεναρίου δε γίνεται να προχωρήσει η διαδικασία.

Μόλις τοποθετηθεί το όνομα, επιλέγει ο χρήστης αν θέλει ή όχι να έχει το σενάριο συνθήκες αισθητήρων. Πάλι αν δεν επιλέξει κάτι δε θα προχωρήσει στο επόμενο βήμα. Σε περίπτωση που δεν επιθυμεί να έχει συνθήκες αισθητήρων το δηλώνει μέσω της επιλογής «χωρίς». Αντίθετα, με την επιλογή κάποιου αισθητήρα εμφανίζεται πλαίσιο που πρέπει να δοθούν τα όρια των τιμών. Πάλι, η μη συμπλήρωση των κενών αποτελεί τροχοπέδη για το επόμενο βήμα. Μόλις συμπληρωθούν με επιτυχία οι τιμές, τότε ρωτάται ο χρήστης αν επιθυμεί να χρησιμοποιήσει και άλλον αισθητήρα. Στην περίπτωση θετικής επιβεβαίωσης ο χρήστης καλείται να διαλέξει τη σχέση που θα έχει ο επόμενος αισθητήρας με τον προηγούμενο, δηλαδή αν θα είναι «Η» (OR) είτε «ΚΑΙ» (AND) είτε «ΧΩΡΙΣ» όπου στην τελευταία περίπτωση, αυτόματα η διαδικασία διακόπτεται και μπορεί να συνεχίσει στο επόμενο βήμα. Για κάθε αισθητήρα που προσθέτει επαναλαμβάνεται η διαδικασία μέχρι να μην υπάρχει άλλος αισθητήρας ή να μην επιθυμεί άλλη προσθήκη. Εν συνεχεία ενεργοποιείται η επιλογή «Επόμενο βήμα». Η παραπάνω περιγραφή απεικονίζεται στην εικόνα 75, όπου τα στιγμιότυπα με έντονο περίγραμμα αποτελούν τελικές καταστάσεις (δηλαδή ενεργοποίηση επόμενου βήματος).



Εικόνα 75: Δημιουργία σεναρίου - Βήμα 1

Όπως έχει αναφερθεί και παραπάνω υπάρχουν συνεχείς έλεγχοι τόσο από script της javascript όσο και κατά την επεξεργασία των δεδομένων για να αποθηκευτούν στη βάση. Γι' αυτό και σε κάθε πεδίο που απαιτείται συμπλήρωση κενού, ο χρήστης έχει περιορισμούς στο τι μπορεί να εισάγει. Σε περίπτωση αριθμού μπορεί να εισάγει μόνο αριθμό (με τη χρήση της

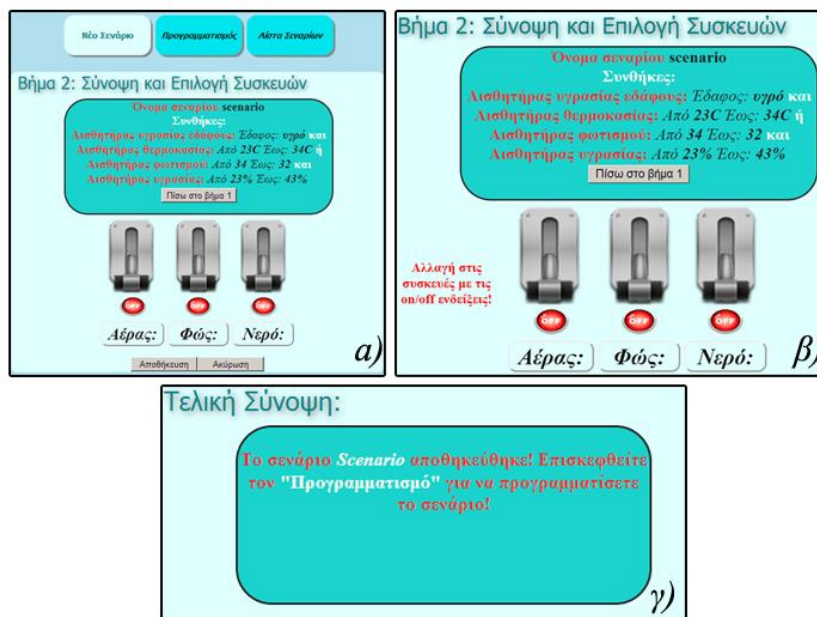
“is_numeric”), σε περίπτωση κειμένου μόνο κείμενο χωρίς παρενθέσεις και ερωτηματικά (ανάλογα με ποια γλώσσα προβολής του ιστοχώρου χρησιμοποιεί ο χρήστης, επιλέγεται και το περιεχόμενο της “preg_match”, είτε για αγγλικά “/[^\.,_A-Za-z0-9\ ']/” είτε για ελληνικά “/[^\.,_A-Za-zάΈΕήΗίΙόΌύΎώΩΖζΧχΨψωΩβΒνΝμΜλλΚκΞξΗηΓγΦφΔδΣσΑαζΕεΡρΤτΥυΘθΙιΟοπΠ0-9\ ']/”) , ενώ υπάρχουν περιορισμοί και στα μεγέθη των τιμών των αισθητήρων, ώστε να ανταποκρίνονται σε πραγματικά δεδομένα (εικόνα 76). Με αυτόν τον τρόπο εξασφαλίζεται σε μεγάλο ποσοστό η εύρυθμη λειτουργία του ιστοχώρου και γενικά του συστήματος, τόσο σε επίπεδο ασφάλειας όσο και επίπεδο συμπεριφοράς και εκτέλεσης των εντολών του χρήστη.

Όνομα σεναρίου:	\$_POST()	Σφάλμα δεδομένων!
Αισθητήρας		
υγρασία	Από: 34 % Έως: nikos	Μόνο αριθμοί!
υγρασία	Από: 34 % Έως: 101	Όχι μεγαλύτερο από 100%!

Εικόνα 76: Έλεγχος δεδομένων

- Βήμα 2^ο – Σύνοψη και επιλογή συσκευών:** Στο δεύτερο βήμα ο χρήστης αρχικά μπορεί να δει τις συνθήκες αισθητήρων που έχει επιλέξει στο προηγούμενο βήμα μέσω του πεδίου «Σύνοψη» (σε περίπτωση που δεν έχει επιλέξει συνθήκες το πεδίο είναι κενό) όπως εμφανίζεται στην εικόνα 77 α). Σε περίπτωση που παρατηρήσει στο πεδίο «Σύνοψη» κάποιο σφάλμα στις συνθήκες μπορεί να επιστρέψει στο προηγούμενο βήμα. Αντίθετα, αν είναι ικανοποιημένος από αυτές, ορίζει την κατάσταση στην οποία θα βρίσκονται οι συσκευές δράσης, δηλαδή είτε ενεργοποιημένες είτε απενεργοποιημένες. Για τη διευκόλυνση του χρήστη υπάρχει μήνυμα βοήθειας για τον τρόπο χειρισμού των μοχλών (εικόνα 77 β).

Στη συνέχεια αν ο χρήστης επιλέξει «Αποθήκευση» το σενάριο θα αποθηκευτεί στον πίνακα “scenarios” (παράγραφος 3.2) και θα είναι έτοιμο προς χρήση, εμφανίζοντας και το σχετικό μήνυμα (εικόνα 77 γ). Στις περιπτώσεις που επιλέξει είτε «Άκυρο» είτε δεν επιλέξει τίποτα και περιηγηθεί εκτός της διαδικασίας αυτής, το σενάριο δεν αποθηκεύεται και δε μπορεί να χρησιμοποιηθεί στη συνέχεια κατά τον προγραμματισμό.



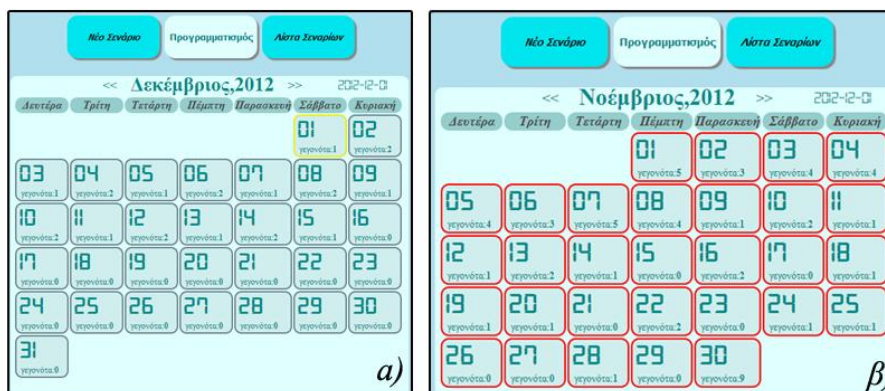
Εικόνα 77: Δημιουργία Σεναρίου - Βήμα 2

Η διαδικασία που περιγράφηκε παραπάνω, βελτιστοποιήθηκε στη μορφή αυτή μετά από πολλές πειραματικές χρήσεις. Έχει δημιουργηθεί με σκοπό να είναι ευέλικτη για το χρήστη, με την εναλλαγή των βημάτων, αλλά σε περίπτωση που κάποιος δεν αντιληφθεί από την αρχή τον τρόπο λειτουργίας της, μέσω των βοηθητικών μηνυμάτων, παίρνει τις κατάλληλες κατευθύνσεις για να μαθαίνει όπως τη χρησιμοποιεί.

3.4.6 Λειτουργία προβολής ημερολογίου γεγονότων

Με τη χρήση της επιλογής «Προγραμματισμός» που βρίσκεται στον τομέα «Πρόγραμμα», εμφανίζεται στο χρήστη ένα ημερολόγιο με τα γεγονότα (σενάρια) που έχει καταχωρήσει. Υπάρχει η δυνατότητα περιήγησης στους επόμενους και προηγούμενους μήνες όπως και η δυνατότητα άμεσης επιστροφής στην τρέχουσα ημερομηνία (ημερομηνία στιγμιότυπου: 01/ 12/ 2012).

Οι ημερομηνίες οι οποίες έχουν προηγηθεί της τρέχουσας, έχουν κόκκινο περίγραμμα (εικόνα 78 β), η τρέχουσα έχει κίτρινο ενώ οι επόμενες έχουν ανοιχτό μπλε (εικόνα 78 α). Με τον τρόπο αυτό ο χρήστης μπορεί χωρίς να εστιάσει στο περιεχόμενο των κελιών να γνωρίζει που μπορεί να προσθέσει σενάρια και που όχι.

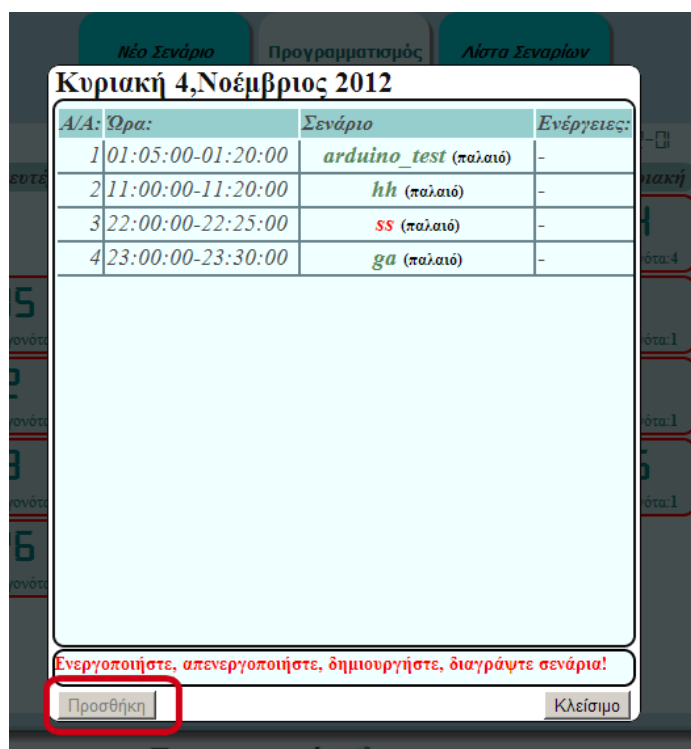


Εικόνα 78: Ημερολόγιο γεγονότων

3.4.7 Λειτουργία προβολής ημερήσιας λίστας γεγονότων

Ο χρήστης επιλέγοντας κάποια ημερομηνία από την προβολή ημερολογίου γεγονότων, μεταφέρεται στην αναλυτική λίστα με τα σενάρια που έχουν προγραμματιστεί. Η λίστα αυτή αφορά μόνο την ημερομηνία που επέλεξε. Τα σενάρια τα οποία είναι ενεργοποιημένα έχουν πράσινο χρώμα, ενώ εκείνα που είναι απενεργοποιημένα έχουν κόκκινο. Οι λειτουργίες που μπορούν να εκτελεστούν, χωρίζονται σε δύο κατηγορίες ανάλογα με το αν η επιλεγμένη ημερομηνία προηγείται ή όχι της τρέχουσας:

1. **Προηγούμενη ημερομηνία:** Στην περίπτωση που ο χρήστης επιλέξει ημερομηνία που προηγείται της τρέχουσας, τότε δε μπορεί, όπως είναι φυσικό να επεξεργαστεί τα σενάρια που βρίσκονται στη λίστα. Επίσης, η επιλογή προσθήκη σεναρίου είναι απενεργοποιημένη ενώ υπάρχει και σχετική σημείωση σε κάθε σενάριο ότι είναι «παλαιό», όπως απεικονίζεται στην εικόνα 79.



A/A:	Ωρα:	Σενάριο	Ενέργειες:
1	01:05:00-01:20:00	arduino_test (παλαιό)	-
2	11:00:00-11:20:00	hh (παλαιό)	-
3	22:00:00-22:25:00	ss (παλαιό)	-
4	23:00:00-23:30:00	ga (παλαιό)	-

Ενεργοποιήστε, απενεργοποιήστε, δημιουργήστε, διαγράψτε σενάρια!

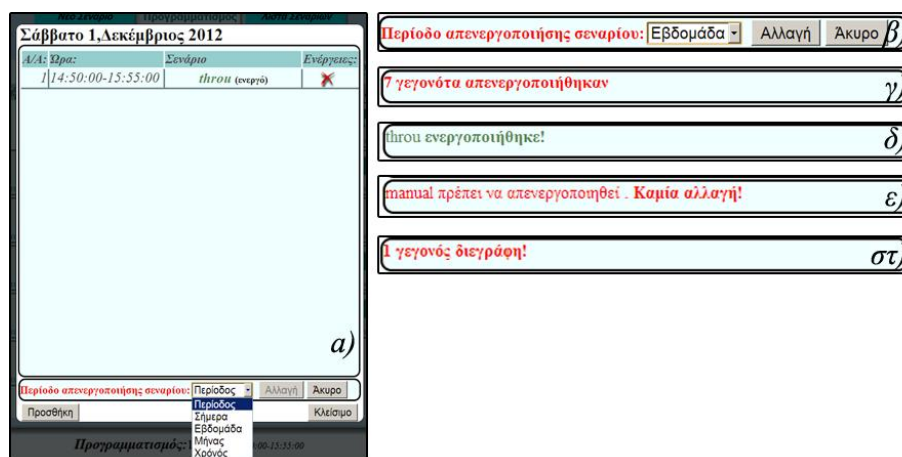
Προσθήκη Κλείσιμο

Εικόνα 79: Λίστα προηγούμενης ημερομηνίας

2. **Τρέχουσα ή επομένη ημερομηνία:** Στην περίπτωση που ο χρήστης επιλέξει την τρέχουσα ημερομηνία ή επόμενη της (εικόνα 80), τότε μπορεί στα σενάρια τα οποία εκτελούνται ή πρόκειται να εκτελεστούν, να εφαρμόσει συγκεκριμένες λειτουργίες, όπως:
 - i. Απενεργοποίηση σεναρίου: Στην περίπτωση αυτή, το σενάριο, το οποίο είναι αρχικά ενεργοποιημένο, επιλέγεται και εμφανίζεται μήνυμα για την περίοδο απενεργοποίησής του (εικόνα 80 α). Ανάλογα με την περίοδο που θα διαλέξει ο χρήστης ενημερώνεται εν συνεχεία για το πλήθος των σεναρίων που απενεργοποιήθηκαν (εικόνα 80 β & γ). Σε περίπτωση που δε διαλέξει περίοδο η διαδικασία δε μπορεί να ολοκληρωθεί.
 - ii. Ενεργοποίηση σεναρίου: Στην περίπτωση αυτή, το σενάριο, το οποίο είναι αδρανές αρχικά, επιλέγεται και αν δεν υπάρχει σενάριο που να επικαλύπτει ή να επικαλύπτεται χρονικά, ενεργοποιείται (εικόνα 80 δ)

ή σε αντίθετη περίπτωση εμφανίζεται αντίστοιχο μήνυμα ύπαρξης σεναρίου (εικόνα 80 ε).

- iii. Διαγραφή σεναρίου: Στην περίπτωση αυτή, το σενάριο, είτε είναι ενεργό είτε αδρανές, επιλέγεται και εμφανίζεται μήνυμα για την περίοδο διαγραφής του. Ανάλογα με την περίοδο που θα διαλέξει ο χρήστης ενημερώνεται εν συνεχεία για το πλήθος των σεναρίων που διαγράφηκαν (εικόνα 80 στ). Σε περίπτωση που δε διαλέξει περίοδο η διαδικασία δε μπορεί να ολοκληρωθεί.



Εικόνα 80: Λίστα τρέχουσας - επόμενη ημερομηνίας

Επίσης, ο χρήστης έχει τη δυνατότητα να προσθέσει σενάριο στη λίστα. Η διαδικασία αυτή αναλύεται σε παρακάτω υποενότητες.

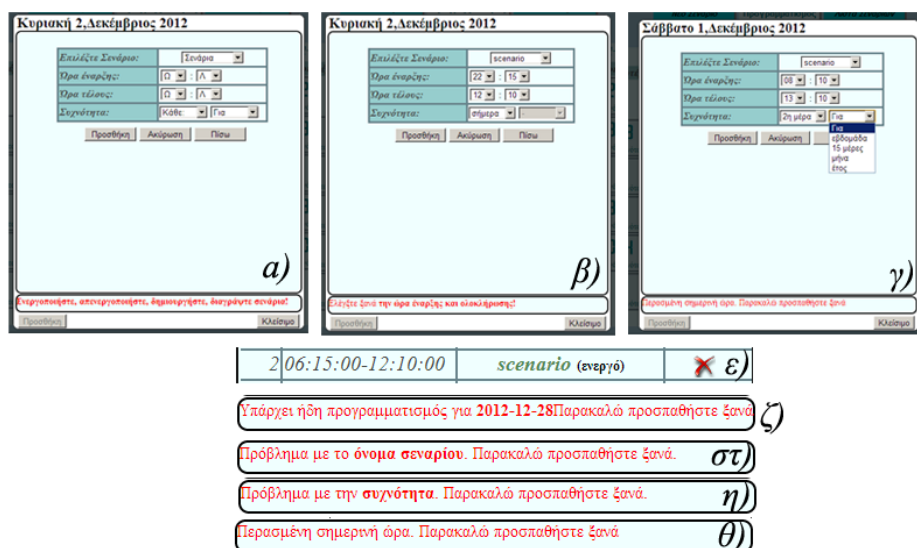
3.4.8 Λειτουργία προσθήκης σεναρίου στον προγραμματισμό

Ο χρήστης έχει τη δυνατότητα, εφόσον βρίσκεται σε τρέχουσα ή επόμενη ημερομηνία, να προγραμματίσει τα σενάρια τα οποία έχει φτιάξει. Η υλοποίησή της απαιτεί την επιλογή του πεδίου «Προσθήκη» από την προβολή ημερήσιας λίστας γεγονότων (εικόνα 81 α). Η διαδικασία η οποία χρειάζεται να εκτελεστεί αποτελείται από τη συμπλήρωση των επόμενων πεδίων:

1. **«Επιλέξτε Σενάριο»**: Ο χρήστης επιλέγει ποιο σενάριο θέλει να προγραμματίσει.
2. **«Ωρα έναρξης» - «Ωρα τέλους»**: Στα πεδία αυτά ο χρήστης δηλώνει τη χρονική περίοδο που θα λάβει χώρα η εκτέλεση του σεναρίου. Παρακάτω εξετάζονται οι περιορισμοί που υπάρχουν κατά τη συμπλήρωση των πεδίων αυτών:
 - i. Συνέπεια έναρξης και λήξης: Σε περίπτωση που δεν υπάρχει χρονική συνέπεια μεταξύ της εναρκτήριας ώρας και του τερματισμού, τότε εμφανίζεται αντίστοιχο μήνυμα (εικόνα 81 β).
 - ii. Συνέπεια ύπαρξης ενός μόνο σεναρίου: Σε περίπτωση που ο χρήστης θέσει κάποιο χρονικό περιθώριο που επικαλύπτει ή επικαλύπτεται από άλλο σενάριο υπάρχει αντίστοιχο μήνυμα (εικόνα 81 ζ).
 - iii. Συνέπεια τρέχουσας ώρας: Σε περίπτωση που ο χρήστης προγραμματίσει για την τρέχουσα ημερομηνία, αλλά για περασμένη ώρα, εμφανίζεται αντίστοιχο μήνυμα (εικόνα 81 θ).
3. **«Συχνότητα»**: Στο πεδίο αυτό ο χρήστης επιλέγει αν πρόκειται για προγραμματισμό που αντιστοιχεί μόνο στη συγκεκριμένη ημερομηνία ή αν πρόκειται για προγραμματισμό που θα εκτελείται επαναλαμβανόμενα κάποιο

διάστημα (εικόνα 81 γ). Σε περίπτωση που ένα σενάριο προγραμματίζεται εξακολουθητικά, τότε για όποια ημερομηνία υπάρχει ήδη καταχώριση εμφανίζεται το αντίστοιχο μήνυμα (εικόνα 81 ζ).

Η διαδικασία ολοκληρώνεται με την επιλογή «Αποθήκευση» (εικόνα 81 ε). Αν ο χρήστης επιλέξει είτε «Ακύρωση» είτε «Πίσω» δε γίνεται καμία καταχώριση στο ημερολόγιο. Επίσης, αν ο χρήστης δε συμπληρώσει κάποιο από τα παραπάνω πεδία υπάρχει αντίστοιχο μήνυμα (εικόνα 81 στ & η).

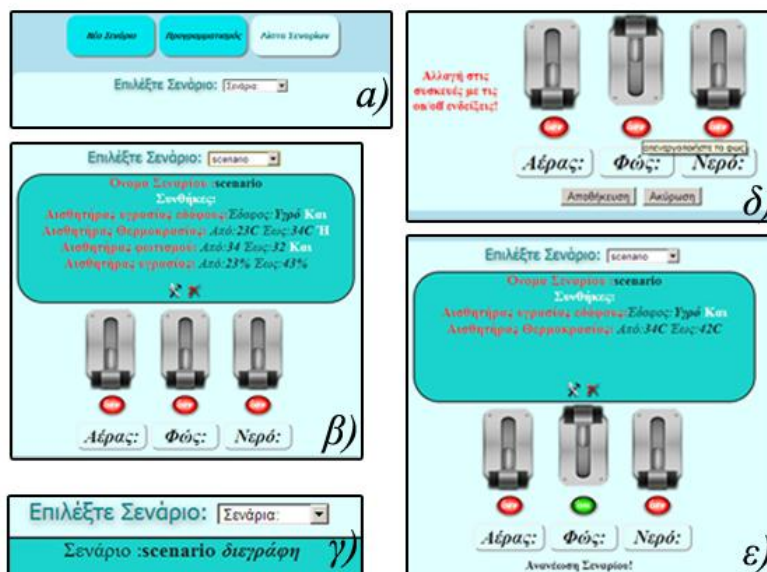


Εικόνα 81: Προσθήκη σεναρίου

3.4.9 Λειτουργία προβολής – επεξεργασίας σεναρίου

Το πεδίο «Λίστα Σεναρίων» προσφέρει στο χρήστη τη δυνατότητα να προβάλει και να επεξεργάζεται τα σενάρια που έχει δημιουργήσει. Με αυτόν τον τρόπο δε θα χρειάζεται να επαναλαμβάνει τη διαδικασία δημιουργίας νέου σεναρίου, ενώ θα μπορεί για ήδη προγραμματισμένα σενάρια να κάνει αλλαγές χωρίς να επηρεάζεται ο προγραμματισμός τους. Οι δυνατότητες που δίνει το πεδίο αυτό είναι οι εξής:

1. **Προβολή:** Ο χρήστης εφόσον επιλέξει από την λίστα κάποιο σενάριο (εικόνα 82 α), τότε εκείνο προβάλλεται όπως στο 2^ο βήμα της δημιουργίας σεναρίου (εικόνα 82 β). Στην προβολή ο χρήστης δε μπορεί να κάνει κάποια αλλαγή.



Εικόνα 82: Προβολή - επεξεργασία - διαγραφή σεναρίου

2. **Επεξεργασία:** Ο χρήστης εφόσον επιλέξει την επεξεργασία από το χαρακτηριστικό εικονίδιο, μπορεί να πραγματοποιήσει αλλαγές είτε στις συνθήκες των αισθητήρων είτε στις συσκευές δράσης. Πιο αναλυτικά:
 - i. Επεξεργασία αισθητήρων: Δίπλα από τη γραμμή κάθε αισθητήρα εμφανίζεται η επιλογή «Αλλαγή» και «X» (διαγραφή), μέσω των οποίων μπορεί είτε να αλλάξει τα όρια στον αισθητήρα και τη σχέση του με τον επόμενο είτε να διαγράψει τη συνθήκη εντελώς (εικόνα 83 α & γ). Σε περίπτωση που διαγράψει τη συνθήκη, τότε η σχέση του προηγούμενου με τον επόμενο αισθητήρα μένει ως έχει συμπληρωθεί (εικόνα 83 β). Αν η συνθήκη που διαγράφεται είναι η τελευταία, τότε στην προηγούμενή της διαγράφεται η μεταβλητή λογικής πράξης (εικόνα 83 δ). Σε κάθε πεδίο που απαιτείται συμπλήρωση υπάρχουν και οι αντίστοιχοι έλεγχοι περιεχομένου όπως ακριβώς και στο 1^ο βήμα της δημιουργίας σεναρίου, ενώ δεν υπάρχει επιλογή προσθήκης αισθητήρα (εικόνα 83 ε).



Εικόνα 83: Επεξεργασία αισθητήρων

- ii. Επεξεργασία συσκευών: Ο χρήστης μπορεί ενεργοποιώντας ή απενεργοποιώντας τους μοχλούς των συσκευών να αλλάξει την

κατάσταση τους. Η διαδικασία είναι όμοια του 2^{ου} βήματος της δημιουργίας σεναρίου (εικόνα 82 δ).

3. **Διαγραφή:** Ο χρήστης μπορεί να διαγράψει το σενάριο που έχει επιλέξει για προβολή επιλέγοντας το «X» (διαγραφή) όπως φαίνεται στην εικόνα 82 γ). Μόνο το σενάριο manual δε μπορεί να διαγραφεί όπως έχει αναφερθεί και παραπάνω.

Για την αποθήκευση των αλλαγών ο χρήστης επιλέγει την «Αποθήκευση». Η ενημέρωση είναι άμεση (εικόνα 82 ε). Σε περίπτωση που ο χρήστης επιλέξει είτε «Ακύρωση» είτε δεν επιλέξει κάτι, τότε οι αλλαγές στο σενάριο δεν αποθηκεύονται.

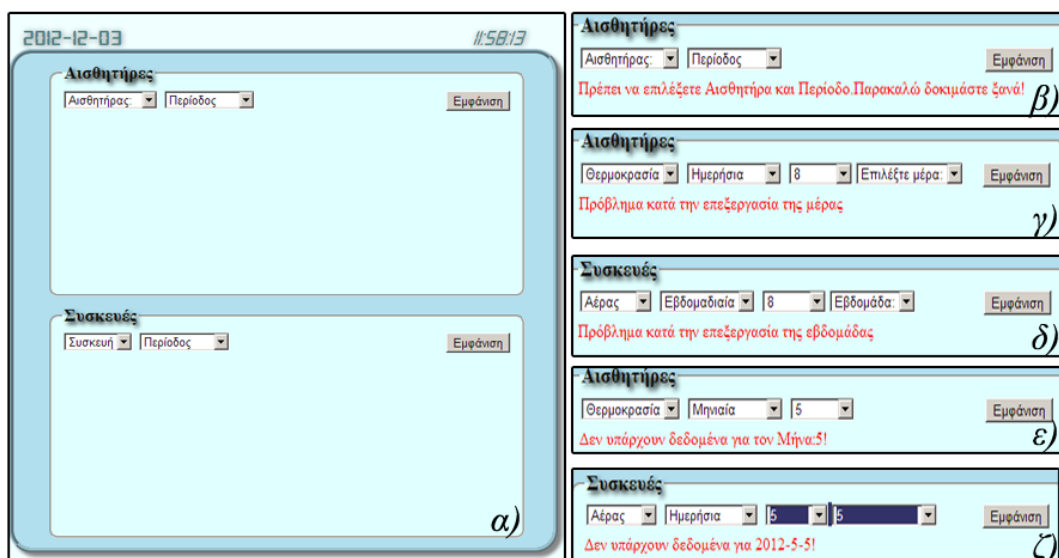
3.4.10 Λειτουργία διαγραμμάτων χρήσης

Στον τομέα «Διαγράμματα» ο χρήστης έχει τη δυνατότητα της προβολής στατιστικών στοιχείων σχετικά με τις συσκευές δράσης και τους αισθητήρες (εικόνα 84 α). Η λειτουργία αυτή είναι ιδιαίτερα σημαντική για δύο λόγους:

1. Ο χρήστης μπορεί να συγκρίνει τα στατιστικά στοιχεία των αισθητήρων, αλλά και των συσκευών δράσης και από σοδειά σε σοδειά να βελτιστοποιεί την παραγωγή βάση της συνολικής λειτουργίας του θερμοκηπίου. Επίσης, μπορεί ταυτόχρονα να καλλιεργεί τα ίδια φυτά, αλλά με διαφορετική αντιμετώπιση από ομάδα σε ομάδα και να διαγνώσει την καλύτερη μέθοδο ανάπτυξής τους.
2. Ο χρήστης σε περίπτωση που θέλει να καλλιεργήσει εποχιακά φυτά σε διαφορετικές περιόδους, τότε μπορεί να πάρει πληροφορίες σχετικά με τις κατάλληλες συνθήκες επώασής τους και να προσπαθήσει να τις προσομοιώσει στο θερμοκήπιο. Με αυτόν τον τρόπο μπορεί να έχει καλοκαιρινά φρούτα και λαχανικά το χειμώνα και αντίστροφα το καλοκαίρι.

Για να γίνει εμφάνιση των διαγραμμάτων είτε για τους αισθητήρες είτε για τις συσκευές δράσης, ο χρήστης πρέπει να βρίσκεται στον τομέα «Διαγράμματα». Εφόσον, συμβαίνει αυτό η διαδικασία που πρέπει να ακολουθήσει είναι η εξής:

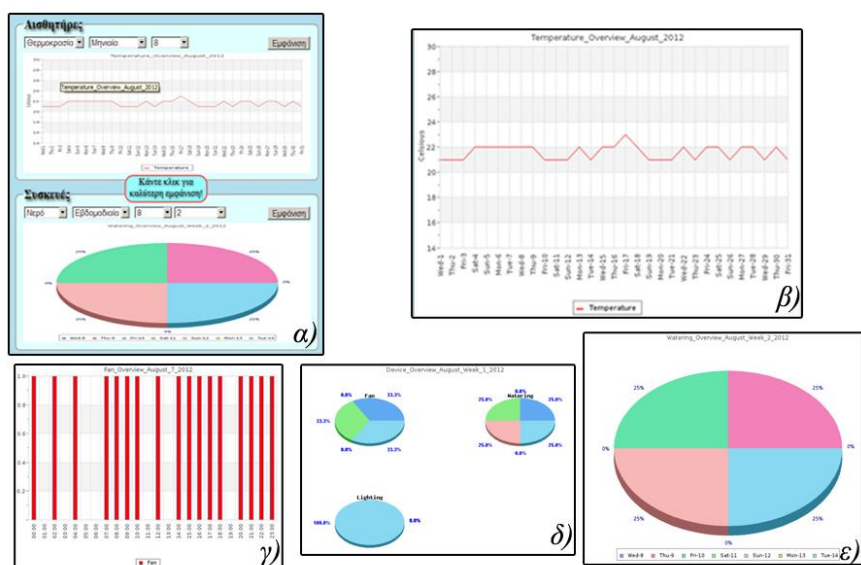
1. **Βήμα 1^ο:** Επιλογή αισθητήρα ή αντίστοιχα συσκευής, καθώς και περιόδου. Αν ο χρήστης δεν επιλέξει κάποιο από τα δύο πεδία, τότε με την επιλογή «Εμφάνιση» θα του εμφανιστεί το αντίστοιχο μήνυμα (εικόνα 84 β).
2. **Βήμα 2^ο:** Ο χρήστης ανάλογα με την περίοδο που έχει διαλέξει πρέπει να προσδιορίσει συγκεκριμένες λεπτομέρειες. Ακολουθεί αναλυτική περιγραφή:
 - i. Περίοδος Ημερήσια: Εμφανίζεται πεδίο όπου επιλέγεται ο μήνας. Εφόσον επιλεγθεί ο μήνας εμφανίζεται πεδίο που επιλέγεται συγκεκριμένη μέρα.
 - ii. Περίοδος Εβδομαδιαία: Εμφανίζεται πεδίο όπου επιλέγεται ο μήνας. Εφόσον επιλεγθεί ο μήνας εμφανίζεται πεδίο που επιλέγεται συγκεκριμένη εβδομάδα.
 - iii. Περίοδος Μηνιαία: Εμφανίζεται πεδίο όπου επιλέγεται ο μήνας.
 - iv. Περίοδος Ετήσια: Εμφανίζεται πεδίο όπου επιλέγεται ο χρόνος, βάση τα καταχωρημένα έτη λειτουργίας της συσκευής.
3. **Βήμα 3^ο:** Ο χρήστης επιλέγει το πεδίο «Εμφάνιση». Σε περίπτωση που κάποιο από τα παραπάνω πεδία δεν έχει συμπληρωθεί δεν εξάγεται κάποιο διάγραμμα, αλλά εμφανίζεται μήνυμα λάθους (εικόνα 84 γ, δ). Επίσης, στην περίπτωση που δεν υπάρχουν καταχωρήσεις για τη συγκεκριμένη περίοδο εμφανίζεται αντίστοιχο μήνυμα (εικόνα 84 ε, ζ).



Εικόνα 84: Διαγράμματα διαδικασία

Εφόσον δεν υπάρχει κάποιο σφάλμα κατά την εκτέλεση των βημάτων και υπάρχουν δεδομένα προς προβολή, εμφανίζεται το διάγραμμα (εικόνα 85 α). Ο χρήστης έχει τη δυνατότητα μεγιστοποίησης του για καλύτερη προβολή. Η μορφή των διαγραμμάτων αυτών αλλάζει ανάλογα με τι δεδομένα που εμφανίζονται και την περίοδο που αντιστοιχούν. Γι' αυτό και υπάρχουν οι παρακάτω περιπτώσεις:

1. Διαγράμματα αισθητήρων: Τα διαγράμματα αυτά είναι γραμμικά (εικόνα 85 β). Στην περίπτωση που ο χρήστης επιλέξει την εμφάνιση όλων των αισθητήρων, τότε το διάγραμμα που εμφανίζεται έχει διαφορετικούς κάθετους άξονες αντίστοιχα με τα δεδομένα τους καθενός.
2. Διαγράμματα συσκευών δράσης: Η εμφάνιση των διαγραμμάτων αυτών διαφέρει ανάλογα την περίοδο. Όταν πρόκειται για την ημερήσια εμφάνιση χρήσης τους, οι τιμές είναι απόλυτες δηλαδή ή ενεργοποιημένες ή απενεργοποιημένες (εικόνα 85 γ). Σε αντίθετη περίπτωση τα διαγράμματα είναι υπό μορφή «πίτας» και εμφανίζουν το κατανομημένο ποσοστό χρήσης τους (εικόνα 85 ε). Όπως και στα διαγράμματα αισθητήρων, με την επιλογή όλων των συσκευών γίνεται ταυτόχρονη εμφάνιση των διαγραμμάτων (εικόνα 85 δ).



Εικόνα 85: Διαγράμματα προβολή

Προγραμματιστικά, η εμφάνιση των διαγραμμάτων δεν απαιτεί τη φόρτωση εκ νέου της σελίδας, καθώς χρησιμοποιείται η μέθοδος AJAX. Με αυτόν τον τρόπο η εμφάνιση γίνεται άμεσα ενώ σε περίπτωση που το διάγραμμα έχει προβληθεί και στο παρελθόν, δεν ξανασχεδιάζεται, αλλά φορτώνεται αυτόματα από τη βάση.

3.4.11 Λειτουργία προβολής και επεξεργασίας στοιχείων χρήστη

Ο χρήστης έχει τη δυνατότητα, στον τομέα «Προφίλ», να προβάλει και να επεξεργαστεί τα προσωπικά του στοιχεία. Η απαίτηση αυτή όπως είχε διατυπωθεί παραπάνω (παράγραφος 3.1), ικανοποιείται μέσω της επιλογής «Στοιχεία Χρήστη» (εικόνα 85 α). Εφόσον ο χρήστης επιλέξει το πεδίο αυτό, φορτώνονται τα προσωπικά του στοιχεία όπως το email του και το πλήθος των μηνυμάτων που έχει στείλει στο διαχειριστή.

Εκτός από την προβολή των παραπάνω πεδίων ο χρήστης μπορεί να επεξεργαστεί τα στοιχεία του, όπως τον κωδικό και την ηλεκτρονική του διεύθυνση. Η διαδικασία είναι η εξής:

Αρχικά ο χρήστης πρέπει να πληκτρολογήσει τον παλιό του κωδικό. Ο λόγος ύπαρξης του ελέγχου αυτού, είναι η ταυτοποίηση του χρήστη εκ νέου, καθώς υπάρχει η περίπτωση να υπάρχει αποθηκευμένος ο κωδικός στο browser και να μπαίνει στον ιστοχώρο όποιος χρήστης βρίσκεται στο συγκεκριμένο υπολογιστή. Για αυτό πραγματοποιείται ξανά ταυτοποίηση, ώστε να εξασφαλιστεί ότι ο χρήστης είναι ενήμερος για τις αλλαγές που ακολουθούν. Αν ο κωδικός είναι σωστός, τότε μπορούν να γίνουν αλλαγές όπως φαίνεται στην εικόνα 85 γ, διαφορετικά τα πεδία που συμπληρώνονται μένουν ανενεργά (εικόνα 85 β).

Οι αλλαγές που μπορούν να γίνουν είναι τροποποίηση κωδικού και email. Στην πρώτη περίπτωση ο χρήστης εισάγει το νέο κωδικό και παράλληλα παρουσιάζεται και το επίπεδο δυσκολίας του κωδικού (εικόνα 85 ε). Εν συνεχεία πληκτρολογεί ξανά το νέο κωδικό για επιβεβαίωση, ώστε να είναι σίγουρος για το περιεχόμενό του. Μόνο στην περίπτωση που επιβεβαιωθεί μπορεί να αποθηκεύσει τις αλλαγές (εικόνα 85 δ).

Στη δεύτερη περίπτωση ο χρήστης μπορεί να αλλάξει το email του. Το περιεχόμενο που καλείται να συμπληρώσει ο χρήστης ελέγχεται, ώστε να είναι μόνο υπό μορφή email (εικόνα 85 ζ). Μόνο τότε μπορεί να αποθηκεύσει τις αλλαγές.

Επίσης, ο χρήστης μπορεί να αλλάξει και τον κωδικό και το email του ταυτόχρονα. Πάλι οι έλεγχοι που περιγράφηκαν παραπάνω ισχύουν, ώστε να αποθηκευτούν οι αλλαγές.

Για να ενημερωθεί η βάση για τις αλλαγές ο χρήστης επιλέγει το πεδίο «Αλλαγή». Τότε αποθηκεύονται οι αλλαγές στη βάση και εμφανίζεται το αντίστοιχο μήνυμα, ενώ σε περίπτωση που έχει αλλαχθεί το email του χρήστη η αλλαγή είναι άμεση στον πίνακα που εμφανίζεται (εικόνα 85 στ). Αν ο χρήστης επιλέξει «Άκυρο» ή δεν επιλέξει τίποτα, τότε δεν αποθηκεύονται οι αλλαγές στη βάση.

Στοιχεία Χρήστη
Ρυθμίσεις Συσκευής
Επικοινωνία
Σφάλματα

Email:	gnnpls.nikos@gmail.com	
Απεσταλμένα:	11	
Ισχύον Κωδικός:		Πατήστε enter
Νέος Κωδικός:		
Επιβεβαίωση:		
Νέο email:		Πατήστε enter

Αλλαγή Ακύρω

α)

Email:	gnnpls.nikos@gmail.com	
Απεσταλμένα:	11	
Ισχύον Κωδικός:		Κωδικός Λάθος!
Νέος Κωδικός:		
Επιβεβαίωση:		
Νέο email:		Πατήστε enter

Αλλαγή Ακύρω

β)

Νέος Κωδικός:		Κωδικός μέτριος! ε)
Νέο email:	wrong@wrong	Μη έγκυρο email!

Email:	gnnpls.nikos@gmail.com	
Απεσταλμένα:	11	
Ισχύον Κωδικός:		Κωδικός Δεκτός!
Νέος Κωδικός:		
Επιβεβαίωση:		
Νέο email:		Πατήστε enter

Αλλαγή Ακύρω

γ)

Email:	new_mail@zafora.con	
Απεσταλμένα:	11	
Ισχύον Κωδικός:		Πατήστε enter
Νέος Κωδικός:		
Επιβεβαίωση:		
Νέο email:		Πατήστε enter

Αλλαγή Ακύρω

Ενημέρωση κωδικού!Ενημέρωση email!

στ)

Email:	gnnpls.nikos@gmail.com	
Απεσταλμένα:	11	
Ισχύον Κωδικός:		Κωδικός Δεκτός!
Νέος Κωδικός:		Κωδικός καλός!
Επιβεβαίωση:		Δεκτό!
Νέο email:		Πατήστε enter

Αλλαγή Ακύρω

δ)

Εικόνα 86: Προβολή επεξεργασία στοιχείων χρήστη

3.4.12 Λειτουργία αυτόματης επικοινωνίας συστήματος με χρήστη

Στις απαιτήσεις χρήστη, που αναλύθηκαν παραπάνω (παράγραφος 3.1), αναφέρεται η απαίτηση για αλληλεπίδραση του συστήματος με τον κάτοχο του θερμοκηπίου δίχως τη συνεχή παρουσία του στον ιστόχωρο. Η δυνατότητα αυτή δίνεται, χωρίς να είναι αυτόματα επιλεγμένη, αλλά ο χρήστης έχει τη δυνατότητα να επιλέξει πως θα γίνεται μέσω του πεδίου «Ρυθμίσεις Συσκευής» που βρίσκεται στον τομέα «Προφίλ», όπως απεικονίζεται στην εικόνα 87. Μέσω των ρυθμίσεων αυτών υλοποιείται η λειτουργία αυτόματης επικοινωνίας συστήματος με το χρήστη.

Στοιχεία Χρήστη
Ρυθμίσεις Συσκευής
Επικοινωνία
Σφάλματα

Ρυθμίσεις αισθητήρων:	Ειδοποιήσεις: ☐ Ξαφνική ατμοσφαιρική αλλαγή ☒ Αισθητήρας εκτός λειτουργίας Αποστολή αναφοράς κάθε: Ημέρα
Ρυθμίσεις συσκευών:	Αποστολή αναφοράς κάθε: Εβδομάδα
Γενικά:	☒ Απώλεια σύνδεσης
Επικοινωνία:	Επαφή μέσω ηλεκτρονικής αλληλογραφίας

Αποθήκευση Ακύρωση

Εικόνα 87: Ρυθμίσεις συσκευής

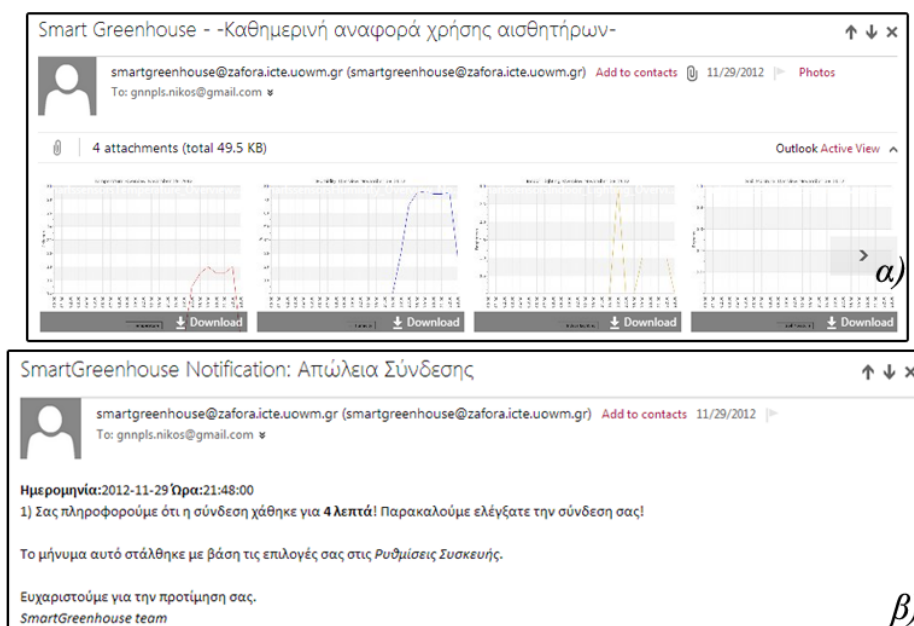
Το σύστημα έχει κάποιες προκαθορισμένες επιλογές για επικοινωνία με το χρήστη. Κάθε φορά που εισέρχεται στο πεδίο αυτό εμφανίζονται οι προεπιλεγμένες ρυθμίσεις που έχει ορίσει ο ίδιος. Υπάρχουν δύο κατηγορίες μηνυμάτων που

στέλνονται στην ηλεκτρονική διεύθυνση του κατόχου του θερμοκηπίου. Οι κατηγορίες αυτές είναι οι εξής:

1. **Μηνύματα σφαλμάτων:** Τα μηνύματα αυτά τα λαμβάνει ο χρήστης αν υπάρχει:
 - i. Ξαφνική ατμοσφαιρική αλλαγή: Σε περίπτωση που διαδοχικές μετρήσεις αισθητήρα παρουσιάσουν μεγάλη διαφορά, τότε είτε υπάρχει πρόβλημα στο κύκλωμα είτε υπάρχει πρόβλημα στον περιβάλλοντα χώρο του θερμοκηπίου.
 - ii. Αισθητήρας εκτός λειτουργίας: Σε περίπτωση που στη βάση καταχωρηθεί τιμή σφάλματος αισθητήρα (αναλυτικά: παράγραφος 3.5.2).
 - iii. Απώλεια σύνδεσης: Σε περίπτωση που οι διαδοχικές μετρήσεις έχουν διαφορά στην ενημέρωση τους ανώτερη του ενός λεπτού.
2. **Μηνύματα αναφοράς:** Τα μηνύματα αναφοράς περιέχουν τα διαγράμματα είτε των αισθητήρων είτε των συσκευών δράσης. Ο χρήστης μπορεί να τα λαμβάνει σε ημερήσια, εβδομαδιαία, μηνιαία και ετήσια βάση. Φυσικά υπάρχει και η επιλογή να μην τα λαμβάνει.

Χαρακτηριστικά, στην εικόνα 87, ο χρήστης έχει επιλέξει να λαμβάνει email όταν κάποιος αισθητήρας είναι εκτός λειτουργίας ή όταν χαθεί η σύνδεση. Επίσης, έχει επιλέξει να λαμβάνει ημερήσιες αναφορές αισθητήρων και εβδομαδιαίες αναφορές για τις συσκευές δράσης. Οι αλλαγές γίνονται με απλή επιλογή των πεδίων και εφόσον στο τέλος επιλεγεί η «Αποθήκευση». Αν ο χρήστης επιλέξει είτε «Άκυρο» είτε δεν επιλέξει κάτι, δεν αποθηκεύονται οι αλλαγές.

Ο χρήστης επιλέγει αν θέλει να έχει αλληλεπίδραση με τη συσκευή και σε ποιες περιπτώσεις. Με αυτόν τον τρόπο ο χρήστης δε χρειάζεται να επισκέπτεται συνεχώς τον ιστοχώρο διότι λαμβάνει τις ειδοποιήσεις που τον ενδιαφέρουν μέσω των email, ενώ σε περίπτωση που δε θέλει να λαμβάνει μηνύματα δεν επιβαρύνεται με ειδοποιήσεις που δεν τις θέλει. Στην εικόνα 88 α), απεικονίζεται ένα email αναφοράς χρήσης αισθητήρων σε καθημερινή βάση, ενώ στην εικόνα 88 β) απεικονίζεται ένα μήνυμα σφάλματος και πιο συγκεκριμένα τύπου απώλειας σύνδεσης.

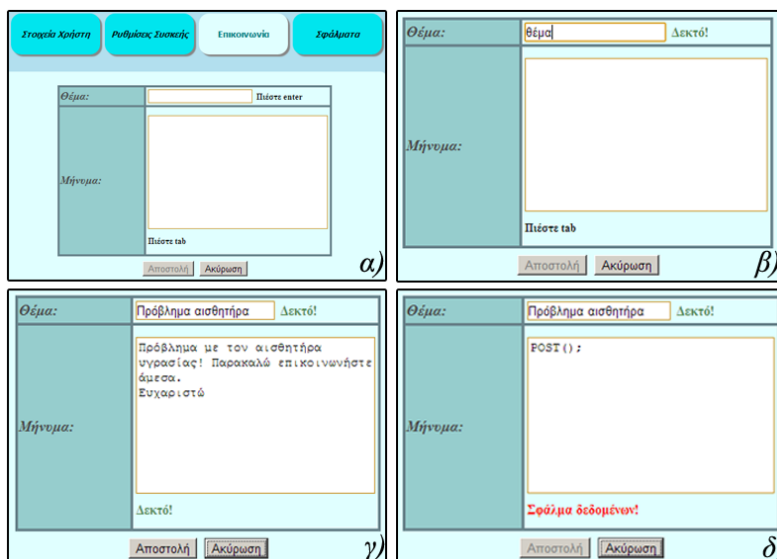


Εικόνα 88: Email συστήματος

3.4.13 Λειτουργία επικοινωνίας με το διαχειριστή

Ο χρήστης έχει τη δυνατότητα να επικοινωνεί με το διαχειριστή. Η λειτουργία αυτή έχει αναπτυχθεί ως προς τη πλευρά του διαχειριστή (παράγραφος 3.3.6). Μέσω του πεδίου «Επικοινωνία» που βρίσκεται στον τομέα «Προφίλ» (εικόνα 89 α), ο χρήστης μπορεί να στείλει μηνύματα στο διαχειριστή με τυχόν απορίες ή αναφορές σφαλμάτων.

Ο χρήστης πρέπει να συμπληρώσει τα θέμα και το περιεχόμενο του μηνύματος. Αν δε συμπληρώσει κάποιο από τα δύο, τότε δε μπορεί να σταλεί το μήνυμα (εικόνα 89 β). Επίσης, υπάρχουν έλεγχοι ως προς το περιεχόμενο, ώστε να μην υπάρχει περίπτωση κακόβουλης επίθεσης. Οι έλεγχοι γίνονται τόσο σε επίπεδο javascript όσο και σε php, πριν καταγραφεί το μήνυμα στη βάση. Αν κάποιο πεδίο δεν είναι έγκυρο, τότε δε μπορεί να σταλεί το μήνυμα (εικόνα 89 δ). Εφόσον όμως και τα δύο πεδία είναι έγκυρα και συμπληρωμένα ο χρήστης επιλέγοντας «Αποστολή», στέλνει το μήνυμα (εικόνα 89 γ). Αν επιλέξει «Άκυρο» ή δεν επιλέξει τίποτα, το μήνυμα δε στέλνεται.



α) **Θέμα:** **Μήνυμα:**

β) **Θέμα:** **Μήνυμα:**

γ) **Θέμα:** **Μήνυμα:**

δ) **Θέμα:** **Μήνυμα:**

Εικόνα 89: Επικοινωνία με το διαχειριστή

3.4.14 Λειτουργία προβολής σφαλμάτων

Ο χρήστης έχει τη δυνατότητα να ελέγχει για τυχόν σφάλματα της συσκευής. Με αυτόν τον τρόπο γνωρίζει για το πότε και τι σφάλμα συνέβη με απόρροια την καλύτερη εξαγωγή σφαλμάτων για τη χρηστικότητα της, αλλά και για τυχόν προτεινόμενες προς τους διαχειριστές βελτιώσεις. Η λειτουργία αυτή είναι όμοια με εκείνη για το διαχειριστή που έχει αναλυθεί παραπάνω (παράγραφος 3.3.5). Η μόνη διαφορά είναι στο γεγονός ότι εδώ προβάλλονται μόνο τα σφάλματα του χρήστη που έχει πρόσβαση στον ιστόχωρο και όχι όπως είναι φυσικό όλων των χρηστών. Εκτός αυτού οι υπόλοιπες δυνατότητες είναι ίδιες.

Στοιχεία Χρήστη			
Ρυθμίσεις Συσκευής			
Επικοινωνία			
Σφάλματα			
Πρόβλημα	Επιλέξτε τύπο	Συνολικά: 13	
A/A	Ημερομηνία	Ωρα	Περιγραφή
1	2012-12-01	12:18:00	Απώλεια επικοινωνίας για 3 λεπτά
2	2012-11-30	22:44:00	Απώλεια επικοινωνίας για 67 λεπτά
3	2012-11-30	15:18:00	Απώλεια επικοινωνίας για 3 λεπτά
4	2012-11-30	10:18:00	Απώλεια επικοινωνίας για 4 λεπτά
5	2012-11-29	21:48:00	Απώλεια επικοινωνίας για 4 λεπτά
6	2012-11-29	16:49:00	Απώλεια επικοινωνίας για 4 λεπτά
7	2012-11-29	14:10:00	Απώλεια επικοινωνίας για 15 λεπτά
8	2012-11-29	11:03:00	Απώλεια επικοινωνίας για 3 λεπτά

Εικόνα 90: Προβολή σφαλμάτων χρήστη

3.5 Προγραμματισμός αισθητήρων, συσκευών δράσης και Arduino

Στην ενότητα αυτή θα γίνει αναφορά για τον τρόπο με τον οποίο η συσκευή αντιλαμβάνεται τις περιβαλλοντικές συνθήκες και δρα σε αυτές. Επίσης, θα περιγραφεί ο τρόπος με τον οποίο επικοινωνεί με τη βάση δεδομένων και κατ' επέκταση συνολικά με τον ιστοχώρο. Για καλύτερη κατανόηση του κώδικα που θα παρατεθεί θα γίνει αρχικά αναφορά σε συναρτήσεις και εργαλεία που παρείχε το Arduino IDE και συνολικά η αρχιτεκτονική Arduino.

3.5.1 Βασικές συναρτήσεις, βιβλιοθήκες και εργαλεία

Η αρχιτεκτονική Arduino βασίζεται σε κώδικα C. Επομένως υπάρχουν πολλές βιβλιοθήκες, συναρτήσεις και εργαλεία τα οποία χρησιμοποιούνται για την ανάπτυξη κώδικα. Εκτός αυτών λόγω της ιδιαιτερότητας του προγραμματισμού που απαιτεί η αρχιτεκτονική Arduino, προσφέρονται από το σύστημα της και πρόσθετα στοιχεία. Παρακάτω γίνεται αναφορά σε ότι αξιοποιήθηκε προγραμματιστικά κατά την ανάπτυξη του κώδικα του συστήματος “Smart Greenhouse” όσον αφορά το προγραμματισμό των αισθητήρων, συσκευών δράσης και Arduino:

setup(): Η συνάρτηση `setup()` καλείται όταν ξεκινά το πρόγραμμα. Χρησιμοποιείται για να προετοιμάσει τις μεταβλητές, τους τρόπους με τους οποίους συνδέονται τα pin κλπ. Εκτελείται μόνο μία φορά, μετά από κάθε ενεργοποίηση ή επανεκκίνηση του Arduino.

loop(): Η συνάρτηση `loop()` είναι ένας διαδοχικός βρόχος, που επιτρέπει στο πρόγραμμα να επαναλαμβάνεται.

#include <Ethernet.h>: Πρόκειται για βιβλιοθήκη που επιτρέπει σε μια πλακέτα Arduino να συνδέεται στο διαδίκτυο. Μπορεί να χρησιμεύσει είτε ως διακομιστής που δέχεται εισερχόμενες συνδέσεις είτε ως πελάτης. Η βιβλιοθήκη υποστηρίζει έως και τέσσερις ταυτόχρονες συνδέσεις (εισερχόμενες, εξερχόμενες ή συνδυασμός). Παρακάτω γίνεται αναλυτική περιγραφή των δομών και εντολών που χρησιμοποιήθηκαν:

1) **Ethernet.begin():** Αρχικοποιεί τη βιβλιοθήκη ethernet και τις ρυθμίσεις του δικτύου. Με την έκδοση 1.0, η βιβλιοθήκη υποστηρίζει DHCP. Χρησιμοποιώντας Ethernet.begin(mac) με τη σωστή εγκατάσταση του δικτύου, το Ethernet shield αποκτά αυτόματα μια διεύθυνση IP. Αυτό όμως αυξάνει το μέγεθος προγράμματος σημαντικά.

- Σύνταξη:
 - Ethernet.begin (mac);
 - Ethernet.begin (mac, ip);
 - Ethernet.begin (mac, ip, dns);
 - Ethernet.begin (mac, ip, dns, gateway);
 - Ethernet.begin (mac, ip, dns, gateway, subnet);
- Παράμετροι:
 - mac: η MAC (Media Access Control) διεύθυνση της συσκευής (πίνακας 6 bytes) αποτελεί την Ethernet διεύθυνση υλικού του shield. Τα νεότερα Arduino Shields Ethernet περιλαμβάνουν ένα αυτοκόλλητο με τη διεύθυνση MAC της συσκευής.
 - ip: η διεύθυνση IP της συσκευής (πίνακας 4 bytes).
 - DNS: Η διεύθυνση για ένα διακομιστή DNS.
 - gateway: η διεύθυνση IP της πύλης του δικτύου (πίνακας 4 bytes).
 - subnet: η μάσκα υποδικτύου του δικτύου (πίνακας 4 bytes).

2) **Ethernet.localIP():** Αποκτά την IP διεύθυνση του Ethernet Shield. Είναι χρήσιμο όταν η διεύθυνση αποδίδεται αυτόματα μέσω του DHCP.

- Σύνταξη: Ethernet.localIP();

3) **IPAddress ():** Ορίζει μια διεύθυνση IP. Μπορεί να χρησιμοποιηθεί για να δηλώσει τόσο τις τοπικές όσο και τις απομακρυσμένες διευθύνσεις.

- Σύνταξη: IPAddress (address);
- Παράμετροι:
 - address: μια λίστα οριοθετημένη με κόμματα που αντιπροσωπεύει τη διεύθυνση (4 bytes, πχ 192, 168, 1, 1).

4) **Server():** Δημιουργεί ένα διακομιστή που ακούει για εισερχόμενες συνδέσεις στη συγκεκριμένη θύρα.

- Σύνταξη: Server(port);
- Παράμετροι:
 - port: Η θύρα στην οποία ακούει (int).

5) **begin():** Ξεκινά η διαδικασία από την πλευρά του διακομιστή για εύρεση νέων συνδέσεων.

- Σύνταξη: server.begin();

6) **available():** Παίρνει δεδομένα που είναι διαθέσιμα για ανάγνωση από έναν πελάτη που συνδέεται με το διακομιστή. Η σύνδεση παραμένει όταν τα επιστρεφόμενα δεδομένα από τον πελάτη τελειώσουν.

- Σύνταξη: server.available();

7) **write():** Γράφει δεδομένα για όλους τους πελάτες που συνδέονται σε έναν διακομιστή.

- Σύνταξη: server.write(data);
- Παράμετροι:
 - data: η τιμή που γράφει (byte ή char).

8) **print() και println():** Εκτυπώνει δεδομένα σε όλους τους πελάτες που συνδέονται σε έναν διακομιστή. Η println() προσθέτει και μια αλλαγή γραμμής.

- Σύνταξη:
 - server.print(data);
 - server.print (data, BASE);
- Παράμετροι:
 - data: τα δεδομένα προς εκτύπωση (char, byte, int, long, ή string).
 - BASE (προαιρετικό): η βάση στην οποία θα εκτυπωθούν οι αριθμοί: BIN για δυαδικό (βάση 2), DEC για δεκαδικό (βάση 10), OCT για οκταδικό (βάση 8), HEX για δεκαεξαδικό (βάση 16).

9) **Client():** Δημιουργεί έναν πελάτη ο οποίος μπορεί να συνδεθεί στο Internet μέσω μιας καθορισμένης διεύθυνσης IP και θύρας.

- Σύνταξη: Client(ip, port);
- Παράμετροι:
 - ip: η διεύθυνση IP με την οποία θα συνδεθεί ο πελάτης (πίνακας 4 bytes).
 - port: η θύρα με την οποία θα συνδεθεί ο πελάτης (int).

10) **connected():** Το αποτέλεσμα της συνάρτησης αυτής είναι η σύνδεση ή μη του πελάτη.

- Σύνταξη: client.connected();

11) **connect():** Συνδέεται με τη διεύθυνση IP και τη θύρα που καθορίστηκε κατά τη δημιουργία. Η τιμή επιστροφής δείχνει την επιτυχία ή την αποτυχία.

- Σύνταξη: client.connect();

12) **write():** Γράφει δεδομένα στο διακομιστή στον οποίο ο πελάτης είναι συνδεδεμένος.

- Σύνταξη: client.write(data);
- Παράμετροι:
 - data: η τιμή που πρόκειται να γραφτεί.

13) **available():** Επιστρέφει τον αριθμό των bytes που διατίθενται για ανάγνωση (δηλαδή, η ποσότητα των δεδομένων που έχει γραφτεί για τον πελάτη από το διακομιστή που είναι συνδεδεμένος).

- Σύνταξη: client.available();

14) **read():** Διαβάζει τα επόμενα bytes που λαμβάνονται από το διακομιστή στον οποίο είναι συνδεδεμένος ο πελάτης.

- Σύνταξη: client.read();
- 15) **flush()**: Απορρίπτει τα bytes που έχουν γραφτεί για τον πελάτη, αλλά δεν έχει ακόμη διαβαστεί.
 - Σύνταξη: client.flush();

16) **stop()**: Αποσυνδέεται από το διακομιστή.

- Σύνταξη: client.stop();

#include <SPI.h>: Αυτή η βιβλιοθήκη επιτρέπει την επικοινωνία με άλλες SPI συσκευές, με το Arduino ως κύρια συσκευή. Σειριακή περιφερική διεπαφή (SPI) είναι ένα σύγχρονο σειριακό πρωτόκολλο δεδομένων που χρησιμοποιείται από μικροελεγκτές για την επικοινωνία με μία ή περισσότερες περιφερειακές συσκευές σε μικρές αποστάσεις. Μπορεί επίσης, να χρησιμοποιηθεί για την επικοινωνία μεταξύ των δύο μικροεπεξεργαστών.

3.5.2 Προγραμματισμός αισθητήρων

Στην ενότητα αυτή περιγράφεται ο τρόπος με τον οποίο οι αισθητήρες μετατρέπουν τις μετρήσεις τους σε δεδομένα κατανοητά για το χρήστη, αλλά και για τον προγραμματιστή του συστήματος. Κάθε εξάρτημα που χρησιμοποιήθηκε, απαιτούσε και διαφορετική αντιμετώπιση ως προς το προγραμματισμό του ανάλογα πάντα με το τι μεγέθη μετρούσε (πχ υγρασία, θερμοκρασία κτλ), αλλά και βάση συνδεσμολογίας (η συνδεσμολογία κάθε αισθητήρα περιγράφηκε στο κεφάλαιο 2). Παρακάτω ακολουθεί ανάλυση του κώδικα που αναπτύχθηκε:

1) Αισθητήρας φωτός:

```
1  int PhotocellPin = 2;
2  int IL;
3  void setup() {
4      pinMode(PhotocellPin, INPUT);
5  }
6
7
8  void loop() {
9      int PhotocellReading = analogRead(PhotocellPin);
10     IL = PhotocellReading;
11 }
```

Εικόνα 91: Κώδικας αισθητήρας φωτός

- Γραμμή 1: Ορίζεται ότι ο αισθητήρας φωτός συνδέεται με την αναλογική είσοδο του arduino, 2.
- Γραμμή 2: Ορίζεται η μεταβλητή ακεραίου IL, η οποία αντιπροσωπεύει την ένταση του φωτός στον εσωτερικό χώρο.
- Γραμμή 4: Ορίζεται κατά την εκκίνηση της συσκευής ότι η αναλογική θέση 2 είναι μεταβλητή εισόδου.
- Γραμμή 9: Με τη χρήση της συνάρτησης analogRead, λαμβάνονται οι μετρήσεις της αναλογικής εισόδου 2.
- Γραμμή 10: Η μεταβλητή IL λαμβάνει την τιμή της έντασης του φωτός.

2) Αισθητήρας υγρασίας:

```
1  int HumidityPin = 3; //humidity sensor pin
2  double H;
3
4
5
6  void setup() {
7      pinMode(HumidityPin, INPUT);
8  }
9
10
11 void loop() {
12     double Humidity = analogRead(HumidityPin);
13     //humidity check
14     if ( Humidity == 0){
15         H = 0;
16     }
17     else
18     {
19         double voltageValue = ((4.82)*Humidity)/1023;
20         double rh = (voltageValue - 0.788)/0.0314;
21         H = rh;
22     }
```

Εικόνα 92: Κώδικας αισθητήρα υγρασίας

- ii. Γραμμή 1: Ορίζεται ότι ο αισθητήρας υγρασίας συνδέεται με την αναλογική είσοδο του arduino, 3.
- iii. Γραμμή 2: Ορίζεται η μεταβλητή δεκαδικού αριθμού H, η οποία αντιπροσωπεύει την υγρασία στον εσωτερικό χώρο.
- iv. Γραμμή 7: Ορίζεται κατά την εκκίνηση της συσκευής ότι η αναλογική θέση 3 είναι μεταβλητή εισόδου.
- v. Γραμμή 12: Με τη χρήση της συνάρτησης analogRead, λαμβάνονται οι μετρήσεις της αναλογικής εισόδου 3.
- vi. Γραμμή 14-21: Αρχικά γίνεται ένας έλεγχος για ύπαρξη μηδενικής τιμής. Η περίπτωση αυτή αντιπροσωπεύει το γεγονός μη καλής λειτουργίας του αισθητήρα, καθώς πάντα υπάρχει στο χώρο σχετική υγρασία έστω και σε πολύ χαμηλά επίπεδα. Αν δεν είναι μηδενική η τιμή, τότε πολλαπλασιάζεται η τιμή αυτή επί την τάση εισόδου (4,82 μετρήθηκε με τη χρήση πολυμέτρου) και ακολουθούν πράξεις διαίρεσης και αφαίρεσης οι οποίες χρειάζονται για τη μετατροπή της μέτρησης σε ποσοστό υγρασίας.

3) Αισθητήρας Θερμοκρασίας:

```
1  int TemperaturePin = 4;
2  float T;
3
4  void setup() {
5
6      pinMode(TemperaturePin, INPUT);
7
8  }
9
10 void loop() {
11     //temperature check
12     if ( Temperature == 0){
13         T= 0;
14     }
15     else
16     {
17         float VoltBoost = (Temperature *4.81); // convert
18         float Boost = (VoltBoost *100); //convert 2
19         float Kelvin = (Boost/1023); //Kelvin
20         float Celsius = Kelvin -275; //Celsius
21         T= Celsius;
22     }
```

Εικόνα 93: Κώδικας αισθητήρα θερμοκρασίας

- i. Γραμμή 1: Ορίζεται ότι ο αισθητήρας θερμοκρασίας συνδέεται με την αναλογική είσοδο του arduino, 4.
- ii. Γραμμή 2: Ορίζεται η μεταβλητή δεκαδικού αριθμού T, η οποία αντιπροσωπεύει τη θερμοκρασία στον εσωτερικό χώρο.
- iii. Γραμμή 6: Ορίζεται κατά την εκκίνηση της συσκευής ότι η αναλογική θέση 4 είναι μεταβλητή εισόδου.
- iv. Γραμμή 12-21: Αρχικά γίνεται ένας έλεγχος για ύπαρξη μηδενικής τιμής. Η περίπτωση αυτή αντιπροσωπεύει το γεγονός μη καλής λειτουργίας του αισθητήρα, καθώς πάντα υπάρχει στο χώρο σχετική θερμοκρασία έστω και σε πολύ χαμηλά επίπεδα. Αν δεν είναι μηδενική η τιμή, τότε πολλαπλασιάζεται η τιμή αυτή επί την τάση εισόδου και ακολουθούν πράξεις διαίρεσης και αφαίρεσης οι οποίες χρειάζονται για τη μετατροπή της μέτρησης σε ποσοστό θερμοκρασίας. Η κατάληξη στα συγκεκριμένα νούμερα έγινε κατόπιν πειραματικών μετρήσεων με αναλογικό, αλλά και ψηφιακό θερμόμετρο.

4) Αισθητήρας υγρασίας εδάφους:

```
1  int MoistureSensorPin = 5;
2  int M;
3
4  void setup() {
5      pinMode(MoistureSensorPin, INPUT);
6  }
7
8  void loop() {
9      //moisture sensor check
10     if ( WaterNeed < 0 && WaterNeed >=1300){
11         M=3;
12     }
13     else
14     {
15         if (WaterNeed<690)
16         {
17             M=0;
18         }
19         else
20         {
21             M=1;
22         }
23     }
```

Εικόνα 94: Κώδικας αισθητήρα εδάφους

- i. Γραμμή 1: Ορίζεται ότι ο αισθητήρας θερμοκρασίας συνδέεται με την αναλογική είσοδο του arduino, 5.
- ii. Γραμμή 2: Ορίζεται η μεταβλητή δεκαδικού αριθμού M, η οποία αντιπροσωπεύει την υγρασία εδάφους στον εσωτερικό χώρο.
- iii. Γραμμή 5 Ορίζεται κατά την εκκίνηση της συσκευής ότι η αναλογική θέση 5 είναι μεταβλητή εισόδου.
- iv. Γραμμή 10-23: Αρχικά γίνεται ένας έλεγχος για την ομαλή λειτουργία του αισθητήρα. Αν η τιμή που ληφθεί δεν είναι μέσα στα πλαίσια που ορίζονται, τότε ο αριθμό 3 αντιπροσωπεύει τη διαπίστωση προβλήματος. Διαφορετικά οι τιμές θα είναι ή 0 ή 1, ξηρό ή υγρό αντίστοιχα. Για την εξαγωγή της κατάστασης του εδάφους, πραγματοποιήθηκαν πειράματα για την σωστή διαπίστωση της έννοιας του υγρού ή ξηρού εδάφους, στο θερμοκηπίου της μακέτας.

3.5.3 Προγραμματισμός συσκευών δράσης

Οι συσκευές δράσης που ενσωματώθηκαν στο σύστημα “Smart Greenhouse”, συνδέονταν με τους ψηφιακούς ακροδέκτες του Arduino όπως έχει αναφερθεί και παραπάνω. Η κατάσταση που μπορούν να βρεθούν είναι είτε ενεργοποιημένοι (HIGH) είτε απενεργοποιημένοι (LOW). Παρακάτω απεικονίζεται (εικόνα 95) ο προγραμματισμός της κατάστασης στην οποία βρίσκονται οι συσκευές δράσης:

```
1  #define VALVE 8
2  #define FAN 7
3  #define LED 6
4
5  int F;
6  int V;
7  int L;
8
9  void setup() {
10     pinMode(FAN, OUTPUT);
11     pinMode(VALVE, OUTPUT);
12     pinMode(LED, OUTPUT);
13 }
14
15 void loop() {
16     //Led condition
17     if (L==1)
18     { digitalWrite(LED,HIGH); } else { digitalWrite(LED,LOW); }
19     //fan condition
20     if (F==1)
21     { digitalWrite(FAN,HIGH); } else { digitalWrite(FAN,LOW); }
22     if (V==1)
23     { digitalWrite(VALVE,HIGH); } else { digitalWrite(VALVE,LOW); }
24 }
```

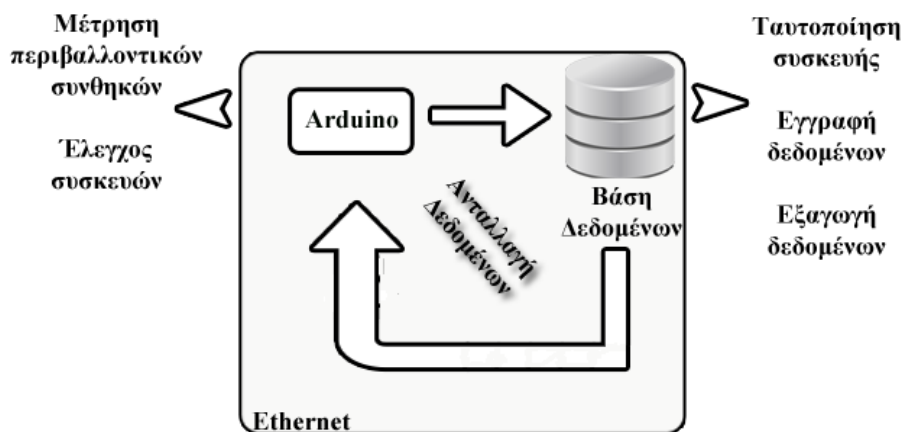
Εικόνα 95: Κώδικας συσκευών δράση

- i. Γραμμή 1-3: Ορίζονται οι θέσεις των συσκευών δράσης. Χαρακτηριστικά στη θέση 8 είναι η ηλεκτροβάνα, στη θέση 7 ο ανεμιστήρας και στη θέση 6 το UV LED.
- ii. Γραμμή 5-7: Ορίζονται ακέραιες μεταβλητές για κάθε συσκευή ξεχωριστά σύμφωνα με τα αρχικά των ονομάτων τους. Οι μεταβλητές αυτές μπορεί να είναι 1 ή 0.
- iii. Γραμμή 9-13: Ορίζεται κατά την εκκίνηση της συσκευής ότι οι θέσεις 6, 7, 8, αποτελούν εξόδους του συστήματος.
- iv. Γραμμή 15-24: Ανάλογα με τις τιμές που έχουν οι μεταβλητές των συσκευών δράσης αντίστοιχα ενεργοποιούν ή απενεργοποιούν το ηλεκτρικό σήμα που ευθύνεται για τη ροή ή διακοπή του ρεύματος στα N-MOSFET, όπως περιγράφηκε στο κεφάλαιο 2, η συνδεσμολογία.

Οι μεταβλητές των συσκευών δράσης είναι υπεύθυνες ουσιαστικά για το αν θα ενεργοποιηθούν ή όχι οι συσκευές. Η τιμή των μεταβλητών αυτών καθορίζεται από τα δεδομένα που λαμβάνει από τη βάση και τις παραμέτρους που εξετάζει το arduino, που προέρχονται από τον ιστοχώρο του συστήματος. Ο τρόπος με τον οποίο γίνεται η επικοινωνία αυτή αναλύεται παρακάτω.

3.5.4 Επικοινωνία arduino και ιστοχώρου

Η ενότητα αυτή ασχολείται με τον τρόπο με τον οποίο επικοινωνεί η συσκευή του συστήματος “Smart Greenhouse”, με τον ιστοχώρο που έχει δημιουργηθεί. Ουσιαστικά η επικοινωνία γίνεται με τα δεδομένα της βάσης. Το παρακάτω σχήμα (εικόνα 96) περιγράφει τη διαδικασία αυτή:



Εικόνα 96: Επικοινωνία arduino και βάσης

Αρχικά, κατά την εκκίνηση της συσκευής ή την επανεκκίνηση, προσπαθεί η συσκευή arduino να συνδεθεί με το διαδίκτυο. Εφόσον συνδεθεί με το διαδίκτυο η επικοινωνία γίνεται στον server (διακομιστή) "zafora.ict.e.uowm.gr" όπου φιλοξενείται και ο ιστοχώρος. Όταν συνδεθεί με τον server η ανταλλαγή των δεδομένων γίνεται με το αρχείο «arduino.php». Η επικοινωνία γίνεται με τη μέθοδο \$_POST, με την αποστολή συγκεκριμένων μεταβλητών. Οι μεταβλητές device_id και device_password, εξετάζονται από το αρχείο «arduino.php», για να διαπιστωθεί αν είναι καταχωρημένες στο σύστημα, ενώ η μεταβλητή option κατευθύνει τη συσκευή σε συγκεκριμένη περιοχή του αρχείου. Παρακάτω γίνεται μία ανάλυση των πεδίων αυτών:

1. **Device_id:** Για κάθε συσκευή που χορηγείται στο χρήστη, δημιουργείται και ένας μοναδικός αριθμός που τη χαρακτηρίζει. Με αυτόν τον τρόπο αποκλείεται οποιαδήποτε φαινόμενο ο χαρακτηριστικός αυτός αριθμός να δοθεί σε δύο διαφορετικών χρηστών συσκευές.
2. **Device_password:** Για κάθε device_id έχει δημιουργηθεί δυναμικά και ένας κωδικός. Με αυτόν τον τρόπο αυξάνεται η ασφάλεια του συστήματος και επιτυγχάνεται ταυτοποίηση της συσκευής, με απόρροια τη μη εύκολη πρόσβαση κακόβουλων συσκευών στο σύστημα. Ο κωδικός αυτός δημιουργείται με τη μέθοδο κρυπτογράφησης hash [29].
3. **Option:** Το πεδίο αυτό κατευθύνει το arduino σε συγκεκριμένη περιοχή του κώδικα του αρχείου arduino.php. Οι τιμές που μπορεί να πάρει είναι οι εξής:
 - i. **1:** Η περιοχή που αντιστοιχεί στην τιμή αυτή είναι υπεύθυνη για τον έλεγχο των συσκευών. Αρχικά ελέγχει τον πίνακα schedule_scenarios αν υπάρχει σενάριο. Σε περίπτωση που υπάρχει παράγει την παρακάτω συμβολοσειρά:

Παράδειγμα συμβολοσειράς:		
Θέση:	Χαρακτήρας:	Περιγραφή:
0	*	Ο χαρακτήρας αυτός δηλώνει την έναρξη της συμβολοσειράς. Χρησιμοποιείται κατά το διάβασμα της από το arduino.
1	1	Έλεγχος ανεμιστήρα. Η τιμή 1 είναι για ενεργοποίηση, η τιμή 0 είναι για απενεργοποίηση, ανάλογα με τις συνθήκες του σεναρίου.
2	0	Έλεγχος ηλεκτροβάνας. Η τιμή 1 είναι για ενεργοποίηση, η τιμή 0 είναι για απενεργοποίηση, ανάλογα με τις συνθήκες του σεναρίου.
3	0	Έλεγχος φωτισμού. Η τιμή 1 είναι για ενεργοποίηση, η

		τιμή 0 είναι για απενεργοποίηση, ανάλογα με τις συνθήκες του σεναρίου.
Οι παρακάτω 8 χαρακτήρες δηλώνουν πληροφορίες για συγκεκριμένες συνθήκες ενός αισθητήρα. Κάθε αισθητήρας αναπαριστάται στη συμβολοσειρά με 8 χαρακτήρες. Οπότε το μέγεθός της είναι μεταβλητό, δηλαδή αν υπάρχει σενάριο θα είναι 13(ένας αισθητήρας), 21 (δύο αισθητήρες), 29 (3 αισθητήρες), 37 (4 αισθητήρες) και 5 (κανένας αισθητήρας - manual).		
4 (1)	1	Η 1 ^η θέση των χαρακτήρων που αναφέρονται στις συνθήκες αισθητήρα δηλώνουν το είδος του αισθητήρα. Η τιμή 1 είναι για τη θερμοκρασία, η τιμή 2 για την υγρασία, η τιμή 3 για την ένταση του φωτός και η τιμή 4 για την υγρασία του εδάφους. Στην προκειμένη περίπτωση πρόκειται για τον αισθητήρα θερμοκρασίας.
5 (2)	0	Οι τρεις αυτές θέσεις θέτουν το μικρότερο όριο. Στην προκειμένη περίπτωση είναι 20 C.
6 (3)	2	
7 (4)	0	
8 (5)	0	Οι τρεις αυτές θέσεις θέτουν το μεγαλύτερο όριο. Στην προκειμένη περίπτωση είναι 30 C.
9 (6)	3	
10 (7)	0	
11 (8)	1	Η τιμή αυτής της θέσης είναι ιδιαίτερα σημαντική, καθώς δηλώνει τις σχέσεις εξάρτησης των αισθητήρων μεταξύ τους. Δηλαδή αν θα ισχύουν οι συνθήκες του ενός ή (or) του επόμενου είτε και (and) των δύο. Η προτεραιότητα είναι σειριακή, καθώς από αισθητήρα σε αισθητήρα αλλάζει η μεταβλητή, ανάλογα τις προτιμήσεις του χρήστη.
Οι επόμενοι 8 χαρακτήρες αφορούν συνθήκες νέου αισθητήρα.		
12 (1)	4	Αισθητήρας υγρασίας εδάφους.
13 (2)	0	Στην περίπτωση του αισθητήρα υγρασίας εδάφους, δεν υπάρχει κάτω ή πάνω όριο για να δημιουργηθεί το πεδίο τιμών. Απλά ελέγχει αν είναι υγρό (τιμή 1) ή αν είναι ξηρό (τιμή 2).
14 (3)	0	
15 (4)	1	
16 (5)	0	
17 (6)	0	
18 (7)	0	
19 (8)	0	Η τιμή 0 υποδηλώνει τη μη ύπαρξη κανόνα προτεραιότητας εκτέλεσης. Ουσιαστικά αυτό συμβαίνει μόνο στις συνθήκες του τελευταίου αισθητήρα.
20	@	Ο χαρακτήρας αυτός δηλώνει το τέλος της συμβολοσειράς. Χρησιμοποιείται κατά το διάβασμα της από το arduino.

Πίνακας 19: Συμβολοσειρά σεναρίου

Αν δεν υπάρχει σενάριο, τότε η συμβολοσειρά είναι ***000@**. Σε κάθε περίπτωση, πάντως, η αποκωδικοποίησή της γίνεται με τον παρακάτω τρόπο:

Μόλις εντοπισθεί το σύμβολο έναρξης της συμβολοσειράς, καταγράφονται σε έναν πίνακα τα δεδομένα που ακολουθούν μέχρι το σύμβολο τέλους. Αν δεν υπάρχει σενάριο οι συσκευές απενεργοποιούνται. Αντίθετα αν υπάρχει σενάριο, αρχικοποιείται μια μεταβλητή result τύπου boolean με τιμή false. Εξετάζει αν υπάρχουν συνθήκες αισθητήρων ή αν το σενάριο είναι το manual. Αν υπάρχουν συνθήκες αισθητήρων, ανάλογα με τις τιμές τους, η μεταβλητή γίνεται true ή false και σειριακά (δηλαδή από αισθητήρα σε αισθητήρα)

εκτελούνται οι λογικές πράξεις or/ and. Αν γίνει true, τότε εφαρμόζονται οι επιλογές για τις συσκευές δράσης. Διαφορετικά δεν εφαρμόζονται και παραμένουν απενεργοποιημένες. Στην περίπτωση που δεν υπάρχουν συνθήκες αισθητήρα, τότε η μεταβλητή result γίνεται true και εφαρμόζονται οι επιλογές για τις συσκευές δράσης. Χαρακτηριστικά παρατίθεται κομμάτι του κώδικα αποκωδικοποίησης της συμβολοσειράς (εικόνα 97):

```

233 if (os[0]!='\0')
234 {
235     if (i==3)
236     {
237         result=true;
238     }
239     else
240     {
241         b=1;
242         for(a=3;a<=i-1;a=a+8)
243         {
244             if (os[a]=='\0')
245             {
246                 float limit1=float(100*os[a+1] +10*os[a+2]+1*os[a+3]);
247                 float limit2=float(100*os[a+4] +10*os[a+5]+1*os[a+6]);
248                 if (T>=limit1 && T<=limit2)
249                 {
250                     resultT=true;
251                 }
252                 else
253                 {
254                     resultT=false;
255                 }
256                 if (b!=1)
257                 {
258                     if (os[a-1]!='\0')
259                     {
260                         result |=resultT;
261                     }
262                     else
263                     {
264                         result &=resultT;
265                     }
266                 }
267                 else
268                 {
269                     result=resultT;
270                 }
271             }
272             b++;
273         }
274     }
275 }
276
277 if (result==true)
278 {
279     F=os[0];
280     V=os[1];
281     L=os[2];
282 }
283
284 else
285 {
286     F=0;
287     V=0;
288     L=0;
289 }

```

Εικόνα 97: Κώδικας αποκωδικοποίησης

Γραμμή 233: Έλεγχος ύπαρξης συμβολοσειράς.

Γραμμή 235-238: Περίπτωση σεναρίου manual.

Γραμμή 241: Μεταβλητή b. Αρίθμηση αισθητήρα.

Γραμμή 242: Εφαρμογή βρόγχου για ανάγνωση των δεδομένων του πίνακα σε ομάδες των 8 χαρακτήρων.

Γραμμή 244: Αναγνώριση αισθητήρα.

Γραμμή 245-255: Έλεγχος πεδίου τιμών μεταβλητής αισθητήρα.

Γραμμή 256 - 273: Αν το b είναι 1, τότε σημαίνει ότι είναι ο πρώτος αισθητήρας που ελέγχεται και εφαρμόζεται το result του, στο τελικό result. Αν όχι, τότε γίνεται η λογική πράξη μεταξύ των δύο result και εξάγεται το τελικό και εν συνεχεία αυξάνεται το b.

Γραμμή 363-374: Ανάλογα με την τιμή του τελικού result εφαρμόζεται ή όχι η λογική του σεναρίου όσον αφορά την ενεργοποίηση και απενεργοποίηση των συσκευών δράσης.

Ο τρόπος αυτός επικοινωνίας του arduino με τη βάση χρησιμοποιήθηκε, ώστε να μην είναι ορατά με γυμνό μάτι τα δεδομένα των σεναρίων, αλλά και να είναι μεταβλητό το μέγεθος των δεδομένων με στόχο την ελαχιστοποίηση του.

- ii. 2: Η περιοχή που αντιστοιχεί στην τιμή αυτή είναι υπεύθυνη για την καταχώρηση των περιβαλλοντικών συνθηκών και της κατάστασης των

συσκευών δράσης στη βάση δεδομένων. Παρακάτω (εικόνα 98) απεικονίζεται ο τρόπος με τον οποίο στέλνει το arduino τα δεδομένα:

```
403 void connectAndSend() {  
404     if (client2.connect(serverName, 80)) {  
405         delay(1000);  
406         Serial.println("connected 2");  
407         String data2="device_id=1&device_password=bda5accf070d5938804296ecaab15978d2c45e9c&option=2";  
408         data2+="&temp=";  
409         data2+=(int)T;  
410         data2+="&hum=";  
411         data2+=(int)H;  
412         data2+="&somo=";  
413         data2+=M;  
414         data2+="&inli=";  
415         data2+=IL;  
416         data2+="&f=";  
417         data2+=F;  
418         data2+="&v=";  
419         data2+=V;  
420         data2+="&l=";  
421         data2+=L;  
422         data2+="&date=";  
423         data2+=year;  
424         data2+="-";  
425         data2+=month;  
426         data2+="-";  
427         data2+=day;  
428         data2+="&time=";  
429         data2+=hour;  
430         data2+=":";  
431         data2+=mi;  
432         data2+=":";  
433         data2+="00";  
434         client2.println("POST /~ictest00120/arduino.php HTTP/1.1");  
435         client2.println("Host: zafora.ict.e.uowm.gr");  
436         client2.println("Content-Type: application/x-www-form-urlencoded");  
437         client2.println("Connection: close");  
438         client2.print("Content-Length: ");  
439         client2.println(data2.length());  
440         client2.println();  
441         client2.println(data2);  
442         client2.println();  
443         delay(100);  
444         client2.stop();  
445         delay(100);  
446         client2.flush();
```

Εικόνα 98: Δεδομένα αποστολής arduino

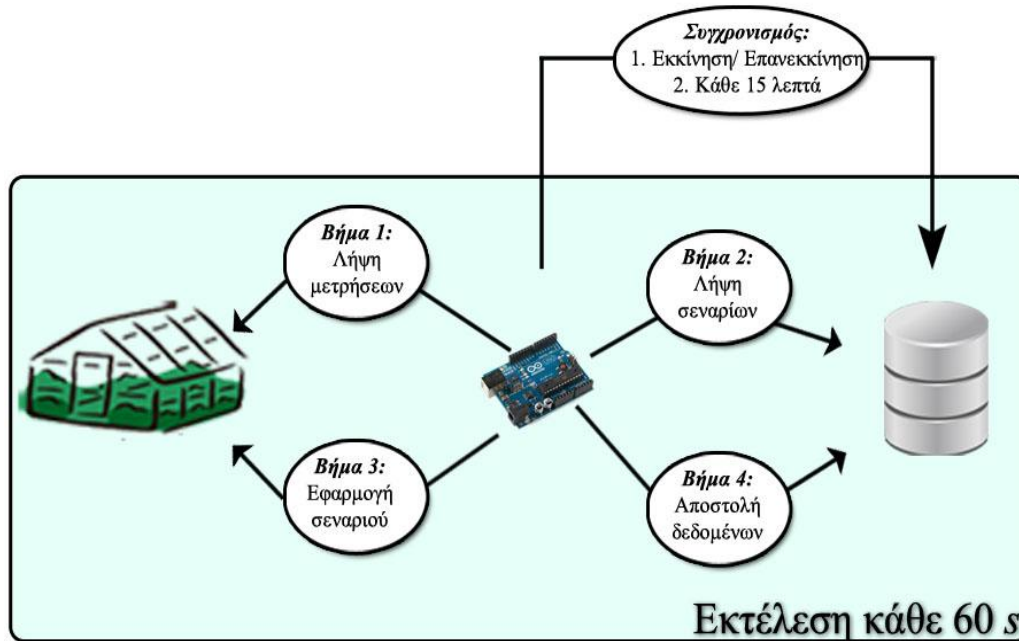
Η μέθοδος αποστολής είναι η POST, για την οποία έχει γίνει αναφορά στο κεφάλαιο 1. Στις γραμμές 434-446 της εικόνας 98, αποτυπώνεται ο τρόπος με τον οποίο πραγματοποιείται η επικοινωνία με το αρχείο μέσω της μεθόδου POST. Για κάθε μεταβλητή πριν καταχωρηθεί στη βάση γίνεται έλεγχος αν είναι δεκτή ανάλογα το είδος της και το προεπιλεγμένο τύπο δεδομένων που πρέπει να έχει. Αν κάποιος αισθητήρας βγει εκτός λειτουργίας, βάση των τιμών σφάλματος που αναφέρθηκαν παραπάνω, καταχωρείται στον πίνακα troubleshoots η αντίστοιχη τιμή σφάλματος. Ανάλογα με τις επιλογές του χρήστη μπορούν να αποσταλούν και τα αντίστοιχα email (ηλεκτρονικού ταχυδρομείου μηνύματα). Επίσης, εφόσον οι τιμές καταχωρηθούν στη βάση γίνεται έλεγχος με την αμέσως προηγούμενη για τυχόν σφάλματα ή και για ενδεχόμενες διακοπές επικοινωνίας.

Η περιοχή αυτή είναι υπεύθυνη για την εκκαθάριση της βάσης δεδομένων και την εξαγωγή των μέσων όρων ώρας, μέρας, μήνα που χρησιμοποιούνται στο σχεδιασμό των διαγραμμάτων. Με αυτόν τον τρόπο μειώνεται το μέγεθος της βάσης, αλλά και αυξάνεται η ταχύτητα σχεδίασης των διαγραμμάτων.

- iii. **3:** Η τιμή αυτή καλείται από το arduino σε περίπτωση εκκίνησης ή επανεκκίνησης της συσκευής. Η περιοχή που αντιστοιχεί στην τιμή αυτή είναι υπεύθυνη για τον συγχρονισμό του ρολογιού της συσκευής με τον server. Επίσης, εκτελείται κάθε 15 λεπτά για καλύτερο συγχρονισμό και ελαχιστοποίηση χρονικών αποκλίσεων, λόγω της βαρύνουσας σημασίας των χρονικών δεδομένων. Σε περίπτωση που

δεν επιτευχθεί συγχρονισμός σε κάποια από τις κλήσεις η συσκευή πραγματοποιεί επανεκκίνηση.

Οι τρεις αυτές τιμές του πεδίου option που αναφέρθηκαν παραπάνω είναι υπεύθυνες για την επικοινωνία του arduino με τη βάση δεδομένων και κατ' επέκταση με τον ιστοχώρο. Ο τρόπος που δουλεύει ο κώδικας της συσκευής στο σύνολό του, παρατίθεται παρακάτω (εικόνα 99):



Εικόνα 99: Λειτουργία κώδικα Arduino

Αρχικά το arduino συλλέγει τις τιμές των αισθητήρων. Στη συνέχεια επικοινωνεί με τη βάση όπου γίνεται ο έλεγχος για την ύπαρξη σεναρίου. Έπειτα, εφαρμόζει αν υπάρχει τη λογική του σεναρίου βάσει των τιμών των αισθητήρων και ενεργοποιούνται ή απενεργοποιούνται ανάλογα οι συσκευές. Αν δεν υπάρχει σενάριο οι συσκευές παραμένουν απενεργοποιημένες. Στο τέλος, αποστέλλονται οι τιμές των αισθητήρων, καθώς και η κατάσταση των συσκευών δράσης, ώστε να εμφανιστούν μέσω του ιστοχώρου στο χρήστη. Η διαδικασία αυτή επαναλαμβάνεται κάθε 60 δευτερόλεπτα (1 λεπτό), με σκοπό την αλληλεπίδραση σε πραγματικό χρόνο.

Κεφάλαιο 4

Μελλοντικές επεκτάσεις συστήματος

Το σύστημα “Smart Greenhouse” αποτελεί ένα ενσωματωμένο σύστημα. Στηρίζει την αρχιτεκτονική του, στην ενσωμάτωση αισθητήρων στο θερμοκήπιο και την επικοινωνία τους με τον ιστοχώρο, ώστε ο χρήστης που έχει πρόσβαση, μέσω διαδικτύου να έχει τη δυνατότητα πληροφόρησης αλλά και αλληλεπίδρασης με το χώρο. Η έννοια των ενσωματωμένων συστημάτων, είναι σχετικά πρόσφατα γνωστή στο ευρύ κοινό, αλλά σε ερευνητικό επίπεδο αποτελεί αντικείμενο έρευνας τα τελευταία 60 χρόνια.

Όπως κάθε σύστημα που στηρίζεται σε αισθητήρες, μικροελεγκτές, αλλά και στο διαδίκτυο υπάρχουν σημαντικά θέματα λειτουργίας που λόγω της τεχνολογικής εξέλιξης, απαιτούν συνεχή βελτίωση. Ιδιαίτερα όταν πρόκειται για συστήματα που έχουν πρακτική εφαρμογή, τότε η λειτουργία τους θα πρέπει να είναι τέτοια ώστε να ανταποκρίνεται και στις πιο απαιτητικές συνθήκες, όπως και στους πιο απαιτητικούς χρήστες. Το σύστημα “Smart Greenhouse”, που ανήκει στην κατηγορία που αναφέραμε παραπάνω, βρίσκεται ήδη ερευνητικά στη διαδικασία βελτιστοποίησης. Οι μελλοντικές επεκτάσεις του συστήματος, που αποτελούν το αντικείμενο αναφοράς του κεφαλαίου αυτού, ασχολούνται με:

1. Θέματα τροφοδοσίας
2. Θέματα επικοινωνίας
3. Θέματα χρήσης αισθητήρων
4. Θέματα γενικής χρήσης

4.1 Μελλοντικές επεκτάσεις σε θέματα τροφοδοσίας

Η συσκευή που έχει αναπτυχθεί συνολικά, όπως έχει αναφερθεί και παραπάνω, καταναλώνει 0,072 kWh την ημέρα. Η κατανάλωση αυτή αφορά αποκλειστικά τους μικροελεγκτές και τους αισθητήρες, όχι τις συσκευές δράσης καθώς από χρήστη σε χρήστη διαφέρουν οι συσκευές αυτές ενώ συνήθως τα φορτία τους είναι πολλαπλάσια του παραπάνω μεγέθους.

Η αρχική έκδοση του συστήματος χρησιμοποιεί ηλεκτρικό ρεύμα μέσω του αντίστοιχου φορέα παροχής, για την τροφοδοσία της. Όπως είναι όμως φυσιολογικό, η χρήση του είναι μεν αξιόπιστη αλλά δεν καλύπτει το χρήστη απόλυτα. Παρακάτω, αναλύονται ιδέες για τη βελτιστοποίηση του συστήματος όσο αφορά την τροφοδοσία του.

4.1.1 Χρήση μπαταρίας

Η χρήση μπαταρίας, είτε αλκαλικής είτε λιθίου, αποτελεί μία βελτιστοποίηση του συστήματος. Ο λόγος που χρειάζεται η παρουσία της στο σύστημα, προέκυψε κατά τη διάρκεια των πειραματικών δοκιμών και πιο συγκεκριμένα κατά τη διακοπή παροχής ρεύματος στο σύστημα μέσω πρίζας. Η διακοπή παροχής ρεύματος στο σύστημα έχει τις παρακάτω επιπτώσεις:

1. Το σύστημα δε μπορεί να λειτουργήσει ούτε να ελέγξει τις συσκευές δράσης. Σε περίπτωση που οι συσκευές δράσης, τροφοδοτούνται από διαφορετική παροχή, η οποία δε σταματήσει να τροφοδοτείται με ρεύμα, τότε η κατάστασή τους θα παραμείνει σύμφωνα με την τελευταία λήψη εντολών από τη βάση. Το γεγονός αυτό μπορεί να προκαλέσει αντίθετα από τα επιθυμητά αποτελέσματα, όπως αυξημένη υγρασία εδάφους ή υπέρμετρη ηλιακή ακτινοβολία (βάση των συσκευών δράσης της μακέτας).
2. Ο χρήστης, που βρίσκεται εκτός του περιβάλλοντα χώρου του θερμοκηπίου, με το που διαπιστωθεί η απώλεια σύνδεσης, δε μπορεί να γνωρίζει ποια είναι η αιτία, δηλαδή αν πρόκειται για πρόβλημα στην επικοινωνία ή στην παροχή ρεύματος.

Για να ελαχιστοποιηθούν οι πιθανότητες εμφάνισης τέτοιων παθογενειών στο σύστημα, η χρήση μπαταρίας αποτελεί μία λύση. Με τη παρουσία της, θα μπορεί να υπάρχει η λειτουργία της ειδοποίησης για διακοπή ρεύματος και ενεργοποίησης συγκεκριμένων ρυθμίσεων από το χρήστη. Φυσικά, μεγάλο ρόλο παίζει και ο τρόπος που γίνεται η επικοινωνία με τη βάση, αλλά αυτό θα αναλυθεί παρακάτω στο κεφάλαιο αυτό. Επομένως, ο χρήστης θα λαμβάνει ειδοποίηση (αν φυσικά το επιθυμεί) για το αν έχει διακοπεί το ρεύμα και θα έχει ρυθμίσει τη συμπεριφορά της συσκευής σε ανάλογη περίπτωση (αδρανοποίηση των συσκευών δράσης κτλ). Στόχος της χρήσης της μπαταρίας δεν είναι η ολική αποκατάσταση της τροφοδοσίας, αλλά η ειδοποίηση του χρήστη για το γεγονός της διακοπής, που μπορεί να έχει σημαντικές επιπτώσεις, ανάλογα με τα εργαλεία που χρησιμοποιεί, στην ανάπτυξη των φυτών.

4.1.2 Χρήση ανανεώσιμων πηγών ενέργειας

Η χρήση ανανεώσιμων πηγών ενέργειας, για την κάλυψη των αναγκών της συσκευής του συστήματος “Smart Greenhouse”, προέκυψε μετά την καταγραφή της κατανάλωσης, η οποία αναφέρεται παραπάνω. Η χρήση λοιπόν της ηλιακής ή της αιολικής ενέργειας, ανάλογα πάντα και την τοποθεσία την οποία εγκαθίσταται το σύστημα μπορεί να αντικαταστήσει την παροχή ρεύματος από τους σχετικούς φορείς. Φυσικά υπάρχει το κόστος της αρχικής εγκατάστασης, αλλά η απόσβεση είναι

δεδομένη. Εκτός αυτού η συσκευή γίνεται αυτόνομη όσο αφορά τις ενεργειακές της ανάγκες, με αποτέλεσμα η επικοινωνία με τη βάση να μην απειλείται από θέματα παροχής ρεύματος.

Η χρήση ανανεώσιμων πηγών ενέργειας απαιτεί μελέτη και παρακολούθηση των συνθηκών της ευρύτερης περιοχής που θα εγκατασταθεί το σύστημα. Δεν είναι δυνατόν, να γίνει λόγος για επιλογή κάποιας συγκεκριμένης πηγής αν δεν υπάρχει ένδειξη ότι θα καλύψει απόλυτα τις ενεργειακές ανάγκες της συσκευής. Επίσης, μπορεί να γίνεται και συνδυασμός όσο αφορά με τη χρήση ηλεκτρικού ρεύματος από τον αντίστοιχο φορέα παροχής. Με αυτόν τον τρόπο, ανάλογα με τα επίπεδα ενέργειας που μπορεί να τροφοδοτήσει η ανανεώσιμη πηγή, θα μπορεί να συνεισφέρει στο σύστημα και η ηλεκτροδότηση. Εκτός αυτού, θα μπορεί ο χρήστης να διαλέγει ποια πηγή θέλει να έχει ως κύρια και ποια ως εφεδρική.

4.2 Μελλοντικές επεκτάσεις σε θέματα επικοινωνίας

Η επικοινωνία της συσκευής με τη βάση, σε αυτήν την έκδοση του συστήματος γίνεται με τη χρήση του πρωτοκόλλου Ethernet, μέσω του Ethernet Shield. Η υλοποίηση αυτή, απαιτεί την ύπαρξη διαδικτύου στο χώρο που βρίσκεται το θερμοκήπιο και την ύπαρξη δρομολογητή (router) για τη σύνδεση της συσκευής. Όμως η ύπαρξη σύνδεσης με το διαδίκτυο ή ακόμα και τηλεφωνικής υποδομής δεν είναι πάντα εφικτή. Για το λόγο αυτό παρακάτω γίνεται προσέγγιση εναλλακτικών μεθόδων επικοινωνίας της συσκευής με τη βάση ανάλογα πάντα και με την τοποθεσία που βρίσκεται το θερμοκήπιο.

4.2.1 Χρήση Wifi

Η χρήση Wifi, αποτελεί μία λύση στην περίπτωση που το θερμοκήπιο βρίσκεται εντός εμβέλειας ασύρματου δικτύου. Η αρχιτεκτονική arduino έχει προβλέψει για το γεγονός αυτό και υπάρχει το αντίστοιχο shield, όπως έχει αναφερθεί και παραπάνω (Πίνακας 3, εικόνα 13). Με τη χρήση του εξαρτήματος αυτού, η συσκευή μπορεί να επικοινωνεί ασύρματα με το διαδίκτυο χωρίς να χρειάζεται καλώδιο που να συνδέεται με το δρομολογητή.

Επίσης, η σύνδεση με το wifi δίκτυο μπορεί να γίνει και με το ethernet shield. Στην περίπτωση αυτή χρησιμοποιείται ένα switch, το οποίο συνδέεται με το wifi και έχει εξόδους σύνδεση μέσω ethernet. Το ethernet shield έχει το μειονέκτημα ότι μπορεί να συνδεθεί μόνο με switch ή router.

4.2.2 Χρήση GPRS

Η χρήση GPRS, μπορεί να εξαλείψει, αρχικά την ανάγκη ύπαρξης σύνδεσης διαδικτύου στο χώρο εγκατάστασης της συσκευής. Η arduino αρχιτεκτονική έχει προβλέψει για το γεγονός αυτό και υπάρχει αντίστοιχο shield, όπως αυτό της εικόνας 16 στον πίνακα 3. Στην αγορά υπάρχουν διάφορα μοντέλα συμβατά με το arduino αλλά διαφορετικών τεχνικών χαρακτηριστικών. Κοινή ανάγκη όλων, είναι η χρήση SIM card και κεραίας GSM.

Η επικοινωνία μέσω GPRS μπορεί να γίνει με τη χρήση μηνυμάτων (sms), με τη σύνδεση σε TCP/IP δίκτυο, αλλά και με την αναγνώριση χρόνου κλίσης. Ο τρίτος τρόπος έχει το πλεονέκτημα ότι δεν έχει κάποιο κόστος, ενώ οι δύο πρώτοι έχουν χρέωση ανάλογα με τον πάροχο κινητής επικοινωνίας. Εκτός αυτού κατάλληλη

προσαρμογή χρειάζεται και η βάση ώστε να μπορεί να δέχεται αλλά και να λαμβάνει τα δεδομένα από τη συσκευή.

4.3 Μελλοντικές επεκτάσεις ως προς τη χρήση αισθητήρων

Στόχος του συστήματος “Smart Greenhouse”, είναι όπως έχει αναφερθεί και παραπάνω, η αντικατάσταση της ανθρώπινης παρουσίας. Με τους αισθητήρες μπορεί ο χρήστης να έχει μερική αντίληψη του περιβάλλοντα χώρου. Ο χαρακτηρισμός «μερική», στην προηγούμενη πρόταση, δηλώνει ότι ο χρήστης δεν είναι δυνατόν να γνωρίζει απόλυτα την κατάσταση που βρίσκεται το θερμοκήπιο.

Η χρήση, όμως, επιπλέον αισθητήρων θα βελτιώνει το ποσοστό αντίληψης της συσκευής και κατ’ επέκτασης του συστήματος. Η χρήση αισθητήρων όπως, ανίχνευσης του διοξειδίου του άνθρακα, χρωματικής ανίχνευσης, μέτρησης του pH του εδάφους, μέτρηση του μεγέθους του καρπού και της φυλλωσιάς του φυτού, καθώς και άλλων αισθητήρων θα σύμβαλε στην πλήρη γνώση των συνθηκών του θερμοκηπίου ακόμα και εκείνων που δεν αναγνωρίζονται απευθείας από τον άνθρωπο (δηλαδή χρειάζεται κάποια συσκευή για τη μέτρηση τους). Με τη χρήση πολλών και διαφορετικών αισθητήρων αλλά αντίστοιχα και συσκευών δράσης (θερμάστρες, ρελέ, ηλεκτρικό άνοιγμα/ κλείσιμο των παραθύρων, ανάμειξη λιπασμάτων), θα υπήρχε η δυνατότητα της πλήρους αντικατάστασης της ανθρώπινης παρουσίας. Με αυτόν τον τρόπο θα υπήρχαν οι προϋποθέσεις για αυτοματοποίηση της γεωργικής καλλιέργειας με στόχο την ειδοποίηση για θερισμό της σοδειάς μέσω ενός ηλεκτρονικού μηνύματος.

Επίσης, μπορεί να χρησιμοποιηθεί και σύστημα παρακολούθησης μέσω κάμερας. Ο χρήστης θα μπορεί να έχει τη δυνατότητα να την ενεργοποιεί και σε πραγματικό χρόνο να μπορεί να βλέπει τι συμβαίνει. Εκτός αυτού, θα μπορεί να προγραμματίζει συγκεκριμένες ώρες και μέρες να λαμβάνει στιγμιότυπα από τα διάφορα φυτά που καλλιεργεί ώστε να έχει ακόμα καλύτερη αντίληψη για την ανάπτυξη τους, δημιουργώντας το αντίστοιχο αρχείο που θα το βοηθούσε να εξάγει συμπεράσματα για την παραγωγή του.

4.4 Μελλοντικές επεκτάσεις γενικής χρήσης συστήματος

Το σύστημα “Smart Greenhouse”, απαιτεί για την εγκατάσταση του σε κάθε χρήστη, ξεχωριστή αντιμετώπιση. Το γεγονός αυτό προκαλείται από το πλουραλισμό αισθητήρων και συσκευών δράσης που υπάρχουν στην αγορά, αλλά και από το γεγονός της διαφορετικής αντίληψης της καλλιέργειας ακόμα και των ίδιων φυτών από άνθρωπο σε άνθρωπο. Ανάλογα με τους αισθητήρες και τις συσκευές δράσης που θέλει ο χρήστης γίνεται και ο αντίστοιχος προγραμματισμός του ιστοχώρου.

Η διαφορετική αντιμετώπιση του κάθε χρήστη προγραμματιστικά είναι ένα πρόβλημα για το δημιουργό του συστήματος, καθώς συνέχεια θα χρειάζεται να επεμβαίνει τόσο στο λογισμικό μέρος όσο και στο υλικό μέρος του συστήματος. Για αυτό μια λύση θα ήταν η ανάπτυξη μιας πλατφόρμας η οποία θα αποτελούνταν από τον ιστοχώρο και τη συσκευή. Η διαφορά της με την υπάρχουσα μορφή του συστήματος, θα ήταν ο τρόπος εγκατάστασης των αισθητήρων και τη σύνδεση με τις συσκευές δράσης που θα γινόταν από το χρήστη. Η διαδικασία θα ήταν η εξής:

Ο χρήστης θα λάβει μια συσκευή η οποία δεν θα έχει αισθητήρες, αλλά υποδοχές για μια συγκεκριμένη λίστα αισθητήρων που θα καλύπτουν τις αυξημένες

ανάγκες για μέτρηση και μελέτη των συνθηκών ενός θερμοκηπίου. Φυσικά θα υπάρχει και η ανάλογη υποστήριξη τόσο σε επίπεδο συνδεσμολογίας όσο και σε επίπεδο εμφάνισης στον ιστοχώρο. Με το που θα συνδέσει τους αισθητήρες και τις συσκευές, θα γίνει αυτόματη αναγνώριση από το σύστημα ή μέσω επιλογής λίστας εξαρτημάτων από το χρήστη, ώστε να εμφανιστούν στον ιστόχωρο τα κατάλληλα πεδία με τις μετρήσεις. Δηλαδή το σύστημα θα αφήνει ένα ποσοστό ελευθερίας στο χρήστη ως προς τους αισθητήρες και τις συσκευές. Όλες οι λειτουργίες του συστήματος θα ενημερωθούν σχετικά με τα είδη των αισθητηρίων και τις συσκευές.

Η διαδικασία που περιγράφηκε παραπάνω μπορεί να εξελιχθεί στην ανάπτυξη πλατφόρμας ανοιχτού λογισμικού. Οι χρήστες ανάλογα με τις απαιτήσεις τους θα μπορούν να ανανεώνουν τη λίστα αισθητήρων και συσκευών και να μοιράζονται διαδικτυακά τη γνώση, συμβάλλοντας στην ανάπτυξη του συστήματος τόσο σε επίπεδο υλικού όσο και σε επίπεδο λογισμικού.

Κεφάλαιο 5

Συμπεράσματα

Η διπλωματική εργασία που παρουσιάστηκε στις παραπάνω επιμέρους παραγράφους, υλοποίησε ένα πρότυπο ενσωματωμένο σύστημα παρακολούθησης περιβαλλοντικών συνθηκών και απομακρυσμένου ελέγχου, ενός θερμοκηπίου. Βάση των αναγκών των απλών χρηστών αλλά και των διαχειριστών αναπτύχθηκαν και οι αντίστοιχες λειτουργίες που παρουσιάστηκαν παραπάνω.

Ο απλός χρήστης έχει τη δυνατότητα με το που εισέλθει στον ιστοχώρο να μελετήσει σε πραγματικό χρόνο τις περιβαλλοντικές συνθήκες. Επίσης μπορεί να ενεργοποιεί και απενεργοποιεί τις συσκευές δράσης ή να έχει προγραμματίσει τα σενάρια του με τέτοιο τρόπο, ώστε να αντικαθίσταται η ανθρώπινη παρουσία. Σε περίπτωση που ο χρήστης δε θέλει να έχει μόνιμη επαφή με τον ιστοχώρο, τότε ανάλογα με τις ρυθμίσεις που έχει επιλέξει μπορεί να λαμβάνει email σφαλμάτων ή αναφοράς, ανάλογα με τις περιόδους που έχει επιλέξει. Φυσικά, υπάρχει και η δυνατότητα επικοινωνίας με τους διαχειριστές του συστήματος για τυχόν απορίες, όσο και η αλλαγή των προσωπικών του δεδομένων (email, κωδικός).

Ο διαχειριστής έχει τη δυνατότητα να δημιουργεί, επεξεργάζεται και να διαγράφει απλούς χρήστες από το σύστημα. Εκτός αυτού, έχει τη δυνατότητα επίβλεψης των σφαλμάτων όλων των συσκευών αλλά και επιμέρους κατηγοριοποίησής τους. Αντίστοιχα, έχει τη δυνατότητα ανάγνωσης των μηνυμάτων που λαμβάνονται από τους χρήστες. Φυσικά, υπάρχει και η δυνατότητα της απεικόνισης στατιστικών δεδομένων για τον ιστοχώρο και τους χρήστες.

Η δημιουργία τόσο απλών χρηστών όσο και διαχειριστών, έγινε με σκοπό την καλύτερη λειτουργία του ιστοχώρου, μέσω της βοήθειας της δεύτερης ομάδας στην

πρώτη. Με την επικοινωνία μεταξύ τους, αλλά και με την επίβλεψη των σφαλμάτων, το σύστημα μπορεί να βελτιστοποιηθεί προγραμματιστικά παρέχοντας εν τέλει καλύτερες υπηρεσίες στον απλό χρήστη.

Η δημιουργία του συστήματος "Smart Greenhouse", λοιπόν, έγινε με σκοπό την αντικατάσταση της ανθρώπινης παρουσίας, όσο είναι δυνατό στον τομέα της γεωργίας, από το περιβάλλον ενός θερμοκηπίου. Απώτερος σκοπός είναι, η εξέλιξη του ενσωματωμένου συστήματος να είναι τέτοια, ώστε να μπορεί ο χρήστης να καλλιεργεί τα δικά του προϊόντα ελαχιστοποιώντας το κόστος μετακίνησης στο χώρο του θερμοκηπίου, με φιλική συμπεριφορά προς το περιβάλλον, αλλά και πολύ καλύτερη ποιότητα ζωής λόγω της τροφής που λαμβάνει. Ιδίως στη σημερινή κοινωνία όπου οι άνθρωποι έχουν έλλειψησχόλης λόγω του διευρυμένου ωραρίου εργασίας, αλλά και έλλειψη χρημάτων λόγω των οικονομικών, πολιτικών, κοινωνικών μεταπτώσεων, η χρήση του συστήματος που περιγράφεται στο παρών κείμενο, αποτελεί μία λύση για μια καλύτερη και φιλικότερη προς το περιβάλλον ζωή.

Βιβλιογραφία

- [1] HTML, <http://el.wikipedia.org/wiki/HTML>, Νοέμβριος 2012
- [2] CSS, <http://el.wikipedia.org/wiki/CSS>, Νοέμβριος 2012
- [3] W3Schools, http://www.w3schools.com/js/js_intro.asp, Νοέμβριος 2012
- [4] Ανάπτυξη διαδικτυακών εφαρμογών, Παναγιώτη Κεντέρλης 2009, Νοέμβριος 2012
- [5] W3Schools, http://www.w3schools.com/ajax/ajax_intro.asp, Νοέμβριος 2012
- [6] Silberschatz, Korth, Sudarshan, Συστήματα Βάσεων Δεδομένων έκδοση 4, Εκδόσεις Μ. Γκιούρδας, 2009, Νοέμβριος 2012
- [7] Arduino playground, <http://www.arduino.cc/>, Νοέμβριος 2012
- [8] Arduino GPRS info, <http://labs.arduino.cc/GPRS/Index>, Νοέμβριος 2012
- [9] Notepad++, <http://notepad-plus-plus.org/>, Νοέμβριος 2012
- [10] Xampp, <http://www.apachefriends.org/en/xampp.html>, Νοέμβριος 2012
- [11] Arduino IDE, <http://www.arduino.cc/en/Main/software>, Νοέμβριος 2012
- [12] Photoshop, <http://www.photoshop.com/>, Νοέμβριος 2012
- [13] Fritzing, <http://fritzing.org/>, Νοέμβριος 2012
- [14] Wiring, <http://www.wiring.org.co/>, Νοέμβριος 2012
- [15] C, [http://en.wikipedia.org/wiki/C_\(programming_language\)](http://en.wikipedia.org/wiki/C_(programming_language)), Νοέμβριος 2012
- [16] JAVA, [http://en.wikipedia.org/wiki/Java_\(programming_language\)](http://en.wikipedia.org/wiki/Java_(programming_language)), Νοέμβριος 2012
- [17] Avr gcc, http://www.avrfreaks.net/wiki/index.php/Documentation:AVR_GCC, Νοέμβριος 2012
- [18] Arduino Ethernet Shield, <http://www.arduino.cc/en/Main/ArduinoEthernetShield>, Νοέμβριος 2012
- [19] LM 335, http://probyte.fi/oscom/product_info.php?products_id=47, Νοέμβριος 2012
- [20] 808h5V5, <http://www.sensolution.co.kr/pdf/808H5V5.pdf>, Νοέμβριος 2012
- [21] Photocell, <http://download.instructables.com/EIU/5ET1/FVS81JJ6/Photocell-tutorial.pdf>, Νοέμβριος 2012
- [22] MOSFET, <http://www.materials.uoc.gr/el/undergrad/courses/ETY482/notes/mosfet.pdf>, Νοέμβριος 2012
- [23] IRF 520, <http://www.datasheetcatalog.org/datasheet/supertex/IRF521.pdf>, Νοέμβριος 2012
- [24] IN 4007, <http://www.alldatasheet.com/view.jsp?Searchword=IN4007>, Νοέμβριος 2012
- [25] Σύνδεση με N-MOSFET, <http://bildr.org/2012/03/rfp30n06le-arduino/>, Νοέμβριος 2012
- [26] EAGLE, <http://www.cadsoftusa.com/>, Νοέμβριος 2012
- [27] Perhydrol, <http://pl.wikipedia.org/wiki/Perhydrol>, Νοέμβριος 2012
- [28] Υδροχλωρικό οξύ, http://el.wikipedia.org/wiki/υδροχλωρικό_οξύ/, Νοέμβριος 2012
- [29] Hash, http://en.wikipedia.org/wiki/Hash_function, Νοέμβριος 2012
- [30] Crontabs, <http://en.wikipedia.org/wiki/Cron>, Νοέμβριος 2012
- [31] Xbee, <http://en.wikipedia.org/wiki/XBee>, Δεκέμβριος 2012