

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1393

**Σχεδίαση και υλοποίηση συστήματος
εποπτείας και ελέγχου θερμοκηπίου πάνω από
το διαδίκτυο, βασισμένο στην αναπτυξιακή
πλατφόρμα Arduino.**

Κανέλλης Νικόλαος

Κατεργάρης Ευάγγελος

Εισηγητές:

Τοπάλης Ευάγγελος

Καψάλης Βασίλειος

Πάτρα, Ιούνιος 2015

Πρόλογος

Η ανάπτυξη της τεχνολογίας τα τελευταία έτη, έχει προκαλέσει σπουδαίες καινοτομίες στον τομέα της πληροφορικής, οι οποίες είχαν σαν αποτέλεσμα την αύξηση της επεξεργαστικής ισχύος των υπολογιστικών συστημάτων και συγχρόνως τη μείωση των διαστάσεων τους αλλά και του κόστους τους. Εφεξής η απόκτηση ενός υπολογιστικού συστήματος με ευρύ φάσμα δυνατοτήτων το οποίο μπορεί να χρησιμοποιηθεί σε ποικίλες εφαρμογές είναι απόλυτα προσιτή.

Το λογισμικό ανοικτού κώδικα βοήθησε αρκετά στην δημιουργία εργαλείων ανάπτυξης λογισμικού, δεδομένου ότι σαν φιλοσοφία, μέθοδος διανομής και επαναχρησιμοποίησης πηγαίου κώδικα, παρέχει τη δυνατότητα πρόσβασης σε εφαρμογές δωρεάν τόσο σε αυτές όσο και στον πηγαίο κώδικα. Αποτέλεσμα αυτών είναι ότι η ανάπτυξη εφαρμογών αλλά και η βελτιστοποίησή τους επιτυγχάνεται με ευκολία και ταχύτητα εξαιτίας της ελευθερίας που παρέχει το ανοικτού κώδικα λογισμικό.

Επακόλουθο των παραπάνω είναι η υλοποίηση εφαρμογών που είναι σε θέση να εκτελεστούν σε οποιοδήποτε λειτουργικό σύστημα, οι οποίες οδηγούν στη δημιουργία εργαλείων ανεξαρτήτου πλατφόρμας. Έτσι καταλήγουμε στο ότι τα εργαλεία ανάπτυξης του λειτουργικού συστήματος μας, φέρουν τις ίδιες λειτουργίες και είναι πλήρως συμβατά σε διάφορα λειτουργικά συστήματα.

Πλέον η απόκτηση ενός μικροελεγκτή με σκοπό την ανάπτυξη εφαρμογών είναι αρκετά προσιτή καθότι το κόστος τους είναι πολύ χαμηλό. Στην αγορά υπάρχει ποικιλία μικροελεγκτών που συνδυάζουν χαμηλή τιμή και συγχρόνως είναι ικανοί να παράγουν πλήθος εφαρμογών. Υπάρχει τέτοια εξέλιξη στον τομέα ανάπτυξης νέων μικροελεγκτών οι οποίοι προσφέρονται για κάθε γενική ή εξειδικευμένη εφαρμογή.

Χάρη στην ευχρηστία των εργαλείων ανάπτυξης εφαρμογών για μικροελεγκτές, η διαδικασία αυτή δεν αποτελεί έργο που μπορεί να φέρει σε πέρας μόνο ένας πεπειραμένος και εξειδικευμένος προγραμματιστής. Συνακόλουθο αυτού είναι η ανάπτυξη εφαρμογών χωρίς να χρειάζεται κάποιος να είναι ειδικός στην διαδικασία αυτή και αυτό διότι πολλές λειτουργίες έχουν καμουφλαριστεί κάτω από γραφικά περιβάλλοντα, τα οποία μέσω εύχρηστων καταλόγων διαθέσιμων επιλογών αυτοματοποιούν τη διαδικασία.

Στην αγορά διατίθενται αισθητήρες οι οποίοι καλύπτουν οποιαδήποτε ανάγκη μέτρησης ποσοτήτων. Με την εξέλιξη της τεχνολογίας έχουν υποστεί και αυτοί σμίκρυνση για λόγους χρηστικότητας, πράγμα που έχει βοηθήσει στην διαμόρφωση καλύτερων αισθητήρων. Έχει επιτευχθεί μείωση του μεγέθους και του κόστους τους σε μεγάλο βαθμό δίχως να στερούνται τη λειτουργικότητά τους. Συνακόλουθο αυτών είναι η ανάπτυξη αισθητήρων με μεγάλη ακρίβεια και αρκετές δυνατότητες διασύνδεσης.

Οι ασύρματες και ενσύρματες τεχνολογίες που καθιστούν δυνατή τη σύνδεση με το διαδίκτυο πλέον έχουν χαμηλό κόστος και το κυριότερο πλεονέκτημά τους είναι η ευκολία στην χρήση τους.

Περίληψη

Η πτυχιακή εργασία αφορά τη δημιουργία ενός συστήματος εποπτείας και ελέγχου θερμοκηπίου, μέσω κατάλληλου υλικού-λογισμικού, το οποίο θα μπορεί να διασυνδέεται στο διαδίκτυο και μέσω αυτού να γίνεται έλεγχος ενεργοποιητών και ταυτόχρονα παρακολούθηση περιβαλλοντικών συνθηκών του θερμοκηπίου.

Πιο συγκεκριμένα, για την υλοποίηση του συστήματος θα χρησιμοποιηθεί η αναπτυξιακή πλατφόρμα Arduino. Το Arduino είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++).

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για τη διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωσή του σε άλλα κυκλώματα. Σε αυτή την πλατφόρμα μπορούν να συνδεθούν ενσύρματα ή ασύρματα διάφοροι αισθητήρες ή ενεργοποιητές (συσκευές δράσης), ενώ παράλληλα δίνει τη δυνατότητα διασύνδεσης και με το διαδίκτυο. Οι αισθητήρες θα καταγράφουν θερμοκρασία και υγρασία αέρα, υγρασία εδάφους, φωτεινότητα και ύψος στάθμης υγρού σε δεξαμενή, ενώ μπορεί διάφορους ελέγχους, όπως έλεγχο ποτίσματος, κ.λπ.

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες, και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring.

Επίσης θα δημιουργηθεί ένας δυναμικός ιστοχώρος στον οποίο θα απεικονίζονται όλες οι μετρήσεις των αισθητήρων, θα γίνεται έλεγχος των κατάλληλων ενεργοποιητών, τόσο με εντολή του χειριστή όσο και αυτόματα, βάσει σεναρίων. Ο σκοπός αυτού του συστήματος είναι η αντικατάσταση της ανθρώπινης παρουσίας όσο βέβαια αυτό είναι εφικτό αλλά και η καταγραφή των συνθηκών σε ένα θερμοκήπιο για την εξαγωγή χρήσιμων συμπερασμάτων.

Τέλος, για την καλύτερη προσέγγιση των πραγματικών συνθηκών και αναγκών ενός θερμοκηπίου θα κατασκευαστεί μια μακέτα.

Περιεχόμενα

Πρόλογος.....	1
Περίληψη.....	2
Περιεχόμενα.....	3
Κατάλογος πινάκων.....	5
Κατάλογος εικόνων.....	6
Κεφάλαιο 1 : Θεωρητικό υπόβαθρο.....	8
1.1 Προγραμματισμός διαδικτύου.....	8
1.1.1 Javascript.....	8
1.1.2 HTML.....	9
1.1.3 CSS.....	12
1.1.4 PHP.....	13
1.1.5 AJAX.....	15
1.1.6 MySQL.....	16
1.2 Αρχιτεκτονική Arduino.....	17
1.2.1 Γενικές πληροφορίες.....	17
1.2.2 Πλεονεκτήματα Arduino.....	18
1.2.3 Εκδόσεις μικροελεγκτών Arduino.....	19
1.2.4 Arduino shields.....	22
1.2.5 Τεχνικά χαρακτηριστικά Arduino redboard.....	24
1.2.5.1 Μικροελεγκτής ATmega 328.....	25
1.2.5.2 Τροφοδοσία Arduino Redboard.....	27
1.2.5.3 Προστασία από υπέρταση.....	28
1.2.5.4 Μνήμη Arduino Redboard.....	28
1.2.5.5 Επικοινωνία Arduino Redboard.....	28
1.2.5.6 Αυτόματη επαναφορά software.....	29
1.2.5.7 Ακροδέκτες Arduino Redboard.....	30
1.2.5.8 Ενσωματωμένα κουμπιά και Led.....	32
1.3 Εργαλεία ανάπτυξης λογισμικού.....	33
1.3.1 Notepad++.....	33
1.3.2 Xampp.....	34
1.3.3 Fritzling.....	34
1.3.4 Arduino IDE.....	35
1.3.4.1 Ολοκληρωμένο περιβάλλον ανάπτυξης.....	35
1.3.4.2 Σειριακή οθόνη.....	36

1.3.4.3 Ανέβασμα σχεδίων.....	37
1.3.4.4 Γλώσσα και δομή προγράμματος.....	38
1.3.4.5 Βιβλιοθήκες.....	41
Κεφάλαιο 2 :Υλικό μέρος συστήματος.....	43
2.1 Επικοινωνία με το διαδίκτυο.....	45
2.2 Περιβαλλοντικοί αισθητήρες.....	50
2.2.1 Αισθητήρας έντασης φωτός.....	51
2.2.2 Αισθητήρας βροχής.....	54
2.2.3 Αισθητήρας υγρασίας χώματος.....	55
2.2.4 Αισθητήρας θερμοκρασίας - υγρασίας.....	56
2.2.5 Αισθητήρας απόστασης υπερήχων.....	60
2.3 Συσκευές δράσης.....	63
2.3.1 Ανεμιστήρας.....	63
2.3.2 Ηλεκτροβάννα.....	64
2.3.3 Διπολικό τρανζίστορ.....	64
2.3.4 Μέθοδος χειρισμού συσκευών δράσης.....	67
2.3.5 Κατασκευή μακέτας – Κόστος συστήματος.....	68
Κεφάλαιο 3 : Λογισμικό μέρος συστήματος.....	71
3.1 Απαιτήσεις συστήματος.....	72
3.2 Λειτουργίες ιστοχώρου.....	73
3.2.1 Αρχική σελίδα.....	73
3.2.2 Σελίδα εποπτείας και ελέγχου.....	76
3.2.3 Σελίδες απεικόνισης μετρήσεων σε πίνακα.....	79
3.2.4 Σελίδες απεικόνισης μετρήσεων σε διάγραμμα.....	82
3.3 Βάση δεδομένων.....	87
3.4 Προγραμματισμός Arduino Redboard.....	90
3.4.1 Βασικές βιβλιοθήκες.....	90
3.4.2 Προγραμματισμός αισθητήρων.....	93
3.4.3 Προγραμματισμός συσκευών δράσης.....	94
3.4.4 Αποστολή μετρήσεων στη βάση δεδομένων.....	96
3.4.5 Επικοινωνία Arduino με την ιστοσελίδα εποπτείας και ελέγχου.....	97
Κεφάλαιο 4 : Συμπεράσματα – Μελλοντικές επεκτάσεις.....	100
Βιβλιογραφία	102
Παράρτημα Α.....	104
Παράρτημα Β.....	110
Παράρτημα Γ.....	121

Κατάλογος πινάκων :

Πίνακας 01:Ετικέτες HTML.....	11
Πίνακας 02:Βασικές μεταβλητές PHP.....	14
Πίνακας 03:Εκδόσεις Arduino.....	19
Πίνακας 04:Arduino shields.....	22
Πίνακας 05:Τεχνικά χαρακτηριστικά Arduino.....	22
Πίνακας 06:Εργαλεία του περιβάλλοντος ανάπτυξης IDE.....	35
Πίνακας 07:Δομές και λειτουργίες προγραμματισμού.....	39
Πίνακας 08:Βιβλιοθήκες Arduino.....	42
Πίνακας 09:Ενημερωτικές φωτεινές ενδείξεις Arduino ethernet shield.....	47
Πίνακας 10:Σύνδεση SPI.....	47
Πίνακας 11:Τεχνικά χαρακτηριστικά αισθητήρα Photocel.....	51
Πίνακας 12:Βαθμονόμιση εξόδου αισθητήρα φωτεινότητας.....	53
Πίνακας 13:Βαθμονόμιση εξόδου αισθητήρα βροχής.....	54
Πίνακας 14:Βαθμονόμιση εξόδου αισθητήρα υγρασίας χώματος.....	55
Πίνακας 15:Επισκόπηση τεχνικών χαρακτηριστικών του DHT11.....	56
Πίνακας 16:Ηλεκτρικά Χαρακτηριστικά για VDD = 5V , T = 25 °C.....	56
Πίνακας 17:Λεπτομερής πίνακας τεχνικών χαρακτηριστικών του DHT11.....	57
Πίνακας 18:Ακροδέκτες αισθητήρα HC-SR04.....	60
Πίνακας 19:Τεχνικά χαρακτηριστικά αισθητήρα HC - SR04.....	61
Πίνακας 20:Τεχνικά χαρακτηριστικά ανεμιστήρα.....	63
Πίνακας 21:Τεχνικά χαρακτηριστικά ηλεκτροβάνας.....	64
Πίνακας 22:Ηλεκτρικά χαρακτηριστικά συσκευών δράσης.....	68
Πίνακας 23:Κόστος εξαρτημάτων.....	69
Πίνακας 24:Πίνακας sensors βάσης δεδομένων.....	87
Πίνακας 25:Πίνακας καταστάσεων.....	93

Κατάλογος εικόνων :

Εικόνα 01:Λογότυπα τεχνολογιών προγραμματισμού διαδικτύου.....	8
Εικόνα 02:Λογότυπο JS.....	8
Εικόνα 03:Παράδειγμα κώδικα Javascript.....	9
Εικόνα 04:Λογότυπο HTML.....	9
Εικόνα 05:Βασικά tags της HTML.....	10
Εικόνα 06:Δέντρο HTML στοιχείων.....	10
Εικόνα 07:Λογότυπο CSS.....	12
Εικόνα 08:Κανόνας σύνταξης CSS.....	12
Εικόνα 09:Παράδειγμα σύνταξης CSS.....	12
Εικόνα 10:Λογότυπο PHP.....	13
Εικόνα 11:Μεταβλητή και δεδομένα.....	13
Εικόνα 12:Αρχιτεκτονική PHP.....	14
Εικόνα 13:Λογότυπο AJAX.....	15
Εικόνα 14:Αρχιτεκτονική AJAX.....	15
Εικόνα 15:Λογότυπο MySQL.....	16
Εικόνα 16:Αρχιτεκτονική MySQL.....	16
Εικόνα 17:Arduino UNO και Redboard.....	17
Εικόνα 18:Κάτοψη μικροελεγκτή ATmega328.....	25
Εικόνα 19:Διάγραμμα ακροδεκτών μικροελεγκτή ATmega 328.....	26
Εικόνα 20:Σχηματικό διάγραμμα ATmega 328.....	26
Εικόνα 21:Ακροδέκτες τροφοδοσίας Redboard.....	27
Εικόνα 22:Κυκλώματα σταθεροποίησης τάσης του Redboard.....	27
Εικόνα 23:Κύκλωμα FTDI FT232RL.....	29
Εικόνα 24: Ακροδέκτες Redboard.....	30
Εικόνα 25:Αναλογικοί ακροδέκτες.....	31
Εικόνα 26:Μέρη του Redboard.....	32
Εικόνα 27:Περιβάλλον Notepad++.....	33
Εικόνα 28:Λογότυπα υποστηριζόμενα από το XAMPP.....	34
Εικόνα 29:Περιβάλλον Fritzning.....	34
Εικόνα 30:Ολοκληρωμένο περιβάλλον ανάπτυξης arduino IDE.....	36
Εικόνα 31:Σειριακή οθόνη arduino IDE.....	36
Εικόνα 32:Επιλογή τύπου πλακέτας.....	37
Εικόνα 33:Επιλογή θύρας usb.....	37
Εικόνα 34:Επιτυχής ανέβασμα σχεδίου (κονσόλα μηνυμάτων).....	37
Εικόνα 35:Δομή προγράμματος.....	38
Εικόνα 36:Διάγραμμα ροής λειτουργίας Wiring.....	39
Εικόνα 37:Τμήματα του συστήματος.....	43
Εικόνα 38:Γενική μορφή συστήματος.....	44
Εικόνα 39:Βασικό υλικό συστήματος.....	44
Εικόνα 40:Κύκλωμα συστήματος.....	45
Εικόνα 41:Τοποθέτηση Ethernet shield.....	45
Εικόνα 42:Arduino redboard – ethernet shield.....	46
Εικόνα 43:Επικοινωνία Ethernet–microSD κάρτα.....	46
Εικόνα 44:Μπλοκ διάγραμμα W5100.....	48
Εικόνα 45:Μέθοδοι πραγματοποίησης σύνδεσης.....	48
Εικόνα 46:Κατασκευή Photocell.....	51
Εικόνα 47:Photocell.....	51

Εικόνα 48:Ευαισθησία – μήκος κύματος.....	52
Εικόνα 49:Αντίσταση – φωτεινή ισχύς.....	52
Εικόνα 50:Σύνδεση αισθητήρα photocell.....	52
Εικόνα 51:Ηλεκτρολογικό σχέδιο σύνδεσης photocell.....	52
Εικόνα 52:Αισθητήρας βροχής FC-37.....	54
Εικόνα 53:Σύνδεση αισθητήρα βροχής.....	54
Εικόνα 54:Αισθητήρας βροχής FC-28.....	55
Εικόνα 55:Σύνδεση αισθητήρα βροχής.....	55
Εικόνα 56:Αισθητήρας DHT11.....	56
Εικόνα 57:Διάγραμμα σύνδεσης αισθητήρα DHT11.....	57
Εικόνα 58:Συνολική διαδικασία επικοινωνίας.....	58
Εικόνα 59:Σήμα εκκίνησης από MCU και απάντησης από DHT11.....	59
Εικόνα 60:Ακροδέκτες αισθητήρα DHT11.....	59
Εικόνα 61:Σύνδεση αισθητήρα DHT11.....	59
Εικόνα 62:Αισθητήρας HC-SR04.....	60
Εικόνα 63:Σύνδεση αισθητήρα απόστασης HC-SR04.....	60
Εικόνα 64:Χρονικό διάγραμμα αισθητήρα HC-SR04.....	61
Εικόνα 65:Διαστάσεις και γωνία μέτρησης αισθητήρα HC-SR04.....	61
Εικόνα 66:Ανεμιστήρας μακέτας.....	63
Εικόνα 67:Ηλεκτροβάννα.....	64
Εικόνα 68:Τομή ηλεκτροβάννας.....	64
Εικόνα 69:Δομή και κυκλωματικό σύμβολο τρανζίστορ.....	65
Εικόνα 70:Πόλωση τρανζίστορ και ροή ελεύθερων ηλεκτρονίων και οπών.....	66
Εικόνα 71:Τρανζίστορ TIP120.....	67
Εικόνα 72:Μέθοδος ελέγχου συσκευών.....	67
Εικόνα 73:Φωτογραφία μακέτας θερμοκηπίου 1.....	69
Εικόνα 74:Φωτογραφία μακέτας θερμοκηπίου 2.....	69
Εικόνα 75:Φωτογραφία μακέτας θερμοκηπίου 3.....	69
Εικόνα 76:Φωτογραφία μακέτας θερμοκηπίου 4.....	69
Εικόνα 77:Λογισμικά μέρη συστήματος.....	70
Εικόνα 78:Αρχική σελίδα.....	72
Εικόνα 79:Σελίδα εποπτείας και ελέγχου.....	75
Εικόνα 80:Πίνακας εποπτείας.....	75
Εικόνα 81:Αυτόματα σενάρια.....	77
Εικόνα 82:Πίνακας ελέγχου συσκευών δράσης.....	77
Εικόνα 83:Σελίδα απεικόνισης μετρήσεων σε πίνακα.....	78
Εικόνα 84:Κουμπι μεταφοράς στην σελίδα απεικόνισης μετρήσεων σε πίνακα.....	78
Εικόνα 85:Κουμπία σελίδων διαγραμμάτων.....	81
Εικόνα 86:Κουμπία μεταφοράς στις σελίδες διαγραμμάτων.....	81
Εικόνα 87:Ιστοσελίδα απεικόνισης αισθητήρα φωτεινότητας σε διάγραμμα.....	82
Εικόνα 88:Ιστοσελίδα απεικόνισης μετρήσεων βροχής σε διάγραμμα.....	82
Εικόνα 89:Ιστοσελίδα απεικόνισης μετρήσεων υγρασίας χώματος σε διάγραμμα.....	83
Εικόνα 90:Ιστοσελίδα απεικόνισης μετρήσεων θερμοκρασίας – υγρασίας σε διάγραμμα.....	83
Εικόνα 91:Ιστοσελίδα απεικόνισης μετρήσεων ύψους δεξαμενής σε διάγραμμα.....	84
Εικόνα 92:Λογότυπο εταιρίας Amcharts.....	84
Εικόνα 93:Δημιουργία βάσης δεδομένων.....	86
Εικόνα 94:Query δημιουργίας πίνακα.....	86
Εικόνα 95:Πίνακας sensors βάσης δεδομένων.....	87

Κεφάλαιο 1: Θεωρητικό υπόβαθρο

Το αντικείμενο μελέτης του κεφαλαίου αυτού, αποτελείται από το θεωρητικό υπόβαθρο που ήταν απαραίτητο για την εκπόνηση της πτυχιακής εργασίας. Πιο συγκεκριμένα αναλύονται όλες οι γνώσεις και συνάμα τα εργαλεία που χρησιμοποιήθηκαν, για το σχεδιασμό και την υλοποίηση της παρούσας εργασίας. Τρεις ενότητες απαρτίζουν το κεφάλαιο αυτό και ασχολούνται με τον προγραμματισμό του διαδικτύου, την αρχιτεκτονική Arduino που επιλέχθηκε για τη σχεδίαση του συστήματος και τέλος με τις πλατφόρμες ανάπτυξης λογισμικού που χρησιμοποιήθηκαν.

1.1 Προγραμματισμός διαδικτύου

Ο προγραμματισμός διαδικτύου έχει να κάνει με τη γνώση γύρω από διαφορετικές τεχνολογίες που επιλέχθηκαν για τον σχεδιασμό και τη σύσταση του ιστοχώρου διαχείρισης του θερμοκηπίου, ο οποίος αποτελεί σημαντικό παράγοντα, στην ορθή λειτουργία του συστήματος καταγραφής και εποπτείας των περιβαλλοντικών συνθηκών. Σε αυτή την ενότητα γίνεται αναφορά στις γλώσσες προγραμματισμού που μπορούν να φέρουν εις πέρας ένα τέτοιο έργο.



Εικόνα 1: Λογότυπα τεχνολογιών προγραμματισμού διαδικτύου.

1.1.1 Javascript

Η JavaScript είναι μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web) και ταυτόχρονα η δημοφιλέστερη scripting γλώσσα στον κόσμο. 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. Αρχικά όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα αυτή διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές.



Εικόνα 2: Λογότυπο JS

Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των πλαισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web. Είναι η τυπική γλώσσα που χρησιμοποιείται σε ιστοσελίδες, αλλά χρησιμοποιείται ευρέως επίσης από desktop εφαρμογές, εφαρμογές κινητής τηλεφωνίας, καθώς και διακομιστές διαδικτύου.

Η αρχική έκδοση της Javascript βασίστηκε στη σύνταξη στη γλώσσα προγραμματισμού C, αν και έχει εξελιχθεί, ενσωματώνοντας πια χαρακτηριστικά από νεότερες γλώσσες. Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο φυλλομετρητής (browser) του χρήστη, και χαρακτηρίστηκε σαν client-side γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML.

Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού). Η JavaScript έχει τη δυνατότητα δυναμικής αλλαγής ετικετών και περιεχομένου της HTML, δηλαδή μπορεί να διαβάσει και να αλλάξει το περιεχόμενο των στοιχείων της. Επίσης μπορεί να χειριστεί το CSS. Μπορεί ακόμα να χρησιμοποιηθεί για την επικύρωση δεδομένων, όπως η επικύρωση μιας φόρμας εισόδου όπως και για την αποθήκευση και ανάκτηση πληροφοριών στον υπολογιστή του χρήστη. Ακόμα είναι δυνατόν να ρυθμιστεί ώστε να εκτελείται όταν συμβαίνει κάτι, όπως όταν ένας χρήστης κάνει κλικ σε ένα στοιχείο HTML. Και τέλος μπορεί να επικοινωνήσει με php, xml, json αρχεία και με άλλες ιστοσελίδες. Ο κώδικας Javascript μιας σελίδας περικλείεται από τις ετικέτες της HTML `<script type="text/javascript">` και `</script>`. Ακολουθεί ένα απλό παράδειγμα κώδικα ενός πλαισίου διαλόγου με το κείμενο "Hello world!":

```
<script type="text/javascript">
alert('Hello world!');
</script>
```

Εικόνα 3: Παράδειγμα κώδικα Javascript.

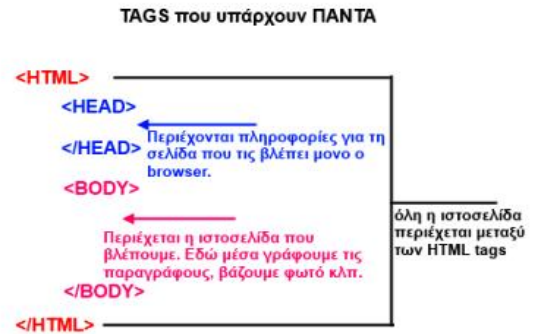
1.1.2 HTML

Η HTML (HyperText Markup Language, ελλ. Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας.



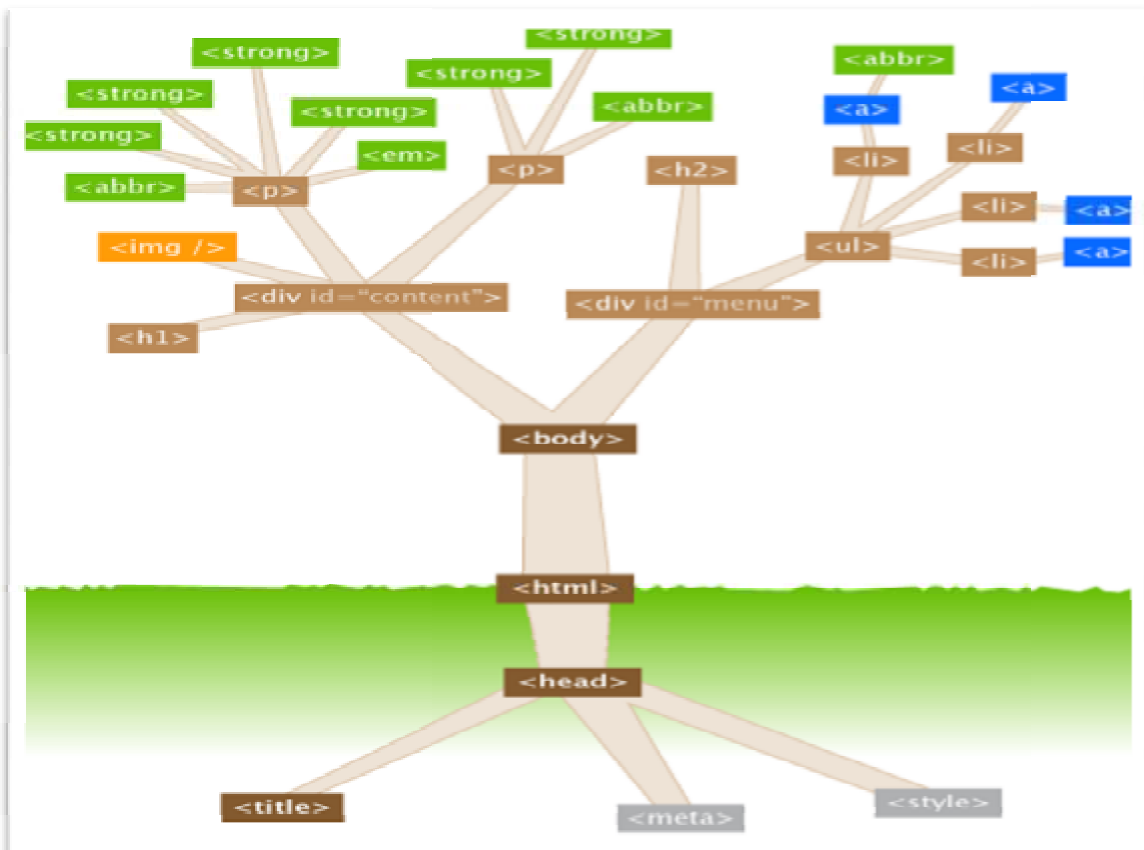
Εικόνα 4: Λογότυπο HTML.

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 διαβάσει τα έγγραφα HTML και τα συνθέσει 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.



Εικόνα 5: Βασικά tags της HTML

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.



Εικόνα 6: Δέντρο HTML στοιχείων

Μερικά στοιχεία που χρησιμοποιήθηκαν φαίνονται στον παρακάτω πίνακα:

Στοιχείο	Λειτουργία
<code><!DOCTYPE></code>	Η δήλωση του DOCTYPE πρέπει να είναι η πρώτη που γίνεται σε ένα έγγραφο HTML. Πρόκειται για μια εντολή στο πρόγραμμα περιήγησης που έχει να κάνει με την έκδοση της HTML στο έγγραφο.
<code><html></code>	Με την ετικέτα <code><html></code> αρχίζουμε πάντα τον κώδικα μας και με την ετικέτα <code></html></code> τον τερματίζουμε. Έτσι πληροφορούμε τον browser ότι υπάρχει κώδικας γραμμένος σε γλώσσα HTML.
<code><head></code>	Αποτελούν τον πρόλογο για την HTML σελίδα. Είναι βασική ετικέτα αφού στο πεδίο της ορίζονται δεδομένα όπως το είδος των χαρακτήρων που χρησιμοποιούνται, εξωτερικές πηγές αρχείων κ.α
<code><title></code>	Περιλαμβάνεται στην ενότητα <code><head></code> και καθορίζει τον τίτλο της σελίδας ο οποίος εμφανίζεται στο πάνω μέρος του παραθύρου του web browser.
<code><body></code>	Ορίζει το κυρίως περιεχόμενο της σελίδας μέσα στο οποίο γράφουμε το κείμενο που θέλουμε να εμφανιστεί μαζί με τις HTML ετικέτες που το μορφοποιούν. Στην ενότητα αυτή τοποθετούμε επίσης εικόνες, video κ.α
<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><h1> - <h6></code>	Οι HTML επικεφαλίδες είναι κείμενο που εμφανίζετε με μεγάλα και έντονα γράμματα. Ορίζονται από τις ετικέτες <code><h1></code> έως <code><h6></code> . Με την <code><h1></code> ορίζουμε την μεγαλύτερη ετικέτα ενώ με την <code><h6></code> την μικρότερη.
<code><p></code>	Οι παράγραφοι ορίζονται από το ζευγάρι ετικετών <code><p></code> και <code></p></code>
<code>
</code>	Η ετικέτα <code>
</code> χρησιμοποιείται όταν θέλουμε να τελειώσουμε μια γραμμή κειμένου και να αρχίσουμε μια καινούργια.
<code><hr></code>	Η οριζόντια γραμμή είναι ένα απλό γραφικό που μπορούμε να χρησιμοποιήσουμε στην σελίδα μας (κυρίως σαν διαχωριστικό). Με την ετικέτα <code><hr></code> τοποθετούμε μια οριζόντια γραμμή στην σελίδα μας.
<code><table></code>	Ορίζουμε έναν πίνακα. Το ζεύγος των ετικετών <code><tr>...</tr></code> ορίζει μια γραμμή του πίνακα, το <code><td>...</td></code> ορίζουν ένα κελί στην γραμμή του πίνακα και το <code><th></code> ορίζει μια επικεφαλίδα σε μια στήλη του πίνακα.
<code><form></code>	Για να δημιουργήσουμε μια φόρμα χρησιμοποιούμε τις ετικέτες <code><form></code> και <code></form></code> . Οι κυριότερες ιδιότητες της ετικέτας είναι οι name, method και action
<code><input></code>	Εισάγουμε τα περισσότερα στοιχεία της φόρμας. Οι κυριότερες ιδιότητες της ετικέτας είναι η type η οποία καθορίζει τον τύπο του στοιχείου της Φόρμας (Πεδίο Κειμένου ή Περιοχή Κειμένου ή Κουμπί Επιλογών κ.α)

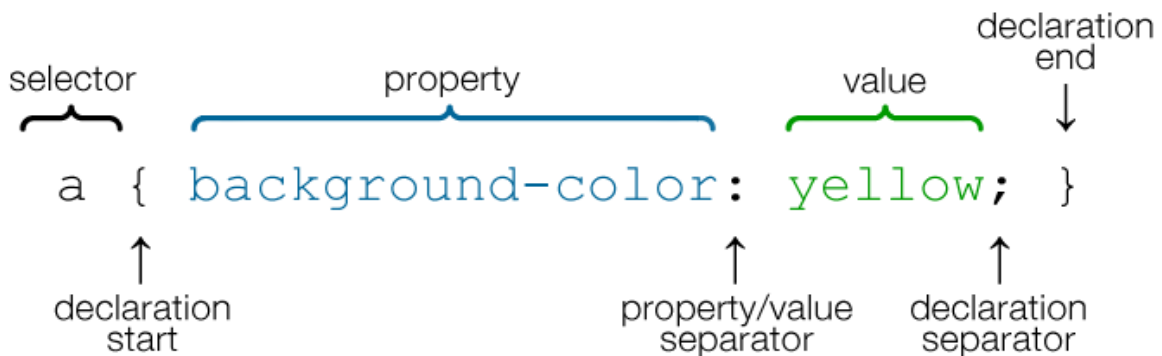
Πίνακας 1: Ετικέτες HTML

1.1.3 CSS

Η CSS (Cascading Style Sheets-Διαδοχικά Φύλλα Στιλ) ή (αλληλουχία φύλλων στιλ) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στιλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. 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 στιλιστικά μια ιστοσελίδα 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. Το πρώτο μέρος είναι ο επιλογέας (selector) και το δεύτερο μέρος είναι οι δηλώσεις (declaration). Ο επιλογέας αφορά το στοιχείο το οποίο είναι προς μορφοποίηση και είναι είτε μία ετικέτα html είτε κάποιο συγκεκριμένο αντικείμενο html. Οι δηλώσεις αποτελούνται από τις ιδιότητες που χαρακτηρίζουν τον επιλογέα και τις τιμές που αυτές παίρνουν. Η παρακάτω εικόνα απεικονίζει ένα πρότυπο σύνταξης κανόνα CSS:



Εικόνα 7: Λογότυπο CSS.



Εικόνα 8: Κανόνας σύνταξης CSS

```
h1 { color: white;
      background: orange;
      border: 1px solid black;
      padding: 0 0 0 0;
      font-weight: bold;
}
/* begin: seaside-theme */

body {
  background-color: white;
  color: black;
  font-family: Arial, sans-serif;
  margin: 0 4px 0 0;
  border: 12px solid;
}
```

Εικόνα 9: Παράδειγμα σύνταξης CSS

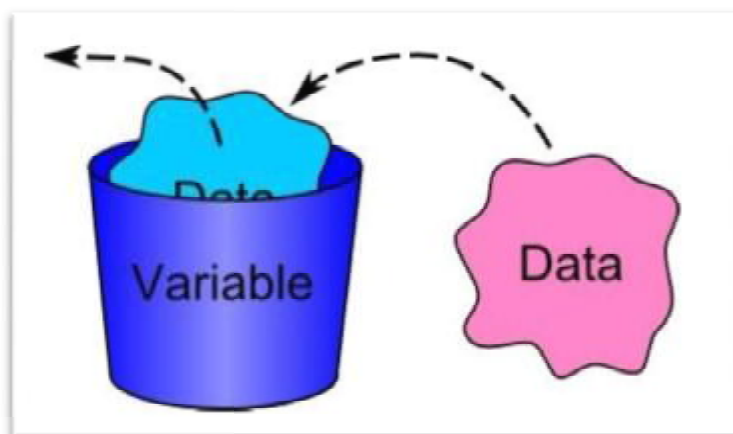
1.1.4 PHP

Η PHP (Hypertext PreProcessor), είναι μια γλώσσα προγραμματισμού βασισμένη σε σενάρια που χρησιμοποιείται ευρέως και η οποία μπορεί να ενσωματωθεί στον κώδικα της HTML. Πρόκειται για μια γλώσσα προγραμματισμού με σκοπό τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. 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. Ενσωματώνοντας μικρά προγράμματα σεναρίων σε έναν HTML κώδικα, αποκτούμε τη δυνατότητα να ελέγχουμε τι θα εμφανίζεται στην οθόνη του φυλλομετρητή με τρόπο πολύ πιο ευέλικτο από ότι με τη χρήση της HTML.



Εικόνα 10: Λογότυπο PHP

Η γλώσσα αυτή εκτελείται στην πλευρά του εξυπηρετητή, κάτι που την κάνει να ξεχωρίζει από άλλες γλώσσες όπως η JavaScript, και κάτι που σημαίνει ότι ο εξυπηρετητής ιστού τη διερμηνεύει προτού ακόμα τη στείλει στο φυλλομετρητή. Σχεδιάστηκε με σκοπό την ανάπτυξη διαδικτυακών εφαρμογών και τη δημιουργία δυναμικού περιεχομένου ιστοσελίδων. Το πιο σημαντικό κομμάτι της, όμως είναι το γεγονός ότι υποστηρίζει ένα ευρύ σύνολο από βάσεις δεδομένων, κάτι που κάνει πολύ πιο εύκολη και απλή τη δημιουργία μιας ιστοσελίδας η οποία επικοινωνεί με βάσεις δεδομένων. Όπως αναφέρθηκε παραπάνω η γλώσσα php περιλαμβάνει κάποιες βασικές μεταβλητές οι οποίες έχουν σκοπό να βοηθούν στην ανάπτυξη διαδικτυακών δυναμικών εφαρμογών.

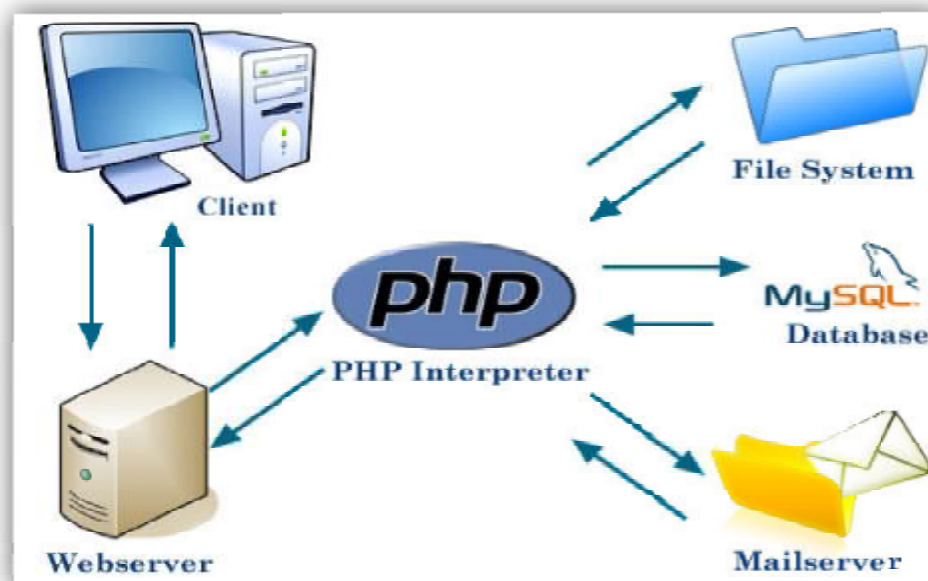


Εικόνα 11: Μεταβλητή και δεδομένα

Έχει το ιδιαίτερο χαρακτηριστικό ότι ο κώδικάς της πρώτα μεταγλωττίζεται στον server και μετά φορτώνεται σαν ένα κανονικό html έγγραφο, χωρίς ο χρήστης να είναι σε θέση να δει τον αρχικό κώδικα. Η PHP είναι δωρεάν για χρήση, και είναι η δημοφιλέστερη για τους hosts Unix και Linux, αν και υπάρχουν εκδόσεις διαθέσιμες για τα Windows. Στον παρακάτω πίνακα αναλύονται οι μεταβλητές αυτές:

\$_GET	Η προκαθορισμένη μεταβλητή <code>\$_GET</code> 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.
\$_POST	Η προκαθορισμένη μεταβλητή <code>\$_POST</code> χρησιμοποιείται για τη συλλογή τιμών σε μια φόρμα (method= "POST"). Οι πληροφορίες που αποστέλλονται από μια φόρμα με την μέθοδο POST είναι αόρατες στους άλλους και δεν έχει όρια για τον όγκο των πληροφοριών κατά την αποστολή.
\$_REQUEST	Περιέχει όλες τις μεταβλητές που στέλνονται μέσω των HTTP GET, HTTP POST και HTTP cookies. Αποτελεί στην ουσία το ισοδύναμο του συνδυασμού των <code>\$_GET</code> , <code>\$_POST</code> και <code>\$_COOKIE</code> και είναι λιγότερο επικίνδυνος από τη χρήση του <code>\$GLOBALS</code> . Όμως, καθώς περιέχει όλες τις μεταβλητές από μη έμπιστες πηγές, όπως είναι οι επισκέπτες της ιστοσελίδας μας, θα πρέπει να τον χρησιμοποιούμε με επιφύλαξη.
\$_COOKIE	Ένα cookie είναι ένα μικρό αρχείο που ενσωματώνει ο server στον υπολογιστή του χρήστη. Η συνάρτηση <code>setcookie()</code> χρησιμοποιείται για να ορίσει ένα cookie.
\$_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 αγорών, 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.

Πίνακας 2: Βασικές μεταβλητές PHP



Εικόνα 12: Αρχιτεκτονική PHP

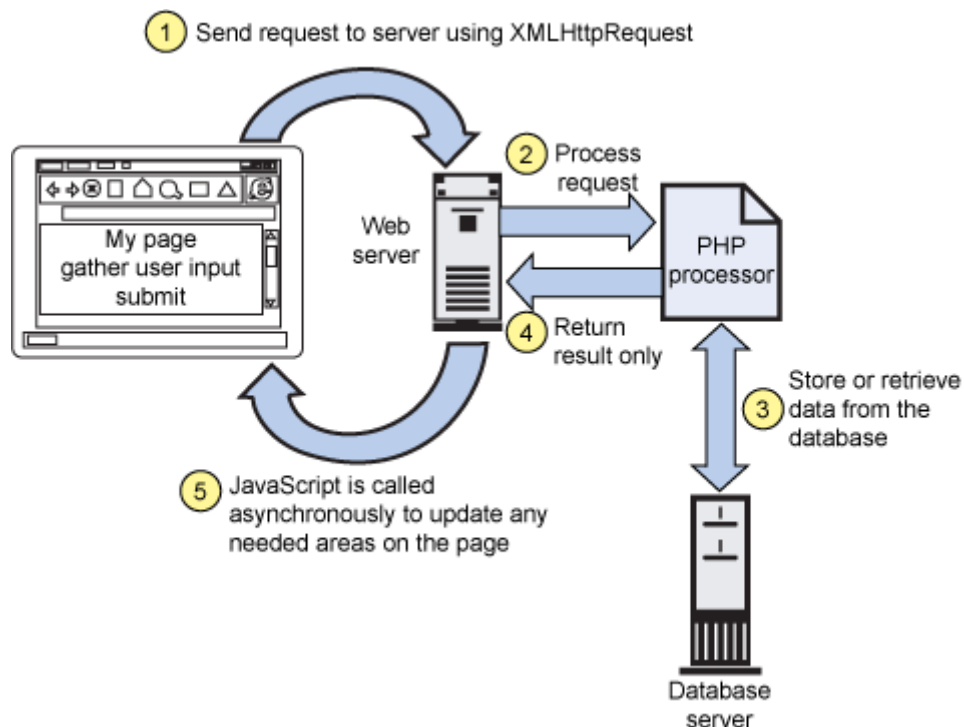
1.1.5 AJAX

Η AJAX δεν είναι γλώσσα προγραμματισμού όπως η JavaScript, αλλά ούτε και γλώσσα χαρακτηρισμού κειμένου όπως η HTML. Δεν θεωρείται καν γλώσσα. AJAX σημαίνει Asynchronous Javascript And XML και είναι η τεχνική με την οποία μπορούμε να δημιουργήσουμε πολύ πιο γρήγορες και δυναμικές ιστοσελίδες περιορίζοντας τον όγκο δεδομένων που ανταλλάσσει ο server με τον browser του επισκέπτη. Η ιδιαίτερη τεχνική αυτή επιτρέπει την ανανέωση περιεχομένων μιας ιστοσελίδας χωρίς αυτή να ανανεωθεί ολόκληρη. Παραδείγματα των εφαρμογών που χρησιμοποιούν AJAX: Google Maps, Gmail, YouTube, Facebook.



Εικόνα 13: Λογότυπο AJAX

Η υλοποίηση της τεχνικής AJAX γίνεται με τον συνδυασμό του αντικειμένου (object) XMLHttpRequest (πραγματοποιεί την ασύγχρονη επικοινωνία με τον server), την Javascript/DOM (αλληλεπιδρά με τα δεδομένα και τα εμφανίζει), την CSS (μορφοποιεί τα προς εμφάνιση δεδομένα) και την XML (συχνά χρησιμοποιείται για την μεταφορά δεδομένων). Η παρακάτω εικόνα παρουσιάζει με τη μορφή σχεδιαγράμματος τον τρόπο λειτουργίας του AJAX.



Εικόνα 14: Αρχιτεκτονική AJAX

1.1.6 MySQL

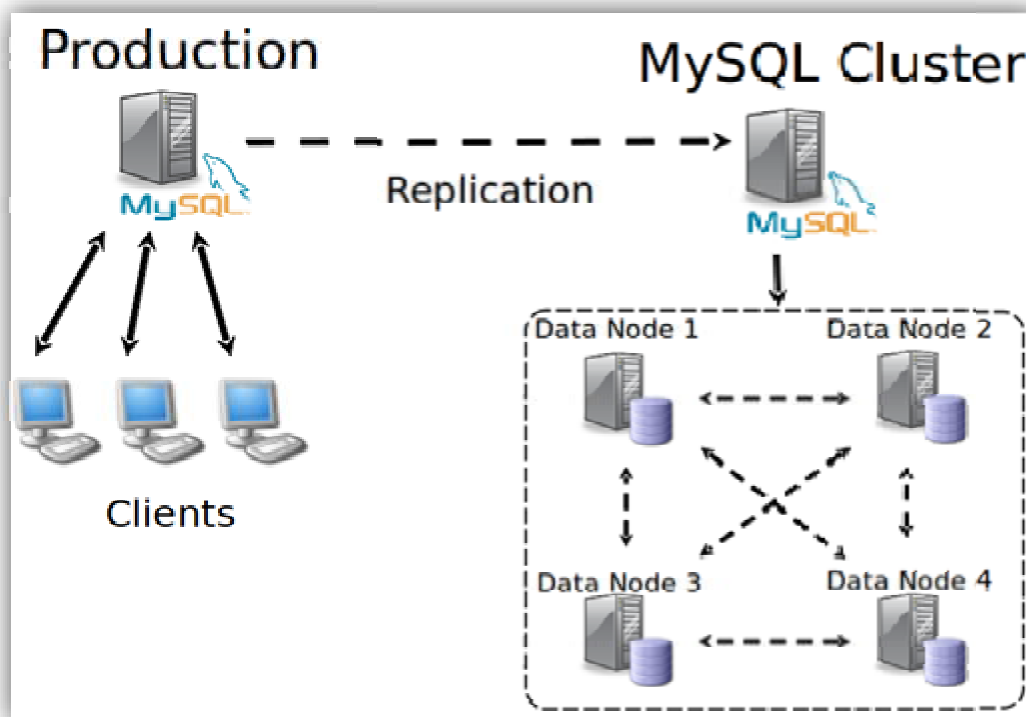
Η MySQL είναι ένα πολύ γρήγορο σε απόδοση και ισχυρό σε δυνατότητες υλοποίησης, σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντου Βιντένιους, τη Μάι.



Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Ο κωδικός του εγχειρίματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιοκτητών συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.

Εικόνα 15: Λογότυπο MySQL

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. Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες.



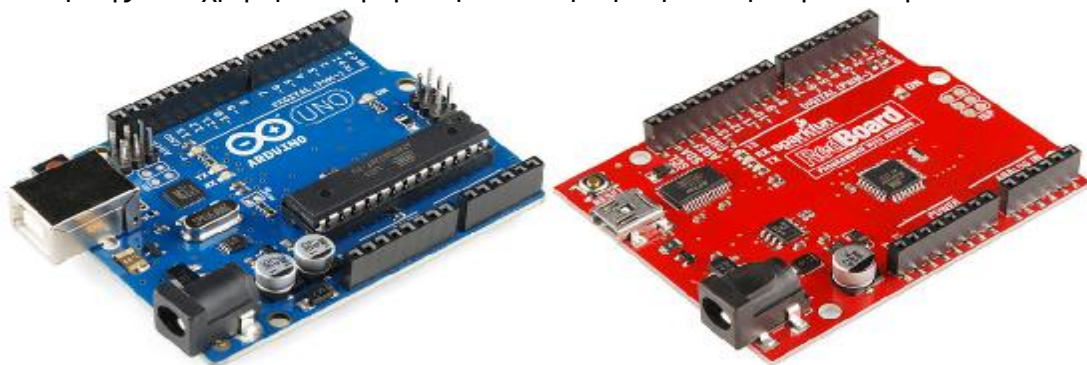
Εικόνα 16: Αρχιτεκτονική MySQL

1.2 Αρχιτεκτονική Arduino

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

1.2.1 Γενικές πληροφορίες

Το Arduino είναι ένας single-board μικροελεγκτής, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Έτσι έχει τη δυνατότητα ο προγραμματιστής-χρήστης να δημιουργήσει λογισμικό το οποίο αυτοματοποιεί τις δραστηριότητες που εκείνος επιθυμεί. Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων που αλληλεπιδρούν με το περιβάλλον κάνοντας λήψη σημάτων μέσα από μία ποικιλία αισθητήρων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες. Αλλά υπάρχουν και εκδόσεις για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους αφού τα σχηματικά και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα. Στην ενότητα αυτή παρουσιάζεται ο μικροελεγκτής arduino uno ο οποίος αποτελεί την πιο διαδεδομένη έκδοση, ο arduino redboard που χρησιμοποιήθηκε στην εργασία, τα χαρακτηριστικά τους, διαθέσιμα εξαρτήματα με δυνατότητα απ'ευθείας σύνδεσης αλλά και ανάλυση αυτών που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος. Οι διαφορές των δύο μικροελεγκτών είναι λιγότες καθώς ο μικροελεγκτής που χρησιμοποιήθηκε πρόκειται για μια βελτιωμένη έκδοση του arduino uno.



Εικόνα 17: Arduino UNO και Redboard.





1.2.2 Πλεονεκτήματα Arduino

Υπάρχει μεγάλη ποικιλία στο εμπόριο σε αναπτυξιακές πλατφόρμες και μικροελεγκτές, ώστε να ασχοληθεί κάποιος και να υλοποιήσει εφαρμογές. Πολλά δε απ' αυτά, παρέχουν στο χρήστη εργαλεία, τα οποία είναι απλοποιημένα και κρύβουν τις δύσκολες λεπτομέρειες της αρχιτεκτονικής, επιτρέποντας τον άμεσο προγραμματισμό του μικροελεγκτή. Ακόμα υπάρχουν πακέτα «έτοιμα προς χρήση», τα οποία είναι σχεδιασμένα για αρχάριους χρήστες. Το Arduino σε σχέση με τις υπόλοιπες αναπτυξιακές πλατφόρμες διαφέρει σε αρκετούς τομείς. Κατ' αρχάς απλοποιεί ιδιαίτερα την διαδικασία ανάπτυξης εφαρμογών με μικροελεγκτές. Επίσης παρέχει επιπλέον πλεονεκτήματα που ενδείκνυνται για χρήση από χρήστες με ελάχιστες γνώσεις στον τομέα αυτό, τα οποία είναι:

- Ø Τρέχει σε διάφορα λειτουργικά συστήματα: Ενώ οι περισσότερες αναπτυξιακές πλατφόρμες μικροελεγκτών περιορίζονται στα Windows, το περιβάλλον προγραμματισμού του Arduino είναι διαθέσιμο για Windows, Macintosh OSX και λειτουργικά συστήματα Linux.
- Ø Απλό προγραμματιστικό περιβάλλον: Το περιβάλλον προγραμματισμού του Arduino είναι ευέλικτο για προχωρημένους χρήστες, αλλά ταυτόχρονα ενδείκνυται και για αρχάριους.
- Ø Ανοιχτού λογισμικού: Το λογισμικό του Arduino διανέμεται με τη μορφή ανοιχτού λογισμικού και είναι διαθέσιμο προς επέκταση. Η γλώσσα προγραμματισμού του μπορεί να επεκταθεί με τη βοήθεια βιβλιοθηκών της C++ και της AVR C που είναι για τον προγραμματισμό των μικροελεγκτών Atmel, γλώσσα στην οποία βασίστηκε το λογισμικό του Arduino. Με τον ίδιο τρόπο μπορεί να προστεθεί κώδικας AVR C που έχει ήδη γραφτεί, σε πρόγραμμα Arduino.
- Ø Ανοιχτού υλικού: Η αναπτυξιακή πλατφόρμα Arduino βασίζεται στον μικροελεγκτή ATmega328 της Atmel. Η εταιρεία Creative Commons διαθέτει άδεια χρήσης των σχηματικών της αναπτυξιακής πλακέτας, αλλά επιτρέπει στους σχεδιαστές να κατασκευάσουν το δικό τους, εξελίσσοντας το ήδη υπάρχον χωρίς να έχουν νομικά προβλήματα. Επίσης υπάρχει στο εμπόριο πληθώρα περιφερειακών, τα οποία είναι κατασκευασμένα αποκλειστικά για να συνεργάζονται με την πλακέτα Arduino.
- Ø Χαμηλό κόστος: Σε σύγκριση με άλλες πλατφόρμες μικροελεγκτών, οι πλακέτες Arduino είναι εξαιρετικά φθηνές. Επίσης υπάρχει η δυνατότητα κατασκευής μίας φθηνότερης εκδοχής Arduino, καθώς τα σχηματικά βρίσκονται δωρεάν στο διαδίκτυο.

1.2.3 Εκδόσεις μικροελεγκτών arduino

Το πρωτότυπο υλικό του Arduino κατασκευάζεται από την Ιταλική εταιρία Smart Projects. Κάποιες πλακέτες με την μάρκα του Arduino έχουν σχεδιαστεί από την Αμερικάνικη εταιρία SparkFun Electronics. Οι διαφορές τους συνήθως είναι στους ακροδέκτες (pins), στην τάση εισόδου και εξόδου, στον επεξεργαστή και στα χαρακτηριστικά των συστημάτων που υλοποιούν. Δεκαέξι εκδοχές του Arduino Hardware έχουν χρησιμοποιηθεί εμπορικά μέχρι τώρα. Στον παρακάτω πίνακα γίνεται παρουσίαση των μοντέλων Arduino που κυκλοφορούν στην αγορά, με αναφορά στα κύρια χαρακτηριστικά τους.

<p><u>Arduino Uno/Duemilanove</u></p> <p>Η πιο γνωστή πλακέτα με χαμηλό κόστος και αρκετούς ακροδέκτες για απλές εφαρμογές. Το Arduino Uno, χρησιμοποιεί την ίδια τεχνολογία ATmega328 όπως το τελευταίο μοντέλο Duemilanove.</p>	
<p><u>Arduino Esplora</u></p> <p>Είναι ένα έτοιμο προς χρήση χειριστήριο με εμφάνιση που παραπέμπει σε χειριστήριο κονσόλας βιντεοπαιχνιδιών με joystick και ενσωματωμένους αισθητήρες για ήχο, φως, θερμοκρασία και επιτάχυνση.</p>	
<p><u>Arduino Leonardo</u></p> <p>Το Arduino Leonardo, με ένα Atmega 32U4 chip που εξαλείφει την ανάγκη για συνδεσιμότητα μέσω USB και μπορεί να χρησιμοποιηθεί ως ψηφιακό πληκτρολόγιο ή ποντίκι.</p>	
<p><u>Arduino Ethernet</u></p> <p>Επιτρέπει την σύνδεση με το διαδίκτυο μέσω ενός καλωδίου RJ45. Επίσης υπάρχει έκδοση όπου η τροφοδοσία γίνεται μέσω του καλωδίου Ethernet και υποδοχή για microSD κάρτα.</p>	

Arduino YUN

Είναι ένας συνδυασμός του Arduino και Linux. Πιο συγκεκριμένα είναι βασισμένο στο Leonardo με ενσωματωμένη κεραία Wi-Fi αλλά και υποδοχή για καλώδιο RJ45.



Arduino Bluetooth

Υποστηρίζει ασύρματη σειριακή επικοινωνία μέσω bluetooth (αλλά δεν είναι συμβατό με Bluetooth ακουστικά ή άλλες συσκευές ήχου).



Arduino Due

Είναι η πρώτη πλακέτα Arduino που βασίζεται σε έναν 32-bit πυρήνα ARM μικροελεγκτή. Διαθέτει 54 ψηφιακές εισόδους / εξόδους και η τάση λειτουργίας στις ακίδες είναι 3,3V και όχι 5V



Arduino ADK

Βασισμένο στο Arduino Mega. Έχει μια θύρα USB για σύνδεση με τα τηλέφωνα Android.



Arduino Mega






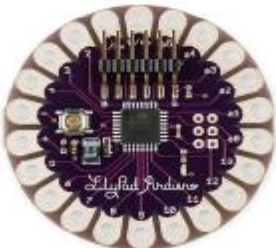
Το Arduino Mega χρησιμοποιεί τεχνολογία surface-mounted ATmega2560 φέρνοντας την ολική μνήμη στα 256kB. Εκτός από τη μνήμη το κύριο πλεονέκτημα του είναι οι 54 ψηφιακές ακίδες εισόδου / εξόδου οι 12 αναλογικές εισόδους και ένα ρολόι 84 MHz.



Arduino Pro

Μικρό σε μέγεθος και χωρίς ακίδες για να έχει τη δυνατότητα προσαρμογής σε μικρά έργα. Υπάρχει σε 2 εκδόσεις με τάση λειτουργίας στις ακίδες 3,3V ή 5V.












<p style="text-align: center;"><u>Arduino Fio</u></p> <p>Προορίζεται για ασύρματες εφαρμογές. Σύνδεση με καλώδιο FTDI.Επιπλέον, με τη χρήση τροποποιημένου προσαρμογέα USB-to-XBee υπάρχει δυνατότητα φόρτωσης κώδικα ασύρματα.Διαθέτει συνδέσεις για μπαταρία Lithium</p>	
<p style="text-align: center;"><u>Arduino Micro</u></p> <p>Μικρό μέγεθος με δυνατότητες παρόμοιες με το Leonardo και ενσωματωμένη επικοινωνία USB.</p>	
<p style="text-align: center;"><u>Arduino Mini</u></p> <p>Το Arduino Pro Mini προορίζεται για ημιμόνιμη εγκατάσταση σε αντικείμενα ή εκθέσεις. Έρχεται χωρίς προτοποθετημένες ακίδες, επιτρέποντας τη χρήση των διαφόρων τύπων των συνδέσμων ή άμεση συγκόλληση των καλωδίων.</p>	
<p style="text-align: center;"><u>Arduino Nano</u></p> <p>Ένα ακόμα μικρό Arduino, με τροφοδοσία μέσω usb. Έχει περισσότερο ή λιγότερο την ίδια λειτουργικότητα του Arduino Duemilanove, αλλά σε διαφορετικό μέγεθος.</p>	
<p style="text-align: center;"><u>Serial Arduino</u></p> <p>Συνδιάζεται με άλλες πλακέτες Arduino οι οποίες δεν έχουν τη δυνατότητα επικοινωνίας μέσω usb.</p>	
<p style="text-align: center;"><u>Arduino LilyPad</u></p> <p>Συνδυάζεται με άλλες πλακέτες Arduino οι οποίες δεν έχουν τη δυνατότητα επικοινωνίας μέσω usb.</p>	

Πίνακας 3:Εκδόσεις Arduino

1.2.4 Arduino shields

Η έννοια του Shield περιγράφει την ενσωμάτωση επιπλέον υλικού στον μικροελεγκτή που του προσδίδει νέες ιδιότητες. Τα Arduino χρησιμοποιούν την τεχνολογία των shields, τυπωμένων boards επεκτάσεων κυκλωμάτων που συνδέονται στα κανονικά παρεχόμενα Arduino pin-headers. Τα shields μπορούν να παρέχουν έλεγχο κινητήρων, επικοινωνία με το διαδίκτυο κ.α. Παρακάτω φαίνονται τα πιο δημοφιλή Shields.





<p><u>Ethernet Shield</u></p> <p>Συνδέει το Arduino στο διαδίκτυο με ένα RJ45 καλώδιο. Διαθέτει υποδοχή για microSD η οποία χρησιμοποιείται για αποθήκευση αρχείων. Υπάρχει έκδοση με Power Over Ethernet (PoE) μονάδα η οποία επιτρέπει την τροφοδοσία μέσω της σύνδεσης Ethernet.</p>	 A blue printed circuit board (PCB) with various electronic components. It features a USB Type-A port on the left, a microSD card slot, and a standard RJ45 Ethernet port. The board is populated with several integrated circuits, capacitors, and other surface components.
<p><u>GSM Shield</u></p> <p>Το Arduino GSM Shield επιτρέπει σε μια πλακέτα Arduino να συνδεθεί στο Internet, να κάνει ή να δέχονται φωνητικές κλήσεις και να στέλνει ή να λαμβάνει μηνύματα SMS. Η Ασπίδα χρησιμοποιεί ένα radio modem M10 από Quectel.</p>	 A blue PCB with a prominent silver metal shield covering a central component, likely the GSM module. It has a microUSB port and a microSD card slot. The board is populated with various electronic components.
<p><u>Wifi Shield</u></p> <p>Το Arduino WiFi Shield επιτρέπει σε μια πλακέτα Arduino να συνδεθεί στο Internet ασύρματα σύμφωνα με τα πρότυπα 802.11b και 802.11g.</p>	 A blue PCB with a central square chip, likely the WiFi module. It features a microUSB port and a microSD card slot. The board is populated with various electronic components.
<p><u>Motor Shield</u></p> <p>Ελέγχει επαγωγικά φορτία, όπως ρελέ και DC ή βηματικούς κινητήρες. Αυτό επιτρέπει έλεγχο δύο κινητήρων συνεχούς ρεύματος, μέτρηση ταχύτητας, ρεύματος του κινητήρα και έλεγχο φοράς.</p>	 A blue PCB with several yellow plastic connectors (headers) for motor control. It features a microUSB port and a microSD card slot. The board is populated with various electronic components.

<p style="text-align: center;"><u>LCD Shield</u></p> <p>Επιτρέπει σε μια πλακέτα Arduino να απεικονίσει πληροφορίες σε μια οθόνη LCD. Επίσης διαθέτει πέντε βοηθητικά κουμπιά για ρύθμιση αντίθεσης ή πλοήγηση στο μενού.</p>	
<p style="text-align: center;"><u>Relay Shield</u></p> <p>Παρέχει έναν εύκολο τρόπο για τον έλεγχο της υψηλής τάσης χάρη σε 4 ανεξάρτητα κανάλια που είναι εξοπλισμένα με Relay. Η μέγιστη ισχύς μεταγωγής είναι 1250VA AC ή DC 150W.</p>	
<p style="text-align: center;"><u>XBEE Shield</u></p> <p>Επιτρέπει σε μια πλακέτα Arduino να επικοινωνεί ασύρματα, χρησιμοποιώντας Zigbee. Η μονάδα μπορεί να επικοινωνεί μέχρι και 30,5 μέτρα σε εσωτερικούς χώρους ή σε εξωτερικούς χώρους 91,5 μέτρα.</p>	
<p style="text-align: center;"><u>Wireless SD Shield</u></p> <p>Επιτρέπει σε μια πλακέτα Arduino να επικοινωνεί ασύρματα με μια ασύρματη μονάδα. Η μονάδα μπορεί να επικοινωνήσει έως και 30 μέτρα σε εσωτερικούς χώρους ή 90 μέτρα σε εξωτερικούς. Η μονάδα περιλαμβάνει μια θύρα υποδοχής SD.</p>	
<p style="text-align: center;"><u>Wireless Proto Shield</u></p> <p>Επιτρέπει σε μια πλακέτα Arduino να επικοινωνεί ασύρματα με μια ασύρματη μονάδα. Η μονάδα μπορεί να επικοινωνήσει έως και 30 μέτρα σε εσωτερικούς χώρους ή 90 μέτρα σε εξωτερικούς. Η μονάδα δεν περιλαμβάνει μια θύρα υποδοχής SD.</p>	

Πίνακας 4: Arduino Shields

1.2.5 Τεχνικά χαρακτηριστικά arduino redboard

Στο σύστημα που αναπτύχθηκε χρησιμοποιήθηκε η έκδοση Arduino Redboard. Το Arduino Redboard είναι παρόμοιο με το Arduino Uno, το οποίο είναι και το πιο διαδεδομένο Arduino. Ο λόγος που χρησιμοποιήθηκε αυτή η έκδοση είναι η δυνατότητα προσθαφαίρεσης των αισθητήρων και των συσκευών δράσης αλλά και ο αριθμός αναλογικών και ψηφιακών εισόδων-εξόδων. Εκτός αυτού η συγκεκριμένη έκδοση έχει πολύ χαμηλό κόστος και χαμηλή κατανάλωση ρεύματος. Οι ομοιότητες του Redboard και του Uno είναι περισσότερες από τις διαφορές τους. Αυτό φαίνεται στον παρακάτω πίνακα όπου φαίνονται τα τεχνικά χαρακτηριστικά τους.

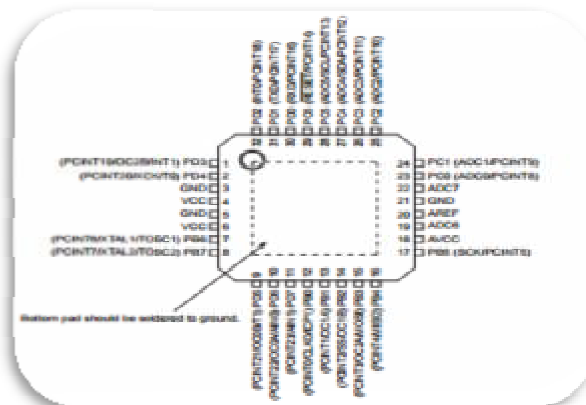
Χαρακτηριστικά	Redboard	Arduino Uno
Πάνω όψη		
Κάτω όψη		
Διαστάσεις	2.7 x 2.1" (68.58 x 53.34mm)	
Υποδοχή USB	Mini-B	Type B
USB-to-Serial Chip	FTDI FT232RL	ATmega16U4 w/ custom firmware
Οδηγοί (Windows)	FTDI VCP Drivers	Arduino USB Driver
Συμβατότητα Windows	8 (32 & 64-bit), 7 (32 & 64-bit), Vista (32 & 64-bit), XP (32 & 64-bit), 2000, 98 *	8 (32 & 64-bit), 7 (32 & 64-bit), Vista (32 & 64-bit), XP (32 & 64-bit)
Συμβατότητα Mac	OS X, OS 9, OS 8	OS X
Συμβατότητα Linux	Ναι	
Μικροελεγκτής	ATmega328	
MCU PTH ή SMD	SMD	PTH
Διακύμανση τάσης εισόδου	7-12V	
Όρια τάσης εισόδου	6-20V	

Τάση λειτουργίας	5V	
Ψηφιακές είσοδοι-έξοδοι	14(6 PWM έξοδοι)	
Ισχύς συνεχόμενου ρεύματος ανά ακροδέκτη	40mA	
Ισχύς συνεχόμενου ρεύματος για ακροδέκτη τάσης 3.3V	50mA	
Αναλογικές είσοδοι	6	
Επιλογή στο IDE	Arduino Uno	
Λιανική Τιμή	\$24.95	\$29.95
Μνήμη flash	32KB (ATMEGA328)	
Μνήμη SRAM	2KB (ATMEGA328)	
Μνήμη EEPROM	1KB (ATMEGA328)	
Ταχύτητα ρολογιού	16MHz	

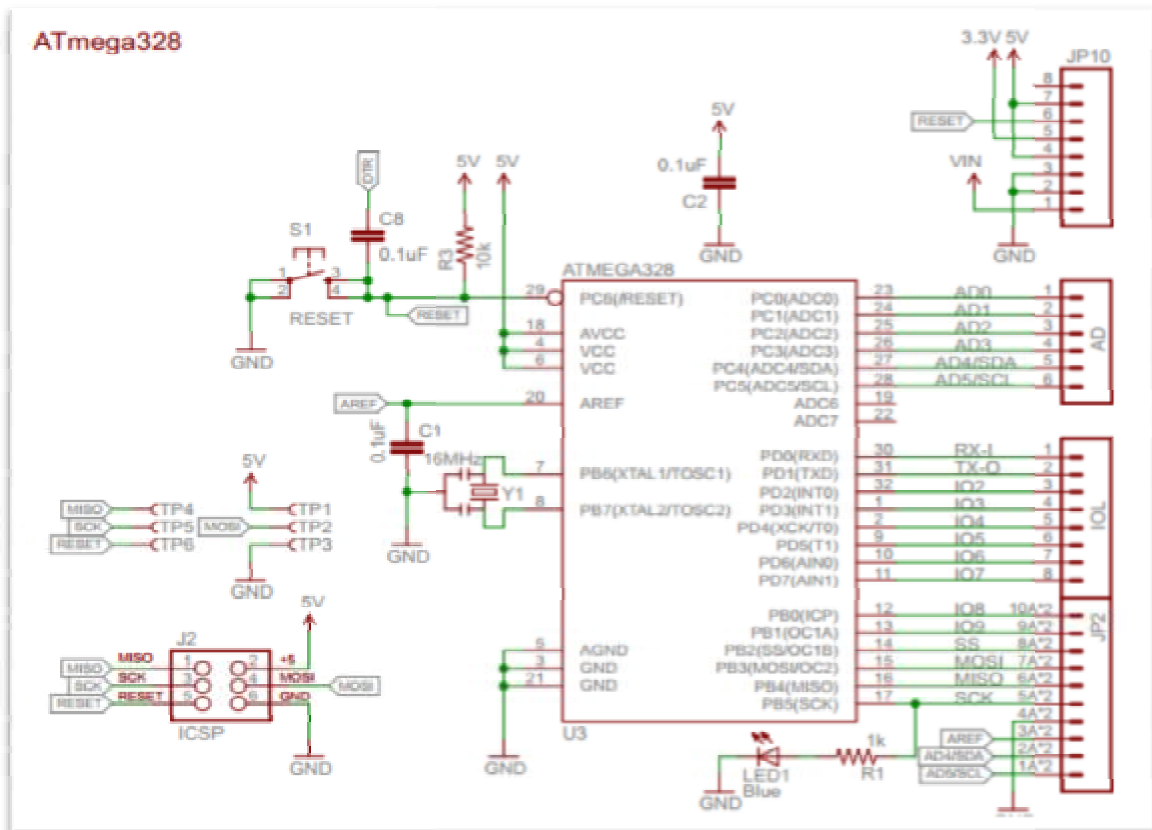
Πίνακας 5:Τεχνικά χαρακτηριστικά Arduino

1.2.5.1 Μικροελεγκτής ATmega328

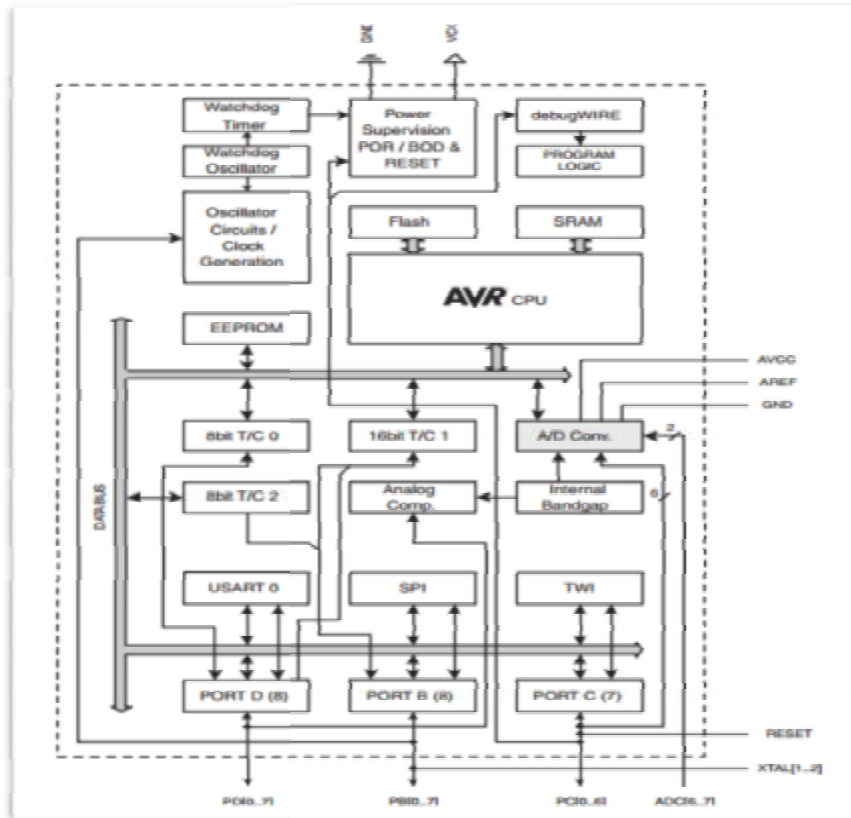
Το υψηλής απόδοσης Atmel 8-bit AVR που βασίζεται στην αρχιτεκτονική RISC συνδυάζει μνήμη flash 32KB ISP με δυνατότητες ανάγνωσης-εγγραφής, 1KB EEPROM, 2KB SRAM, 23 γραμμές εισόδων/εξόδων γενικής χρήσης, 32 γενικού σκοπού καταχωρητές, τρεις ευέλικτους μετρητές με καταστάσεις σύγκρισης, εσωτερικές και εξωτερικές διακοπές, σειριακά προγραμματιζόμενη USART, 2 καλωδίων σειριακή διασύνδεση, σειριακή θύρα SPI, 6 καναλιών 10-bit A / D μετατροπέα (8-κανάλια TQFP και QFN / MLF πακέτα) προγραμματιζόμενο χρονόμετρο φυλάκων με εσωτερικό ταλαντωτή, και πέντε επιλογές λειτουργιών λογισμικού εξοικονόμησης ενέργειας. Η συσκευή λειτουργεί μεταξύ 01.08 - 05.05 βολτ.Εκτελώντας οδηγίες σε ένα μόνο κύκλο ρολογιού, η συσκευή επιτυγχάνει διακίνηση η οποία πλησιάζει το 1 MIPS ανά MHz, εξισορροπώντας την κατανάλωση ενέργειας και την ταχύτητα επεξεργασίας.



Εικόνα 18:Κάτοψη μικροελεγκτή ATmega328



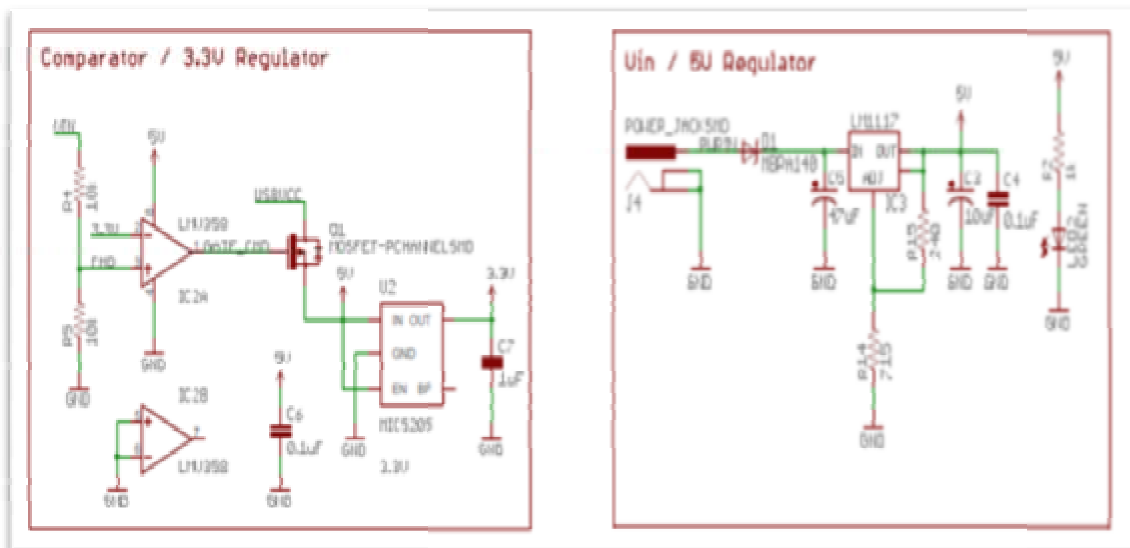
Εικόνα 19: Διάγραμμα ακροδεκτών μικροελεγκτή ATmega 328



Εικόνα20:Σχηματικό διάγραμμα ATmega 328

1.2.5.2 Τροφοδοσία arduino redboard

Η αναπτυξιακή πλατφόρμα Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από εξωτερική τροφοδοσία, είτε απευθείας από τον υπολογιστή μέσω σύνδεσης USB. Η επιλογή της πηγής γίνεται αυτόματα. Η εξωτερική τροφοδοσία μπορεί να είναι είτε μία μπαταρία, είτε ένας μετασχηματιστής. Αν επιλεγθεί για τροφοδοσία της πλακέτας μία μπαταρία, αυτή πρέπει να συνδεθεί στις υποδοχές Vin και GND όπου τοποθετούνται ο θετικός και ο αρνητικός πόλος αντίστοιχα. Σε περίπτωση τροφοδοσίας από μετασχηματιστή, υπάρχει βύσμα των 2.1mm που βρίσκεται στην κάτω γωνία για απευθείας σύνδεση. Η πλακέτα μπορεί να λειτουργήσει με εξωτερική πηγή από 6V έως 20V. Όμως για την αποφυγή προβλημάτων, η εξωτερική τροφοδοσία θα πρέπει να είναι από 7V ως 12V. Αυτό διότι αν η τροφοδοσία είναι λιγότερη από 7 V οι ακίδες εξόδου δεν θα καταφέρουν να εξαγάουν τάση 5V, ενώ αν τροφοδοτηθεί με περισσότερα από 12V θα υπερθερμανθεί ο σταθεροποιητής τάσης της πλακέτας και υπάρχει περίπτωση να καταστραφεί. Συμπερασματικά μία ιδανική τάση είναι τα 9V. Η εικόνα 21 παρουσιάζει τις εισόδους και εξόδους τροφοδοσίας του Arduino Redboard.



Εικόνα 22:Κυκλώματα σταθεροποίησης τάσης του Redboard.

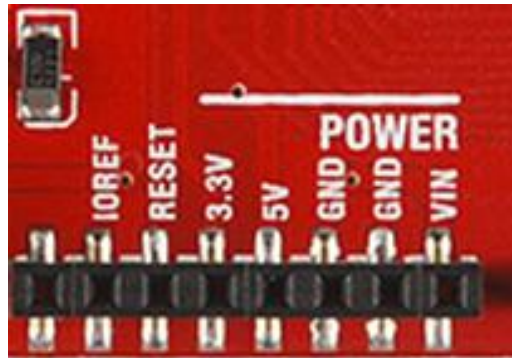
Οι ακροδέκτες τροφοδοσίας είναι οι ακόλουθοι:

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

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

5V: Ακροδέκτης σταθεροποιημένης τάσης 5V. Χρησιμοποιείται από διάφορα ηλεκτρονικά στοιχεία και το μικροελεγκτή. 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.

GND: Ακροδέκτες γείωσης



Εικόνα 21: Ακροδέκτες τροφοδοσίας Redboard

1.2.5.3 Προστασία από υπέρταση

Η πλακέτα του Arduino διαθέτει μία ασφάλεια η οποία προκαλεί επαναφορά (reset), με σκοπό την προστασία της θύρας USB του ηλεκτρονικού υπολογιστή, από βραχυκυκλώματα και υψηλές τιμές τάσεις. Η ασφάλεια αυτή παρέχει ένα επιπλέον επίπεδο προστασίας, καθώς οι περισσότεροι υπολογιστές παρέχουν αυτή την προστασία εσωτερικά. Η σύνδεση θα διακοπεί αυτόματα, σε περίπτωση κατά την οποία περισσότερα από 500mA εφαρμοστούν στην θύρα USB, έως ότου το βραχυκύκλωμα ή η υπέρταση να σταματήσουν.

1.2.5.4 Μνήμη Ardino RedBoard

Ο μικροεπεξεργαστής ATmega328 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.

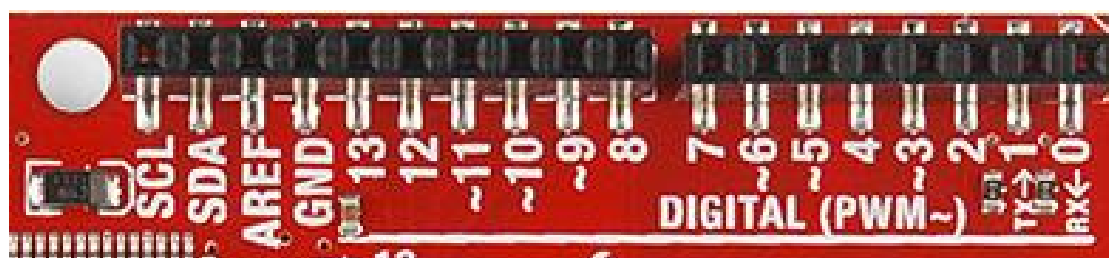
32KB μνήμης Flash: 2 KB χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware είναι αναγκαίο για την εγκατάσταση προγραμμάτων στο μικροελεγκτή μέσω της θύρας USB. Τα υπόλοιπα

Ο λόγος που το λογισμικό Arduino χρησιμοποιεί αυτή τη δυνατότητα είναι για να επιτρέψει στο χρήστη να φορτώσει κώδικα, μόνο με το πάτημα του κουμπιού upload στο περιβάλλον ανάπτυξης. Αυτό δηλώνει ότι ο bootloader μπορεί να έχει μικρότερο χρόνο εκκίνησης, όπως μπορεί και η γραμμή ελέγχου ροής υλικού να είναι συντονισμένη με την έναρξη της μεταφόρτωσης (upload). Έχει προγραμματιστεί ώστε να φιλτράρει και να αγνοεί ακατάλληλα δεδομένα, δηλαδή οτιδήποτε εκτός από την αποστολή νέου κώδικα και παρακολουθεί τα πρώτα byte δεδομένων που του αποστέλλονται και στη συνέχεια ανοίγει τη σύνδεση.

Το Arduino έχει ένα ίχνος χαλκού (trace), το οποίο μπορεί να κοπεί για να απενεργοποιηθεί η λειτουργία αυτόματης επαναφοράς. Για να απενεργοποιήσουμε τη λειτουργία αυτή, οι ακροδέκτες του ίχνους χαλκού πρέπει να είναι συγκολλημένοι. Τέλος για την ακύρωση της επιλογής αυτόματης επαναφοράς, αρκεί να συνδεθεί μεταξύ της τροφοδοσίας 5V και της γραμμής επαναφοράς (reset pin) μία αντίσταση 110Ω ή μεταξύ της γραμμής γείωσης και της γραμμής επαναφοράς (reset pin) έναν πυκνωτή 10μF.

1.2.5.7 Ακροδέκτες Arduino RedBoard

Στην πάνω πλευρά του Arduino βρίσκονται 14 ψηφιακοί ακροδέκτες, αριθμημένα από 0 ως 13. Κάθε ένας από τους 14 ψηφιακούς ακροδέκτες στο Arduino Uno μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος. Όλοι αυτοί οι ακροδέκτες λειτουργούν στα 5V με την δυνατότητα να παρέχουν ή να δέχονται ένταση της τάξεως των 40 mA. Σε κάθε ακροδέκτη υπάρχει εσωτερικά μία pull-up αντίσταση των 20-50 kΩ. Ως ψηφιακή έξοδος, ένα από αυτούς του ακροδέκτες μπορεί να τεθεί από το πρόγραμμα σε κατάσταση HIGH ή LOW, οπότε το Arduino θα καταλάβει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο ακροδέκτη. Αν πάλι ρυθμιστεί ένα από αυτούς του ακροδέκτες ως ψηφιακή είσοδος μέσα από το πρόγραμμα, μπορεί με την κατάλληλη εντολή να διαβάσει την κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχει συνδεθεί σε αυτόν τον ακροδέκτη διοχετεύει ή όχι ρεύμα στον ακροδέκτη. Μερικά από αυτούς τους 14 ακροδέκτες, εκτός από ψηφιακές εισοδοί/έξοδοι έχουν και δεύτερη λειτουργία.



Εικόνα 24: Ακροδέκτες Redboard

Ακροδέκτες 0 και 1: Οι ακροδέκτες 0 και 1 λειτουργούν ως RX και TX της σειριακής θύρας όταν το πρόγραμμα ενεργοποιεί τη σειριακή θύρα. Έτσι, όταν το πρόγραμμα στέλνει δεδομένα στη σειριακή θύρα, αυτά προωθούνται και στη θύρα USB μέσω του 25 ελεγκτή Serial-Over-USB, αλλά και στον ακροδέκτη 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή. Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμα

ενεργοποιήσει το σειριακό interface, χάνει 2 ψηφιακές εισόδους/εξόδους η πλατφόρμα.

Ακροδέκτες 2 και 3: Λειτουργούν και ως εξωτερικές διακοπές (interrupt) (interrupt 0 και 1 αντίστοιχα). Ρυθμίζονται μέσα από το πρόγραμμα ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοί 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 ψευδο-αναλογικές έξοδοι με το σύστημα 8-bit PWM (Pulse Width Modulation).

Ακροδέκτες 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK): Αυτοί οι ακροδέκτες επιτρέπουν την επικοινωνία SPI, η οποία και παρέχεται από το hardware.

Ακροδέκτης 13: Σε αυτόν τον ακροδέκτη υπάρχει ένα ενσωματωμένο LED. Όταν ο ακροδέκτης έχει τιμή HIGH, LED το ανάβει.

AREF: Η θύρα αυτή προσφέρει την τάση αναφοράς και μπορεί να χρησιμοποιηθεί μαζί με την συνάρτηση analogReference().

SDA και SCL: Υποστηρίζουν το πρωτόκολλο I2C χρησιμοποιώντας βιβλιοθήκες της γλώσσας προγραμματισμού Wiring.

GND: Ακροδέκτης γείωσης

Αναλογικοί ακροδέκτες:

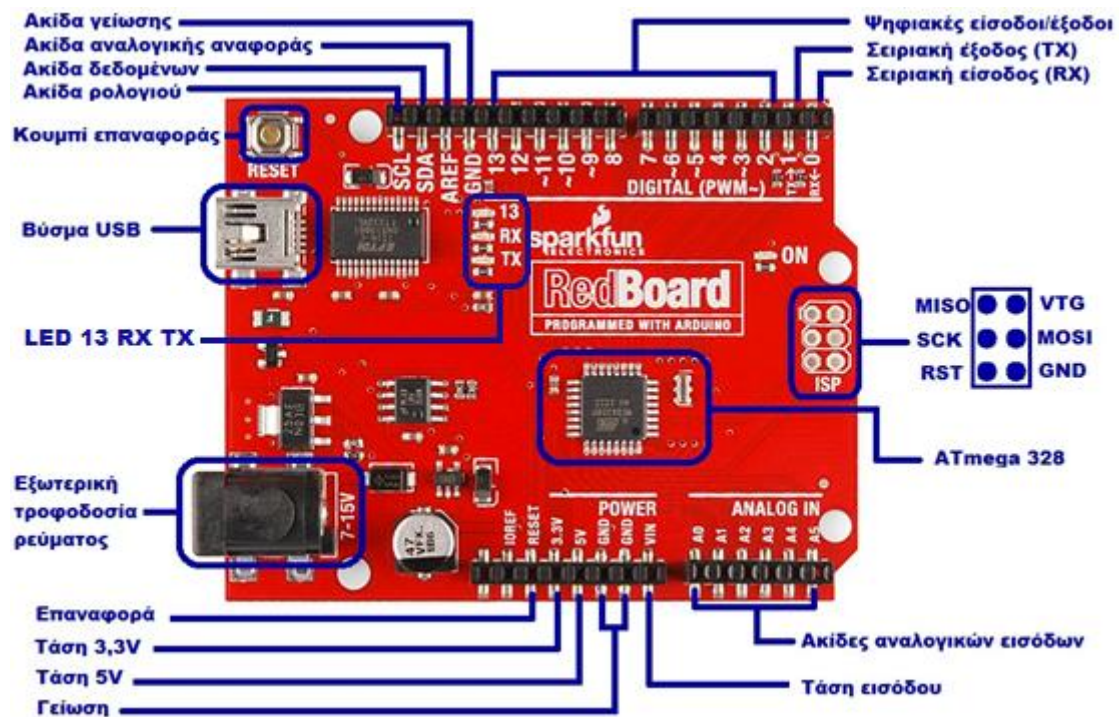
Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN όπως φαίνεται και στην εικόνα 25, υπάρχει μια ακόμη σειρά από 6 ακροδέκτες, αριθμημένους από το 0 ως το 5. Πρόκειται για τις 6 αναλογικές εισόδους του Redboard. Αυτοί μπορούν να διαβάσουν αναλογικές τιμές όπως η τάση μίας μπαταρίας και να τις μετατρέψουν σε μία τιμή από το 0 έως το 1023 (10 bit). Η μέτρηση της τάσης γίνεται από προκαθορισμένα επίπεδα από 0 έως 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. Έτσι, 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.



Εικόνα 25: Αναλογικοί ακροδέκτες.

1.2.5.8 Ενσωματωμένα κουμπιά και LED

Πάνω στην πλακέτα του Arduino υπάρχει ένας διακόπτης micro-switch και 4 μικροσκοπικά LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι προφανής. Τα δύο LED με τις σημάσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει (αντίστοιχα) δεδομένα μέσω USB. Τα LED αυτά ελέγχονται από τον ελεγκτή Serial-over-USB και συνεπώς δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται αποκλειστικά μέσω των ψηφιακών pin 0 και 1.



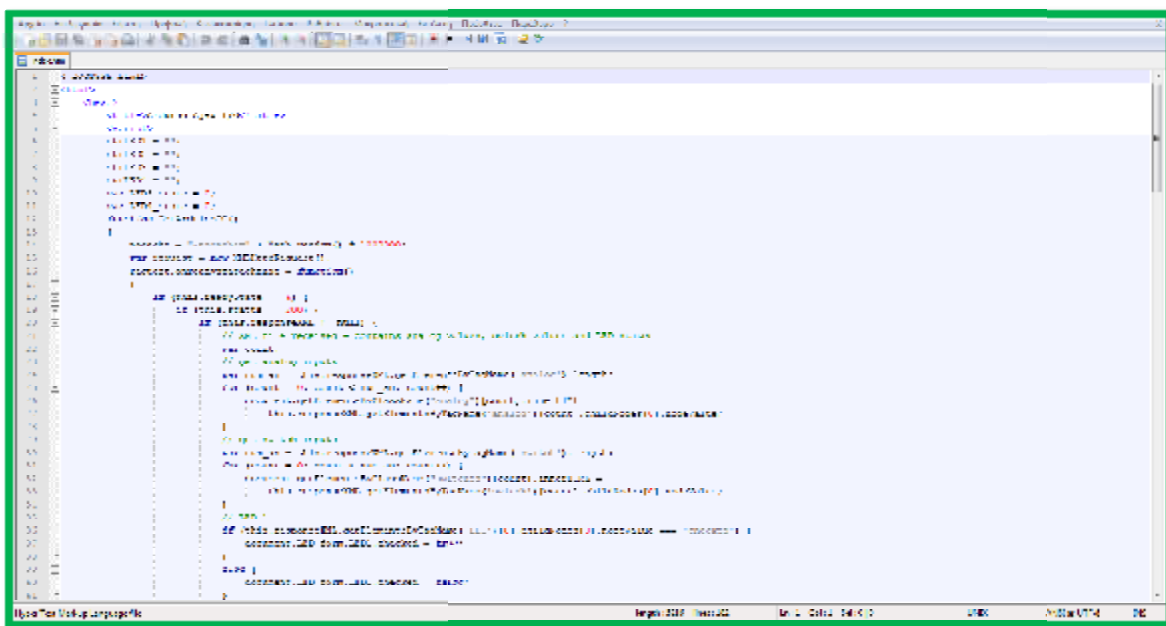
Εικόνα 26: Μέρη του Redboard

1.3 Εργαλεία ανάπτυξης λογισμικού

Χρησιμοποιήθηκαν εργαλεία ανοιχτού λογισμικού. Για τη συγγραφή του κώδικα που αφορά τον προγραμματισμό και την κατασκευή του ιστοχώρου χρησιμοποιήθηκε το notepad++ και συγχρόνως έγινε χρήση του τοπικού διακομιστή Xampp. Όσον αφορά τον προγραμματισμό του arduino χρησιμοποιήθηκε το λογισμικό arduino IDE για την φόρτωση του κώδικα. Η δημιουργία των εικόνων έγινε με το πρόγραμμα Photoshop και η σχεδίαση κυκλωμάτων σε εικόνες πραγματοποιήθηκε με το πρόγραμμα Fritzing. Στη συνέχεια της ενότητας αναλύονται τα παραπάνω εργαλεία.

1.3.1 Notepad++

Πρόκειται για ανοικτού λογισμικού πρόγραμμα το οποίο χρησιμοποιείται για συγγραφή κώδικα. Πρόκειται για τον πιο γνωστό επεξεργαστή κειμένου (text editor). Είναι απαραίτητο εργαλείο συγγραφής των προγραμματιστών, καθώς αποτελεί ένα ελαφρύ πρόγραμμα το οποίο όμως έχει πάρα πολλές δυνατότητες. Το περιβάλλον εργασίας είναι εξαιρετικά απλό και λειτουργικό. Στο επάνω μέρος του υπάρχει μία εργαλειοθήκη με τις κυριότερες λειτουργίες. Όλο το υπόλοιπο μέρος του καταλαμβάνεται από τον χώρο συγγραφής του κώδικα. Γύρω από τον χώρο αυτόν, μπορούν να εμφανιστούν διάφορα πλαίσια τα οποία παρουσιάζουν διάφορες πληροφορίες ανάλογα με τα plugins από τα οποία προέρχονται. Το σημαντικό του χαρακτηριστικό είναι το λεγόμενο syntax highlighting, το οποίο χρωματίζει τον κώδικα ανάλογα με την δομή του και διευκολύνει την ανάπτυξη του. Για κάθε γλώσσα προγραμματισμού υπάρχει διαφορετικό χρωματικό στυλ. Το πρόγραμμα έχει έτοιμα στυλ για πολλές γλώσσες και δίνει την δυνατότητα να τα ρυθμίσεις σύμφωνα με τις ανάγκες του προγραμματιστή. Στην παρούσα εργασία χρησιμοποιήθηκε για τη συγγραφή όλων των script σε php, javascript, html, css, ajax και mysql.



Εικόνα 27: Περιβάλλον Notepad++

1.3.2 Xampp

Το ΧΑΜΡΡ είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, περιέχει εξυπηρετητή ιστοσελίδων http Apache, βάση ιστοσελίδων MySQL και ένα διερμηνέα για σενάρια γραμμένα σε γλώσσες προγραμματισμού PHP και Perl. Χρησιμοποιείται για να τρέχουν σε αυτόν τόσο στατικές, όσο και δυναμικές ιστοσελίδες. Χρησιμοποιήθηκε για την ανάπτυξη του ιστοχώρου με την απεικόνιση που πρόσφερε όντας πρόγραμμα που προσομοιώνει τη φιλοξενία ενός ιστοχώρου σε διακομιστή δικτύου (web server).



Εικόνα 28:Λογότυπα υποστηριζόμενα από το ΧΑΜΡΡ

1.3.3 Fritzing

Το Fritzing, είναι ένα εύκολο στη χρήση λογισμικό ηλεκτρονικού σχεδιασμού ανοικτού κώδικα που έχει σχεδιαστεί για να βοηθήσει στη μετάβαση από ένα πρωτότυπο σε ένα τελικό σχέδιο. Απευθύνεται σε χρήστες που θέλουν να παράγουν έγγραφα ή να τα χωρίσουν σε κατηγορίες και να πειραματιστούν. Έχει τη δυνατότητα να δημιουργίας σχηματικών PCB αλλά και παράγωγα αρχεία PCB. Το Fritzing λοιπόν, είναι μια εφαρμογή που μπορεί να συνδυαστεί με τη τεχνολογία του μέλλοντος, το Arduino ώστε να δημιουργηθούν τ' απαραίτητα κυκλώματα που τις περισσότερες φορές είναι σχετικά απλά ώστε να ελέγχουμε ηλεκτρικές συσκευές από ένα pc.









Εικόνα 29:Περιβάλλον Fritzing

1.3.4 Arduino IDE.

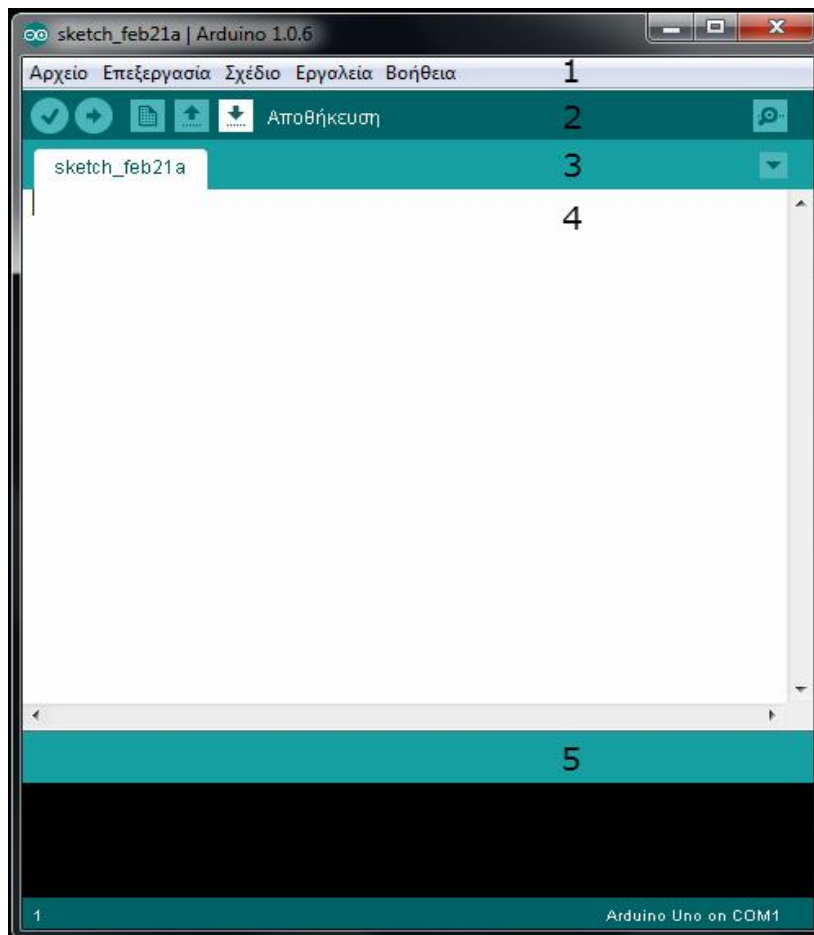
Το Arduino Uno έχει τη δυνατότητα επικοινωνίας με υπολογιστή, άλλες ίδιες πλακέτες ή άλλους μικροελεγκτές. Παρέχεται η δυνατότητα σειριακής επικοινωνίας μέσω USB θύρας και τους FTDI οδηγούς (drivers). Το λογισμικό που παρέχεται μας επιτρέπει να στέλνουμε εντολές ή δεδομένα γενικά από και προς την πλακέτα και να φορτώσουμε το πρόγραμμα στον bootloader, το οποίο θα εκτελεί αυτόματα η πλακέτα. Για τη διαχείριση του Arduino από τον υπολογιστή χρησιμοποιείται το Arduino IDE.

1.3.4.1 Ολοκληρωμένο περιβάλλον ανάπτυξης.

Το Arduino IDE είναι ένα περιβάλλον ανάπτυξης το οποίο περιέχει μια περιοχή επεξεργασίας κειμένου για τη συγγραφή κώδικα, μια περιοχή μηνυμάτων, ένα μενού, μια γραμμή εργαλείων με κουμπιά για κοινές λειτουργίες, καθώς και μια σειρά από μενού. Συνδέεται με το υλικό Arduino για τη φόρτωση προγραμμάτων και για να επικοινωνούν μεταξύ τους. Ένα ολοκληρωμένο πρόγραμμα συνήθως ονομάζεται sketch. Αυτό το sketch είναι γραμμένο με το πρόγραμμα επεξεργασίας κειμένου. Έχει δυνατότητες για την αντιγραφή/επικόλληση και για την αναζήτηση/αντικατάσταση κειμένου. Η κονσόλα απεικονίζει την έξοδο του κειμένου από το περιβάλλον Arduino συμπεριλαμβάνοντας πλήρη μηνύματα λάθους και άλλες πληροφορίες. Τα κουμπιά της γραμμής εργαλείων επιτρέπουν τον έλεγχο και το ανέβασμα των προγραμμάτων, τη δημιουργία νέου sketch, το άνοιγμα και την αποθήκευση των sketch και άνοιγμα της σειριακής οθόνης. Στον επόμενο πίνακα παρουσιάζονται τα εργαλεία του περιβάλλοντος ανάπτυξης, υπό μορφή κουμπιών, ενώ στην επόμενη εικόνα φαίνεται το ολοκληρωμένο περιβάλλον ανάπτυξης του arduino.

	Verify / Compile	Ελέγχει για συντακτικά λάθη στον κώδικα.
	Upload	Μεταγλωττίζει τον κώδικα και τον φορτώνει στο Arduino.
	New	Δημιουργεί ένα νέο sketch.
	Open	Παραθέτει ένα μενού με όλα τα sketch. Κάνοντας κλικ σε ένα από αυτά, θα ανοίξει αυτόματα.
	Save	Αποθηκεύει ένα sketch.
	Serial Monitor	Ανοίγει την σειριακή οθόνη.

Πίνακας 6: Εργαλεία του περιβάλλοντος ανάπτυξης IDE.



1. Μενού

2. Εργαλειοθήκη

3. Καρτέλες (Tabs)

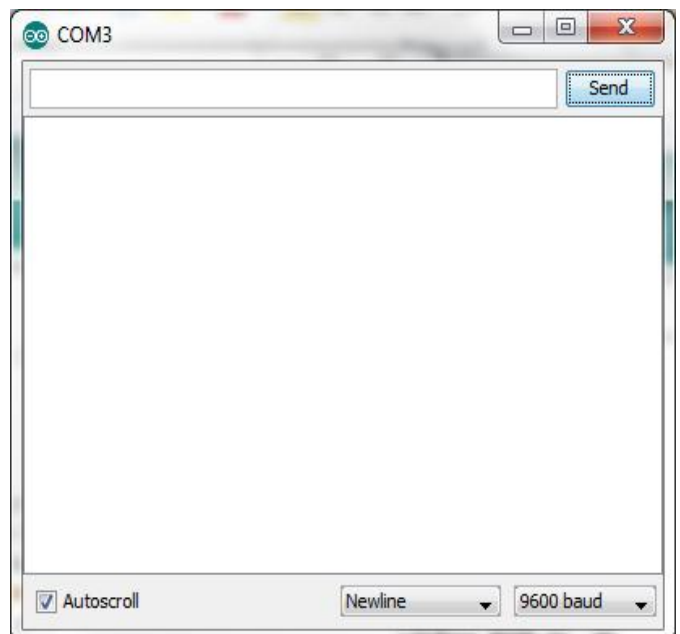
4. Επεξεργαστής
κειμένου

5. Κονσόλα μηνυμάτων

Εικόνα 30: Ολοκληρωμένο περιβάλλον ανάπτυξης arduino IDE

1.3.4.2 Σειριακή οθόνη.

Εμφανίζει τα σειριακά δεδομένα που αποστέλλονται από την πλακέτα Arduino. Πιο συγκεκριμένα, η αποστολή δεδομένων στην πλακέτα γίνεται, εισάγοντας κείμενο και πατώντας το κουμπί send ή πατώντας το Enter. Επίσης, στο κάτω μέρος της σειριακής οθόνης, μπορεί να γίνει η επιλογή της κατάλληλης ταχύτητας (baud) από την λίστα που εμφανίζεται ανάλογα με την τιμή που θα επιλεγεί στο προγραμματισμό του Arduino με το Serial.begin(). Παρακάτω παρουσιάζεται η σειριακή οθόνη (serial monitor).



Εικόνα 31 : Σειριακή οθόνη arduino IDE

Συνοπτικά το Arduino IDE παρέχει:

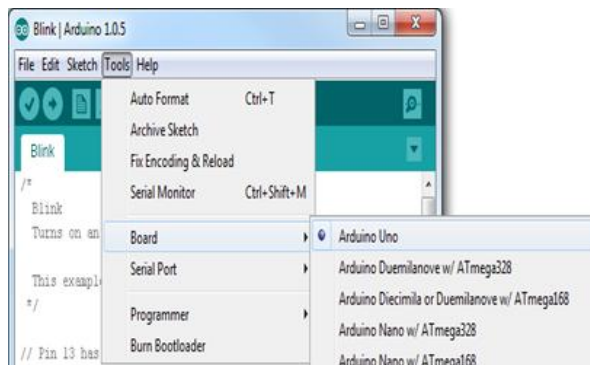
- Ø Ένα πρακτικό περιβάλλον για τη συγγραφή των προγραμμάτων, με συντακτική χρωματική σήμανση.
- Ø Μερικές έτοιμες βιβλιοθήκες για προέκταση της.
- Ø Τον compiler για τη μεταγλώττιση των sketch.
- Ø Μία σειριακή οθόνη (serial monitor) που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για την αποσφαλμάτωση των sketch.
- Ø Την επιλογή για ανέβασμα των μεταγλωττισμένων sketch στο Arduino.

1.3.4.3 Ανέβασμα σχεδίων.

Βήματα για ανέβασμα σχεδίων στο Arduino.

Πριν φορτωθεί το σχέδιο σε οποιοδήποτε Arduino πρέπει να γίνουν τα εξής:

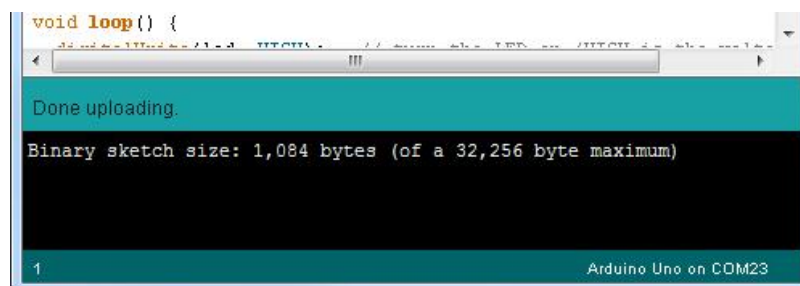
- Ø Έλεγχος για συντακτικά λάθη στον κώδικα με το εργαλείο Verify.
- Ø Επιλογή του τύπου πλακέτας.(Tools→Board)(εικόνα 32)
- Ø Επιλογή θύρας Usb που έχει συνδεθεί το Arduino(Tools→Serial Port)(εικόνα 33)
- Ø Ανέβασμα σχεδίου με το εργαλείο Upload.



Εικόνα 32: Επιλογή τύπου πλακέτας



Εικόνα 33: Επιλογή θύρας usb



Εικόνα 34: Επιτυχής ανέβασμα σχεδίου (κονσόλα μηνυμάτων)

1.3.4.4 Γλώσσα και δομή προγράμματος

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring μια παραλλαγή C/C++ 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 και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Λόγω της καταγωγής της από τη C, στη γλώσσα 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 που γράφονται στο IDE του Arduino ονομάζονται sketch και τους δίνεται αυτόματα όνομα βάση της ημερομηνίας που ξεκίνησαν. Τα αρχεία αυτά έχουν επέκταση INO. Τα σχόλια μιας γραμμής γράφονται μετά από διπλές καθέτους '//'. (//σχόλιο). Σχόλια περισσότερων γραμμών γράφονται εντός '/*' και '*/' . (/* σχόλιο */) Το IDE αναγνωρίζει τα σχόλια και τα χρωματίζει γκριζα, για να δείξει ότι δεν είναι ενεργά στο πρόγραμμα ή στο debugging.

Τα προγράμματα του Arduino διαιρούνται σε τρία μέρη:

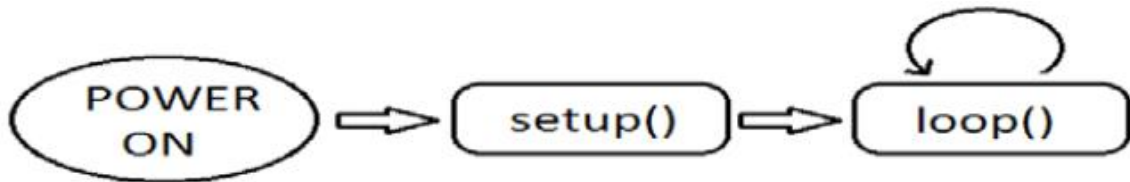
- Ø δομή (structure)
- Ø τιμές (values)
- Ø συναρτήσεις (functions)

Ένα τυπικό πρόγραμμα Arduino έχει την παρακάτω δομή:

```
void setup() {  
  //Κώδικας που εκτελείται μία φορά κατά την εκκίνηση της συσκευής  
}  
  
void loop() {  
  //Κώδικας που εκτελείται συνεχώς  
}
```

Εικόνα 35: Δομή προγράμματος

Υπάρχουν δυο ειδικές συναρτήσεις που είναι μέρος του κάθε sketch του Arduino οι οποίες είναι η setup() και η loop(). Η setup() καλείται μια φορά, όταν το sketch ξεκινά ή όποτε κάνει επαναφορά (reset) η πλατφόρμα Arduino. Κυρίως, σε αυτήν γίνονται οι αρχικοποιήσεις των μεταβλητών, η ρύθμιση της κατάστασης των ακίδων (pins) και η προετοιμασία των βιβλιοθηκών. Αντιθέτως, η συνάρτηση loop() καλείται ξανά και ξανά επιτρέποντας έτσι στο πρόγραμμα να ανταποκριθεί σε εξωτερικά ερεθίσματα.



Εικόνα 36: Διάγραμμα ροής λειτουργίας Wiring

Και οι δυο συναρτήσεις πρέπει να περιλαμβάνονται στο sketch, ακόμα και αν δεν περιέχουν κάτι και να είναι κενές. Μερικές από τις πιο σημαντικές δομές και λειτουργίες που μπορούν να αξιοποιηθούν ως εργαλεία κατά την συγγραφή ενός προγράμματος Arduino επεξηγούνται ανά μέρη στον πίνακα που ακολουθεί:

Δομή.	
setup(), loop()	
Δομές ελέγχου ροής.	
if	Δομή ελέγχου μίας συνθήκης.
if..else	Δομή ελέγχου πολλαπλών συνθηκών.
for	Δομή επαναληπτικού ελέγχου συνθήκης.
do..while	Δομή επαναληπτικού ελέγχου συνθήκης.
while	Δομή επαναληπτικού ελέγχου συνθήκης.
switch case	Δομή ελέγχου περιπτώσεων.
break	Εντολή διακοπής μιας επαναληπτικής δομής.
continue	Εντολή παράλειψης της τρέχουσας επανάληψης.
return	Εντολή επιστροφής από μία συνάρτηση.
go to	Εντολή μετάβασης σε κάποιο σημείο του κώδικα.
Αριθμητικοί τελεστές.	
=, +, -, *, /	Τελεστής εκχώρησης, πρόσθεσης, αφαίρεσης, διαίρεσης.
%	Υπόλοιπο ακέραια διαίρεσης.
Τελεστές σύγκρισης	
<, >, <=, >=	Μικρότερο, Μεγαλύτερο, Μικρότερο ή ίσο, Μεγαλύτερο ή ίσο.
==, !=	Ισότητα, Ανισότητα.
Λογικοί τελεστές.	
&&, , !	Λογική σύζευξη, διάζευξη, άρνηση.
Διαδικοί τελεστές.	
&, 	Διαδική σύζευξη – διάζευξη.
<<, >>	Διαδική αριστερή – δεξιά ολίσθηση.
^	Διαδική αποκλειστική διάζευξη.
~	Διαδική άρνηση.
Τελεστές αύξησης και μείωσης.	
++, --	Αύξηση – μείωση κατά μία ακέραια μονάδα.

Σύνθετοι τελεστές.	
+=, -=, *=, /=, %=	σύνθετοι αριθμητικοί τελεστές.
&=, =, ^=, ~=, <=>=	Σύνθετοι δυαδικοί τελεστές.
Τελεστές δεικτών.	
*, &	Τελεστής απόκτησης περιεχομένου – διεύθυνσης.
Τύποι δεδομένων.	
boolean	Λογική δυαδική τιμή.
char	Προσημασμένος χαρακτήρας 8 ψηφίων.
(unsigned)char	Μη προσημασμένος χαρακτήρας 8 ψηφίων.
byte	Μη προσημασμένος χαρακτήρας 8 ψηφίων.
int	Προσημασμένος ακέραιος 16 ψηφίων.
(unsigned)int	Μη προσημασμένος ακέραιος 16 ψηφίων.
word	Μη προσημασμένος ακέραιος 16 ψηφίων.
long	Προσημασμένος ακέραιος 32 ψηφίων.
(unsigned)long	Μη προσημασμένος ακέραιος 32 ψηφίων.
float, double	Αριθμός κινητής υποδιαστολής απλής ακρίβειας.
string	Αντικείμενο αλφαριθμητικού με χρήσιμες μεθόδους.
Σταθερές.	
HIGH – LOW	Τιμή υψηλής - χαμηλής στάθμης για μία επαφή εισόδου.
INPUT – OUTPUT	Χρησιμοποιείται για τον ορισμό μίας επαφής ως είσοδο – έξοδο.
True-false	Λογικό επίπεδο αλήθειας - ψεύδους σε μία συνθήκη.
A0,...,A5	Συμβολοσταθερές για τις αναλογικές επαφές εισόδου.
Συναρτήσεις εισόδου και εξόδου.	
pinMode()	Ορίζει μια επαφή ως είσοδο ή έξοδο.
Συναρτήσεις αναλογικής εισόδου και εξόδου.	
analogReference()	Ορίζει την τάση αναλογικής αναφοράς.
analogRead()	Διαβάζει από μία αναλογική επαφή εισόδου.
analogWrite()	Γράφει PWM σήματα σε μία επαφή εξόδου.
Συναρτήσεις ψηφιακής εισόδου και εξόδου.	
digitalWrite()	Γράφει σε μία ψηφιακή επαφή εξόδου.
digitalRead()	Διαβάζει από μία ψηφιακή επαφή εισόδου.
Συναρτήσεις Χρόνου.	
millis(), micros()	Διάρκεια εκτέλεσης του προγράμματος σε ms, μs.
delay	Παύση προγράμματος - η διάρκεια δίδεται σε ms.
delayMicroseconds()	Παύση προγράμματος - η διάρκεια δίδεται σε μs.
Προηγμένες συναρτήσεις εισόδου και εξόδου.	
tone(),	Παράγει ένα τετραγωνικό σήμα ορισμένης συχνότητας.
noTone()	Διακόπτει την παραγωγή τετραγωνικών σημάτων.
shiftOut()	Ολισθαίνει τα ψηφία μιας τιμής σε μία επαφή εξόδου.
pulseIn()	Επιστρέφει την διάρκεια σε μs ενός παλμού HIGH ή LOW.

Μαθηματικές και τριγωνομετρικές συναρτήσεις.	
max(), min()	Βρίσκει τον μεγαλύτερο - μικρότερο ανάμεσα σε δύο αριθμούς.
abs()	Επιστρέφει την απόλυτη τιμή ενός αριθμού..
constrain()	Ελέγχει για υπερχείλιση ή υποχείλιση ορίων.
map()	Πραγματοποιεί γραμμικό μετασχηματισμό ορίων.
pow()	Επιστρέφει το αποτέλεσμα μίας δύναμης.
sqrt()	Επιστρέφει την ρίζα ενός αριθμού.
sin(), cos(), tan()	Υπολογίζει το ημίτονο, συνημίτονο , εφαπτομένη ενός αριθμού.
Συναρτήσεις ψευδοτυχαίων αριθμών.	
random()	Δίδεται ένας νέος αριθμός από την γεννήτρια.
randomSeed()	Θέτει τον σπόρο της γεννήτριας παραγωγής.
Συναρτήσεις επεξεργασίας δυαδικών αριθμών.	
lowByte(),highByte()	Επιστρέφει το δεξιότερο - αριστερότερο byte μίας μεταβλητής.
bitRead(),bitWrite()	Διαβάζει - γράφει ένα συγκεκριμένο ψηφίο μίας μεταβλητής.
bitSet(),bitClear()	Γράφει την τιμή 1 - 0 σε κάποιο ψηφίο μίας μεταβλητής.
bit()	Υπολογίζει μία συγκεκριμένη δύναμη με βάση το 2.
Συναρτήσεις χρήσης ρουτινών εξυπηρέτησης διακοπών.	
attachInterrupt()	Ενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής.
detachInterrupt()	Απενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής.
Συναρτήσεις ενεργοποίησης και απενεργοποίησης διακοπών.	
interrupts()	Ενεργοποιεί τα σήματα διακοπής.
noInterrupts()	Απενεργοποιεί τα σήματα διακοπής.
Υποστήριξη σειριακής επικοινωνίας.	
Serial	Αντικείμενο σειριακής επικοινωνίας με χρήσιμες μεθόδους.
Συναρτήσεις μετατροπής τύπων.	
char(), byte(), int(), word(), long(), float(), double()	

Πίνακας 7: Δομές και λειτουργίες προγραμματισμού

1.3.4.5 Βιβλιοθήκες

Η χρήση βιβλιοθηκών προσφέρουν περισσότερο λειτουργικότητα σε συνεργασία με το υλικό και τον χειρισμό των δεδομένων. Για να χρησιμοποιηθεί μια βιβλιοθήκη σε ένα sketch, μπορεί να επιλεγεί από το μενού *Sketch*→ *Import Library*. Αυτό θα εισάγει μια ή περισσότερες βιβλιοθήκες `#include` δηλώσεις στην κορυφή του sketch. Επειδή οι βιβλιοθήκες φορτώνονται στην πλακέτα με το sketch, αυξάνουν το μέγεθος του χώρου που καταλαμβάνεται. Εάν ένα sketch δεν χρειάζεται πλέον μια βιβλιοθήκη, απλά μπορούμε να την διαγράψουμε από την κορυφή του κώδικα. Για την εγκατάσταση των βιβλιοθηκών που δεν υπάρχουν ήδη στο λογισμικό, μπορεί να δημιουργηθεί ένας κατάλογος με την ονομασία *libraries* (βιβλιοθήκες), μέσα στον κατάλογο του *sketchbook*. Στην συνέχεια αποσυμπίεζουμε τη βιβλιοθήκη εκεί. Παρακάτω ακολουθούν μερικές από τις βιβλιοθήκες που υποστηρίζονται από το Arduino.

Καθιερωμένες βιβλιοθήκες	
EEPROM	Ανάγνωση και εγγραφή στην EEPROM.
Ethernet	Σύνδεση με το Internet χρησιμοποιώντας το Ethernet Shield.
WiFi	Σύνδεση με το Internet χρησιμοποιώντας το WiFi Shield.
GSM	Σύνδεση σε ένα δίκτυο GSM/GRPS δικτύου με GSM Shield.
LiquidCrystal	Για τον έλεγχο των οθονών υγρών κρυστάλλων.
SD	Για την ανάγνωση και τη γραφή κάρτες SD.
Servo	Για τον έλεγχο κινητήρων τύπου Servo.
SPI	Επικοινωνία με συσκευές που χρησιμοποιούν το δίαυλο SPI.
SoftwareSerial	Για σειριακή επικοινωνία για οποιαδήποτε ψηφιακή ακίδα.
NewSoftSerial	Βελτιωμένη έκδοση της βιβλιοθήκης SoftwareSerial.
Stepper	Για τον έλεγχο βηματικών κινητήρων.
TFT	Για προβολή κειμένου, εικόνων και σχημάτων στην οθόνη TFT.
Firmata	Για την επικοινωνία με τις εφαρμογές στον υπολογιστή χρησιμοποιώντας ένα τυποποιημένο σειριακό πρωτόκολλο.
Wire	Για την αποστολή και λήψη δεδομένων μέσω δικτύου των συσκευών ή των αισθητήρων. (TWI/I2C).
Επικοινωνίας (δικτύωση και πρωτόκολλα):	
Messenger	Επεξεργασία κειμένου με βάση τα μηνύματα από τον υπολογιστή.
OneWire	Συσκευές ελέγχου που χρησιμοποιούν το πρωτόκολλο one Wire.
PS2Keyboard	Διαβάζει χαρακτήρες από ένα πληκτρολόγιο PS2.
SSerial2mobile	Αποστολή μηνυμάτων κειμένου ή e-mail χρησιμοποιώντας ένα κινητό τηλέφωνο.
Webduino	Επεκτάσιμη βιβλιοθήκη web server (για χρήση με το Arduino Ethernet Shield).
X10	Αποστολή σημάτων X10 μέσω γραμμών εναλλασσόμενου ρεύματος.
Xbee	Για την επικοινωνία με XBees σε λειτουργία API.
Ανίχνευσης	
Capacitive Sensing	Δύο ή περισσότερες ακίδες σε αισθητήρες πυκνωτή.
Debounce	Για την ανάγνωση θορυβώδων ψηφιακών εισόδων.
Εμφάνιση και LED	
GLCD	Γραφικές ρουτίνες για LCD με βάση την KS0108 ή ισοδύναμο chipset.
LedControl	Για τον έλεγχο των LED ή επτά τμημάτων οθόνες με MAX7221 ή MAX7219.
LedControl	Μια εναλλακτική λύση στη βιβλιοθήκη Matrix για την οδήγηση με πολλαπλούς LED.

LedDisplay	Τον έλεγχο της HCMS-29xx οθόνη LED.
Συχνότητα παραγωγής ήχου	
Tone	Αναπαράγει κύματα ήχου συχνότητας στο παρασκήνιο σε κάθε καρφίτσα του μικροελεγκτή.
Χρονοδιάγραμμα	
DateTime	Παρακολουθήση της τρέχουσας ημερομηνίας και ώρας.
MsTime2	Χρησιμοποιεί το χρονόμετρο διακοπής 2 για να ενεργοποιήσει μια δράση κάθε χιλιοστά του δευτερολέπτου N.

Πίνακας 8: Βιβλιοθήκες arduino IDE

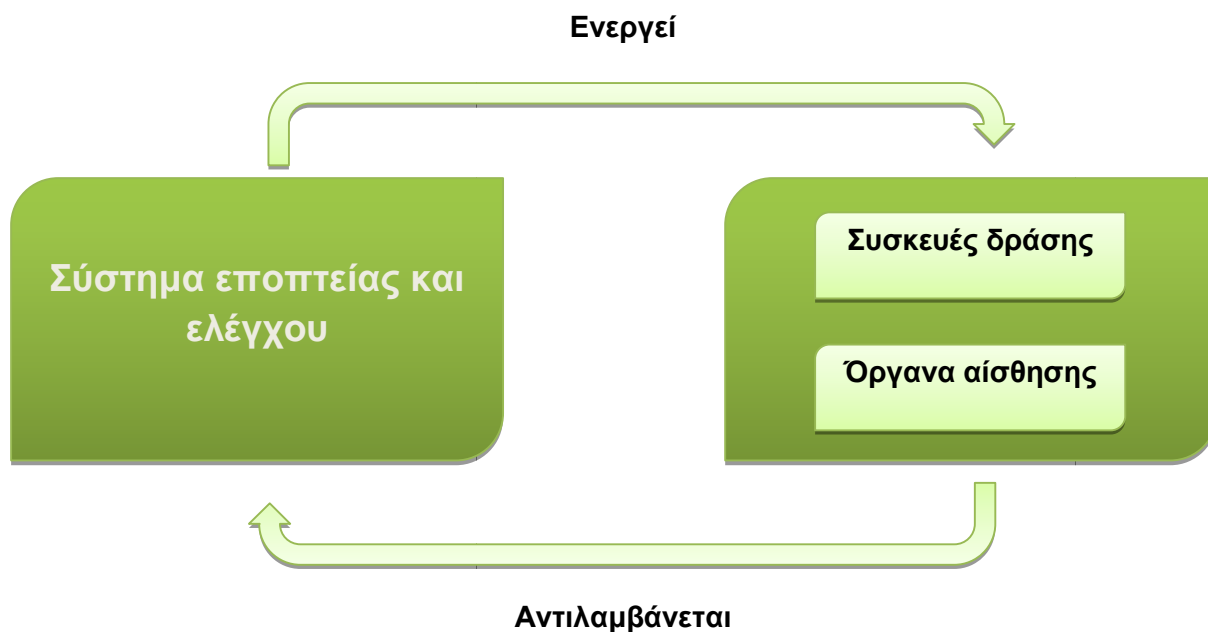
Κεφάλαιο 2: Υλικό μέρος συστήματος

Το σύστημα εποπτείας και ελέγχου ενός θερμοκηπίου πάνω από το διαδίκτυο, είναι βασισμένο στην αναπτυξιακή πλατφόρμα Arduino. Όπως έχει αναφερθεί η αρχιτεκτονική Arduino, μαζί με τον κατάλληλο προγραμματισμό και την σύνδεση διαφόρων εξαρτημάτων, όπως αισθητήρων και συσκευών, επιτρέπει την υλοποίηση του προαναφερθέντος συστήματος. Αυτό το κεφάλαιο ασχολείται με το υλικό μέρος του συστήματος το οποίο είναι βασισμένο στις παραπάνω αρχές. Το υλικό (hardware) μέρος του συστήματος αποτελείται από 5 επιμέρους τμήματα:

- 1 • Μικροελεγκτής
- 2 • Υλικό υλοποίησης επικοινωνίας με το διαδίκτυο
- 3 • Περιβαλλοντικοί αναλογικοί-ψηφιακοί αισθητήρες
- 4 • Ψηφιακά ρυθμιζόμενες συσκευές δράσης
- 5 • Κατασκευή μακέτας θερμοκηπίου εκπαιδευτικού σκοπού

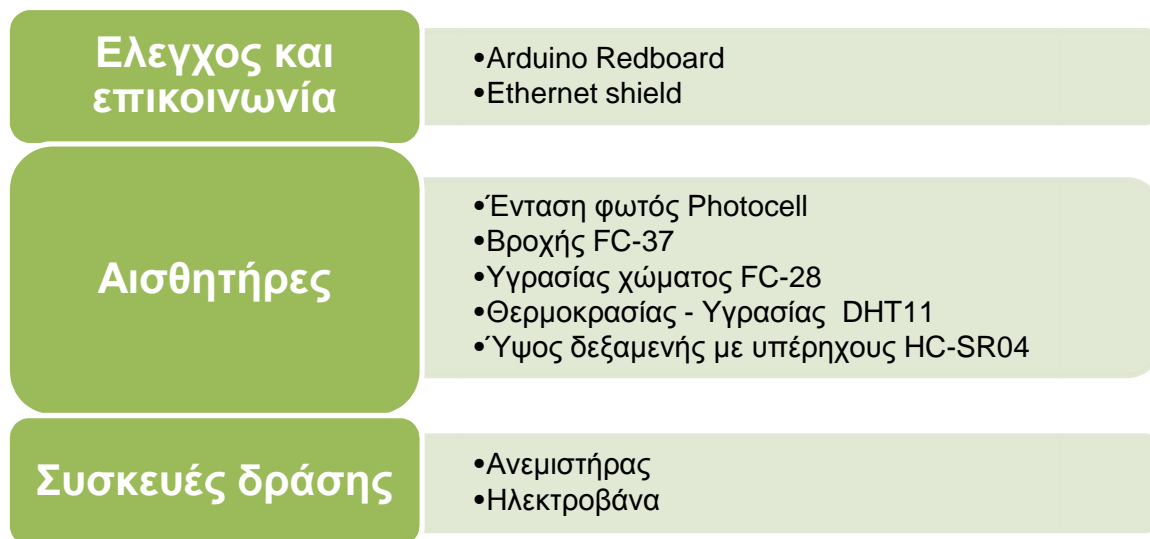
Εικόνα 37: Τμήματα του συστήματος.

Συνοπτικά υπάρχει ένα περιβάλλον (θερμοκήπιο) και το σύστημα εποπτείας και ελέγχου, το οποίο αποτελείται από τον μικροελεγκτή και την πλακέτα επικοινωνίας με το διαδίκτυο. Το σύστημα σε συνεργασία με τα όργανα αίσθησης, δηλαδή τους περιβαλλοντικούς αναλογικούς-ψηφιακούς αισθητήρες αντιλαμβάνεται το ότι συμβαίνει και εν συνεχεία, με τον κατάλληλο προγραμματισμό λαμβάνει τις αποφάσεις για να θέσει σε λειτουργία ή όχι τις ψηφιακά ρυθμιζόμενες συσκευές δράσης.



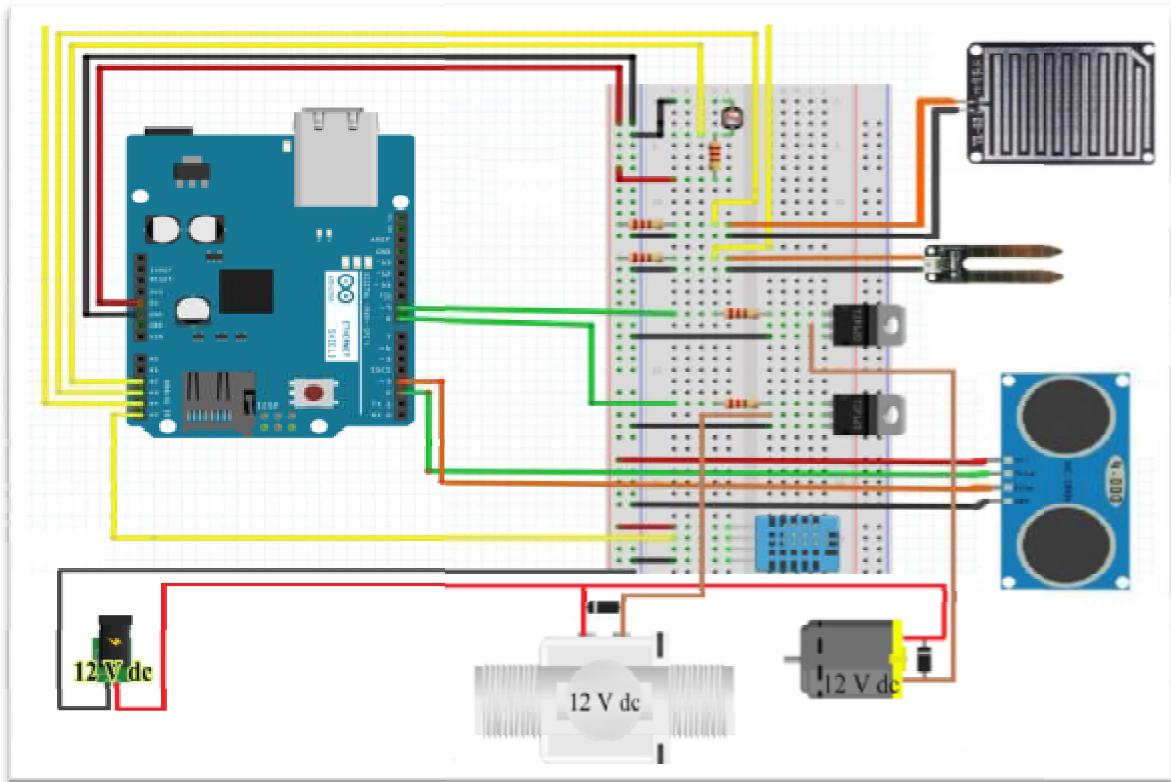
Εικόνα 38: Γενική μορφή συστήματος.

Το κύκλωμα το οποίο κατασκευάστηκε για τη δημιουργία του συστήματος εποπτείας και ελέγχου ενός θερμοκηπίου αποτελείται από τα εξής:



Εικόνα 39: Βασικό υλικό συστήματος.

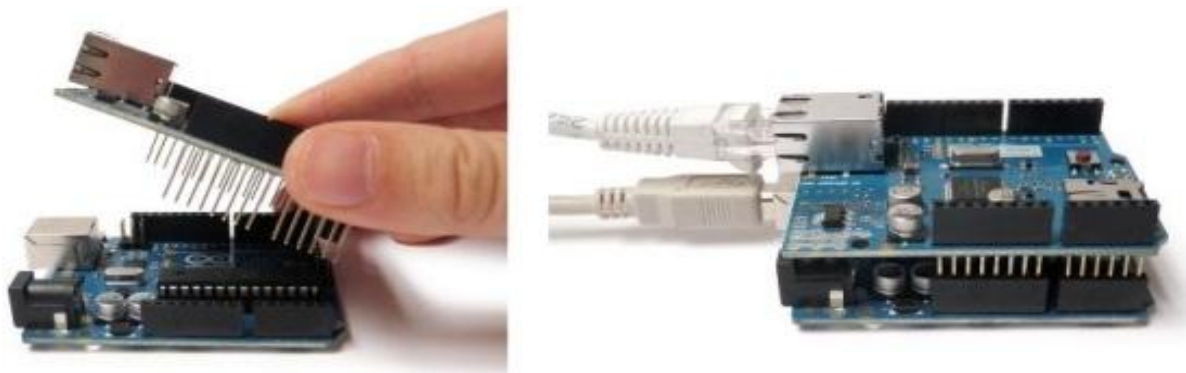
Εκτός από τα βασικά εξαρτήματα, για την υλοποίηση του κυκλώματος χρειάστηκαν ηλεκτρονικά εξαρτήματα τα οποία είναι αντιστάσεις, δίοδοι, τρανζίστορ, καλώδια και ένα breadboard. Στις επόμενες ενότητες αναλύονται λεπτομερειακά οι επιμέρους συνδέσεις αυτών. Στο παρακάτω σχέδιο φαίνονται οι συνδέσεις των εξαρτημάτων του τελικού κυκλώματος του συστήματος.



Εικόνα 40: Κύκλωμα συστήματος.

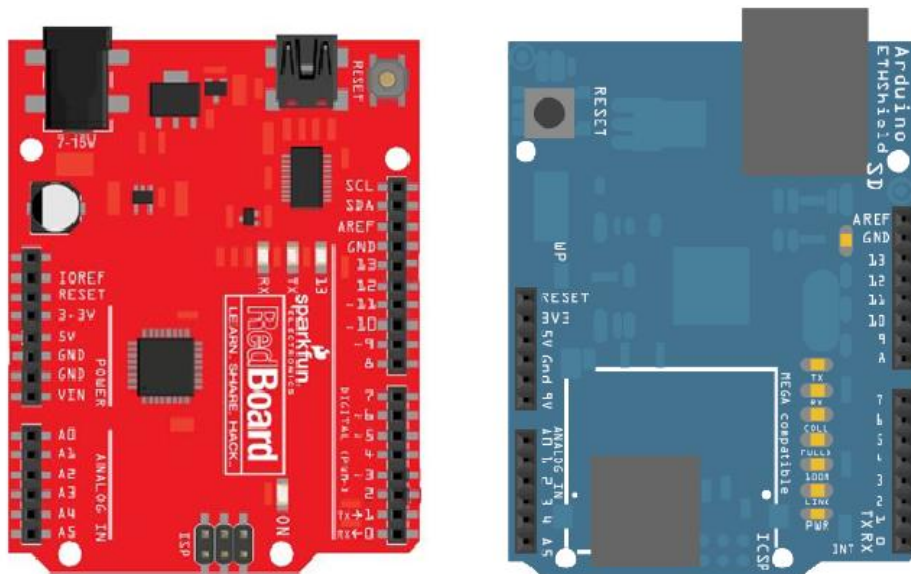
2.1 Επικοινωνία με το διαδίκτυο

Το Arduino redboard σε συνδυασμό με τους αισθητήρες και τις συσκευές δράσης, τα οποία αποτελούν το υλικό μέρος του συστήματος, επικοινωνούν με το διαδίκτυο μέσω της Arduino ethernet shield, η οποία όπως αναφέρθηκε παραπάνω πρόκειται για μία τυποποιημένη πλακέτα που προσθέτει νέες ιδιότητες στο σύστημά μας. Οι ακροδέκτες των δύο πλακετών που συνεργάζονται έχουν ίδια τοποθέτηση, έτσι η μονάδα συνδέεται με την πλακέτα Arduino χρησιμοποιώντας μακριούς συρμάτινους ακροδέκτες, που εκτείνονται διαμέσου αυτής και εισάγονται στις αντίστοιχες υποδοχές του arduino redboard. Αυτό κρατά τη διάταξη στιβαρή και επιτρέπει στη μονάδα να συνδεθεί πάνω από το Arduino.



Εικόνα 41: Τοποθέτηση Ethernet shield.

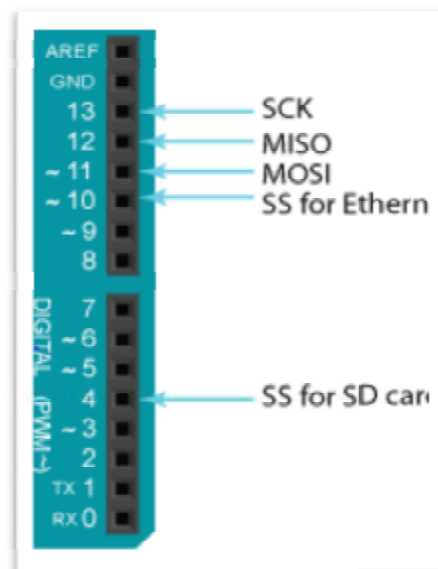
Συνδέεται στο δίκτυο με ένα καλώδιο RJ45. Η τροφοδοσία που απαιτείται από το Arduino ethernet shield είναι 5V dc και απορροφάει ρεύμα 0,7 mA τα οποία παρέχονται από την πλακέτα του Arduino redboard, με αποτέλεσμα να μην απαιτείται επιπλέον τροφοδοσία.



Εικόνα 42: *Arduino redboard – ethernet shield.*

Επίσης υπάρχει μία έκδοση (με ενσωματωμένο μετασχηματιστή γραμμής και Power over Ethernet, PoE) της συγκεκριμένης πλακέτας η οποία επιτρέπει την τροφοδοσία του συστήματος μέσω του καλωδίου δικτύου, η οποία όμως δεν επιλέχθηκε για το σύστημα λόγω μεγαλύτερου κόστους.

Υπάρχει μια υποδοχή κάρτας microSD, η οποία αποθηκεύει αρχεία για την διακίνηση τους μέσω του δικτύου. Είναι συμβατή με το Arduino Uno και Mega (χρησιμοποιώντας τη βιβλιοθήκη Ethernet).Ο ενσωματωμένος αναγνώστης καρτών microSD είναι προσβάσιμος μέσω της βιβλιοθήκης SD.Για επικοινωνία με την SD χρησιμοποιείται η ακίδα 4 ως SS. Το Arduino επικοινωνεί με την Ethernet shield χρησιμοποιώντας το δίαυλο SPI ο οποίος δεσμεύει τις ψηφιακές ακίδες 11, 12, και 13. Η ακίδα 10 χρησιμοποιείται ως SS για επικοινωνία του Ethernet. Τέλος εκτός από τους ακροδέκτες που υπάρχουν στο επάνω μέρος της πλακέτας, υπάρχει και ένα πλήθος φωτεινών ενδείξεων οι οποίες περιγράφονται στον παρακάτω πίνακα :



Εικόνα 43: *Επικοινωνία Ethernet–microSD κάρτα.*

Ενημερωτικές φωτεινές ενδείξεις Arduino ethernet shield	
Ένδειξη	Λειτουργία
PWR	Υποδεικνύει ότι υπάρχει τροφοδοσία στην πλακέτα.
LINK	Παρουσία σύνδεσης δικτύου.
100M	Δείχνει την παρουσία του σύνδεσης με το δίκτυο 100 Mb/s.
FULLD	Υποδεικνύει ότι η σύνδεση δικτύου είναι full duplex.
COLL	Αναβοσβήνει όταν ανιχνεύονται συγκρούσεις (collision) δεδομένων.
RX	Αναβοσβήνει όταν η ασπίδα λαμβάνει δεδομένα.
TX	Αναβοσβήνει όταν η ασπίδα στέλνει δεδομένα.

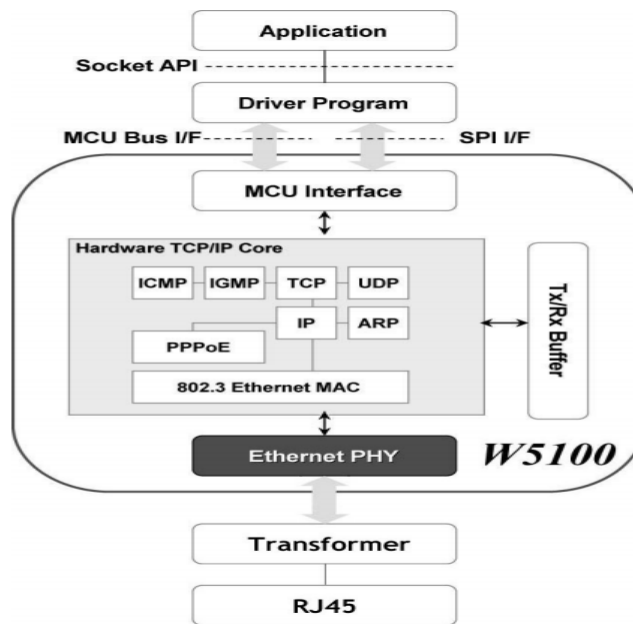
Πίνακας 9: Ενημερωτικές φωτεινές ενδείξεις Arduino ethernet shield.

Η Σειριακή Περιφερική Διεπαφή (SPI) είναι ένα σύγχρονο σειριακό πρωτόκολλο δεδομένων, που χρησιμοποιούνται από μικροελεγκτές για την επικοινωνία με μία ή περισσότερες περιφερειακές συσκευές, γρήγορα σε μικρές αποστάσεις. Μπορεί επίσης να χρησιμοποιηθεί για την επικοινωνία μεταξύ δύο μικροελεγκτών. Με σύνδεση SPI υπάρχει πάντοτε μία συσκευή master (συνήθως ένας μικροελεγκτής), ο οποίος ελέγχει τις περιφερειακές συσκευές. Συνήθως υπάρχουν τρεις γραμμές κοινές για όλες τις συσκευές και μία γραμμή ξεχωριστά για κάθε συσκευή.

Σύνδεση SPI.	
MISO (Master In Slave Out)	Η γραμμή slave για την αποστολή δεδομένων στη Master.
MOSI (Master Out Slave In)	Η γραμμή Master για την αποστολή δεδομένων στα περιφερειακά.
SCK (Serial Clock)	Οι παλμοί ρολογιού, οι οποίοι συγχρονίζουν τη διαβίβαση των δεδομένων που προκύπτουν από την Master συσκευή.
SS (Slave Select)	Η ακίδα σε κάθε συσκευή που η Master μπορεί να χρησιμοποιήσει για να ενεργοποιήσει και να απενεργοποιήσει συγκεκριμένες συσκευές.

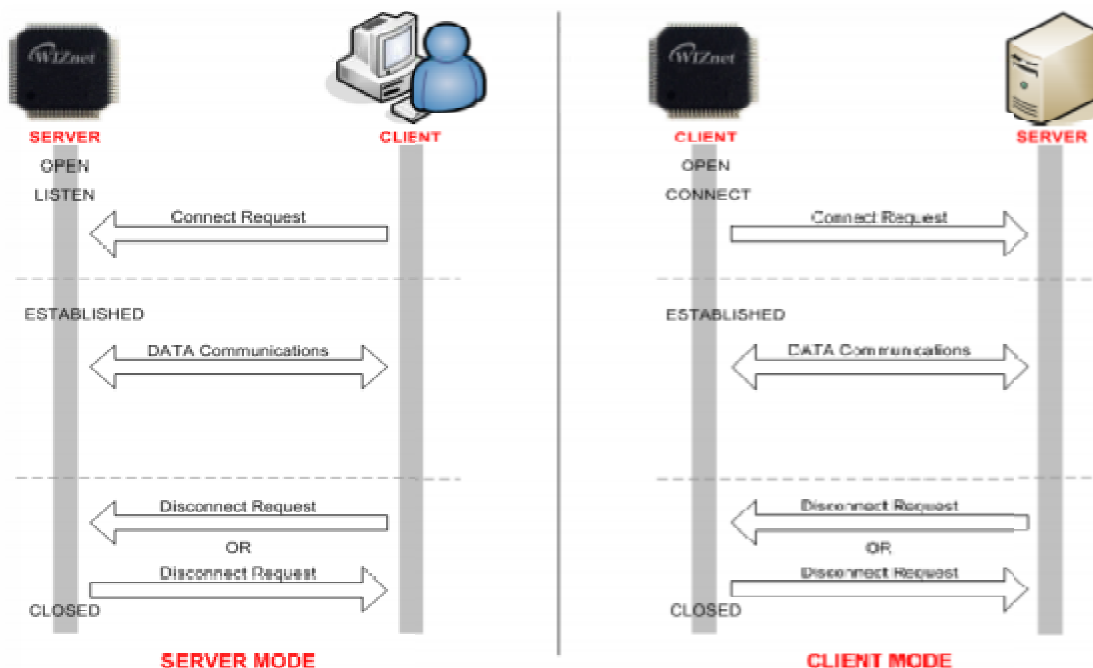
Πίνακας 10: Σύνδεση SPI.

Η arduino ethernet shield βασίζεται στο τσιπ της Wiznet W5100ethernet. Το W5100 είναι ένα πλήρως εξοπλισμένο, single-chip με ελεγκτή Ethernet 10/100, το οποίο έχει σχεδιαστεί για ενσωματωμένες εφαρμογές όπου απαιτούνται η ευκολία της ολοκλήρωσης, η σταθερότητα, η απόδοση, ο έλεγχος και το κόστος του συστήματος. Έχει σχεδιαστεί ώστε να διευκολύνει εφαρμογές σύνδεσης στο Internet χωρίς λειτουργικό σύστημα. Είναι συμβατό με πρωτόκολλο IEEE 802.3 10BASE-T και 802.3u 100BASE-TX.Για τη διαβίβαση δεδομένων περιλαμβάνεται ένας buffer 16Kbytes.Υποστηρίζει έως και τέσσερις ταυτόχρονες συνδέσεις. (εισερχόμενη ή εξερχόμενη ή συνδυασμός).



Εικόνα 44: Μπλοκ διάγραμμα W5100.

Η επικοινωνία δεδομένων είναι διαθέσιμη μέσω TCP, UDP, IP-Raw και MAC-Raw. Η TCP η οποία θα χρησιμοποιηθεί στο σύστημα είναι η σύνδεση βασισμένη σε μεθόδους επικοινωνίας, η οποία εκ των προτέρων δημιουργεί σύνδεση και παραδίδει τα δεδομένα μέσω αυτής χρησιμοποιώντας την διεύθυνση IP και το αριθμό της θύρας (port) των συστημάτων. Υπάρχουν δύο μέθοδοι για την πραγματοποίηση της σύνδεσης. Μία είναι η κατάσταση Server που είναι σε αναμονή για την αίτηση σύνδεσης. Η άλλη είναι η λειτουργία πελάτη (client) η οποία στέλνει αίτηση σύνδεσης σε ένα διακομιστή (server).



Εικόνα 45 : Μέθοδοι πραγματοποίησης σύνδεσης.

Παράδειγμα προγραμματισμού :

Σε αυτό το παράδειγμα, θα χρησιμοποιηθεί η Ethernet Shield και το Arduino με σκοπό να δημιουργηθεί ένας απλός web server. Χρησιμοποιώντας τη βιβλιοθήκη Ethernet, η συσκευή θα είναι σε θέση να απαντήσει σε αίτηση HTTP με την Ethernet shield. Μετά το άνοιγμα ενός browser και πλοήγηση προς τη διεύθυνση IP η οποία θα οριστεί μέσα από το πρόγραμμα, το Arduino θα ανταποκριθεί και θα εμφανιστούν σε γλώσσα html στο πρόγραμμα περιήγησης οι τιμές εισόδου από τις έξι αναλογικές ακίδες.

```
#include <SPI.h>
#include <Ethernet.h>
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 177); // Η διεύθυνση IP θα εξαρτάται από
το τοπικό δίκτυο.
EthernetServer server(80); //(θύρα 80 είναι η προεπιλεγμένη για το
HTTP):
void setup() {
  Serial.begin(9600);
  while (!Serial) { ; }
  Ethernet.begin(mac, ip); // Αρχή σύνδεσης Ethernet

  server.begin();
  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());
}
void loop() {
  EthernetClient client = server.available();
  if (client) {
    Serial.println("new client");
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);

        if (c == '\n' && currentLineIsBlank) {
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: close");
          client.println("Refresh: 5");
          client.println();
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");

          for (int analogChannel = 0; analogChannel < 6; analogChannel++) {
            int sensorReading = analogRead(analogChannel);
            client.print("analog input ");
            client.print(analogChannel);
```

```

        client.print(" is ");
        client.print(sensorReading);
        client.println("<br />");
    }
    client.println("</html>");
    break;
}
if (c == '\n') {
    currentLineIsBlank = true;
}
else if (c != '\r') {
    currentLineIsBlank = false;
}
}
}
delay(1);

client.stop();
Serial.println("client disconnected");
}
}

```

2.2 Περιβαλλοντικοί αισθητήρες

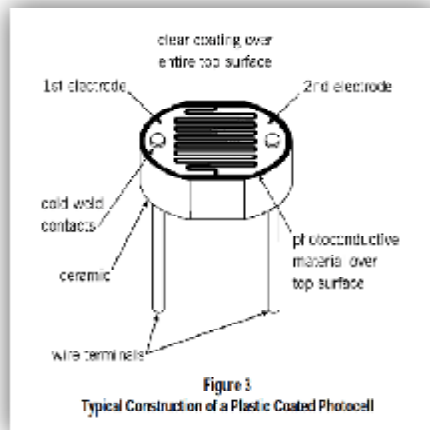
Το σύστημα που υλοποιήθηκε παρακολουθεί και καταγράφει τις περιβαλλοντικές συνθήκες σε ένα θερμοκήπιο. Για να επιτευχθεί αυτό χρησιμοποιήθηκαν αναλογικοί και ψηφιακοί αισθητήρες οι οποίοι συνδέθηκαν στις εισόδους του μικροελεγκτή. Η επιλογή των συγκεκριμένων αισθητήρων έγινε με βάση τα εξής κριτήρια:

1. Όλοι οι αισθητήρες λειτουργούν υπό τάση 5V. Το γεγονός αυτό συμβάλει στη χαμηλή κατανάλωση του συστήματος αλλά και στο ότι δεν είναι απαραίτητη η ύπαρξη κάποιας εξωτερικής πηγής τάσης (εκτός αυτής που τροφοδοτεί το Arduino) αφού το συγκεκριμένο μέγεθος τάσης μπορεί να το παρέχει το Arduino.
2. Οι αισθητήρες ακολουθούν το πρότυπο «plug n play» το οποίο σημαίνει ότι με το που συνδεθούν μπορούν να λειτουργήσουν. Πιο συγκεκριμένα δεν χρειάζεται κάποιος χρόνος απόκρισης ή προθέρμανσης, ώστε να λειτουργήσουν με ακρίβεια 100%.
3. Τέλος επιλέχθηκαν αισθητήρες με χαμηλό κόστος, των οποίων βέβαια τα τεχνικά χαρακτηριστικά τους καλύπτουν απόλυτα τις απαιτήσεις του συστήματος.

Στις επόμενες ενότητες αναλύονται τα επιμέρους εξαρτήματα που χρησιμοποιήθηκαν, τα τεχνικά τους χαρακτηριστικά, ο τρόπος με τον οποίο συνδέθηκαν αλλά και παραδείγματα προγραμματισμού του μικροελεγκτή, ώστε να λαμβάνει τις μετρήσεις από αυτά.

2.2.1 Αισθητήρας έντασης φωτός Photocell

Ο αισθητήρας MiniPhotocell χρησιμοποιήθηκε για την ανίχνευση της έντασης του φωτός στο περιβάλλον του θερμοκηπίου. Ο αισθητήρας αυτός λειτουργεί όπως μία μεταβλητή αντίσταση η οποία μεταβάλλεται ανάλογα με την ποσότητα του φωτός που ανιχνεύει. Ο συγκεκριμένος αισθητήρας χρειάζεται μία αντίσταση να συνδεθεί σε σειρά με αυτόν. Η τιμή αυτής μπορεί να κυμαίνεται από 1kΩ έως και 10kΩ και η επιλογή της γίνεται ανάλογα με τη φωτεινότητα του χώρου όπου τοποθετείται. Η αντίσταση των 1000Ω προορίζεται για εξωτερικούς χώρους όπου υπάρχει μεγάλη ποσότητα φωτός ενώ η αντίσταση των 10kΩ προορίζεται για την ανίχνευση φωτεινών εντάσεων σε εσωτερικούς χώρους.



Εικόνα 46 : Κατασκευή Photocell



Εικόνα 47 : Photocell.

Ο λόγος για τον οποίο επιλέχθηκε ο αισθητήρας αυτός είναι η άμεση ενεργοποίηση του κατά την εκκίνηση του συστήματος, η χαμηλή κατανάλωση ρεύματος (λιγότερο από 1mA κατά μέσω όρο ανεξαρτήτως τάσης εισόδου) αλλά και το πολύ χαμηλό κόστος. Το MiniPhotocell ανιχνεύει την ένταση του φωτός και αυξομειώνει ανάλογα την αντίσταση του. Με αυτόν τον τρόπο αλλάζει το ρεύμα που διατρέχει την αντίσταση του αισθητήρα αλλά και αυτή που είναι σε σειρά με αυτόν, με αποτέλεσμα να μεταβάλλεται και η τάση στα άκρα της αντίστασης. Έτσι λαμβάνεται η μέτρηση και μεταφέρεται σε οποιαδήποτε αναλογική είσοδο του μικροελεγκτή. Στον παρακάτω πίνακα παρουσιάζονται τα τεχνικά χαρακτηριστικά του αισθητήρα Photocell στους 25 °C :

Αντίσταση στο φως στα 10 Lux	8 – 20 KΩ	
Αντίσταση στο σκοτάδι στα 0 Lux	1MΩ	
Τιμή γάμα στα 100 – 10 Lux	0,7	
Μέγιστη τάση	150V	
Φασματική απόκριση αιχμής	540nm	
Εύρος θερμοκρασίας περιβάλλοντος	- 30 έως + 70 °C	

Πίνακας 11: Τεχνικά χαρακτηριστικά αισθητήρα Photocel.

Παράδειγμα προγραμματισμού

```
int photocellPin = 0;
int photocellReading;
void setup(void) {
  Serial.begin(9600);
}
void loop(void) {
  photocellReading = analogRead(photocellPin);
  Serial.print("Analog reading = ");
  Serial.print(photocellReading);      // the raw analog reading
  // We'll have a few thresholds, qualitatively determined
  if (photocellReading < 300) {
    Serial.println(" - Very bright ");
  } else if (photocellReading < 500) {
    Serial.println(" - Bright enough");
  } else if (photocellReading < 700) {
    Serial.println(" - Bright");
  } else {
    Serial.println(" - Dark");
  }
  delay(1000);
}
```

Αναλογική είσοδος	Ένδειξη
0-300	Very bright
300-500	Bright enough
500-700	Bright
700-1023	Dark

Πίνακας 12: Βαθμονόμηση εξόδου αισθητήρα φωτεινότητας.

2.2.2 Αισθητήρας βροχής

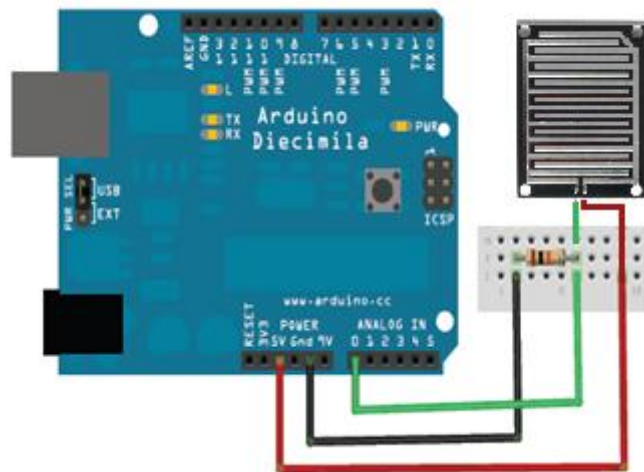
Ο αισθητήρας βροχόπτωσης είναι ένα εύκολο εργαλείο για την καταγραφή της βροχής. Μπορεί να χρησιμοποιηθεί σαν διακόπτης όταν η βροχή πέφτει πάνω στον αισθητήρα, αλλά και σαν μετρητής της έντασης των βροχοπτώσεων. Ο αισθητήρας βροχής ανιχνεύει το νερό το οποίο συμπληρώνει το κύκλωμα των κελιών που τυπώνετε στην πλακέτα του αισθητήρα. Τα κελιά του αισθητήρα λειτουργούν ως μεταβλητή αντίσταση που θα αλλάζει από 100KΩ όταν είναι βρεγμένο σε 2MΩ όταν στεγνώσει. Εν ολίγοις, όταν είναι βρεγμένο θα διέρχεται περισσότερο ρεύμα από όταν είναι στεγνό. Έχει δύο ακροδέκτες από τους οποίους ο ένας συνδέεται στην γείωση και ο άλλος με την αναλογική έξοδο του μικροελεγκτή και σε σειρά με την αντίσταση των 2.2KΩ που καταλήγει στα 5V του μικροελεγκτή. Η συνδεσμολογία απεικονίζεται στην εικόνα.



Εικόνα 52: Αισθητήρας βροχής FC-37.

Παράδειγμα προγραμματισμού:

```
int RaindropPin = 0;
int RaindropReading;
void setup(void) {
    Serial.begin(9600); }
void loop(void) {
    RaindropReading =
    analogRead(RaindropPin);
    Serial.print("Analog reading = ");
    Serial.print(RaindropReading);
    if (RaindropReading < 500) {
        Serial.println(" - Rain");
    } else if (RaindropReading < 800) {
        Serial.println(" - Raindrops");
    } else {
        Serial.println(" - No Rain");
    }
    delay(1000);
}
```



Εικόνα 53: Σύνδεση αισθητήρα βροχής.

Αναλογική είσοδος	Ένδειξη
0-500	Rain
500-800	Raindrops
800-1023	No rain

Πίνακας 13: Βαθμονόμηση εξόδου αισθητήρα βροχής.

2.2.3 Αισθητήρας υγρασίας χώματος

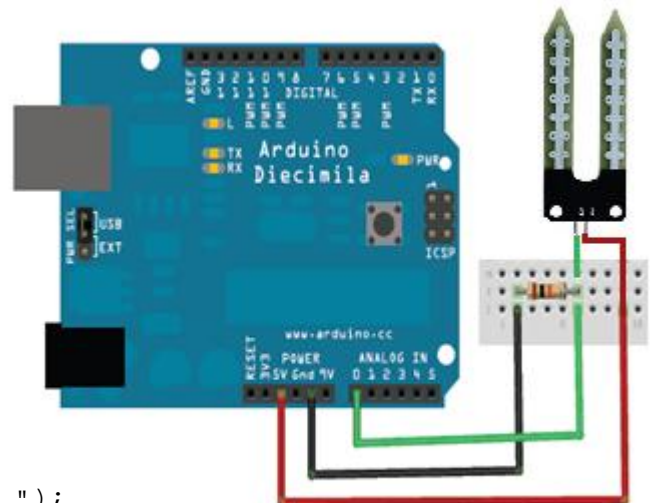
Για την μέτρηση της υγρασίας χώματος χρησιμοποιήθηκε ο αισθητήρας FC-28 ο οποίος απεικονίζεται στην διπλανή εικόνα. Μοιάζει με δύο καρφιά τα οποία τοποθετούμε στο έδαφος. Ανάλογα με την περιεκτικότητα του εδάφους σε υγρασία - νερό, εξάγεται μία τιμή από την οποία μπορούν να βγουν συμπεράσματα για την υγρασία του εδάφους. Το ένα άκρο συνδέεται στη γείωση, το άλλο με την αναλογική έξοδο του μικροελεγκτή και σε σειρά με την αντίσταση των 2.2KΩ που καταλήγει στα 5V του μικροελεγκτή.



Εικόνα 54: Αισθητήρας FC-28.

Παράδειγμα προγραμματισμού:

```
int moisturePin = 0;
int moistureReading;
void setup(void) {
    Serial.begin(9600); }
void loop(void) {
    moistureReading =
    analogRead(moisturePin);
    Serial.print("Analog reading = ");
    Serial.print(moistureReading);
    if (moistureReading < 400) {
        Serial.println(" - Very wet");
    } else if (moistureReading < 700) {
        Serial.println(" - Wet");
    } else if (moistureReading < 800) {
        Serial.println(" - Dry");
    } else {
        Serial.println(" - Very dry");
    }
    delay(1000);
}
```



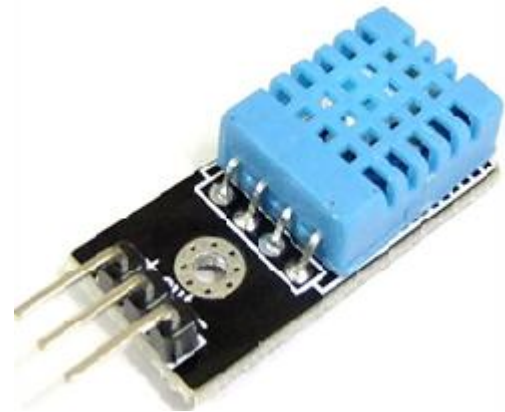
Εικόνα 55: Συνδεση
αισθητήρα βροχής.

Αναλογική είσοδος	Ένδειξη
0-400	Very wet
400-700	Wet
700-800	Dry
800-1023	Very dry

Πίνακας 14: Βαθμονόμηση εξόδου
αισθητήρα υγρασίας χώματος.

2.2.4 Αισθητήρας θερμοκρασίας – υγρασίας

Ο αισθητήρας DHT11 πρόκειται για έναν σύνθετο αισθητήρα ο οποίος έχει τη δυνατότητα μέτρησης θερμοκρασίας και υγρασίας ταυτόχρονα, εξάγοντας ένα βαθμονομημένο ψηφιακό σήμα στην έξοδό του. Με τη χρήση της αποκλειστικής τεχνικής ψηφιακού σήματος και την τεχνολογία ανίχνευσης θερμοκρασίας και υγρασίας, εξασφαλίζει υψηλή αξιοπιστία και εξαιρετικά μακροχρόνια σταθερότητα. Αυτός ο αισθητήρας περιλαμβάνει ένα ωμικού τύπου στοιχείο μέτρησης υγρασίας και ένα NTC στοιχείο μέτρησης θερμοκρασίας τα οποία συνδέονται με ένα υψηλής απόδοσης 8 bit μικροελεγκτή, προσφέροντας άριστη ποιότητα, γρήγορη απόκριση, ικανότητα στο να μένει ανεπηρέαστος στις παρεμβολές και τέλος καλή σχέση κόστους και αποτελεσματικότητας. Κάθε στοιχείο DHT11 βαθμονομείται με εξαιρετική ακρίβεια όσον αφορά την υγρασία. Οι συντελεστές βαθμονόμησης αποθηκεύονται ως προγράμματα στη μνήμη του OTP τα οποία χρησιμοποιούνται από την εσωτερική διαδικασία ανίχνευσης του σήματος του αισθητήρα. Η σειριακή διεπαφή με τις ακίδες καθιστά την ολοκλήρωση του συστήματος εύκολη και γρήγορη. Διακρίνεται για το μικρό μέγεθός του, την χαμηλή κατανάλωση, την πάνω από 20 μέτρα μετάδοση του σήματος, με αποτέλεσμα να είναι η καλύτερη επιλογή όχι μόνο για απλές εφαρμογές αλλά και για πιο απαιτητικές. Το εξάρτημα είναι ένα πακέτο με τέσσερις μονές ακίδες στη σειρά. Είναι βολικό στη σύνδεση και έχει τη δυνατότητα παροχής ειδικών πακέτων σύμφωνα με το αίτημα των χρηστών. Οι παρακάτω πίνακες παρουσιάζουν τις τεχνικές προδιαγραφές του αισθητήρα DHT11. Στον πρώτο πίνακα υπάρχει μία επισκόπηση των τεχνικών προδιαγραφών, στον δεύτερο τα ηλεκτρικά χαρακτηριστικά του και στον τρίτο μία λεπτομερειακή ανάλυση.



Εικόνα 56: Αισθητήρας DHT11.

Εύρος μέτρησης	Ακρίβεια υγρασίας	Ακρίβεια θερμοκρασίας	Ανάλυση	Ακίδες
20 - 90 % RH 0 - 50 °C	±5% RH	±2% °C	1	4 ακίδες σε σειρά

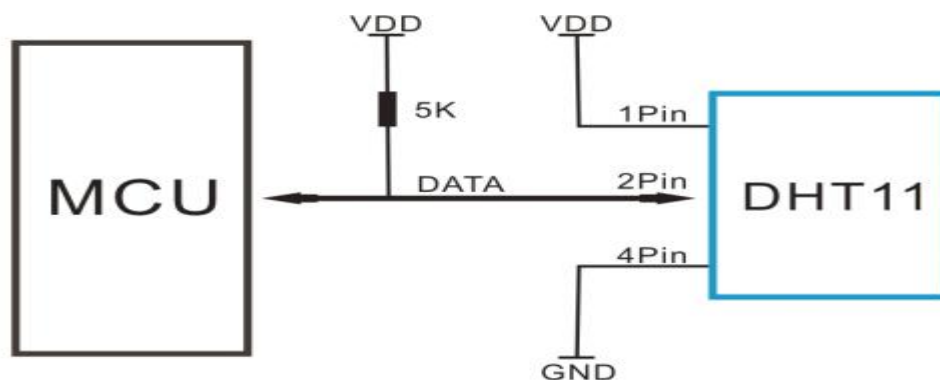
Πίνακας 15: Επισκόπηση τεχνικών χαρακτηριστικών του DHT11.

	Κατάσταση	Ελάχιστο	Τυπικό	Μέγιστο
Τροφοδοσία	DC	3V	5V	5.5V
Παροχή ρεύματος		0.5mA		2.5mA
	μέσος όρος	0.2mA		1mA
	αναμονή	100uA		150uA
Περίοδος δειγματοληψίας	second	1		

Πίνακας 16: Ηλεκτρικά Χαρακτηριστικά για $V_{DD} = 5V$, $T = 25\text{ }^{\circ}\text{C}$.

Παράμετροι	Καταστάσεις	Ελάχιστο	Τυπικό	Μέγιστο
Υγρασία				
Ανάλυση		1%RH	1%RH 8 bit	1%RH
Επαναληψιμότητα			±1%RH	
Ακρίβεια	25°C		±4%RH	
	0-50°C			±5%RH
Εναλλαξιμότητα	Πλήρως Εναλλάξιμα			
Εύρος μέτρησης	0°C	30%RH		90%RH
	25°C	20%RH		90%RH
	50°C	20%RH		80%RH
Χρόνος απόκρισης	1/e(63%)25°C, 1m/s Air	6 S	10 S	15 S
Υστέρηση			±1%RH	
Μακροπρόθεσμη σταθερότητα	Τυπική		±1%RH/year	
Θερμοκρασία				
Ακρίβεια		±1°C		±2°C
Εύρος μέτρησης		0°C		50°C
Χρόνος απόκρισης	1/e(63%)	6 S		30 S

Πίνακας 17: Λεπτομερής πίνακας τεχνικών χαρακτηριστικών του DHT11.

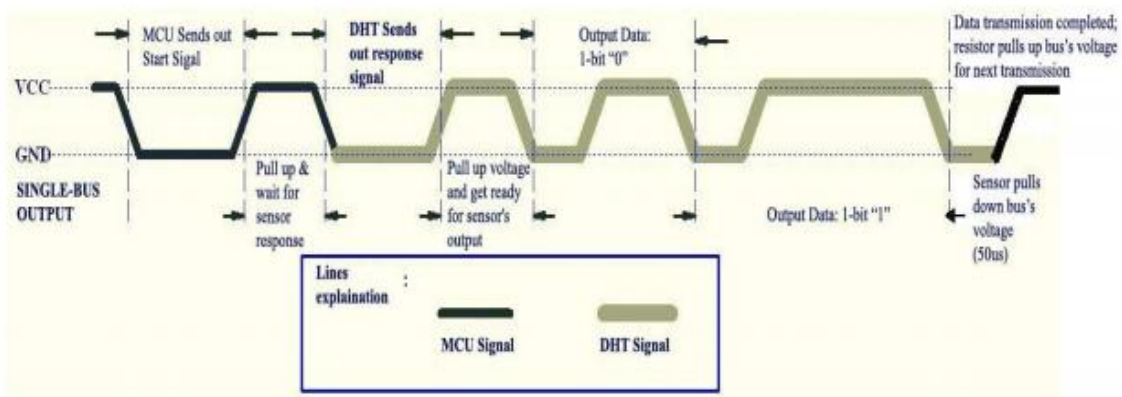


Εικόνα

57:

Διάγραμμα σύνδεσης αισθητήρα DHT11.

Ανάλογα με την απόσταση του καλωδίου σύνδεσης συνιστάται επιλογή κατάλληλης αντίστασης. Η τροφοδοσία του αισθητήρα DHT11 επιβάλλεται να είναι ανάμεσα στα 3 με 3,5V DC. Όταν παρέχεται ρεύμα στον αισθητήρα, δεν πρέπει να σταλεί εντολή προς τον αισθητήρα μέσα σε ένα δευτερόλεπτο, προκειμένου να περάσει την ασταθή κατάσταση. Ένας πυκνωτής 100nF μπορεί να προστεθεί μεταξύ VDD και GND για φιλτράρισμα της τροφοδοσίας. Για την επικοινωνία και τον χρονισμό μεταξύ MCU και του αισθητήρα χρησιμοποιούνται δεδομένα μονού δίαυλου επικοινωνίας. Η διαδικασία επικοινωνίας διαρκεί περίπου 4ms. Τα δεδομένα αποτελούνται από ψηφία και αναπόσπαστα μέρη. Ένα πλήρες σύστημα μετάδοσης δεδομένων είναι 40bit, και η αισθητήρας στέλνει το υψηλότερο bit δεδομένων πρώτα. Όταν η MCU στέλνει ένα σήμα εκκίνησης, ο DHT11 αλλάζει από την κατάσταση χαμηλής κατανάλωσης ενέργειας σε running-mode, περιμένοντας την MCU να ολοκληρώσει το σήμα εκκίνησης. Μόλις ολοκληρωθεί, ο DHT11 στέλνει ένα σήμα ανταπόκρισης των στοιχείων 40-bit που περιλαμβάνει τη σχετική υγρασία και τις πληροφορίες της θερμοκρασίας στην MCU. Οι χρήστες μπορούν να επιλέξουν να συλλέγουν (read) κάποια δεδομένα. Χωρίς το σήμα εκκίνησης από την MCU, ο DHT11 δεν θα δώσει σήμα απάντησης στην MCU. Μόλις συλλέγονται δεδομένα, ο DHT11 θα αλλάξει και θα λειτουργεί σε χαμηλής κατανάλωσης κατάσταση (low-power mode). Έτσι συνεχίζει τη λειτουργία του μέχρι να λάβει ένα σήμα εκκίνησης από MCU ξανά.

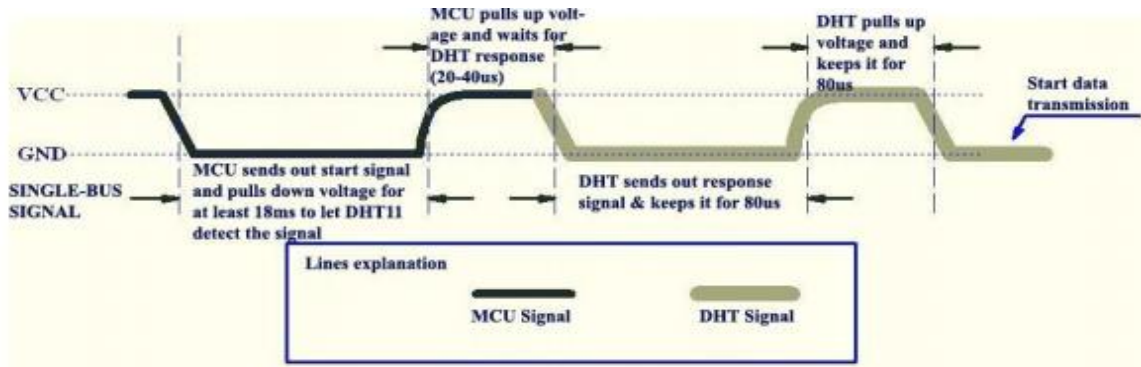


Εικόνα 58: Συνολική διαδικασία επικοινωνίας.

Ο αισθητήρας εκπέμπει 40 bits πληροφορίας τα οποία αντιστοιχούν σε:

- Ø 8 bit για την τιμή της υγρασίας
- Ø 8 bit για την ακέραια τιμή της υγρασίας
- Ø 8 bit για την τιμή της θερμοκρασίας
- Ø 8 bit για την ακέραια τιμή της υγρασίας
- Ø 8 bit για το bit ελέγχου ισοτιμίας δεδομένων

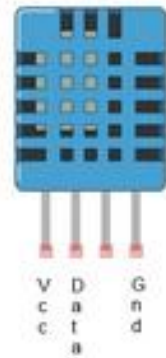
Πριν αναγνωριστεί η ύπαρξη του DHT11 από την MCU η στάθμη της τάσης στο δίαυλο επικοινωνίας είναι υψηλής (high). Όταν η επικοινωνία μεταξύ της MCU και του DHT11 αρχίσει, το πρόγραμμα της MCU θα θέσει στο δίαυλο δεδομένων επίπεδο τάσης από υψηλό (high) σε χαμηλό (low) και αυτή η διαδικασία θα διαρκέσει τουλάχιστον 18ms για να διασφαλιστεί ο εντοπισμός DHT του σήματος MCU. Τότε MCU θα επαναφέρει την τάση σε υψηλή στάθμη και θα περιμένει για απάντηση (περίπου 20-40us) από τον DHT11.



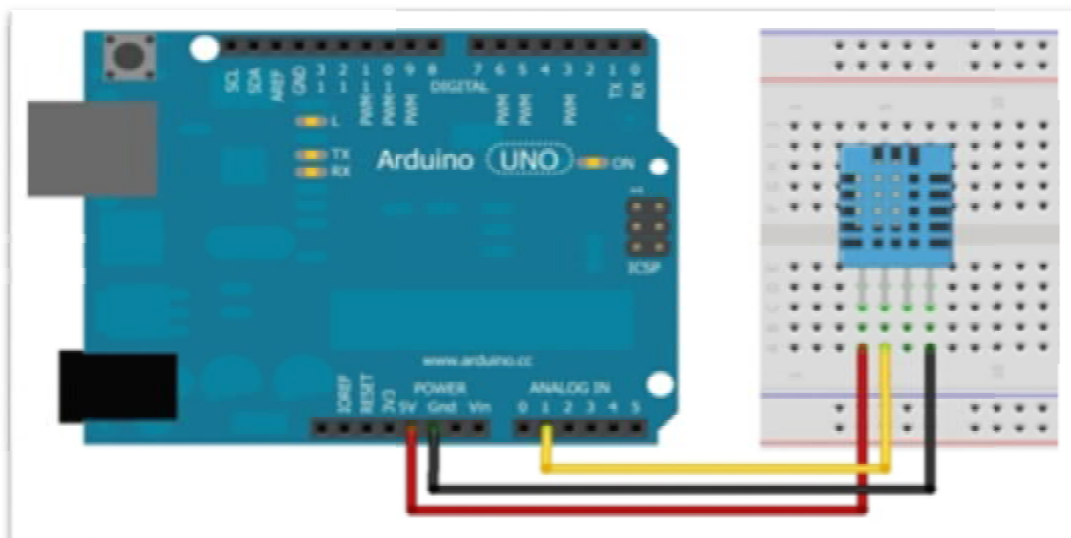
Εικόνα 59: Σήμα εκκίνησης από MCU και απάντησης από DHT11.

Παράδειγμα προγραμματισμού :

```
#include <dht.h>
#define dht_dpin A1
dht DHT;
void setup(){
  Serial.begin(9600);
  Serial.println("Humidity and temperature\n\n");
}
void loop(){
  DHT.read11(dht_dpin);
  Serial.print("Current humidity = ");
  Serial.print(DHT.humidity);
  Serial.print("% ");
  Serial.print("temperature = ");
  Serial.print(DHT.temperature);
  Serial.println("C ");
  delay(800);
}
```



Εικόνα 60: Ακροδέκτες αισθητήρα DHT11.



Εικόνα 61: Σύνδεση αισθητήρα DHT11.

2.2.5 Αισθητήρας απόστασης υπερήχων

Ο HC-SR04 αισθητήρας χρησιμοποιεί σόναρ (παλμούς - κύματα) για να διαπιστώσει την απόσταση ενός αντικειμένου όπως οι νυχτερίδες και τα δελφίνια. Προσφέρει εξαιρετική ανίχνευση με υψηλή ακρίβεια και σταθερών αναγνώσεων σε ένα εύκολο προς χρήση πακέτο. Η λειτουργία του δεν επηρεάζεται από την ηλιοφάνεια ή από τα μαύρα αντικείμενα αλλά είναι δύσκολο να ανιχνεύσει τα μαλακά υλικά όπως ρούχα. Θα χρησιμοποιηθεί για τη μέτρηση της στάθμης νερού σε δεξαμενή. Αποτελείται από έναν πομπό έναν δέκτη υπερήχων και το κύκλωμα ελέγχου.

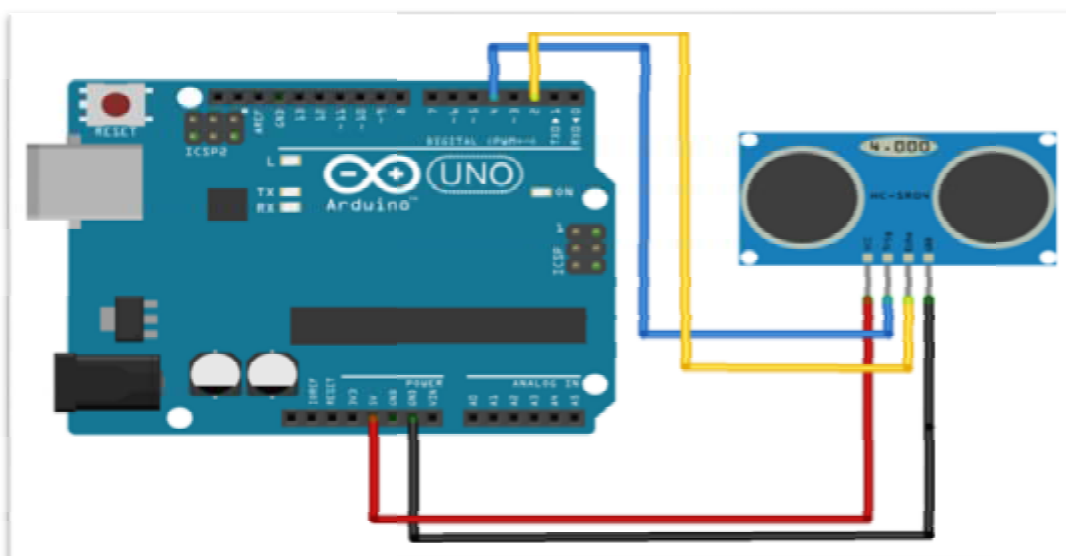


Εικόνα 62: Αισθητήρας HC-SR04.

Στον παρακάτω πίνακα παρουσιάζονται οι τέσσερις ακροδέκτες που διαθέτει ο αισθητήρας.

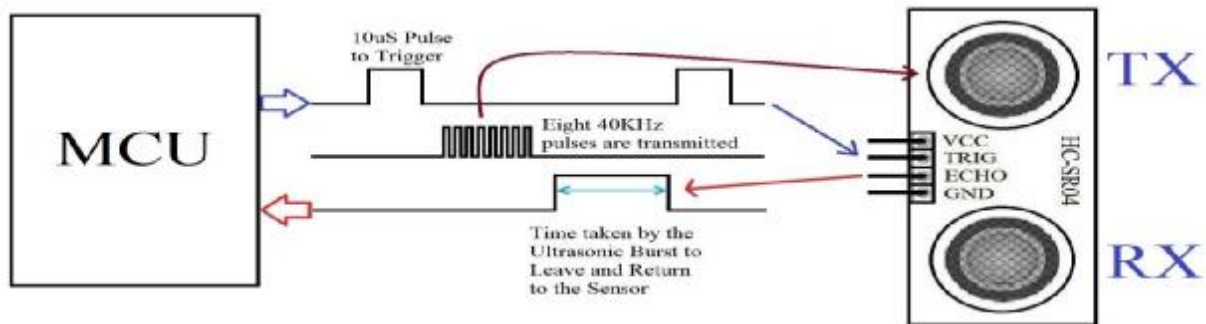
VCC (Τάση)	Συνδέεται στην τάση 5V που παρέχει ο μικροελεγκτής
Trig (Δέκτης)	Συνδέεται στην ψηφιακή είσοδος του μικροελεγκτή.
Echo (Πομπός)	Συνδέεται στην ψηφιακή έξοδο του μικροελεγκτή.
GND (Γείωση)	Συνδέεται με τη γείωση του μικροελεγκτή.

Πίνακας 18: Ακροδέκτες αισθητήρα HC-SR04.



Εικόνα 63: Σύνδεση αισθητήρα απόστασης HC-SR04.

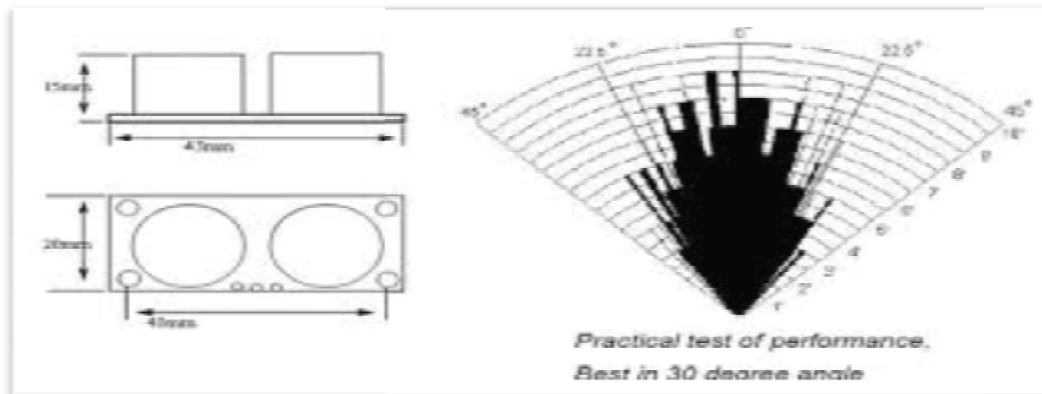
Για να αρχίσει η μέτρηση ο ακροδέκτης trigger του αισθητήρα πρέπει να λάβει παλμό ύψους 5V για τουλάχιστον 10microseconds(us). Έπειτα ο αισθητήρας θα αρχίσει να μεταδίδει 8 κύκλους υπερήχων με συχνότητα 40kHz και θα περιμένει για την αντανάκλαση του υπερήχου. Όταν ο αισθητήρας ανιχνεύσει τον υπέρηχο από τον δέκτη, θα θέσει τον ακροδέκτη echo σε υψηλή στάθμη 5V και θα καθυστερήσει για μία περίοδο η οποία θα είναι ανάλογη της απόστασης. Τα προαναφερόμενα γίνονται κατανοητά με τη βοήθεια της παρακάτω εικόνας στην οποία απεικονίζεται το χρονικό διάγραμμα λειτουργίας του αισθητήρα.



Εικόνα 64: Χρονικό διάγραμμα αισθητήρα HC-SR04.

Τάση λειτουργίας	5V DC
Ρεύμα λειτουργίας	15mA
Ρεύμα αναμονής	< 2mA
Συχνότητα λειτουργίας	40kHz
Παλμός trigger	10µS TTL παλμός
Ελάχιστη απόσταση	2cm
Μέγιστη απόσταση	400cm
Ακρίβεια	0.3cm
Γωνία μέτρησης	15-30 degree
Εύρος θερμοκρασίας	0 έως 70°C
Διαστάσεις	45x20x15mm

Πίνακας 19: Τεχνικά χαρακτηριστικά αισθητήρα HC-SR04.



Εικόνα 65: Διαστάσεις και γωνία μέτρησης αισθητήρα HC-SR04.

Στη συνέχεια ακολουθεί παράδειγμα προγραμματισμού του μικροελεγκτή για μία απλή εφαρμογή μέτρησης απόστασης :

Παράδειγμα προγραμματισμού :

```
const int trigPin = 2;
const int echoPin = 3;
long duration, cm;

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

void loop()
{
    pinMode(trigPin, OUTPUT);
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    pinMode(echoPin, INPUT);
    duration = pulseIn(echoPin, HIGH);

    cm = microsecondsToCentimeters(duration);

    Serial.print(cm);
    Serial.print("cm");
    Serial.println();

    delay(10000);
}

long microsecondsToCentimeters(long microseconds)
{
    return microseconds / 29 / 2;
}
```

2.3 Συσκευές δράσης ψηφιακά ρυθμιζόμενες

Το σύστημα που αναπτύχθηκε μπορεί να αντιλαμβάνεται τι συμβαίνει στο περιβάλλον του θερμοκηπίου και με βάση την νοημοσύνη, που ο χρήστης του έχει εισάγει μέσω του προγραμματισμού του, μπορεί να δρα ανάλογα. Η δράση αυτή επιτυγχάνεται μέσω των συσκευών δράσης.

2.3.1 Ανεμιστήρας

Στη μακέτα χρησιμοποιήθηκε ένας ανεμιστήρας για να καλύψει την ανάγκη μείωσης υγρασίας και θερμοκρασίας. Σε πραγματικές συνθήκες οι ανάγκες του θερμοκηπίου είναι περισσότερες, αλλά σε πρώτο στάδιο έγινε προσπάθεια κάλυψης των βασικών αναγκών. Ακολουθεί μία εικόνα του ανεμιστήρα που χρησιμοποιήθηκε και ένας πίνακας με τα τεχνικά του χαρακτηριστικά.



Εικόνα 66: Ανεμιστήρας μακέτας.

Τάση λειτουργίας	12V DC
Ισχύς	1,56 – 6,6 W
Ταχύτητα	1000 – 2000 rpm +/- 10%
Επίπεδο θορύβου	22 – 37 dB
Διαστάσεις	140x140x25mm

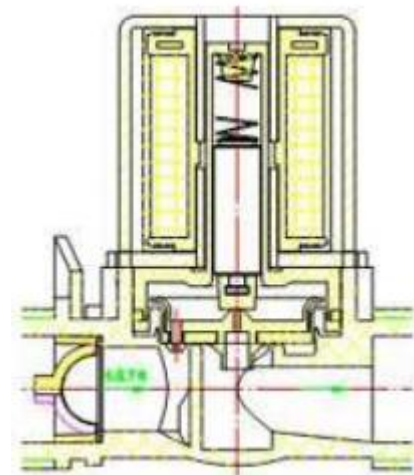
Πίνακας 20: Τεχνικά χαρακτηριστικά ανεμιστήρα.

2.3.2 Ηλεκτροβάνα

Όσον αφορά τον έλεγχο του ποτίσματος επιλέχθηκε μία ηλεκτροβάνα η οποία θα δώσει τη δυνατότητα στον μικροελεγκτή να ελέγχει τη ροή του υγρού. Πρόκειται για μία μαγνητική ελεγχόμενη βαλβίδα υγρού, η οποία όταν τεθεί υπό τάση 12V ανοίγει και επιτρέπει στο υγρό να περάσει. Στη συνέχεια υπάρχουν δύο εικόνες αλλά και τα τεχνικά χαρακτηριστικά της.



Εικόνα 67: Ηλεκτροβάνα.



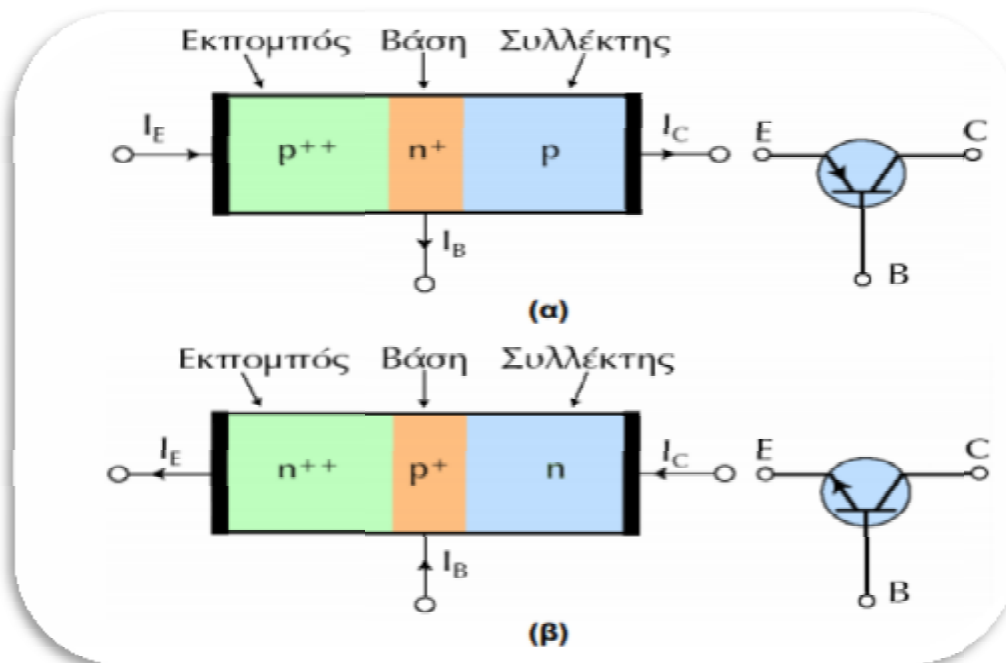
Εικόνα 68: Τομή ηλεκτροβάνας.

Θερμοκρασία λειτουργίας	0 - 100 °C
Πίεση λειτουργίας	400 millimeters water column
Τάση λειτουργίας	12V
Διακύμανση τάσης	15%
Μέγεθος σπειρώματος	3/4 "
Περιβάλλον εργασίας	Νερό, αέριο, λάδι

Πίνακας 21: Τεχνικά χαρακτηριστικά ηλεκτροβάνας.

2.3.3 Διπολικό τρανζίστορ

Το διπολικό τρανζίστορ (bipolar junction transistor-BJT) είναι ένας κρύσταλλος με τρεις περιοχές εμπλουτισμένες με προσμίξεις, δηλαδή αποτελείται από τρία διαδοχικά εναλλασσόμενα στρώματα ημιαγωγού υλικού (sandwich), εκ των οποίων το ενδιάμεσο υλικό είναι είτε τύπου -n (τρανζίστορ τύπου -pnp) ή τύπου -p (τρανζίστορ τύπου -npn). Στην παρακάτω εικόνα υπάρχουν η δομή και το σύμβολο των δύο τύπων του BJT τρανζίστορ.



Εικόνα 69: Δομή και κυκλωματικό σύμβολο τρανζίστορ.

(α) τρανζίστορ- pnp , (β) τρανζίστορ- npn

Ένα τρανζίστορ αποτελείται από 3 περιοχές με διακεκριμένη λειτουργία η καθεμία:

- Εκπομπός (emitter): Είναι μια έντονα εμπλουτισμένη περιοχή (τονίζεται με δύο ++ στην εικ 69) και εκπέμπει φορτία προς τη βάση
- Βάση (base): Είναι μια πολύ λεπτή περιοχή λιγότερο εμπλουτισμένη (τονίζεται με ένα + στην εικ 69), που επιτρέπει στα περισσότερα φορτία που εκπέμπονται από τον εκπομπό να φθάνουν στο συλλέκτη
- Συλλέκτης (collector): Συλλέγει τα φορτία που καταφθάνουν από τον εκπομπό. Το επίπεδο εμπλουτισμού του βρίσκεται μεταξύ αυτών του εκπομπού και της βάσης, ενώ η ισχύς που καταναλώνεται σ' αυτόν είναι μεγαλύτερη από εκείνες που καταναλώνονται στη βάση και στον εκπομπό, γι' αυτό και η περιοχή που καταλαμβάνει ο συλλέκτης είναι μεγαλύτερη. Μπορεί να θεωρηθεί ότι το διπολικό τρανζίστορ αποτελείται από δύο διόδους συνδεδεμένες σε αντίθετη φορά, μια μεταξύ βάσης και εκπομπού (δίοδος ή επαφή εκπομπού) και μια μεταξύ βάσης και συλλέκτη (δίοδος ή επαφή συλλέκτη).

Οι φορείς πλειονότητας σε εκπομπό και συλλέκτη στο τρανζίστορ- pnp είναι οπές, ενώ στο τρανζίστορ- npn είναι ηλεκτρόνια. Αυτό σημαίνει ότι ο ένας τύπος τρανζίστορ είναι συμπληρωματικός του άλλου ή ισοδύναμα ότι κατά τη λειτουργία τους θα έχουν ρεύματα και πολώσεις με αντίθετη φορά. Στη συνέχεια, η μελέτη αναφέρεται στο τρανζίστορ- npn και αντίστοιχα συμπεράσματα θα εξαχθούν – κατ' αναλογία – για το τρανζίστορ- pnp . Όταν δεν εφαρμόζεται πόλωση, τα ελεύθερα ηλεκτρόνια του εκπομπού και του συλλέκτη διαχέονται προς τη βάση, ενώ μέρος των οπών της βάσης από κάθε άκρο της διαχέονται αντίστοιχα

προς τον εκπομπό και το συλλέκτη, με αποτέλεσμα να δημιουργείται περιοχή απογύμνωσης σε κάθε επαφή (επαφή εκπομπού και επαφή συλλέκτη). Κατά μήκος κάθε επαφής αναπτύσσεται φράγμα δυναμικού, το οποίο στους 23°C έχει ύψος 0.3V σε ημιαγωγό Ge και 0.7V σε ημιαγωγό Si (ευρύτατης χρήσης, λόγω των μεγαλύτερων περιορισμών τάσης και έντασης και της μικρότερης εξάρτησης των χαρακτηριστικών του από τη θερμοκρασία).

Αρχή λειτουργίας των διπολικών τρανζίστορ

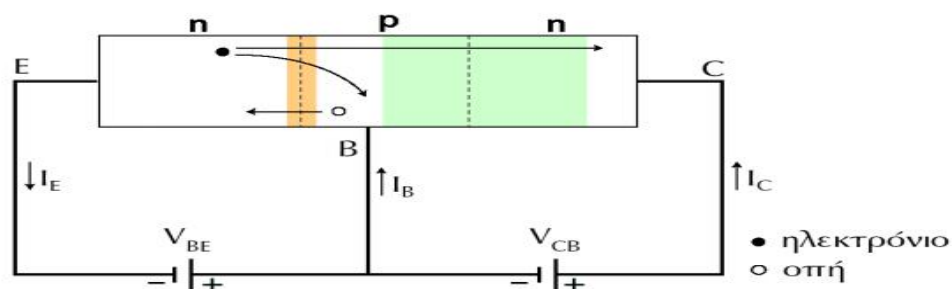
Η λειτουργία ενός διπολικού τρανζίστορ (bipolar ή BJT), βασίζεται στην εκπομπή φορέων από το εκπομπό και τη συλλογή τους από τον συλλέκτη. Σε κανονική λειτουργία, η επαφή εκπομπού είναι πάντα ορθά πολωμένη και η επαφή συλλέκτη ανάστροφα πολωμένη (σε κάποιες ειδικές περιπτώσεις μόνο η επαφή συλλέκτη είναι ορθά πολωμένη), βλ. Σχ. 4. Ανάλογα με την τιμή της τάσης βάσης-εκπομπού (V_{BE}), είναι δυνατές οι ακόλουθες περιπτώσεις:

- Όταν είναι $V_{BE} < 0.7V$ (για τρανζίστορ Si), πρακτικά δεν διέρχεται ρεύμα από τη βάση προς τον εκπομπό.
- Όταν γίνει $V_{BE} > 0.7$, υπάρχει αισθητή ροή ελεύθερων ηλεκτρονίων από τον εκπομπό προς τη βάση και ελεύθερων οπών από τη βάση προς τον εκπομπό.

Η επαφή του συλλέκτη (ανάστροφα πολωμένη) περιορίζει σημαντικά το εύρος της βάσης, με αποτέλεσμα την αύξηση του ποσοστού των ηλεκτρονίων που δεν θα παραμείνουν στο χώρο της βάσης αλλά θα εισέλθουν στο χώρο της επαφής του συλλέκτη. Το ηλεκτρικό πεδίο στην περιοχή φορτίου χώρου του συλλέκτη έχει φορά που ωθεί τα ηλεκτρόνια, που έχουν εισέλθει, προς τον συλλέκτη. Αυτά τα ηλεκτρόνια συλλέγονται από την επαφή του συλλέκτη και δίδουν το ρεύμα συλλέκτη (I_C) (υπάρχει και το ρεύμα ανάστροφης πόλωσης της διόδου συλλέκτη, που θεωρείται αμελητέο). Τα ελεύθερα ηλεκτρόνια, τα οποία παραμένουν στο χώρο της βάσης, μαζί με τις ελεύθερες οπές, οι οποίες εισέρχονται στην περιοχή του εκπομπού, δίνουν το ρεύμα βάσης (I_B). Επειδή το ρεύμα αυτό προκύπτει από αλληλοεξουδετέρωση ελεύθερων ηλεκτρονίων με ελεύθερες οπές, ονομάζεται και ρεύμα επανασύνδεσης (recombination current). Στα περισσότερα τρανζίστορ, περισσότερο από το 95% των φορτίων, που εκπέμπονται από τον εκπομπό, φθάνουν στο συλλέκτη και λιγότερο από το 5% παραμένουν στη βάση και συμβάλλουν στο ρεύμα της βάσης.

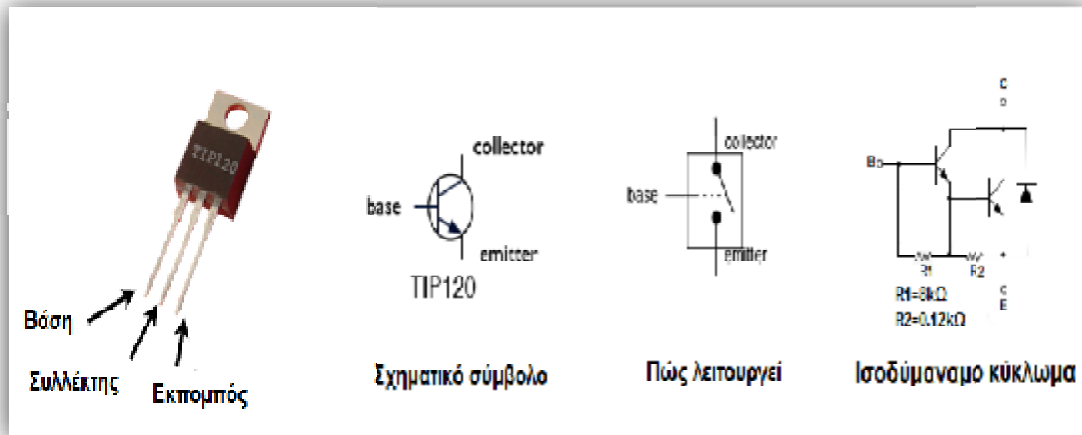
Σε γενικές γραμμές:

- Το ρεύμα συλλέκτη (I_C) είναι περίπου ίσο με το ρεύμα του εκπομπού (I_E).
- Το ρεύμα βάσης (I_B) είναι πολύ μικρό.



Εικόνα 70: Πόλωση τρανζίστορ και ροή ελεύθερων ηλεκτρονίων και οπών.

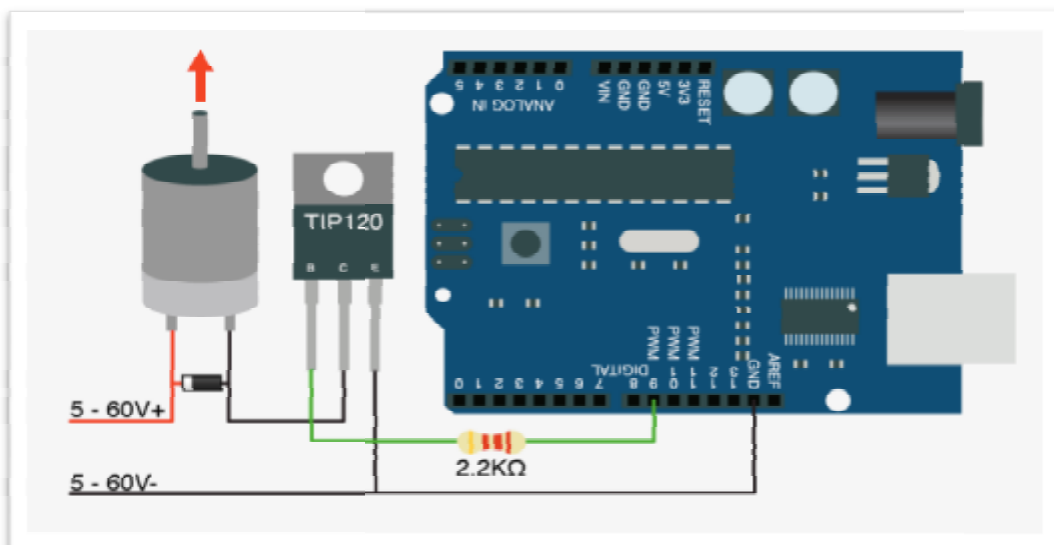
Βάση της κεντρικής ιδέας που αναφέρθηκε παραπάνω, χρησιμοποιήθηκαν 2 ηρη BJT τρανζίστορ, ένα για κάθε συσκευή δράσης που χρησιμοποιείται στο σύστημα που κατασκευάστηκε. Συγκεκριμένα χρησιμοποιήθηκε το μοντέλο TIP 120. Με τον τρόπο αυτό επετεύχθη η σύνδεση συσκευών υψηλότερου φορτίου τάσης (συγκεκριμένα 12 V) με τον μικροελεγκτή που δουλεύει στα 5V. Ουσιαστικά ο μικροελεγκτής ανοίγει ή κλείνει τις συσκευές οι οποίες τροφοδοτούνται μέσω εξωτερικής πηγής, ανεξάρτητα με την αντιστοίχιση του arduino.



Εικόνα 71: Τρανζίστορ TIP120.

2.3.4 Μέθοδος χειρισμού συσκευών δράσης

Για τη σύνδεση των συσκευών, στο κύκλωμα που αλληλεπιδρά με το μικροελεγκτή, εφαρμόστηκε και για τις δύο συσκευές ο ίδιος τρόπος. Εκτός από τα η tip 120 τρανζίστορ που αναλύθηκαν παραπάνω χρησιμοποιήθηκαν δίοδοι και αντιστάσεις των 2.2KΩ. Η εικόνα απεικονίζει τη μέθοδο που χρησιμοποιήθηκε.



Εικόνα 72: Μέθοδος ελέγχου συσκευών.

Στην παραπάνω συνδεσμολογία χρησιμοποιείται μία συσκευή η οποία μπορεί να λειτουργεί υπό τάση από 5 έως και 60 Volt. Ο ακροδέκτης της βάσης του τρανζίστορ, συνδέεται σε μία ψηφιακή έξοδο του μικροελεγκτή, καθώς είναι υπεύθυνος για την ροή του ρεύματος μεταξύ συλλέκτη και εκπομπού. Όταν ο μικροελεγκτής στείλει λογικό 1 στην ψηφιακή έξοδο, τότε η βάση του τρανζίστορ θα επιτρέψει την ροή ρεύματος από το συλλέκτη στον εκπομπό. Αντίθετα όταν στείλει λογικό 0 δεν θα επιτρέπεται η ροή ρεύματος από το συλλέκτη στον εκπομπό. Με αυτόν τον τρόπο ο μικροελεγκτής ελέγχει τη συσκευή.

Ανάμεσα στον συλλέκτη και την ακίδα της ψηφιακής εξόδου του μικροελεγκτή παρεμβάλλεται και μία αντίσταση 2200Ω. Ο ρόλος της είναι πολύ σημαντικός, καθώς όταν ο μικροελεγκτής δεν στέλνει σήμα στην έξοδό του, κρατάει σταθερή και χαμηλή την τάση στο συλλέκτη. Με αυτό τον τρόπο αποφεύγεται κάθε πιθανή επαφή του μικροελεγκτή, με απαγορευτικά για τη λειτουργία του μεγέθη τάσης και ρεύματος.

Παράλληλα με τα παραπάνω ηλεκτρονικά στοιχεία απαιτείται και η χρησιμοποίηση μίας διόδου ανάμεσα στο θετικό και τον αρνητικό πόλο της συσκευής. Αυτό διότι αν η συσκευή που τροφοδοτείται χρησιμοποιεί ηλεκτρικό κινητήρα και είναι σε λειτουργία, όταν σταματήσει να λειτουργεί, για τα επόμενα 5-15ms είναι πολύ πιθανό να επιστρέψει μία τάση, μικρότερη ή ίση σε μέγεθος αλλά με ανάστροφη πολικότητα από αυτή την οποία τροφοδοτήθηκε. Η διάδος είναι τοποθετημένη με τέτοιο τρόπο, ώστε αυτή η τάση να μην εισέρχεται στο συλλέκτη του τρανζίστορ, διότι αυτό θα είχε σαν αποτέλεσμα να καεί και το τρανζίστορ αλλά και ο μικροελεγκτής. Στο κύκλωμα που πραγματοποιήθηκε χρησιμοποιήθηκε μία διάδος του 1A.

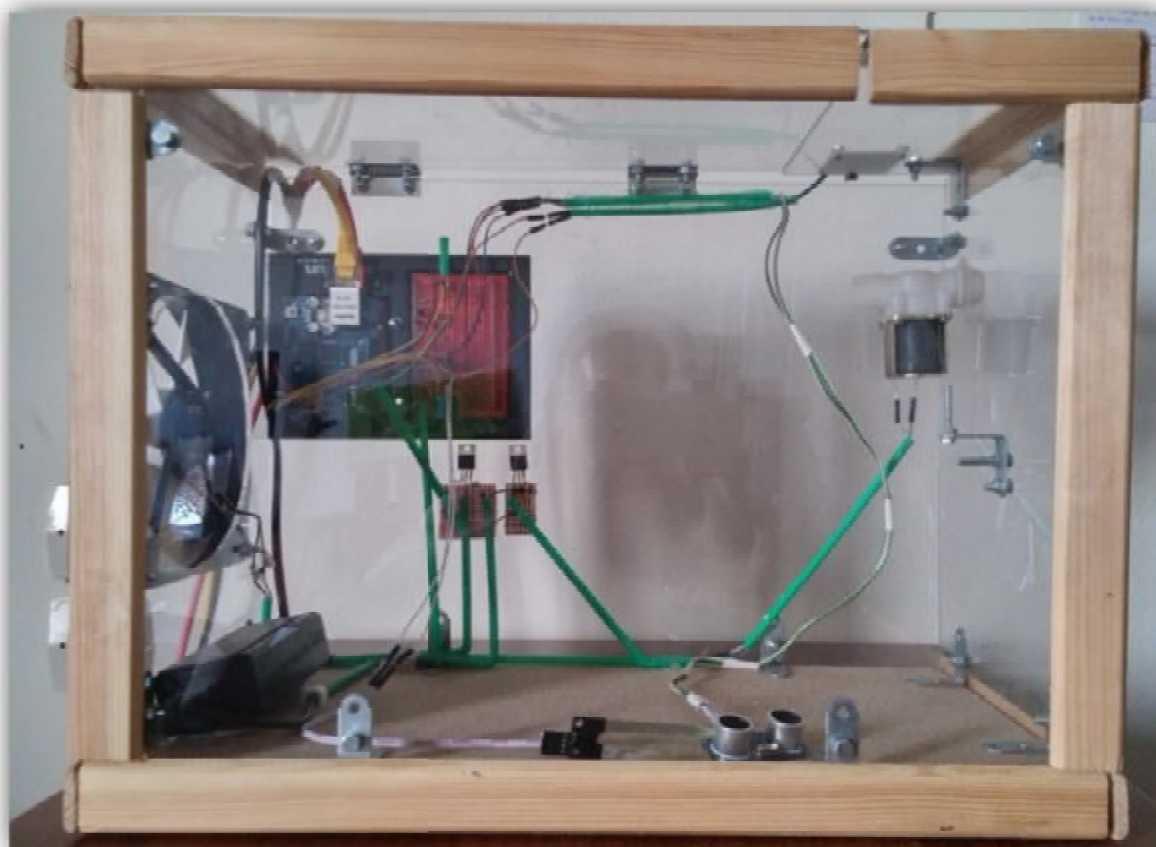
Ηλεκτρικά χαρακτηριστικά συσκευών δράσης			
Συσκευή	Τάση εισόδου	Ρεύμα λειτουργίας	Ισχύς
Ανεμιστήρας	12V	0.13-0.55A	1,56 – 6,6 W
Ηλεκτροβάνα	12V	0.18	2.16W

Πίνακας 22: Ηλεκτρικά χαρακτηριστικά συσκευών δράσης.

Για την τροφοδοσία των συσκευών δράσης επιλέχθηκε κατάλληλο τροφοδοτικό το οποίο δίνει τάση 12V και παρέχει την απαιτούμενη ισχύ που απαιτούν οι συσκευές, για την ορθή λειτουργία τους.

2.3.5 Κατασκευή μακέτας – Κόστος συστήματος

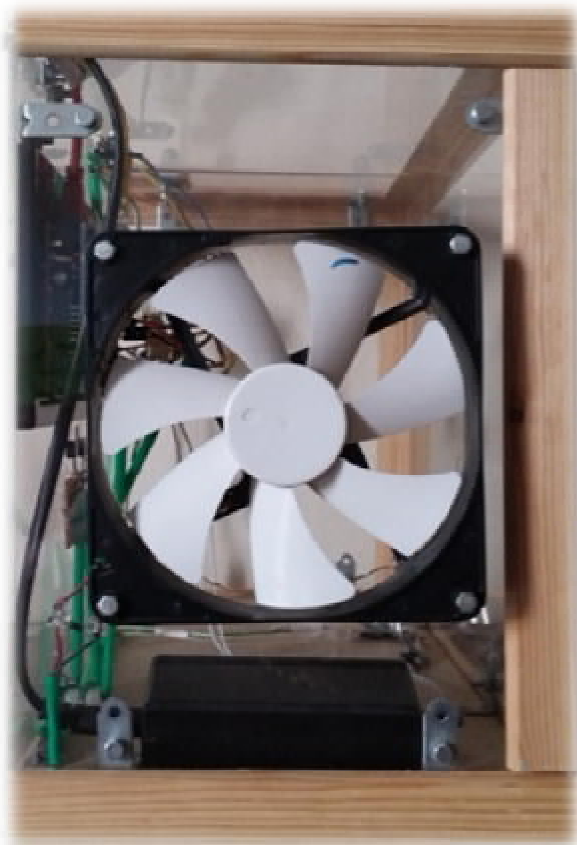
Κατά την υλοποίηση του συστήματος, κατασκευάστηκε μία μακέτα η οποία προσέγγιζε, όσο βέβαια είναι αυτό δυνατό, τις συνθήκες που επικρατούν σε ένα θερμοκήπιο. Αυτό βοήθησε στην κατανόηση των συνθηκών και ταυτόχρονα των αναγκών, που χρειαζόταν να ληφθούν υπ' όψιν, έτσι ώστε αυτό το σύστημα να ανταποκρίνεται σε συνθήκες που πλησιάζουν τις πραγματικές. Το μεγαλύτερο μέρος της μακέτας είναι κατασκευασμένο από plexiglass. Αυτό έγινε ούτως ώστε να υπάρχει η δυνατότητα επαλήθευσης ανάμεσα στα δεδομένα που παρουσιάζονται στον ιστοχώρο και στα πραγματικά. Οι διαστάσεις της μακέτας είναι 50εκ. μήκος, 25 εκ. πλάτος, 32 εκ. ύψος και ζυγίζει περίπου 2 κιλά. Τέλος η κατανάλωση του συστήματος είναι λιγότερη από 4W.



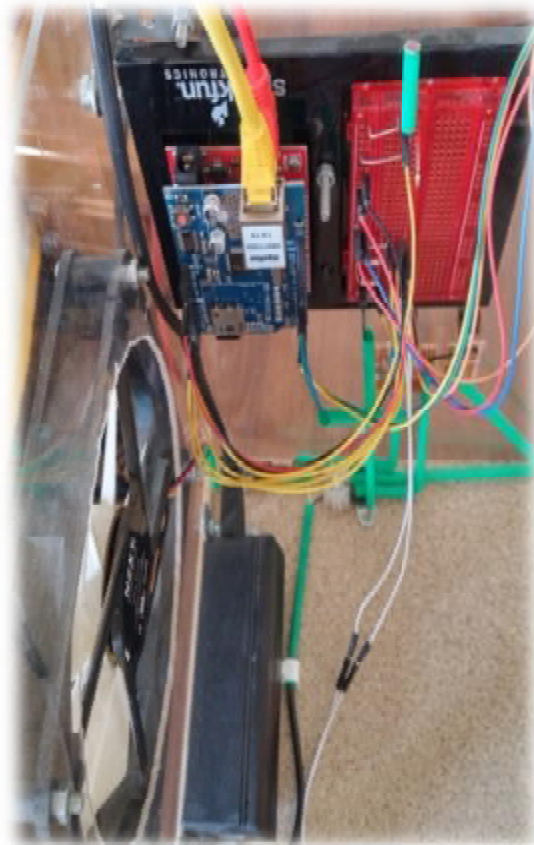
Εικόνα 73: Φωτογραφία μακέτας θερμοκηπίου 1.

Κόστος εξαρτημάτων	
Εξάρτημα	Τιμή
Arduino RedBoard	22€
Arduino Ethernet Shield	10€
Αισθητήρας Θερμοκρασίας-Υγρασίας	3€
Αισθητήρας φωτεινότητας	0,70€
Αισθητήρας υγρασίας εδάφους	2,5€
Αισθητήρας βροχής	2,5€
Αισθητήρας απόστασης με υπερήχους	4€
Ηλεκτροβάνα	8€
Ανεμιστήρας	5€
Μετασχηματιστής	10€
Αντιστάσεις-δίοδοι-ρυθμιστές τάσης	3€
Breadboard-Καλώδια	5€
Υλικά μακέτας	40€
Συνολικό κόστος	115.7€

Πίνακας 23: Κόστος εξαρτημάτων.



Εικόνα 74: Φωτογραφία μακέτας θερμοκηπίου 3.



Εικόνα 75: Φωτογραφία μακέτας θερμοκηπίου 2



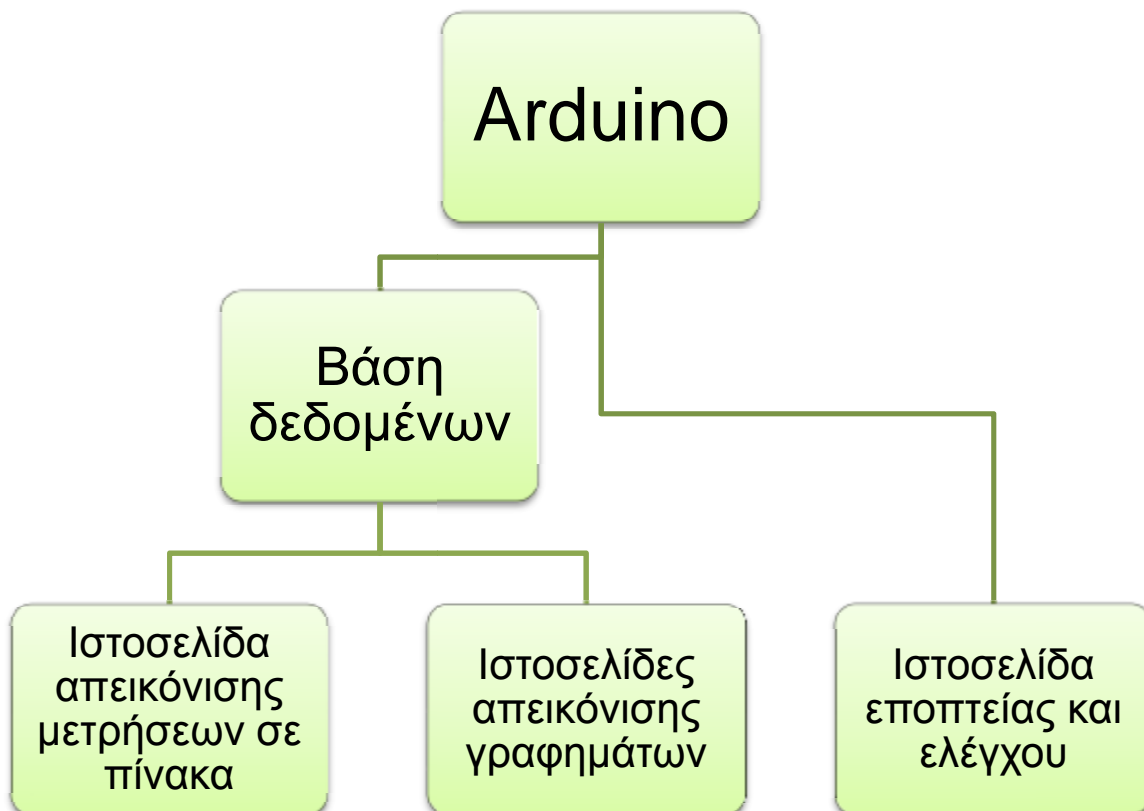
Εικόνα 76: Φωτογραφία μακέτας θερμοκηπίου 4.

Κεφάλαιο 3: Λογισμικό μέρος συστήματος

Σε αυτό το κεφάλαιο θα αναλυθεί η λογική με την οποία υλοποιήθηκε και λειτουργεί το σύστημα εποπτείας και ελέγχου ενός θερμοκηπίου, η οποία αποτελείται από το λογισμικό μέρος του συστήματος. Το σύνολο του λογισμικού μέρους μπορεί να διακριθεί σε δύο βασικά τμήματα. Το πρώτο τμήμα έχει να κάνει με την τεχνική πλευρά του συστήματος και ορίζεται ως «τμήμα μικροελεγκτή». Πιο συγκεκριμένα το τμήμα αυτό είναι υπεύθυνο για την αναγνώριση συσκευών, την σύνδεση μεταξύ τους, αλλά και την σύνδεση του κυκλώματος με τον μικροελεγκτή. Το δεύτερο τμήμα αφορά τις δυνατότητες διαχείρισης του συστήματος από τον χρήστη και ορίζεται ως «διαδικτυακό τμήμα».

Αμεσότητα και ακρίβεια είναι τα απαιτούμενα χαρακτηριστικά, του τρόπου με τον οποίο αλληλεπιδρά - μέσω του ιστοχώρου - το σύστημα με το περιβάλλον στο οποίο είναι εγκατεστημένο. Γίνεται αντιληπτό ότι η υλοποίηση του λογισμικού, των συσκευών και του ιστοχώρου απαιτεί μεγαλύτερη χρονική διάρκεια σε σχέση με τη σύνδεση των διαφόρων εξαρτημάτων. Συνοπτικά στη συνέχεια θα αναλυθεί ο τρόπος προγραμματισμού του Arduino, του ιστοχώρου αλλά και οι λειτουργίες αυτού. Οι ενότητες του λογισμικού μέρους του συστήματος είναι οι ακόλουθες:

- Ø Απαιτήσεις συστήματος
- Ø Βάση δεδομένων
- Ø Λειτουργίες ιστοχώρου
- Ø Προγραμματισμός Arduino



Εικόνα 77: Λογισμικά μέρη συστήματος.

3.1 Απαιτήσεις συστήματος

Το σύστημα που υλοποιήθηκε αντικαθιστά τη φυσική παρουσία του ιδιοκτήτη ενός θερμοκηπίου στο χώρο αυτό. Σκοπός του είναι η επίβλεψη και συγχρόνως η καταγραφή των περιβαλλοντικών συνθηκών που επικρατούν στο χώρο του θερμοκηπίου. Μέσω της δυνατότητας επίβλεψης ο χρήστης έχει τη δυνατότητα να σχηματίσει μία εικόνα για την κατάσταση του θερμοκηπίου σε πραγματικό χρόνο, ενώ η καταγραφή των περιβαλλοντικών συνθηκών βοηθάει στην εξαγωγή χρήσιμων συμπερασμάτων ώστε να επιτευχθεί το βέλτιστο αποτέλεσμα. Η σημαντικότερη όμως λειτουργία όσον αφορά την αντικατάσταση της φυσικής παρουσίας του χρήστη, είναι αυτή του απομακρυσμένου ελέγχου συσκευών. Έτσι λοιπόν ο χρήστης χρησιμοποιώντας το σύστημα εποπτείας και ελέγχου που υλοποιήθηκε, έχει τη δυνατότητα να επιβλέπει περιβαλλοντικές συνθήκες και να ελέγχει συσκευές μέσω διαδικτύου. Το σύστημα καλύπτει τις παρακάτω απαιτήσεις:

- Ø **Φιλικό προς το χρήστη περιβάλλον:** Το σύστημα σχεδιάστηκε με τέτοιο τρόπο ούτως ώστε να είναι φιλικό προς το χρήστη, να μην του προκαλεί σύγχυση η διαχείρισή του, και το περιβάλλον να του θυμίζει γνώριμα εργαλεία και καταστάσεις. Ο λόγος αυτών είναι ότι το σύστημα πιθανόν να χρησιμοποιηθεί και από άτομα με έλλειψη τεχνολογικής κατάρτισης.
- Ø **Αντικατάσταση ανθρώπινης παρουσίας:** Η αντικατάσταση της συνεχούς φυσικής παρουσίας στο χώρο ενός θερμοκηπίου είναι ο σημαντικότερος σκοπός του χρήστη, που χρησιμοποιεί το συγκεκριμένο σύστημα. Για το λόγο αυτό υλοποιήθηκαν σενάρια τα οποία αντικαθιστούν απόλυτα την ανθρώπινη νοημοσύνη και εφαρμόζονται με τη βοήθεια των αισθητήρων. Με αυτόν τον τρόπο ο χρήστης δύναται να καθορίζει τη συμπεριφορά των συσκευών δράσης σύμφωνα με τις περιβαλλοντικές συνθήκες που επικρατούν στο θερμοκήπιο, λαμβάνοντας υπ'όψιν τα δεδομένα των αισθητήρων.
- Ø **Άμεσος έλεγχος συσκευών δράσης:** Ο χρήστης έχει τη δυνατότητα , εκτός από τον προγραμματισμό του συστήματος με σενάρια, να θέτει σε λειτουργία ή να απενεργοποιεί τις συσκευές δράσης, σε πραγματικό χρόνο (real time). Αυτό θεωρείται ως χειροκίνητο (manual) σενάριο και οποιαδήποτε στιγμή ενεργοποιείται από την ιστοσελίδα ελέγχου και εποπτείας.
- Ø **Καταγραφή περιβαλλοντικών συνθηκών:** Δημιουργήθηκαν διαγράμματα, τα οποία καλύπτουν τις απαιτήσεις του χρήστη, και οπτικοποιούν σε σχέση με το χρόνο την κατάσταση του κάθε αισθητήρα. Με αυτόν τον τρόπο ο χρήστης έχει στη διάθεση του στατιστικά τα οποία ανταποκρίνονται στις περιβαλλοντικές συνθήκες που επικρατούσαν στο χώρο του θερμοκηπίου τα περασμένα χρονικά διαστήματα. Εκτός των διαγραμμάτων αυτών, που παρουσιάζουν τις τελευταίες μετρήσεις, σχεδιάστηκε μία σελίδα η οποία αποτελείται από έναν πίνακα, ο οποίος περιέχει όλες τις μετρήσεις που έχουν καταγραφεί.

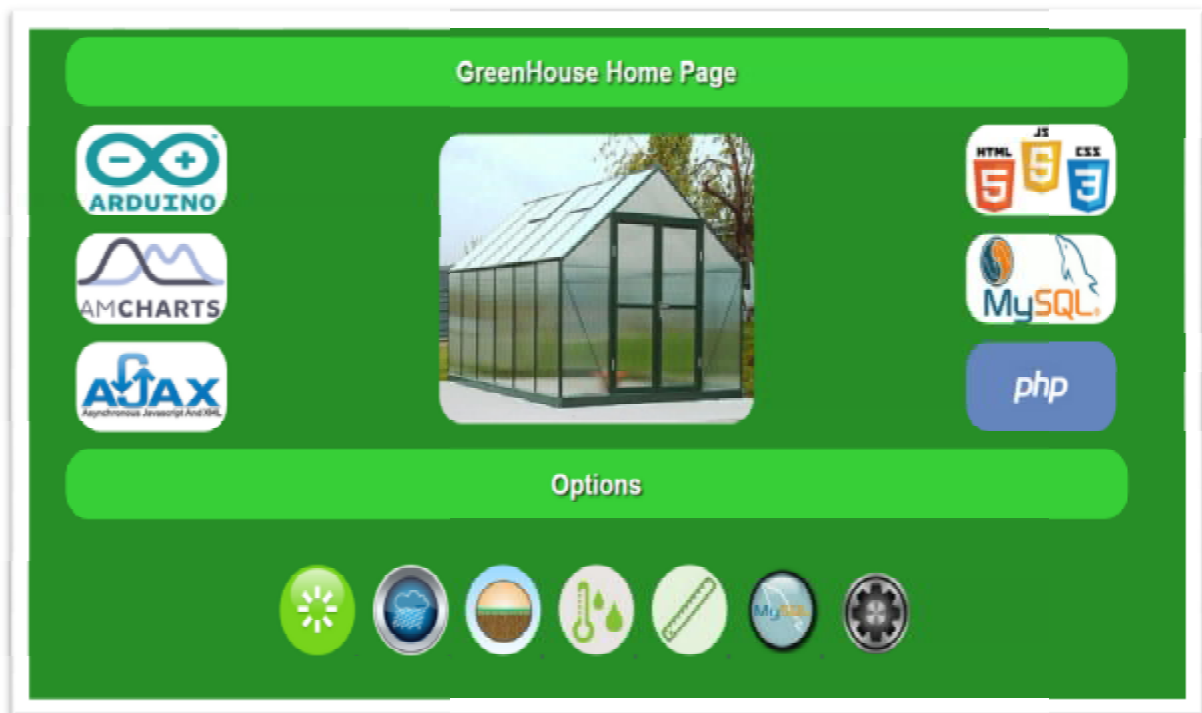
Στις επόμενες ενότητες γίνεται λεπτομερειακή παρουσίαση στον τρόπο αντιμετώπισης των απαιτήσεων αυτών. Στη συνέχεια ακολουθεί η παρουσίαση της βάσης δεδομένων του συστήματος.

3.2 Λειτουργίες ιστοχώρου

Στην ενότητα αυτή γίνεται παρουσίαση των λειτουργιών που προσφέρει το σύστημα εποπτείας και ελέγχου ενός θερμοκηπίου. Το «χτίσιμο» του ιστοχώρου, πραγματοποιήθηκε με σκοπό την ικανοποίηση των απαιτήσεων, οι οποίες επισημάνθηκαν σε προηγούμενη ενότητα.

3.2.1 Αρχική σελίδα

Στην αρχική σελίδα υπάρχουν επτά κουμπιά, τα οποία με το πάτημα τους σε μεταφέρουν στην ιστοσελίδα που τους αντιστοιχεί. Τα πέντε από τα επτά κουμπιά σε μεταφέρουν σε σελίδες οι οποίες απεικονίζουν τα σχεδιαγράμματα από τις μετρήσεις των αισθητήρων, ένα κουμπί σε μεταφέρει στη σελίδα εποπτείας και ελέγχου του Arduino και ένα άλλο στην ιστοσελίδα με τον πίνακα όπου υπάρχουν όλες οι εγγραφές των αισθητήρων.



Εικόνα 78: Αρχική σελίδα.

Στη συνέχεια παρουσιάζεται ο κώδικας της αρχικής σελίδας. Το αρχείο ονομάζεται home.html και είναι αποθηκευμένο στο φάκελο htdocs του τοπικού server.

```
<DOCTYPE html>
<html>
  <style>
    h1 {
      width:1400;
      margin-top:30px;
      margin-right:85px;
      margin-bottom:40px;
      margin-left:85px;
      font-weight: bold;
      font-style:normal;
      text-decoration: none;
      background-color:#32CD32;
      color:white;
      text-shadow: 2px 2px 4px #000000;
      border-radius: 30px;
      padding-top:20px ;
      padding-bottom:20px ;
      font-family: Arial;
      text-align:center;
    }
    body {
      background-color:#228B22;
      text-align: center;
    }
  </style>
  <head>
    <title>GreenHouse Home Page</title>
  </head>
  <body>
    <h1>GreenHouse Home Page</h1>
    <div>
      
      
      
    </div>
    <div>
      
      
    </div>
  </div>

```

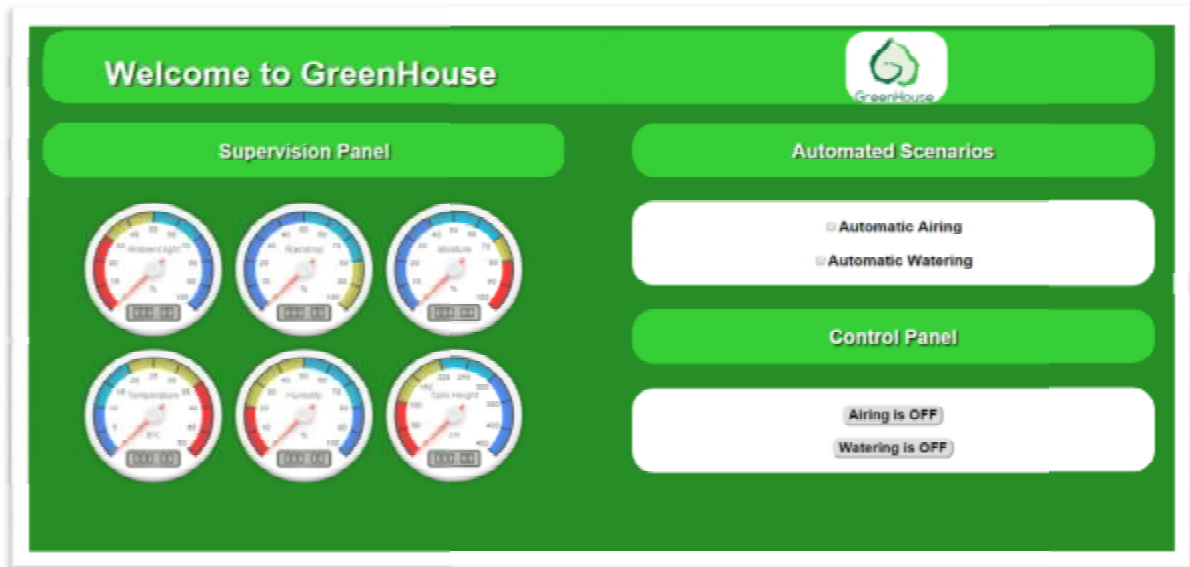
```

        
        
    </div>
    <h1>Options</h1>
    <div style="margin:50px 0px 0px 0px;">
        <a href="charts/light.html">
            
        </a>
        <a href="charts/raindrop.html">
            
        </a>
        <a href="charts/moisture.html">
            
        </a>
        <a href="charts/dht.html">
            
        </a>
        <a href="charts\tank.html">
            
        </a>
        <a href="review_data.php">
            
        </a>
        <a href="https://169.254.208.1/">
            
        </a>
    </div>
</body>
</html>

```

3.2.2 Σελίδα εποπτείας και ελέγχου

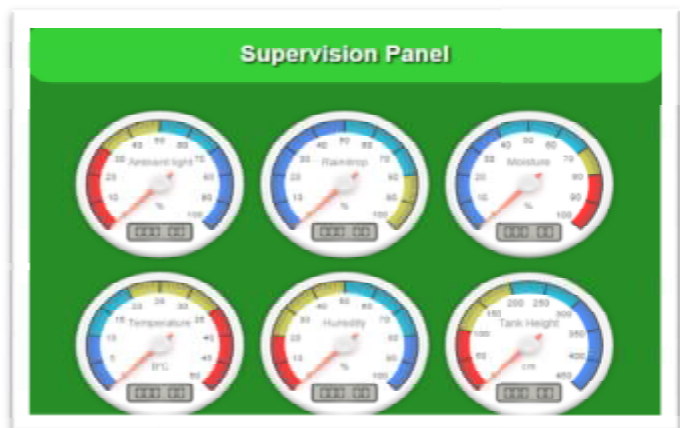
Η σελίδα αυτή αποτελείται από 3 μέρη. Το πρώτο μέρος αφορά την εποπτεία του συστήματος σε πραγματικό χρόνο και ονομάζεται πίνακας εποπτείας (Supervision Panel). Περιλαμβάνει έξι μετρητές οι οποίοι απεικονίζουν με αναλογικό αλλά και ψηφιακό τρόπο την τιμή του κάθε αισθητήρα. Το δεύτερο μέρος ονομάζεται αυτοματοποιημένα σενάρια (automated scenarios) και έχει να κάνει με την ενεργοποίηση – απενεργοποίηση τους. Τέλος το τρίτο μέρος αφορά τον χειρισμό των συσκευών δράσης και ονομάζεται πίνακας ελέγχου.



Εικόνα 79: Σελίδα εποπτείας και ελέγχου.

Πίνακας εποπτείας:

Στον πίνακα εποπτείας υπάρχουν έξι πεδία απεικόνισης τα οποία έχουν τη μορφή ενός μετρητή κινητού πηνίου με βελόνα, ούτως ώστε να απεικονίζονται οι τιμές των αισθητήρων με αναλογικό τρόπο. Αυτό βοηθάει στην γρήγορη ανάγνωση της κατάστασης των αισθητήρων. Το μειονέκτημα αυτού του τρόπου είναι ότι ο χρήστης δεν μπορεί διακρίνει την ακριβής αριθμητική τιμή. Γι αυτό το λόγο υπάρχει και ψηφιακή απεικόνιση των τιμών των αισθητήρων, η οποία έχει τη μορφή μίας ψηφιακής συσκευής απεικόνισης. Ακόμα κάθε μετρητής έχει διαφορετικά χρώματα σε κάθε φάσμα τιμών τα οποία δηλώνουν μία πιο γενική κατάσταση του κάθε αισθητήρα. Στο πάνω μέρος του μετρητή περιγράφεται το μέγεθος το οποίο απεικονίζει, ενώ στο κάτω μέρος τη μονάδα μέτρησης αυτού. Ο κώδικας της σελίδας, λόγω του μεγέθους του το οποίο δεν θα βοηθούσε



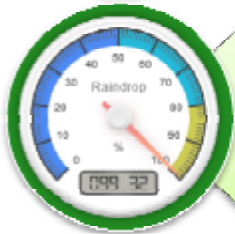
Εικόνα 80: Πίνακας εποπτείας.

στη ροή της εργασίας, βρίσκεται στο παράρτημα Α. Στη συνέχεια υπάρχει μία αναλυτική αναφορά για τον κάθε μετρητή του πίνακα εποπτείας.



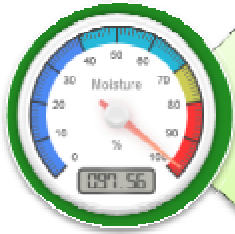
Αισθητήρας φωτεινότητας - μονάδα μέτρησης %

0-30 Κόκκινο: Πολύ φωτεινό
30-50 Πράσινο: Αρκετά φωτεινό
50-70 Γαλάζιο: Φωτεινό
70-100 Μπλε: Σκοτάδι



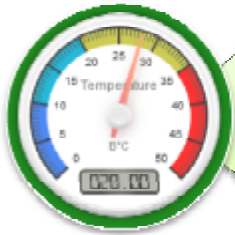
Αισθητήρας βροχής - μονάδα μέτρησης %

0-50 Μπλε: Βροχή
50-80 Γαλάζιο: Ψιχάλες
80-100 Πράσινο: Καθόλου βροχή



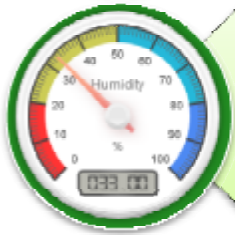
Αισθητήρας υγρασίας χώματος - μονάδα μέτρησης %

0-40 Μπλε: Πολύ υγρό
40-70 Γαλάζιο: Υγρό
70-80 Πράσινο: Ξερό
80-100 Κόκκινο: Πολύ ξερό



Αισθητήρας θερμοκρασίας- μονάδα μέτρησης °C

0-10 Μπλε: Χαμηλή θερμοκρασία
10-20 Γαλάζιο: Κανονική θερμοκρασία
20-35 Πράσινο: Υψηλή θερμοκρασία
35-50 Κόκκινο: Πολύ υψηλή θερμοκρασία



Αισθητήρας υγρασίας- μονάδα μέτρησης %

0-20 Κόκκινο: Χαμηλή υγρασία
20-50 Πράσινο: Κανονική υγρασία
50-80 Γαλάζιο: Υψηλή υγρασία
80-100 Μπλε: Πολύ υψηλή υγρασία

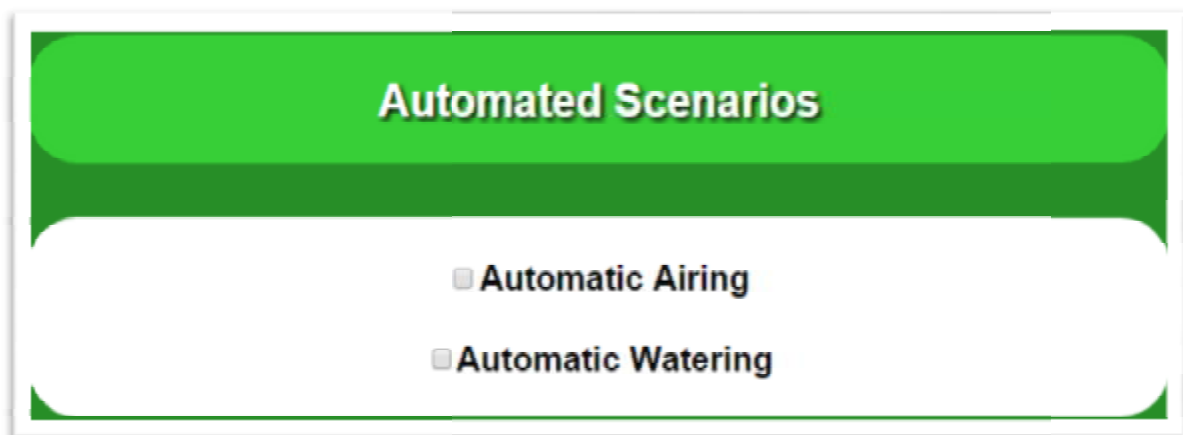


Αισθητήρας υγρασίας- μονάδα μέτρησης cm

0-100 Κόκκινο: Σχεδόν άδεια δεξαμενή
100-200 Πράσινο: Χαμηλή στάθμη δεξαμενής
200-300 Γαλάζιο: Υψηλή στάθμη δεξαμενής
300-450 Μπλε: Γεμάτη δεξαμενή

Αυτόματα σενάρια:

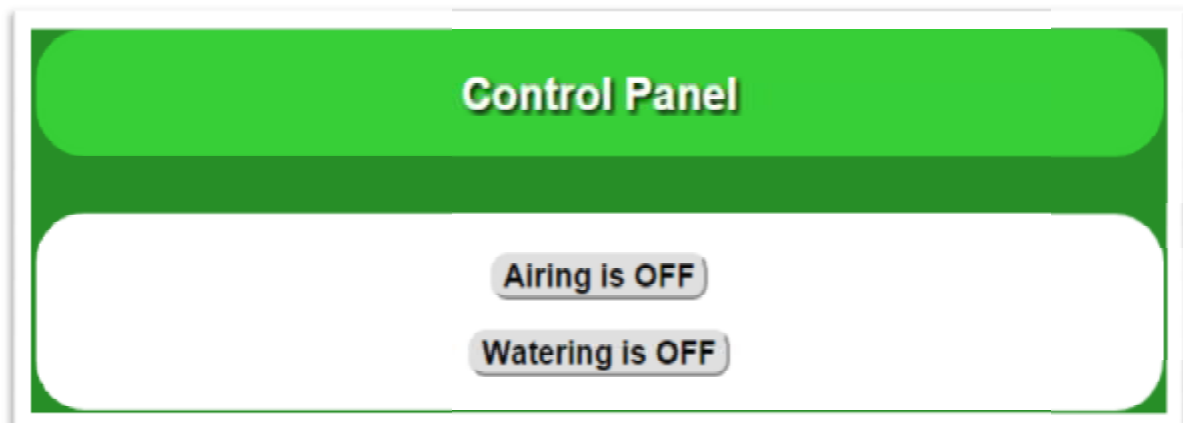
Όπως έχει προαναφερθεί υλοποιήθηκαν δύο αυτόματα σενάρια τα οποία συμβάλουν στην απαίτηση της αντικατάστασης ανθρώπινης παρουσίας στο χώρο του θερμοκηπίου. Το πρώτο σενάριο αφορά τον αυτόματο αερισμό του θερμοκηπίου. Πιο συγκεκριμένα όταν ενεργοποιηθεί αυτό το σενάριο, κάθε φορά που η θερμοκρασία ξεπεράσει μία συγκεκριμένη τιμή, τότε θα ενεργοποιηθεί ο ανεμιστήρας έως ότου η θερμοκρασία επιστρέψει ξανά σε φυσιολογικά επίπεδα. Το δεύτερο σενάριο αφορά το αυτόματο πότισμα του θερμοκηπίου. Αυτό σημαίνει ότι όταν ο αισθητήρας υγρασίας εδάφους αντιληφθεί ότι το χώμα είναι αρκετά ξερό, ο μικροελεγκτής θα θέσει σε λειτουργία την ηλεκτροβάννα, μέχρι το έδαφος να πάρει την απαιτούμενη υγρασία. Αυτά τα δύο σενάρια ενεργοποιούνται τσεκάροντας τα αντίστοιχα κουτάκια(εικ.).



Εικόνα 81: Αυτόματα σενάρια.

Πίνακας ελέγχου:

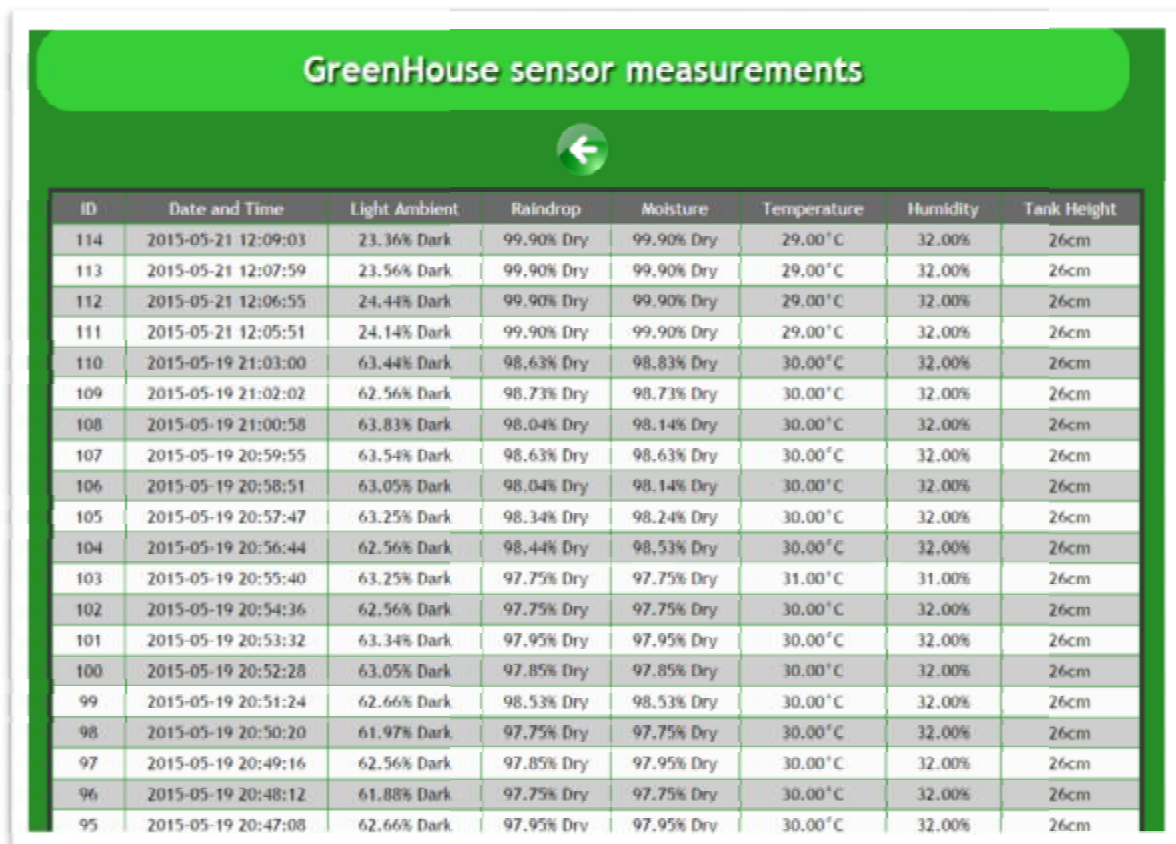
Το τελευταίο μέρος της σελίδας εποπτείας και ελέγχου αφορά τον πίνακα ελέγχου(control panel).Ο πίνακας ελέγχου αποτελείται από δύο κουμπιά τα οποία ενεργοποιούν και απενεργοποιούν τις συσκευές δράσης, δηλαδή τον ανεμιστήρα και την ηλεκτροβάννα. Πατώντας το κάθε κουμπί αλλάζει η κατάσταση της συσκευής.Εφ'όσον γίνει αυτό, αλλάζει και η ένδειξη του κουμπιού και από ON γίνεται OFF και αντίστροφα.



Εικόνα 82: Πίνακας ελέγχου συσκευών δράσης.

3.2.3 Σελίδα απεικόνισης μετρήσεων σε πίνακα

Η σελίδα αυτή δημιουργήθηκε ώστε ο χρήστης να έχει τη δυνατότητα να δει οποιαδήποτε μέτρηση έχει πραγματοποιηθεί. Οι μετρήσεις παρουσιάζονται σε ένα πίνακα, όπου στην κορυφή αυτού βρίσκεται η τελευταία μέτρηση και από κάτω όλες οι υπόλοιπες. Ο χρήστης έχει τη δυνατότητα να μεταφερθεί σε αυτή τη σελίδα, πατώντας το αντίστοιχο κουμπί της αρχικής σελίδας. Επίσης το ίδιο κουμπί υπάρχει και στις πέντε σελίδες οι οποίες απεικονίζουν διαγράμματα καταγεγραμμένων μετρήσεων του κάθε αισθητήρα. Τέλος στην σελίδα απεικόνισης μετρήσεων σε πίνακα, υπάρχει και ένα κουμπί επιστροφής στην αρχική σελίδα.



The screenshot shows a web interface titled "GreenHouse sensor measurements". It features a table with 8 columns: ID, Date and Time, Light Ambient, Raindrop, Moisture, Temperature, Humidity, and Tank Height. The table contains 20 rows of data, with the most recent measurement at the top (ID 114) and older ones below. A back arrow icon is visible above the table.

ID	Date and Time	Light Ambient	Raindrop	Moisture	Temperature	Humidity	Tank Height
114	2015-05-21 12:09:03	23.36% Dark	99.90% Dry	99.90% Dry	29.00°C	32.00%	26cm
113	2015-05-21 12:07:59	23.56% Dark	99.90% Dry	99.90% Dry	29.00°C	32.00%	26cm
112	2015-05-21 12:06:55	24.44% Dark	99.90% Dry	99.90% Dry	29.00°C	32.00%	26cm
111	2015-05-21 12:05:51	24.14% Dark	99.90% Dry	99.90% Dry	29.00°C	32.00%	26cm
110	2015-05-19 21:03:00	63.44% Dark	98.63% Dry	98.83% Dry	30.00°C	32.00%	26cm
109	2015-05-19 21:02:02	62.56% Dark	98.73% Dry	98.73% Dry	30.00°C	32.00%	26cm
108	2015-05-19 21:00:58	63.83% Dark	98.04% Dry	98.14% Dry	30.00°C	32.00%	26cm
107	2015-05-19 20:59:55	63.54% Dark	98.63% Dry	98.63% Dry	30.00°C	32.00%	26cm
106	2015-05-19 20:58:51	63.05% Dark	98.04% Dry	98.14% Dry	30.00°C	32.00%	26cm
105	2015-05-19 20:57:47	63.25% Dark	98.34% Dry	98.24% Dry	30.00°C	32.00%	26cm
104	2015-05-19 20:56:44	62.56% Dark	98.44% Dry	98.53% Dry	30.00°C	32.00%	26cm
103	2015-05-19 20:55:40	63.25% Dark	97.75% Dry	97.75% Dry	31.00°C	31.00%	26cm
102	2015-05-19 20:54:36	62.56% Dark	97.75% Dry	97.75% Dry	30.00°C	32.00%	26cm
101	2015-05-19 20:53:32	63.34% Dark	97.95% Dry	97.95% Dry	30.00°C	32.00%	26cm
100	2015-05-19 20:52:28	63.05% Dark	97.85% Dry	97.85% Dry	30.00°C	32.00%	26cm
99	2015-05-19 20:51:24	62.66% Dark	98.53% Dry	98.53% Dry	30.00°C	32.00%	26cm
98	2015-05-19 20:50:20	61.97% Dark	97.75% Dry	97.75% Dry	30.00°C	32.00%	26cm
97	2015-05-19 20:49:16	62.56% Dark	97.85% Dry	97.95% Dry	30.00°C	32.00%	26cm
96	2015-05-19 20:48:12	61.88% Dark	97.75% Dry	97.75% Dry	30.00°C	32.00%	26cm
95	2015-05-19 20:47:08	62.66% Dark	97.95% Dry	97.95% Dry	30.00°C	32.00%	26cm

Εικόνα 83: Σελίδα απεικόνισης μετρήσεων σε πίνακα.



Εικόνα 84: Κουμπί μεταφοράς στην σελίδα απεικόνισης μετρήσεων σε πίνακα.

Στη συνέχεια ακολουθεί ο κώδικας της σελίδας απεικόνισης μετρήσεων σε πίνακα. Το αρχείο ονομάζεται review_data.php και είναι αποθηκευμένο στο φάκελοhtdocs του τοπικού server.

```
<?php
    // Έναρξη σύνδεσης MySQL χρησιμοποιώντας το αρχείο
dbconnect.php
    include('dbconnect.php');
?>
<html>
    <head>
        <title>GreenHouse sensor measurements</title>

        <style type="text/css">
            .table_titles, .table_cells_odd, .table_cells_even {
                padding-right: 20px;
                padding-left: 20px;
                color: #000;
                text-align: center ;
            }
            .table_titles {
                color: #FFF;
                background-color: #666;
                text-align: center ;
            }
            .table_cells_odd {
                background-color: #CCC;
                text-align: center ;
            }
            .table_cells_even {
                background-color: #FAFAFA;
                text-align: center ;
            }
            table {
                border: 4px solid #333;
                text-align: center ;
                margin-left:290px;
            }
            body { font-family: "Trebuchet MS", Arial;
                text-align: center ;
                background-color:#228B22;
            }
            h1 {
                margin-top:10px;
                margin-right:280px;
                margin-bottom:40px;
                margin-left:280px;
                font-weight: bold;
                font-style:normal;
                text-decoration: none;
                background-color:#32CD32;
                color:white;
                text-shadow: 2px 2px 4px #000000;
                border-radius: 30px;
                padding-top:20px ;
                padding-bottom:20px ;
            }
        </style>
    </head>
    <body>
        <h1>GreenHouse sensor measurements</h1>
    </body>
</html>
```

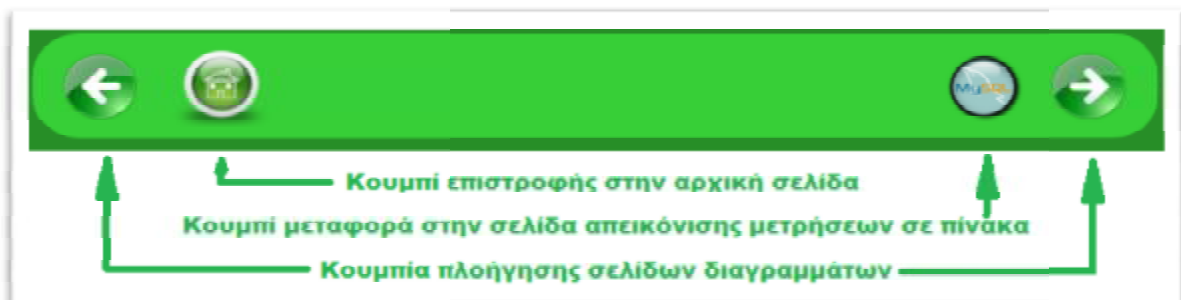
```

    }
  </style>
</head>
<body>
  <h1>GreenHouse sensor measurements</h1>
  <a href="home.html">
    
  </a>
  <table border="0" cellspacing="1" cellpadding="4"
style="margin-top: 10px;">
    <tr>
      <td class="table_titles">ID</td>
      <td class="table_titles">Date and Time</td>
      <td class="table_titles">Light Ambient</td>
      <td class="table_titles">Raindrop</td>
      <td class="table_titles">Moisture</td>
      <td class="table_titles">Temperature</td>
      <td class="table_titles">Humidity</td>
      <td class="table_titles">Tank Height</td>
    </tr>
    <?php
      //Ανάκτηση όλων των αρχείων και προβολή αυτών
      $result = mysql_query("SELECT * FROM sensors ORDER BY id DESC");
      //Επιλογή από πίνακα sensors ταξινομία με βάση το id απ' το
μεγαλύτερο προς το μικρότερο
      $oddrow = true; // Χρησιμοποιείται για την εναλλαγή χρώματος
σειράς
      while( $row = mysql_fetch_array($result) )
      {
        if ($oddrow)
        {
          $css_class=' class="table_cells_odd"';
        }
        else
        {
          $css_class=' class="table_cells_even"';
        }
        $oddrow = !$oddrow;
        echo '<tr>';
        echo ' <td' . $css_class . '>' . $row["id"] . '</td>';
        echo ' <td' . $css_class . '>' . $row["event"] . '</td>';
        echo ' <td' . $css_class . '>' . $row["light"] . '% Dark</td>';
        echo ' <td' . $css_class . '>' . $row["raindrop"] . '% Dry</td>';
        echo ' <td' . $css_class . '>' . $row["moisture"] . '% Dry</td>';
        echo ' <td' . $css_class . '>' . $row["temperature"] . '°C</td>';
        echo ' <td' . $css_class . '>' . $row["humidity"] . '%</td>';
        echo ' <td' . $css_class . '>' . $row["tank"] . 'cm</td>';
        echo '</tr>';
      }
    <?>
  </table>
</body>
</html>

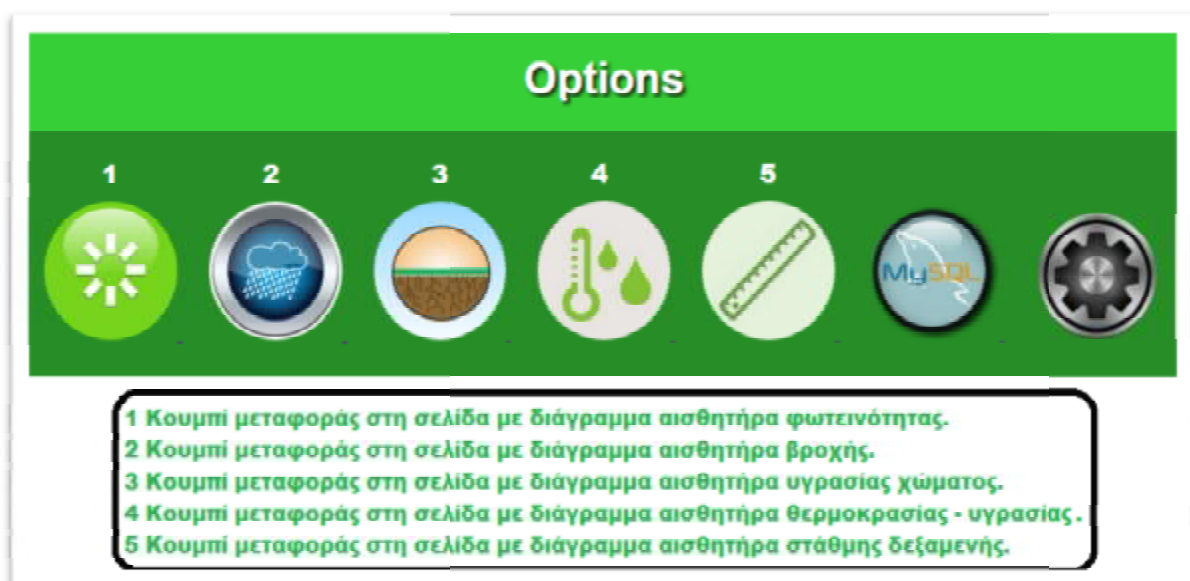
```

3.2.4 Σελίδες απεικόνισης μετρήσεων σε διαγράμματα

Εκτός από τον πίνακα που περιέχει όλες τις μετρήσεις που έχουν καταγραφεί, σχεδιάστηκαν και πέντε σελίδες με διαγράμματα. Μέσα από τον πίνακα ο χρήστης θα ήταν δύσκολο να παρατηρήσει τις μεταβολές των τιμών του κάθε αισθητήρα, ώστε να εξάγει χρήσιμα συμπεράσματα για τις περιβαλλοντικές συνθήκες που επικρατούν. Τα διαγράμματα αυτά οπτικοποιούν τις μετρήσεις του κάθε αισθητήρα σε σχέση με το χρόνο. Κάθε σελίδα αντιστοιχεί σε έναν αισθητήρα, εκτός από μία η οποία περιέχει διάγραμμα με δύο διαφορετικά μεγέθη. Στις σελίδες αυτές απεικονίζονται σε ένα διάγραμμα όλες οι μετρήσεις που έχουν καταγραφεί. Αυτό βέβαια δεν βοηθάει στην παρατήρηση των μεταβολών ενός συγκεκριμένου χρονικού διαστήματος, γι αυτό υπάρχει δυνατότητα επιλογής συγκεκριμένης περιόδου. Η μεταφορά στην κάθε σελίδα μπορεί να γίνει από την αρχική σελίδα πατώντας το αντίστοιχο κουμπί. Επίσης από τη στιγμή που ο χρήστης έχει μεταφερθεί σε μία σελίδα η οποία απεικονίζει ένα διάγραμμα, τότε μπορεί να πλοηγηθεί και στις υπόλοιπες με την βοήθεια των κουμπιών που φαίνονται στην εικόνα 85. Με τη βοήθεια αυτών των κουμπιών, ουσιαστικά αυτές οι πέντε σελίδες συνδέονται μεταξύ τους σε μία κυκλική σειρά. Τέλος υπάρχουν ακόμα δύο κουμπιά στην κάθε σελίδα, τα οποία χρησιμεύουν στον να μεταφερθεί ο χρήστης είτε στην αρχική σελίδα, είτε στη σελίδα απεικόνισης μετρήσεων σε πίνακα.

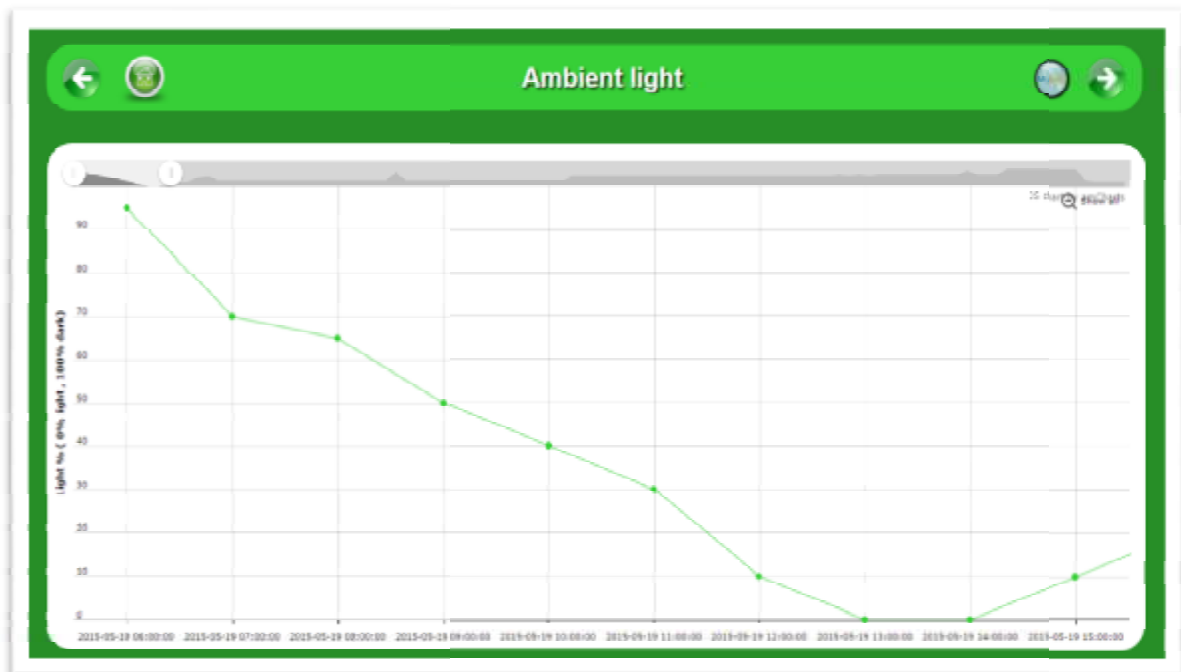


Εικόνα 85: Κουμπιά σελίδων διαγραμμάτων.



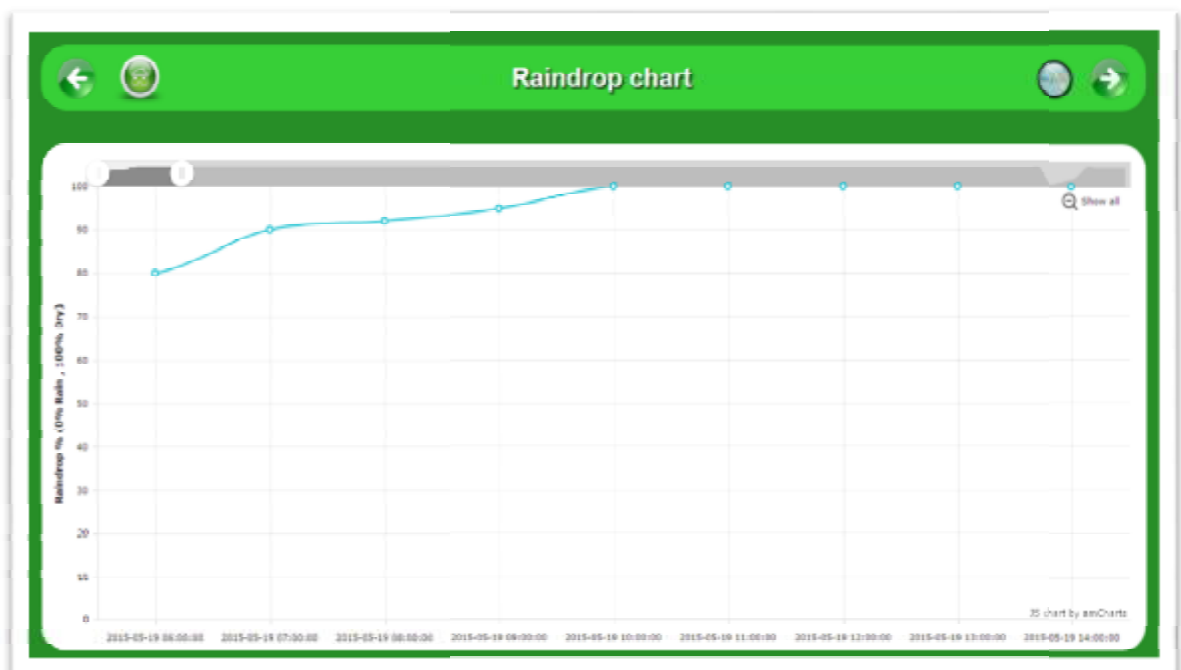
Εικόνα 86: Κουμπιά μεταφοράς στις σελίδες διαγραμμάτων.

Διάγραμμα απεικόνισης μετρήσεων αισθητήρα φωτεινότητας σε σχέση με το χρόνο:



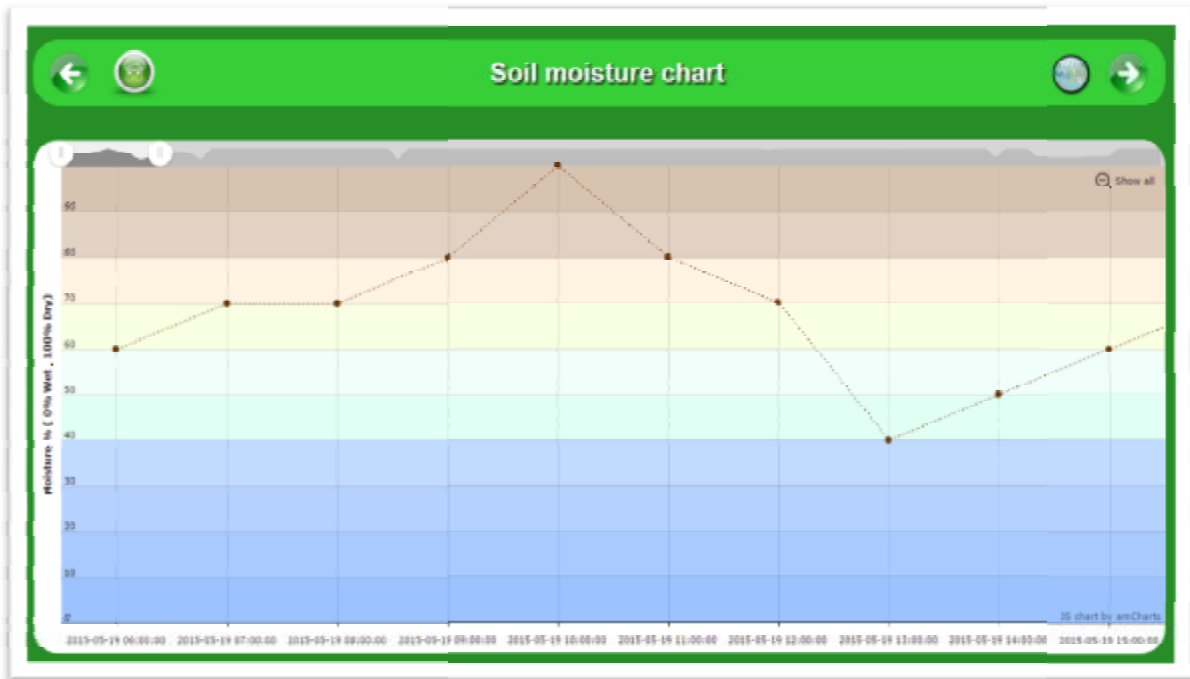
Εικόνα 87: Ιστοσελίδα απεικόνισης αισθητήρα φωτεινότητας σε διάγραμμα.

Διάγραμμα απεικόνισης μετρήσεων αισθητήρα βροχής σε σχέση με το χρόνο:



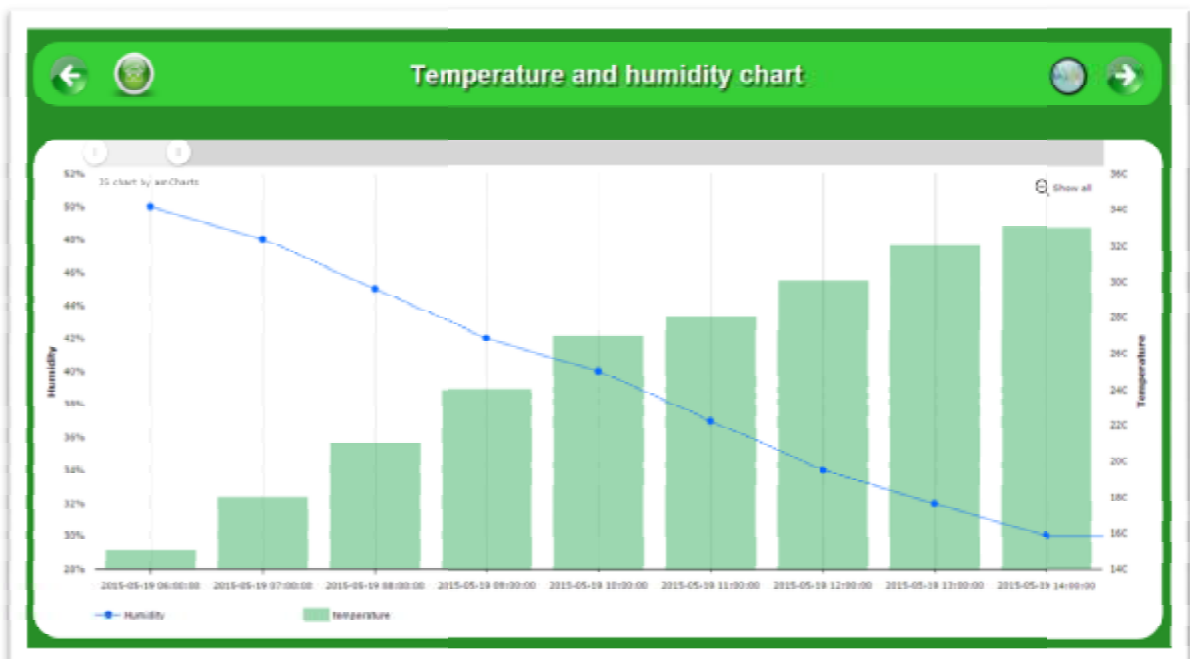
Εικόνα 88: Ιστοσελίδα απεικόνισης μετρήσεων βροχής σε διάγραμμα.

Διάγραμμα απεικόνισης μετρήσεων αισθητήρα υγρασίας χώματος σε σχέση με το χρόνο:



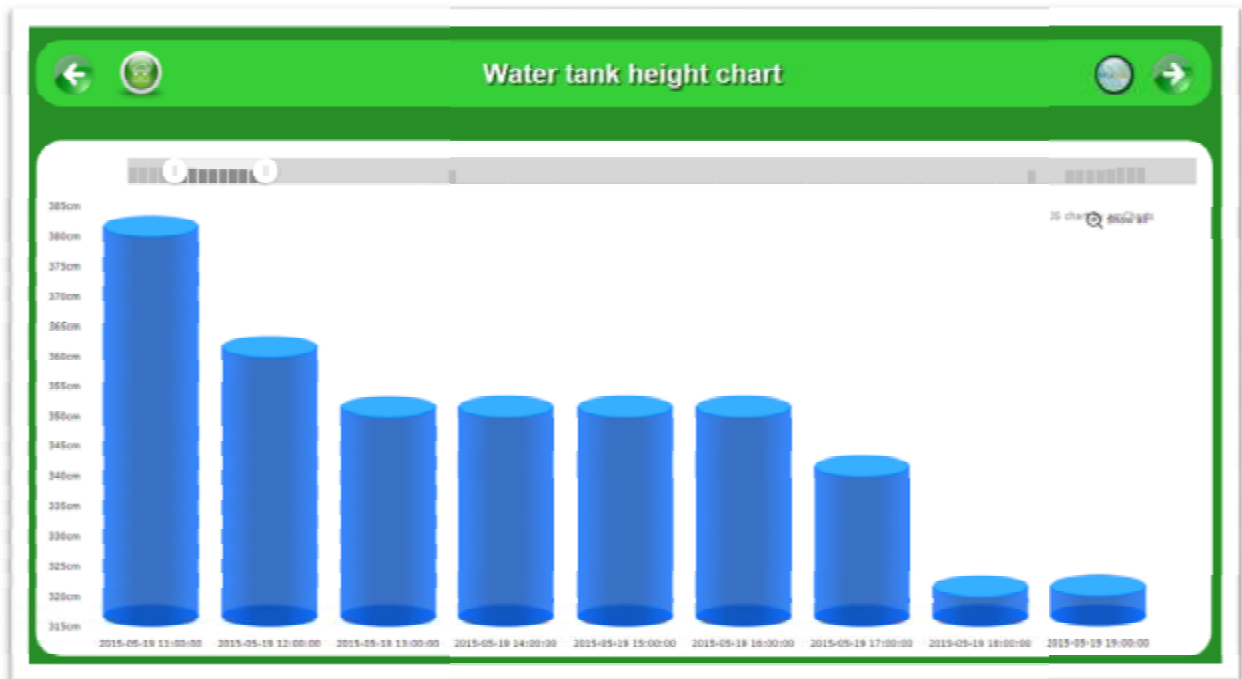
Εικόνα 89: Ιστοσελίδα απεικόνισης μετρήσεων υγρασίας χώματος σε διάγραμμα.

Διάγραμμα απεικόνισης μετρήσεων αισθητήρα θερμοκρασίας - υγρασίας σε σχέση με το χρόνο:



Εικόνα 90: Ιστοσελίδα απεικόνισης μετρήσεων θερμοκρασίας – υγρασίας σε διάγραμμα.

Διάγραμμα απεικόνισης μετρήσεων αισθητήρα μέτρησης ύψους δεξαμενής σε σχέση με το χρόνο:



Εικόνα 91: Ιστοσελίδα απεικόνισης μετρήσεων ύψους δεξαμενής σε διάγραμμα.

Ο κώδικας των ιστοσελίδων απεικόνισης μετρήσεων σε διαγράμματα για να διευκολύνει την ροή της εργασίας βρίσκεται στο παράρτημα Γ. Στον σχεδιασμό και προγραμματισμό των διαγραμμάτων, συνέβαλαν τα διαγράμματα της εταιρίας Amcharts, η οποία παρέχει δωρεάν τον κώδικα μιας μεγάλης ποικιλίας διαφορετικών τύπων διαγραμμάτων. Εκτός αυτού παρέχει αρκετές πληροφορίες – οδηγίες ούτως ώστε οι συγκεκριμένοι κώδικες να τροποποιηθούν ώστε να ανταποκρίνονται στις απαιτήσεις του κάθε χρήστη.



Εικόνα 92: Λογότυπο εταιρίας Amcharts.

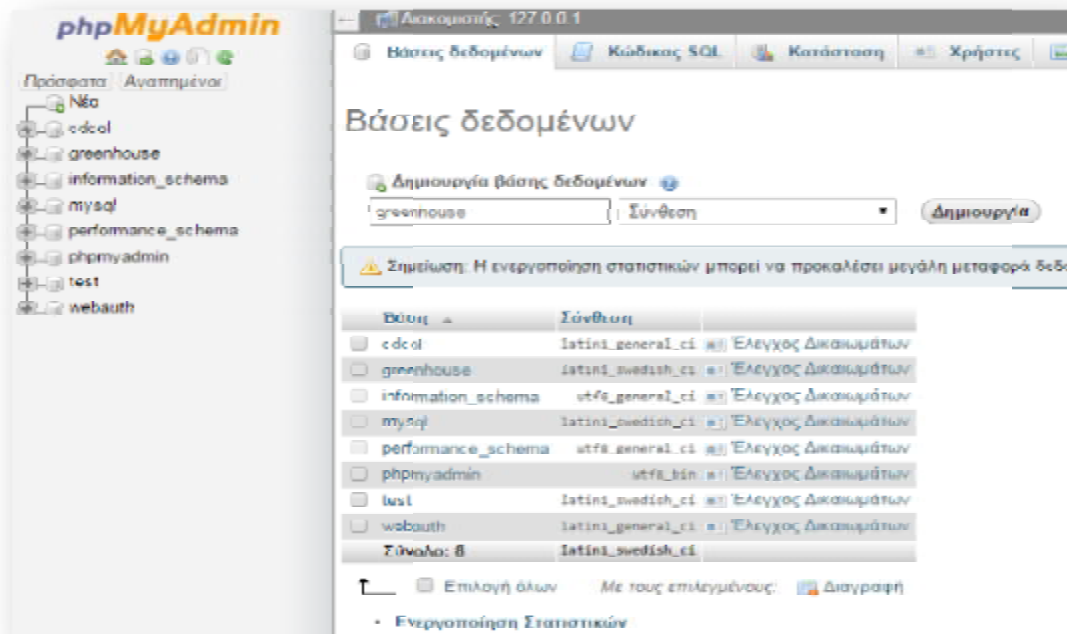
Αρχείο data_all.php

Το συγκεκριμένο αρχείο είναι βοηθητικό και συνεργάζεται με κάθε ένα αρχείο ιστοσελίδας με διάγραμμα. Οι ιστοσελίδες αυτού του είδους καλούν αυτό το αρχείο το οποίο είναι υπεύθυνο για την αποστολή των καταγεγραμμένων μετρήσεων από την βάση δεδομένων στις σελίδες απεικόνισης διαγραμμάτων. Ύστερα αυτές μετατρέπουν τα δεδομένα που τους παρέχει το αρχείο data_all.php σε διαγράμματα.

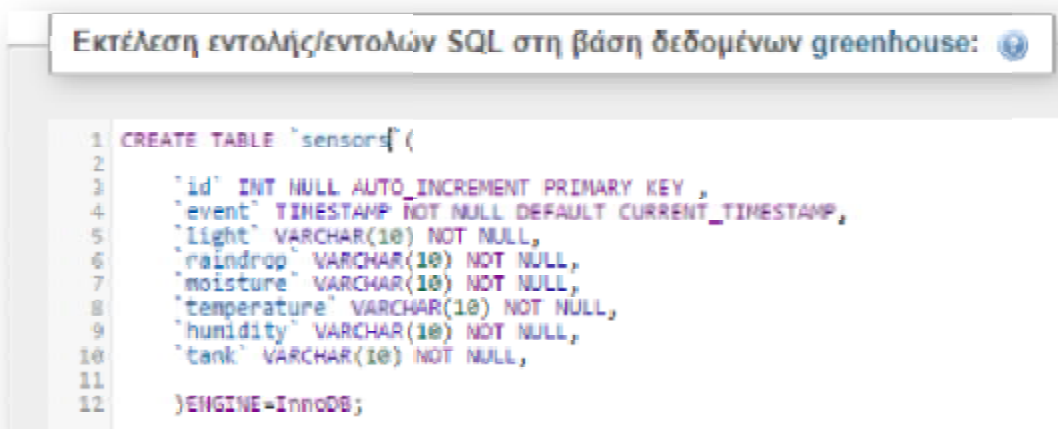
```
<?php
// Connect to MySQL
$link = mysql_connect( 'localhost', 'root', '' );
if ( !$link ) {
    die( 'Could not connect: ' . mysql_error() );
}
// Select the data base
$db = mysql_select_db( 'greenhouse', $link );
if ( !$db ) {
    die ( 'Error selecting database \'greenhouse\' : ' . mysql_error() );
}
// Fetch the data
$query = " SELECT * FROM sensors ORDER BY event ";
$result = mysql_query( $query );
// All good?
if ( !$result ) {
    // Nope
    $message = 'Invalid query: ' . mysql_error() . "\n";
    $message .= 'Whole query: ' . $query;
    die( $message );
}
// Print out rows
$prefix = '';
echo "\n";
while ( $row = mysql_fetch_assoc( $result ) ) {
    echo $prefix . " {\n";
    echo ' "event": "' . $row['event'] . ',' . "\n";
    echo ' "light": ' . $row['light'] . ',' . "\n";
    echo ' "raindrop": "' . $row['raindrop'] . ',' . "\n";
    echo ' "moisture": ' . $row['moisture'] . ',' . "\n";
    echo ' "temperature": "' . $row['temperature'] . ',' . "\n";
    echo ' "humidity": ' . $row['humidity'] . ',' . "\n";
    echo ' "tank": "' . $row['tank'] . ',' . "\n";
    echo " }";
    $prefix = ",\n";
}
echo "\n]";
// Close the connection
mysql_close($link);
?>
```

3.3 Βάση δεδομένων

Η βάση δεδομένων του ιστοχώρου που υλοποιήθηκε, έχει το όνομα “greenhouse” και αποτελείται από ένα πίνακα ο οποίος ονομάζεται “sensors”.Ο συγκεκριμένος πίνακας διαθέτει οκτώ στήλες που χρησιμοποιούνται για την καταγραφή των μετρήσεων. Για τη δημιουργία της βάσης δεδομένων, ανοίγουμε τη σελίδα phpmyadmin, φτιάχνουμε μία νέα βάση δεδομένων δίνοντας της ένα όνομα, και εκτελούμε το παρακάτω query ώστε να δημιουργηθεί ο πίνακας.



Εικόνα 93: Δημιουργία βάσης δεδομένων.



Εικόνα 94: Query δημιουργίας πίνακα.

Ο παρακάτω πίνακας χρησιμοποιείται για την καταχώριση των μετρήσεων των αισθητήρων και την καταγραφή της ώρας κατά την οποία έγινε εγγραφή. Ακολουθεί αναλυτική περιγραφή των πεδίων του:

Όνομα	Τύπος	Σύνθεση	Κενό	Προεπιλογή
id	int(11)		Όχι	Καμία
event	timestamp		Όχι	CURRENT_TIMESTAMP
light	varchar(10)	latin1_swedish_ci	Όχι	Καμία
raindrop	varchar(10)	latin1_swedish_ci	Όχι	Καμία
moisture	varchar(10)	latin1_swedish_ci	Όχι	Καμία
temperature	varchar(10)	latin1_swedish_ci	Όχι	Καμία
humidity	varchar(10)	latin1_swedish_ci	Όχι	Καμία
tank	varchar(10)	latin1_swedish_ci	Όχι	Καμία

Πίνακας 24: Πίνακας sensors βάσης δεδομένων.

- Ø **id:** Αποτελεί το αναγνωριστικό κάθε εγγραφής. Σε κάθε νέα εγγραφή που πραγματοποιείται έχει προγραμματιστεί να αυξάνεται κατά 1 (Πρόσθετη επιλογή AUTO_INCREMENT). Είναι ακέραιος έως 11 ψηφίων.
- Ø **event:** Γίνεται εγγραφή της ακριβούς ημερομηνίας και ώρας (π.χ. 2015-05-19 19:03:12) ώστε ο χρήστης να γνωρίζει ακριβώς πότε έγινε η κάθε καταγραφή δεδομένων.
- Ø **light:** Αποθηκεύει την τιμή της φωτεινότητας του περιβάλλοντος.
- Ø **raindrop:** Αποθηκεύει την τιμή του αισθητήρα βροχής.
- Ø **moisture:** Αποθηκεύει την τιμή της υγρασίας χώματος.
- Ø **temperature:** Αποθηκεύει την τιμή της θερμοκρασίας του αέρα.
- Ø **humidity:** Αποθηκεύει την τιμή της υγρασίας του αέρα.
- Ø **tank:** Αποθηκεύει την τιμή του ύψους της δεξαμενής νερού.

id	event	light	raindrop	moisture	temperature	humidity	tank
1	2015-05-19 19:02:08	363	1009	1009	29.00	32.00	26
2	2015-05-19 19:03:12	358	1016	1016	30.00	32.00	26
3	2015-05-19 19:04:17	319	1010	1010	30.00	32.00	47
4	2015-05-19 19:05:21	358	1001	1001	30.00	32.00	26
5	2015-05-19 19:06:25	345	1002	1002	30.00	32.00	26
6	2015-05-19 19:07:29	362	1013	1013	30.00	32.00	26
7	2015-05-19 19:08:32	355	1008	1008	30.00	32.00	26
8	2015-05-19 19:09:36	347	1010	1010	30.00	32.00	26
9	2015-05-19 19:10:40	356	1013	1013	30.00	32.00	26
10	2015-05-19 19:15:00	34.21	99.12	99.12	30.00	32.00	26

Εικόνα 95: Πίνακας sensors βάσης δεδομένων.

Επικοινωνία βάσης δεδομένων

Η βάση δεδομένων επικοινωνεί με το Arduino χρησιμοποιώντας αρχεία php, ούτως ώστε να λάβει και να καταγράψει τα δεδομένα. Τα συγκεκριμένα αρχεία βρίσκονται αποθηκευμένα στο φάκελο με τα υπόλοιπα αρχεία του τοπικού server. Παρακάτω παρουσιάζονται τα αρχεία αυτά και αναλύεται ο τρόπος συνεργασίας τους.

Αρχείο dbconnect.php.

Χρησιμοποιείται για να αρχίσει η επικοινωνία με τη βάση δεδομένων greenhouse.

```
<?php
$MyUsername = "root"; // Εισαγωγή ονόματος χρήστη για την βάση
mysql
$MyPassword = ""; // Εισαγωγή κωδικού για την βάση mysql
$MyHostname = "localhost"; // Για τοπικό server Xampp localhost
$dbh = mysql_pconnect($MyHostname , $MyUsername, $MyPassword);
$selected = mysql_select_db("greenhouse",$dbh); // Σύνδεση με τη βάση
δεδομένων greenhouse
?>
```

Αρχείο add_data.php.

Το συγκεκριμένο αρχείο συνεργάζεται με το αρχείο dbconnect.php, ούτως ώστε να αρχίσει η επικοινωνία με την βάση δεδομένων, και στη συνέχεια «τραβάει» τα δεδομένα των αισθητήρων από το Arduino και τα αποθηκεύει στη βάση δεδομένων.

```
<?php
// Σύνδεση με τη βάση δεδομένων χρησιμοποιώντας το αρχείο
dbconnect.php
include("dbconnect.php");
//Λήψη δεδομένων και αποθήκευση αυτών στη βάση δεδομένων
$SQL = "INSERT INTO greenhouse.sensors (
light ,
raindrop,
moisture,
temperature,
humidity,
tank)
VALUES ('".$_GET["light"]."',
'".$_GET["raindrop"]."',
'".$_GET["moisture"]."',
'".$_GET["temperature"]."',
'".$_GET["humidity"]."',
'".$_GET["tank"]."')";

// Εκτέλεση δήλωσης SQL
mysql_query($SQL);
?>
```

3.4 Προγραμματισμός Arduino Redboard

Η παρούσα ενότητα αφορά την ανάλυση του τρόπου λειτουργίας της συσκευής. Πιο συγκεκριμένα το πώς συμπεριφέρεται το σύστημα ανάλογα με τις περιβαλλοντικές συνθήκες. Θα γίνει περιγραφή του τρόπου επικοινωνίας με τον ιστοχώρο και τη βάση δεδομένων. Τέλος θα γίνει εμβάθυνση στον κώδικα με τον οποίο προγραμματίστηκε το Arduino, παρουσιάζοντας τις βασικότερες συναρτήσεις, τεχνικές και εργαλεία που χρησιμοποιήθηκαν, με σκοπό την καλύτερη κατανόηση του κώδικα. Στις παρακάτω υποενότητες θα γίνει αναφορά σε βασικά μέρη του κώδικα προγραμματισμού του Arduino, ο οποίος υπάρχει στο παράρτημα Α ολόκληρος.

3.4.1 Βασικές βιβλιοθήκες

Η αρχιτεκτονική Arduino βασίζεται για τον προγραμματισμό της στη γλώσσα C. Αυτό συνεπάγεται αρκετές βιβλιοθήκες και συναρτήσεις για την υλοποίηση κώδικα. Ακόμα εξ αιτίας της ιδιαιτερότητας του προγραμματισμού που απαιτεί αυτή η αρχιτεκτονική, προσφέρει μερικά πρόσθετα στοιχεία. Στη συνέχεια γίνεται μία ανάλυση των στοιχείων που χρησιμοποιήθηκαν προγραμματιστικά, με σκοπό την υλοποίηση του κώδικα του συστήματος εποπτείας και ελέγχου, τα οποία αφορούν τον προγραμματισμό του Arduino, των αισθητήρων και των συσκευών δράσης.

Βιβλιοθήκη Ethernet

Η συγκεκριμένη βιβλιοθήκη καθιστά δυνατή τη σύνδεση μιας πλακέτας Arduino με το διαδίκτυο, μέσω του Ethernet shield. Έχει τη δυνατότητα λειτουργίας είτε ως διακομιστής (server) ο οποίος δέχεται εισερχόμενες συνδέσεις, είτε ως πελάτης (client). Η βιβλιοθήκη παρέχει τη δυνατότητα μέχρι και τεσσάρων ταυτόχρονων συνδέσεων, οι οποίες μπορεί να είναι είτε εισερχόμενες, είτε εξερχόμενες ή ακόμα και συνδυασμοί των παραπάνω. Ακολουθεί μία λεπτομερειακή αναφορά στις εντολές και τις δομές, των οποίων έγινε χρήση:

Ø **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 του προγράμματος σημαντικά.

Σύνταξη: Ethernet.begin (mac, ip, dns, gateway, subnet);

Παράμετροι: mac – η mac (Media Access Control) διεύθυνση της συσκευής (πίνακας 6 bytes) αποτελεί την Ethernet διεύθυνση του υλικού του shield, 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).

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

Σύνταξη: Ethernet.localIP()

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

Σύνταξη: IPAddress (address);

Παράμετροι: address – μία λίστα οριοθετημένη με κόμματα που αντιπροσωπεύει τη διεύθυνση (4 bytes).

- Ø **Server ()**: Δημιουργεί ένα διακομιστή που ακούει για εισερχόμενες συνδέσεις στη συγκεκριμένη θύρα.
Σύνταξη: Server(port);
Παράμετροι: port – Η θύρα στην οποία ακούει (int).
- Ø **begin()**: Ξεκινά η διαδικασία από την πλευρά του διακομιστή για εύρεση νέων συνδέσεων.
Σύνταξη: server.available();
- Ø **available ()**: Για server - 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. Για client - 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: server.available();client.available();
- Ø **write ()**: Γράφει τα δεδομένα για όλους τους πελάτες που συνδέονται σε ένα διακομιστή.
Σύνταξη: client.write(data); , server.write(data);
Παράμετροι: data - Η τιμή που γράφει (byte ή char).
- Ø **print () – println ()**: Εκτυπώνει δεδομένα σε όλους τους πελάτες που συνδέονται σε έναν διακομιστή. Η println() προσθέτει και μια αλλαγή γραμμής.
Σύνταξη: client.print(); και client.println(); ή server.print(data); και server.println(data);
Παράμετροι: data – τα δεδομένα προς εκτύπωση (byte ,char ,int ,long ,string).
- Ø **Client ()**: Είναι η βασική κλάση για όλες τις Ethernet client συναρτήσεις. Δεν καλείται απευθείας, αλλά περιέχεται στην κλήση συνάρτησης που αναφέρεται σε αυτή. Δημιουργεί έναν πελάτη ο οποίος μπορεί να συνδεθεί στο διαδίκτυο μέσω μίας 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).
- Ø **connected ()**: Το αποτέλεσμα της συνάρτησης αυτής είναι η σύνδεση ή μη του πελάτη.
Σύνταξη : client.connected();
- Ø **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();
- Ø **write ()**: Γράφει δεδομένα στο διακομιστή στον οποίο ο πελάτης είναι συνδεδεμένος.
Σύνταξη: client.write(data);
Παράμετροι: data - η τιμή που πρόκειται να γραφτεί.
- Ø **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();
- Ø **read ()**: Διαβάζει τα επόμενα bytes που λαμβάνονται από το διακομιστή στον οποίο είναι συνδεδεμένος ο πελάτης.

Σύνταξη: client.read();

- Ø **flush ()**: Απορρίπτει τα bytes που έχουν γραφτεί για τον πελάτη, αλλά δεν έχουν ακόμη διαβαστεί.

Σύνταξη: client.flush();

- Ø **stop ()**: Αποσυνδέεται από το διακομιστή.

Σύνταξη: client.stop();

- Ø **status ()**: Επιστρέφει ένα ακέραιο αριθμό, ο οποίος δείχνει την κατάσταση στην οποία βρίσκεται το http ερώτημα που έγινε.

Σύνταξη: client.status();

Βιβλιοθήκη SPI

Η βιβλιοθήκη αυτή παρέχει τη δυνατότητα επικοινωνίας μεταξύ άλλων SPI συσκευών, 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.

Βιβλιοθήκη SD

Η βιβλιοθήκη SD επιτρέπει την ανάγνωση και εγγραφή σε κάρτες SD. Η βιβλιοθήκη υποστηρίζει FAT16 και FAT32 συστήματα με τις τυποποιημένες κάρτες SD και SDHC. Χρησιμοποιεί σύντομες ονομασίες 8.3 αρχείων. Τα ονόματα των αρχείων που χρησιμοποιούνται στις λειτουργίες της βιβλιοθήκης SD μπορεί να περιλαμβάνουν διαδρομές που διαχωρίζονται από τα εμπρός-καθέτους, /, π.χ. "κατάλογο / filename.txt". Από την έκδοση 1.0, η βιβλιοθήκη υποστηρίζει το άνοιγμα πολλαπλών αρχείων. Η επικοινωνία μεταξύ του μικροελεγκτή και της κάρτας SD χρησιμοποιεί SPI, η οποία χρησιμοποιεί τις ψηφιακές ακίδες 11, 12, και 13 (για τις περισσότερες εκδόσεις Arduino). Επιπλέον, μία ακόμα ακίδα πρέπει να χρησιμοποιηθεί για να επιλεγεί η κάρτα SD. Αυτή μπορεί να είναι η ακίδα SS (pin 10) η οποία θα πρέπει να οριστεί ως έξοδος, ειδάλλως δεν θα λειτουργήσει η βιβλιοθήκη SD. Ακολουθεί μία λεπτομερειακή αναφορά στις εντολές των οποίων έγινε χρήση:

- Ø **SD.begin ()**: Προετοιμάζει την βιβλιοθήκη SD αλλά και την sd κάρτα. Αρχίζει η χρησιμοποίηση του SPI διαύλου επικοινωνίας και της επιλεγμένης ακίδας.

Σύνταξη: SD.begin (pin);

- Ø **SD.open ()**: Ανοίγει ένα αρχείο από την sd κάρτα.

Σύνταξη: SD.open (filepath);

Βιβλιοθήκη DHT

- Ø Η βιβλιοθήκη dht χρησιμοποιείται για την ανάγνωση των μετρήσεων μίας συγκεκριμένης σειράς αισθητήρων θερμοκρασίας και υγρασίας. Μεταφράζει το αναλογικό σήμα που δέχεται σε πραγματική τιμή θερμοκρασίας και υγρασίας. Ακολουθεί η εντολή που χρησιμοποιήθηκε:

- Ø **DHT.read11()**: Διαβάζει την αναλογική είσοδο μίας ακίδας και δίνει αυτόματα στις μεταβλητές DHT.temperature και DHT.humidity τις αντίστοιχες τιμές θερμοκρασίας και υγρασίας.

Σύνταξη: DHT.read11(pin);

3.4.2 Προγραμματισμός αισθητήρων

Σε προηγούμενη ενότητα στο κεφάλαιο δύο έγινε περιγραφή του τρόπου προγραμματισμού του κάθε αισθητήρα ξεχωριστά. Σε αυτή την ενότητα θα γίνει μία παρόμοια ανάλυση με βάση τον κώδικα που χρησιμοποιήθηκε στον προγραμματισμό του συστήματος.

Για τους αισθητήρες φωτεινότητας, βροχής και υγρασίας χρώματος χρησιμοποιήθηκαν απευθείας οι αναλογικές τιμές εισόδου. Για να εμφανίζεται η τιμή τους σαν ποσοστό επί τις εκατό πραγματοποιήθηκε μια απλή πράξη. Για να γίνει αυτό χρειάζεται η αναλογική είσοδος να διαιρεθεί με τον αριθμό 10.23. Στη συνέχεια υπάρχει ο κώδικας που χρησιμοποιήθηκε από το πρόγραμμα όταν χρειαζόταν να κάνει οποιαδήποτε ενέργεια που περιλαμβάνει τις τιμές των τριών αισθητήρων.

```
(analogRead(2))/10.23// Αισθητήρας φωτεινότητας  
(analogRead(3))/10.23// Αισθητήρας βροχής  
(analogRead(4))/10.23// Αισθητήρας υγρασίας
```

Όσον αφορά τον αισθητήρα θερμοκρασίας – υγρασίας με τη βοήθεια της βιβλιοθήκης DHT η ανάγνωση των μετρήσεων όπως φαίνεται στη συνέχεια είναι πολύ εύκολη.

```
DHT.read11(dht_dpin);
```

Ύστερα από την εκτέλεση αυτής της εντολής οι παρακάτω μεταβλητές παίρνουν τις αντίστοιχες τιμές.

```
DHT.temperature  
DHT.humidity
```

Η ανάγνωση της απόστασης από τον αισθητήρα υπερήχων είναι λίγο πιο πολύπλοκη. Η διαδικασία μετατροπής των ψηφιακών εισόδων σε απόσταση έχει αναλυθεί σε προηγούμενη ενότητα. Το πρόγραμμα όταν χρειαστεί να χρησιμοποιήσει τη μέτρηση του αισθητήρα απόστασης χρησιμοποιεί τη μεταβλητή cm. Στη συνέχεια υπάρχει ο κώδικας που χρησιμοποιήθηκε στην υλοποίηση του συστήματος.

```
const int trigPin = 2;  
const int echoPin = 3;  
long duration, cm;  
pinMode(trigPin, OUTPUT);  
digitalWrite(trigPin, LOW);  
delayMicroseconds(2);  
digitalWrite(trigPin, HIGH);  
delayMicroseconds(10);  
digitalWrite(trigPin, LOW);  
pinMode(echoPin, INPUT);  
duration = pulseIn(echoPin, HIGH);  
cm = microsecondsToCentimeters(duration);  
long microsecondsToCentimeters(long microseconds) {  
    return microseconds / 29 / 2;  
}
```

3.4.3 Προγραμματισμός συσκευών δράσης

Οι συσκευές δράσης ενεργοποιούνται με δύο τρόπους όπως έχει προαναφερθεί. Ο πιο απλός τρόπος αφορά την ενεργοποίηση τους από τα κουμπιά στον πίνακα ελέγχου. Αυτό επιτυγχάνεται στέλνοντας στην ψηφιακή έξοδο που συνδέεται κάθε συσκευή, είτε λογικό 1 (υψηλή στάθμη) για ενεργοποίηση, είτε λογικό 0 (χαμηλή στάθμη) για απενεργοποίηση. Ο δεύτερος τρόπος ενεργοποίησης των συσκευών δράσης αφορά τα αυτόματα σενάρια. Η μόνη διαφορά από τον προηγούμενο είναι ότι έχει προστεθεί στον κώδικα μία εντολή ελέγχου ροής (συγκεκριμένα η if), με την βοήθεια της οποίας γίνεται έλεγχος των τιμών των αισθητήρων. Αν οι τιμές τους ικανοποιούν την συνθήκη, στη συνέχεια ενεργοποιείται η αντίστοιχη συσκευή δράσης. Ακολουθεί μέρος του κώδικα προγραμματισμού του Arduino.

Κατάσταση	Συσκευή	Σενάριο	Σχόλια
state[0]	Ανεμιστήρας	Αυτόματο	Τιμή = 1 Ενεργοποίηση σεναρίου
state[1]	Ηλεκτροβάνα	Αυτόματο	Τιμή = 1 Ενεργοποίηση σεναρίου
state[2]	Ανεμιστήρας	Χειροκίνητο	Τιμή = 1 Ενεργοποίηση συσκευής
state[3]	Ηλεκτροβάνα	Χειροκίνητο	Τιμή = 1 Ενεργοποίηση συσκευής

Πίνακας 25: Πίνακας καταστάσεων.

Το συγκεκριμένο μέρος του κώδικα βρίσκεται μέσα στο Void loop, δηλαδή το κυρίως πρόγραμμα που εκτελείται συνέχεια. Αφορά μόνο την ενεργοποίηση των συσκευών δράσης όταν έχει οριστεί ότι το σύστημα λειτουργεί με ένα τουλάχιστον αυτόματο σενάριο. Ο κώδικας αυτός ελέγχει εάν έχει ενεργοποιηθεί κάποιο σενάριο, και εν συνεχεία αν ισχύει αυτό εξετάζει τις τιμές των αντίστοιχων αισθητήρων, ώστε να προβεί σε ενεργοποίηση ή απενεργοποίηση μίας συσκευής.

Πρώτο σενάριο: Ενεργοποίηση του ανεμιστήρα εφόσον η τιμή της θερμοκρασίας ξεπεράσει τους 35 °C. Ο ανεμιστήρας θα απενεργοποιηθεί όταν η θερμοκρασία κατέβει κάτω από τους 35 °C.

```
if (state[0]) { //Έλεγχος αυτόματου σεναρίου ανεμιστήρα
    if((DHT.temperature)>35){ // Έλεγχος τιμής αισθητήρα
        digitalWrite(8, HIGH);} //Ενεργοποίηση ανεμιστήρα
    else{digitalWrite(8,LOW);} //Απενεργοποίηση ανεμιστήρα
} //Τέλος ελέγχου αυτόματου σεναρίου ανεμιστήρα
```

Δεύτερο σενάριο: Ενεργοποίηση της ηλεκτροβάνας εφόσον η τιμή της υγρασίας χώματος ξεπεράσει το αριθμό 700(το οποίο σημαίνει ξερό έδαφος). Η ηλεκτροβάνα θα απενεργοποιηθεί όταν η τιμή της υγρασίας χώματος γίνει μικρότερη από τον αριθμό 700.

```
if (state[1]) { //Έλεγχος αυτόματου σεναρίου ηλεκτροβάνας
    if(analogRead(4)>700){ // Έλεγχος τιμής ηλεκτροβάνας
        digitalWrite(9, HIGH);} //Ενεργοποίηση ηλεκτροβάνας
    else{digitalWrite(9,LOW);} //Απενεργοποίηση ηλεκτροβάνας
} //Τέλος ελέγχου αυτόματου σεναρίου ηλεκτροβάνας
```

Το επόμενο κομμάτι κώδικα αφορά την υπορουτίνα Set, η οποία επιτρέπει την ενεργοποίηση – απενεργοποίηση συσκευών και σεναρίων. Η συγκεκριμένη υπορουτίνα ελέγχει εάν υπάρχει κάποιο http αίτημα, και εφόσον ανιχνευθεί κάτι τέτοιο, ενεργεί ανάλογο.

```

void Set(void) {
    if (StrContains(HTTP_req, "AFAN=1")) {
        state[0] = 1; // Αποθηκεύει την κατάσταση του σεναρίου
        αυτομάτου αερισμού
    }
    else if (StrContains(HTTP_req, "AFAN=0")) {
        state[0] = 0; // Αποθηκεύει την κατάσταση του σεναρίου
        αυτομάτου αερισμού
        digitalWrite(8, LOW);
    }
    if (StrContains(HTTP_req, "ASOLENOID=1")) {
        state[1] = 1; // Αποθηκεύει την κατάσταση του σεναρίου
        αυτομάτου ποτίσματος
    }
    else if (StrContains(HTTP_req, "ASOLENOID=0")) {
        state[1] = 0; // Αποθηκεύει την κατάσταση του σεναρίου
        αυτομάτου ποτίσματος
    }
    if (StrContains(HTTP_req, "MFAN=1")) {
        state[2] = 1; // Αποθηκεύει την κατάσταση του ανεμιστήρα
        digitalWrite(8, HIGH);
    }
    else if (StrContains(HTTP_req, "MFAN3=0")) {
        state[2] = 0; // Αποθηκεύει την κατάσταση του ανεμιστήρα
        digitalWrite(8, LOW);
    }
    if (StrContains(HTTP_req, "MSOLENOID=1")) {
        state[3] = 1; // Αποθηκεύει την κατάσταση της ηλεκτροβάνας
        digitalWrite(9, HIGH);
    }
    else if (StrContains(HTTP_req, "MSOLENOID4=0")) {
        state[3] = 0; // Αποθηκεύει την κατάσταση της ηλεκτροβάνας
        digitalWrite(9, LOW);
    }
}
}

```

3.4.4 Αποστολή μετρήσεων στη βάση δεδομένων

Το Arduino έχει προγραμματιστεί ώστε να στέλνει τις μετρήσεις στη βάση δεδομένων ανά τακτά χρονικά διαστήματα. Αυτό επιτυγχάνεται με τη βοήθεια της υπορουτίνας db. Η συγκεκριμένη υπορουτίνα καλείται συνέχεια από το κυρίως πρόγραμμα, αλλά εξαιτίας ενός μετρητή που περιέχεται σε αυτή δεν εκτελείται συνέχεια. Ο συγκεκριμένος μετρητής δίνει τη δυνατότητα ελέγχου των χρονικών διαστημάτων που μεσολαβούν ανάμεσα σε κάθε αποστολή δεδομένων. Θα μπορούσε να χρησιμοποιηθεί μία βιβλιοθήκη που μετράει το χρόνο, ώστε να ελέγχονται αυτά τα χρονικά διαστήματα, αλλά λόγω του μεγέθους της επιλέχθηκε ο συγκεκριμένος τρόπος. Στον κώδικα που ακολουθεί έχει οριστεί ο μετρητής να φτάνει στην τιμή 10000 και ύστερα να μηδενίζει, το οποίο αντιστοιχεί σε αποστολή δεδομένων ανά 10 λεπτά.

```
void db () { //Υπορουτίνα επικοινωνίας με βάση δεδομένων
    countdb = (countdb + 1) ;//Αύξηση μετρητή (αφορά την συχνότητα
εγγραφών στη βάση δεδομένων)
    if (countdb >= 10000) {
        if (client.connect(server, 80)) {
            Serial.println("-> Connected");
            client.print( "GET /arduino/add_data.php?");
            client.print("&&");
            client.print("light=");
            client.print( (analogRead(2))/10.23 );
            client.print("&&");
            client.print("raindrop=");
            client.print( (analogRead(3))/10.23 );
            client.print("&&");
            client.print("moisture=");
            client.print( (analogRead(4))/10.23 );
            client.print("&&");
            client.print("temperature=");
            client.print( DHT.temperature );
            client.print("&&");
            client.print("humidity=");
            client.print( DHT.humidity );
            client.print("&&");
            client.print("tank=");
            client.print( cm );
            countdb = 0 ;
            client.println( " HTTP/1.1");
            client.print( "Host: " );
            client.println(server);
            client.println( "Connection: close" );
            client.println();
            client.println();
            client.stop();
        }
        else {
            Serial.println("--> connection failed/n");
        }
    }
}
```

3.4.5 Επικοινωνία Arduino με ιστοσελίδα εποπτείας και ελέγχου

Σε αυτή την υποενότητα παρουσιάζεται ο κώδικας ο οποίος χρησιμοποιείται για τη σύνδεση του μικροελεγκτή με την ιστοσελίδα εποπτείας και ελέγχου.

```
if (client) { // Έλεγχος για εύρεση client
    boolean currentLineIsBlank = true;
    while (client.connected()) {
        if (client.available()) { // Έλεγχος για το αν είναι
            // Διαθέσιμα τα δεδομένα του client
            char c = client.read(); // Διαβάζει 1 byte
            // (χαρακτήρα) από τον client
            if (req_index < (REQ_BUF_SZ - 1)) {
                HTTP_req[req_index] = c; // Αποθηκεύει
                // το HTTP αίτημα
                req_index++;
            }
            if (c == '\n' && currentLineIsBlank) {
                client.println("HTTP/1.1 200 OK");
                if (StrContains(HTTP_req, "GET /logo.jpg")) {
                    webFile = SD.open("logo.jpg");
                    if (webFile) {
                        client.println("HTTP/1.1 200 OK");
                        client.println();
                    }
                    if (webFile) {
                        while(webFile.available()) {
                            client.write(webFile.read()); //
                            // Στέλνει την ιστοσελίδα στον client
                        }
                        webFile.close();
                    }
                }
                if (StrContains(HTTP_req, "ajax_inputs")) {
                    client.println("Content-Type:
                    text/xml");
                    client.println("Connection: keep-
                    alive");
                    client.println();
                    Set();
                    XML_response(client);
                }
                else {
                    client.println("Content-Type:
                    text/html");
                    client.println("Connection: keep-
                    alive");
                    client.println();
                    webFile = SD.open("index.htm");
                }
            }
        }
    }
}
// Άνοιγμα αρχείου ιστοσελίδας
```

```

        if (webFile) {
            while(webFile.available()) {
                client.write(webFile.read()); //
Αποστολή ιστοσελίδας στον client
            }
            webFile.close();
        }
    }
    Serial.print(HTTP_req);
    req_index = 0;
    break;
}
if (c == '\n') {
    currentLineIsBlank = true;
}
else if (c != '\r') {
    currentLineIsBlank = false;
}
} //Τέλος if (client.available())
} // Τέλος του while (client.connected())
delay(1); // Καθυστέρηση η οποία δίνει χρόνο στον
περιηγητή να λάβει τα δεδομένα
client.stop(); // Κλείνει η σύνδεση
} // Τέλος if (client)
}

```

Αφού γίνει η σύνδεση καλείται η υπορουτίνα XML_response η οποία ευθύνεται για την αποστολή δεδομένων από τον μικροελεγκτή στην ιστοσελίδα. Το πρώτο μέρος του κώδικα της υπορουτίνας αφορά την αποστολή των μετρήσεων του κάθε αισθητήρα, ενώ το δεύτερο μέρος έχει να κάνει με την κατάσταση των συσκευών δράσης.

```

void XML_response(EthernetClient cl) {
    int analog_val; // stores value read from analog
inputs
    int count; // used by 'for' loops
    cl.print("<?xml version = \"1.0\" ?>");
    cl.print("<inputs>");
    for (count = 2; count <= 4; count++) { // A2 to A5
        analog_val = (analogRead(count));
        cl.print("<analog>");
        cl.print((analog_val)/10.23);
        cl.println("</analog>");
    }
    cl.print("<analog>");
    cl.print(DHT.temperature);
    cl.print("</analog>");
    cl.print("<analog>");
    cl.print(DHT.humidity);
    cl.println("</analog>");
    cl.print("<analog>");
    cl.print(cm);
    cl.println("</analog>");
    cl.print("<DEVICE>");
    if (state[0]) {

```



```

        cl.print("checked");
    }
    else {
        cl.print("unchecked");
    }
    cl.println("</DEVICE >");
    cl.print("<DEVICE >");
    if (state[1]) {
        cl.print("checked");
    }
    else {
        cl.print("unchecked");
    }
    cl.println("</DEVICE >");
    cl.print("<DEVICE >");
    if (state[2]) {
        cl.print("on");
    }
    else {
        cl.print("off");
    }
    cl.println("</DEVICE >");
    cl.print("<DEVICE >");
    if (state[3]) {
        cl.print("on");
    }
    else {
        cl.print("off");
    }
    cl.println("</DEVICE >");
    cl.print("</inputs>");
}

```

Κεφάλαιο 4: Συμπεράσματα – μελλοντικές επεκτάσεις.

Στην πτυχιακή αυτή εργασία μετά την ανάλυση των απαιτήσεων και την παράλληλη έρευνα για την επιλογή των κατάλληλων εργαλείων ανάπτυξης, έγινε περιγραφή του σχεδιασμού της κατασκευής, της υλοποίησης και της λειτουργίας ενός συστήματος εποπτείας και ελέγχου ενός θερμοκηπίου, πάνω από το διαδίκτυο, το οποίο βασίζεται στην αναπτυξιακή πλατφόρμα Arduino.

Το σύστημα το οποίο σχεδιάστηκε απαιτεί πολύ χαμηλό και προσιτό κόστος σε σύγκριση με άλλα ολοκληρωμένα συστήματα εποπτείας και ελέγχου ενός χώρου που είναι διαθέσιμα στο εμπόριο. Το χαμηλό κόστος υλοποίησης αυτού του συστήματος σε συνδυασμό με τη διαδικτυακή υποστήριξη και σύνδεση σε server καθιστούν μία καλή επιλογή για τη συγκεκριμένη εφαρμογή.

Το συγκεκριμένο σύστημα, παρέχει στο χρήστη τη δυνατότητα μελέτης των περιβαλλοντικών συνθηκών ενός θερμοκηπίου αλλά και αλληλεπίδρασης με αυτό, αρκεί ο χώρος εγκατάστασης να καλύπτεται από ενσύρματη υποδομή πρωτοκόλλου 802.2. Τα οφέλη χρήσης είναι σημαντικά, τα κυριότερα το μέγεθός του, το οποίο είναι ιδιαίτερα μικρό, επίσης οι απαιτήσεις του σε ενέργεια αλλά και σε τεχνογνωσία που απαιτεί για την διαχείριση του είναι επίσης χαμηλές.

Το συγκεκριμένο σύστημα είναι βασισμένο στην καθολική διασύνδεση συσκευών με το «Internet of things» (IoT), το οποίο στα ελληνικά μεταφράζεται ως διαδίκτυο των αντικειμένων και μπορεί κάλλιστα να χαρακτηριστεί ως το διαδίκτυο του μέλλοντος (future internet). Από την πρώιμη εποχή της πρώτης επιτυχημένης μεταφοράς δεδομένων μεταξύ δύο υπολογιστών σε διαφορετικά δωμάτια, η άμεση εξέλιξη των πραγμάτων όριζε, την ιδεατή για την τότε εποχή, δικτύωση όλων των υπολογιστών (τερματικών) πάνω σε ένα δίκτυο. Η ιδέα αυτή όσον αφορά τη σύνδεση (δικτύωση) των προσωπικών υπολογιστών του χρήστη και των υπολογιστικών συστημάτων (mainframew, servers) υλοποιείται εδώ και χρόνια με το ευρέως πλέον γνωστό διαδίκτυο (Internet), το οποίο δεν είναι τίποτα άλλο από τη σύνδεση πολλών δικτύων υπολογιστών μεταξύ τους.

Ένα σύστημα σαν αυτό που παρουσιάζει η παρούσα πτυχιακή εργασία, σίγουρα δέχεται βελτιώσεις και επεκτάσεις. Στη συνέχεια γίνεται αναφορά σε μερικές επιπλέον δυνατότητες, βελτιώσεις και επεκτάσεις που θα μπορούσε να περιλαμβάνει το σύστημα σε θέματα τροφοδοσίας, επικοινωνίας, αισθητήρων και λογισμικού.

Θέματα επικοινωνίας:

- Ø Wifi shield: Σε περίπτωση που το θερμοκήπιο βρίσκεται εντός εμβέλειας ασύρματου δικτύου, θα αποτελούσε λύση η χρήση μιας Wifi shield, η οποία παρέχει τη δυνατότητα στο Arduino να επικοινωνεί ασύρματα με το διαδίκτυο, χωρίς να χρειάζεται να συνδεθεί στον δρομολογητή (router) με καλώδιο.
- Ø GPRS shield: Σε περίπτωση που δεν υπάρχει δυνατότητα σύνδεσης με το διαδίκτυο στο χώρο του θερμοκηπίου, η συγκεκριμένη κάρτα (shield) επιτρέπει την επικοινωνία μέσω GPRS. Πιο συγκεκριμένα η επικοινωνία μπορεί να γίνει με τη χρήση μηνυμάτων

(sms) ή με αναγνώριση χρόνου κλήσης. Ο δεύτερος τρόπος έχει το πλεονέκτημα ότι δεν έχει κόστος σε σχέση με τον πρώτο που υπάρχει μία χρέωση ανάλογα με τον πάροχο κινητής τηλεφωνίας.

Θέματα τροφοδοσίας:

- Ø Μπαταρία: Η χρήση μιας μπαταρίας είτε αλκαλικής , είτε λιθίου θα βοηθούσε σε περίπτωση διακοπής παροχής ρεύματος στον μικροελεγκτή.
- Ø Ανανεώσιμες πηγές ενέργειας: Η τροφοδοσίας του συστήματος θα μπορούσε να προέρχεται από ανανεώσιμες πηγές ενέργειας.

Θέματα λογισμικού:

- Ø Προγραμματισμός: Να έχει αποδοτικότερο προγραμματισμό ώστε να χρησιμοποιεί λιγότερους πόρους του συστήματος.
- Ø Εφαρμογές για κινητά: Να υλοποιηθούν εφαρμογές για κινητά τηλέφωνα Android ή iPhone.

Θέματα αισθητήρων:

- Ø Κάμερα: Να έχει υποστήριξη ιστορικού κάμερας.
- Ø Αριθμός αισθητήρων: Να παρέχει τη δυνατότητα διασύνδεσης με περισσότερους αισθητήρες.

Βιβλιογραφία

Βιβλία:

Αρχές λειτουργίας και σχεδιασμός των Ηλεκτρονικών Συστημάτων Μέτρησης, Καλαϊτζάκης, Κουτρούλης, Κλειδάριθμος, 2010

Ηλεκτρονικά I-Εισαγωγή στα Ηλεκτρονικά, Ι. Χαριτάντης, Πανεπιστημιακές Εκδόσεις Αράκυνθος, 2006

Ηλεκτρονικά II-Αναλογικά Ηλεκτρονικά, Ι. Χαριτάντης, Πανεπιστημιακές Εκδόσεις Αράκυνθος, 2007

Ανάπτυξη εφαρμογών με το Arduino, Παπατζογλου ,Λιωνής , Τζιόλα, 7/2014

Διαδίκτυο:

Προγραμματισμός διαδικτύου, <http://www.w3schools.com/> (Ανάκτηση: Δεκέμβριος 2014)

Προγραμματισμός διαδικτύου, <http://www.codecademy.com/learn> (Ανάκτηση: Δεκέμβριος 2014)

HTML, <http://el.wikipedia.org/wiki/HTML> (Ανάκτηση: Ιανουάριος 2015)

CSS, <http://el.wikipedia.org/wiki/CSS> (Ανάκτηση: Ιανουάριος 2015)

PHP, <http://php.net/> (Ανάκτηση: Μάρτιος 2015)

PHP, <http://el.wikipedia.org/wiki/PHP> (Ανάκτηση: Μάρτιος 2015)

AJAX, [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming)) (Ανάκτηση: Μάρτιος 2015)

MySQL, <https://www.mysql.com/> (Ανάκτηση: Ιούνιος 2015)

MySQL, <http://el.wikipedia.org/wiki/MySQL> (Ανάκτηση: Ιούνιος 2015)

Arduino, <http://el.wikipedia.org/wiki/Arduino> (Ανάκτηση: Ιανουάριος 2015)

Arduino, <http://www.arduino.cc/#> (Ανάκτηση: Ιανουάριος 2015)

Redboard, https://learn.sparkfun.com/tutorials/redboard-vuno?_ga=1.250580352.706255221.1432302589 (Ανάκτηση: Ιανουάριος 2015)

Notepad, <https://notepad-plus-plus.org/> (Ανάκτηση: Ιανουάριος 2015)

Xampp, <http://el.wikipedia.org/wiki/XAMPP> (Ανάκτηση: Μάρτιος 2015)

Xampp, <https://www.apachefriends.org/index.html> (Ανάκτηση: Μάρτιος 2015)

Fritzing, <http://fritzing.org/home/> (Ανάκτηση: Μάρτιος 2015)

Arduino IDE, <http://www.arduino.cc/en/pmwiki.php?n=main/software> (Ανάκτηση: Φεβρουάριος 2015) (Ανάκτηση: Ιανουάριος 2015)

Wiring, <http://wiring.org.co/> (Ανάκτηση: Ιανουάριος 2015)

Ethernet shield, <http://www.arduino.cc/en/Main/ArduinoEthernetShield> (Ανάκτηση: Φεβρουάριος 2015)

Photocell, <http://www.instructables.com/id/Photocell-tutorial/> (Ανάκτηση: Φεβρουάριος 2015)

Raindrop sensor, <http://www.instructables.com/id/Arduino-Modules-Rain-Sensor/> (Ανάκτηση: Φεβρουάριος 2015)

Soil moisture sensor, <http://www.instructables.com/id/Soil-Moisture-Sensor/> (Ανάκτηση: Φεβρουάριος 2015)

DHT11, <http://playground.arduino.cc/main/DHT11Lib> (Ανάκτηση: Φεβρουάριος 2015)

HC-SR04, <http://www.instructables.com/id/Simple-Arduino-and-HC-SR04-Example/> (Ανάκτηση: Φεβρουάριος 2015)

Amcharts, <http://www.amcharts.com/> (Ανάκτηση: Ιούνιος 2015)

Προγραμματισμός Arduino, <http://startingelectronics.org/tutorials/arduino/ethernet-shield-web-server-tutorial/> (Ανάκτηση: Φεβρουάριος 2015)

Παράρτημα Α

Κώδικας προγραμματισμού μικροελεγκτή

```
#include <SPI.h> //Δήλωση βιβλιοθήκης SPI
#include <Ethernet.h> //Δήλωση βιβλιοθήκης Ethernet
#include <SD.h> //Δήλωση βιβλιοθήκης SD
#define REQ_BUF_SZ 60 // Μέγεθος της προσωρινής μνήμης που
χρησιμοποιείται για να συλλάβει τα αιτήματα HTTP
#include <dht.h> //Δήλωση βιβλιοθήκης SPI
#define dht_dpin A5 // Δήλωση αναλογικής ακίδας A5 ως ακίδα μέτρησης
θερμοκρασίας-υγρασίας
dht DHT;
const int trigPin = 2; //Δήλωση ψηφιακής ακίδας 2 ως ακίδα δέκτη
αισθητήρα απόστασης
const int echoPin = 3; //Δήλωση ψηφιακής ακίδας 3 ως ακίδα πομπού
αισθητήρα απόστασης
long duration, cm; //Δήλωση μεταβλητών που χρησιμοποιούνται για τον
υπολογισμό απόστασης
int countdb = 0; //Δήλωση ακέραιας μεταβλητής που χρησιμοποιείται
ως μετρητής
byte mac[] = { 0x54, 0x34, 0x41, 0x30, 0x30, 0x31 }; //Δήλωση mac
address της Ethernet shield
EthernetClient client;
char server[] = "192.168.123.100"; // Διεύθυνση IP τοπικού server
Xampp
IPAddress ip(192,168,123,101); // Διεύθυνση IP Arduino server
EthernetServer server1(80); // Δημιουργία διακομιστή στο port 80
File webFile; // Το αρχείο της ιστοσελίδας στην SD
κάρτα
char HTTP_req[REQ_BUF_SZ] = {0}; // Μηδενίζεται η προσωρινή μνήμη
που χρησιμοποιείται για να συλλάβει τα αιτήματα HTTP
char req_index = 0; // Μηδενίζεται ο δείκτης προσωρινής
μνήμης
boolean state[4] = {0}; // Αποθηκεύει τις καταστάσεις των συσκευών
δράσης

void setup() { // Πραγματοποιείται μία φορά.Αρχικοποιήσεις,
προετοιμασία βιβλιοθηκών, ρύθμιση κατάστασης ακίδων
pinMode(10, OUTPUT); //Απενεργοποίηση Ethernet chip
digitalWrite(10, HIGH); //Ενεργοποίηση Ethernet chip
SD.begin(4); //Έναρξη επικοινωνίας με SD κάρτα
pinMode(8, OUTPUT); // Δήλωση ακροδέκτη 8 ως έξοδος (
Ανεμιστήρας)
pinMode(9, OUTPUT); // Δήλωση ακροδέκτη 9 ως έξοδος (
Ηλεκτροβόνα)
Ethernet.begin(mac);
Ethernet.begin(mac, ip); // Η προετοιμασία της Ethernet shield
server1.begin(); // Έναρξη ανίχνευσης clients
}

void loop() { //Καλείται ξανά και ξανά επιτρέποντας έτσι στο
πρόγραμμα να ανταποκριθεί σε εξωτερικά ερεθίσματα
```

```

db(); //Κλήση υπορουτίνας επικοινωνίας με βάση δεδομένων
if (state[0]) { //Έλεγχος αυτόματου σεναρίου ανεμιστήρα
    if(analogRead(2)>800){ // Έλεγχος τιμής αισθητήρα
        digitalWrite(8, HIGH);} //Ενεργοποίηση ανεμιστήρα
    else{digitalWrite(8,LOW);} //Απενεργοποίηση ανεμιστήρα
} //Τέλος ελέγχου αυτόματου σεναρίου ανεμιστήρα
if (state[1]) { //Έλεγχος αυτόματου σεναρίου ηλεκτροβάνας
    if(analogRead(4)>650){ // Έλεγχος τιμής ηλεκτροβάνας
        digitalWrite(9, HIGH);} //Ενεργοποίηση ηλεκτροβάνας
    else{digitalWrite(9,LOW);} //Απενεργοποίηση ηλεκτροβάνας
} //Τέλος ελέγχου αυτόματου σεναρίου ηλεκτροβάνας
pinMode(trigPin, OUTPUT); //Δήλωση ακίδας πομπού ως έξοδος (
μέτρηση απόστασης)
digitalWrite(trigPin, LOW);//Η ακίδα πομπού παίρνει τιμή χαμηλής
στάθμης ( μέτρηση απόστασης)
delayMicroseconds(2);// Καθυστέρηση 2 μs ( μέτρηση απόστασης)
digitalWrite(trigPin, HIGH);//Η ακίδα πομπού παίρνει τιμή υψηλής
στάθμης ( μέτρηση απόστασης)
delayMicroseconds(10); //Καθυστέρηση 10 μs ( μέτρηση απόστασης)
digitalWrite(trigPin, LOW);//Η ακίδα πομπού παίρνει τιμή χαμηλής
στάθμης ( μέτρηση απόστασης)
pinMode(echoPin, INPUT);//Δήλωση ακίδας δέκτη ως έξοδος (
μέτρηση απόστασης)
duration = pulseIn(echoPin, HIGH);//Το χρονικό διάστημα ( σε
μικροδευτερόλεπτα ) από την αποστολή ( μέτρηση απόστασης)
cm = microsecondsToCentimeters(duration);//Μετατροπή του χρόνου
σε εκάτοστά
DHT.read11(dht_dpın);//Διαβάζει τις τιμές του αισθητήρα DHT11(
μέτρηση θερμοκρασίας-απόστασης)
EthernetClient client = server1.available(); // Προσπαθεί να
ανιχνεύσει client
if (client) { // Έλεγχος για εύρεση client
    boolean currentLineIsBlank = true;
    while (client.connected()) {
        if (client.available()) { // Έλεγχος για το αν είναι
διαθέσιμα τα δεδομένα του client
            char c = client.read(); // Διαβάζει 1 byte
(χαρακτήρα) από τον client
            if (req_index < (REQ_BUF_SZ - 1)) {
                HTTP_req[req_index] = c; // Αποθηκεύει
το HTTP αίτημα
                req_index++;
            }
            if (c == '\n' && currentLineIsBlank) {
                client.println("HTTP/1.1 200 OK");
                if (StrContains(HTTP_req, "GET /logo.jpg")) {
//Έλεγχος αιτήματος εικόνας ιστοσελίδας
                webFile = SD.open("logo.jpg");//Άνοιγμα
εικόνας από sd καρτα
                if (webFile) {
                    client.println("HTTP/1.1 200 OK");
                    client.println();
                }
                if (webFile) {
                    while(webFile.available()) {

```



```

        client.write(webFile.read()); //
        Στέλνει την εικόνα στην ιστοσελίδα
    }
    webFile.close();
}
}
    if (StrContains(HTTP_req, "ajax_inputs")) {
        client.println("Content-Type:
text/xml");
        client.println("Connection: keep-
alive");
        client.println();
        Set();
        XML_response(client);
    }
    else {
        client.println("Content-Type:
text/html");
        client.println("Connection: keep-
alive");
        client.println();
        webFile = SD.open("index.htm");
        // Άνοιγμα αρχείου ιστοσελίδας
        if (webFile) {
            while(webFile.available()) {
                client.write(webFile.read()); //
                Αποστολή ιστοσελίδας στον client
            }
            webFile.close();
        }
    }
    Serial.print(HTTP_req);
    req_index = 0;
    break;
}
if (c == '\n') {
    currentLineIsBlank = true;
}
else if (c != '\r') {
    currentLineIsBlank = false;
}
} //Τέλος if (client.available())
} // Τέλος του while (client.connected())
delay(1); // Καθυστέρηση η οποία δίνει χρόνο στον
περιηγητή να λάβει τα δεδομένα
client.stop(); // Κλείνει η σύνδεση
} // Τέλος if (client)
}

void db () { //Υπορουτίνα επικοινωνίας με βάση δεδομένων
    countdb = (countdb + 1) ;//Αύξηση μετρητή (αφορά την συχνότητα
εγγραφών στη βάση δεδομένων)
    if (countdb >= 1000) {
        if (client.connect(server, 80)) {
            Serial.println("-> Connected");
        }
    }
}

```

```

        client.print( "GET /arduino/add_data.php?");
        client.print("&&");
        client.print("light=");
        client.print( (analogRead(2))/10.23 );
        client.print("&&");
        client.print("raindrop=");
        client.print( (analogRead(3))/10.23 );
        client.print("&&");
        client.print("moisture=");
        client.print( (analogRead(4))/10.23 );
        client.print("&&");
        client.print("temperature=");
        client.print( DHT.temperature );
        client.print("&&");
        client.print("humidity=");
        client.print( DHT.humidity );
        client.print("&&");
        client.print("tank=");
        client.print( cm );
        countdb = 0 ;
        client.println( " HTTP/1.1");
        client.print( "Host: " );
        client.println(server);
        client.println( "Connection: close" );
        client.println();
        client.println();
        client.stop();
    }
else {
    Serial.println("--> connection failed/n");
}
}

void Set(void) {
    if (StrContains(HTTP_req, "AFAN=1")) {
        state[0] = 1; // Αποθηκεύει την κατάσταση του σεναρίου
        αυτόματου αερισμού
    }
    else if (StrContains(HTTP_req, "AFAN=0")) {
        state[0] = 0; // Αποθηκεύει την κατάσταση του σεναρίου
        αυτόματου αερισμού
        digitalWrite(8, LOW);
    }
    if (StrContains(HTTP_req, "ASOLENOID=1")) {
        state[1] = 1; // Αποθηκεύει την κατάσταση του σεναρίου
        αυτόματου ποτίσματος
    }
    else if (StrContains(HTTP_req, "ASOLENOID =0")) {
        state[1] = 0; // Αποθηκεύει την κατάσταση του σεναρίου
        αυτόματου ποτίσματος
    }
    if (StrContains(HTTP_req, "MFAN=1")) {
        state[2] = 1; // Αποθηκεύει την κατάσταση του ανεμιστήρα
        digitalWrite(8, HIGH);
    }
}

```

```

}
else if (StrContains(HTTP_req, "MFAN3=0")) {
    state[2] = 0; // Αποθηκεύει την κατάσταση του ανεμιστήρα
    digitalWrite(8, LOW);
}
if (StrContains(HTTP_req, "MSOLENOID4=1")) {
    state[3] = 1; // Αποθηκεύει την κατάσταση της ηλεκτροβάνας
    digitalWrite(9, HIGH);
}
else if (StrContains(HTTP_req, "MSOLENOID=0")) {
    state[3] = 0; // Αποθηκεύει την κατάσταση της ηλεκτροβάνας
    digitalWrite(9, LOW);
}
}

void XML_response(EthernetClient cl) {
    int analog_val; // stores value read from analog
    inputs
    int count; // used by 'for' loops
    cl.print("<?xml version = \"1.0\" ?>");
    cl.print("<inputs>");
    for (count = 2; count <= 4; count++) { // A2 to A5
        analog_val = (analogRead(count));
        cl.print("<analog>");
        cl.print((analog_val)/10.23);
        cl.println("</analog>");
    }
    cl.print("<analog>");
    cl.print(DHT.temperature);
    cl.print("</analog>");
    cl.print("<analog>");
    cl.print(DHT.humidity);
    cl.println("</analog>");
    cl.print("<analog>");
    cl.print(cm);
    cl.println("</analog>");
    cl.print("<DEVICE>");
    if (state[0]) {
        cl.print("checked");
    }
    else {
        cl.print("unchecked");
    }
    cl.println("</DEVICE >");
    cl.print("<DEVICE >");
    if (state[1]) {
        cl.print("checked");
    }
    else {
        cl.print("unchecked");
    }
    cl.println("</DEVICE >");
    cl.print("<DEVICE >");
    if (state[2]) {
        cl.print("on");
    }
}

```

```

    }
    else {
        cl.print("off");
    }
    cl.println("</DEVICE >");
    cl.print("<DEVICE >");
    if (state[3]) {
        cl.print("on");
    }
    else {
        cl.print("off");
    }
    cl.println("</DEVICE >");
    cl.print("</inputs>");
}

char StrContains(char *str, char *sfind) { // searches for the string
sfind in the string str
// returns 1 if string found
// returns 0 if string not found
    char found = 0;
    char index = 0;
    char len;
    len = strlen(str);
    if (strlen(sfind) > len) {
        return 0;
    }
    while (index < len) {
        if (str[index] == sfind[found]) {
            found++;
            if (strlen(sfind) == found) {
                return 1;
            }
        }
        else {
            found = 0;
        }
        index++;
    }
    return 0;
}

long microsecondsToCentimeters(long microseconds) { // The speed of
sound is 340 m/s or 29 microseconds per centimeter.
// The ping travels out and back, so to find the distance of the
// object we take half of the distance travelled.
    return microseconds / 29 / 2;
}

```

Παράρτημα Β

Κώδικας ιστοσελίδας εποπτείας και ελέγχου

```
<!DOCTYPE html>
<html>
  <head>
    <title>GreenHouse</title>
    <script>
      autoFAN = "";
      autoSOLENOID = "";
      FAN = "";
      SOLENOID = "";
      var FAN_state = 0;
      var SOLENOID_state = 0;
      var light_ambient = 0;
      var raindrop = 0;
      var moisture = 0;
      var temperature = 0;
      var humidity = 0;
      var tank_height = 0;

      eval(function(p,a,c,k,e,r){e=function(c){return(c<a?'':e(parseInt(c/a)))+((c=c%a)>35?String.fromCharCode(c+29):c.toString(36))};if(!''.replace(/^/,String)){while(c-->0)r[e(c)]=k[c]||e(c);k=[function(e){return r[e]};e=function(){return'\\w+'};c=1];while(c-->0)if(k[c])p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c]);return p}('t
W=v(f){W.2t.3T(A);A.B={Z:2u,19:1q,1h:1q,1J:U,1b:D,K:0,V:[\\'0\\',\\'20\\',\\'40\\',\\'2A\\',\\'2B\\',\\'D\\'],2r:10,2C:M,1E:U,2q:{2D:3,2E:2},2H:M,1c:{2j:10,2m:3y,2O:\\'3v\\'},J:{2P:\\'#4h\\',V:\\'#31\\',2r:\\'#3k\\',1J:\\'#37\\',1E:\\'#37\\',3e:\\'#31\\',1t:{2s:\\'1e(3Y, 3d, 3d, 1)\\',3c:\\'1e(1Y,5I,5E,.9)\\'}}},1o:[{1n:20,1D:2A,1F:\\'#3j\\'},{1n:2A,1D:2B,1F:\\'#36\\'},{1n:2B,1D:D,1F:\\'#5n\\'}]};tg=0,1p=A,N=0,1S=0,1G=U;A.5d=v(a){N=f.1c?g:a;tb=(f.1bf.K)/D;1S=a>f.1b?1S=f.1b+b:a<f.K?f.Kb:a;g=a;f.1c?3l():A.1g();C A};A.3m=v(a){N=g;a;A.1g();C A};A.4T=v(){g=N=1S=A.B.K;A.1g();C A};A.4R=v(){C g};A.13=v(){};v 2k(a,b){Q(t i 4P b){z(1H b[i]==\"1W\"&&!(40.4y.2V.4p(b[i]))==\\'[1W 4n]\\')&&i!=\\'Z\\')}{z(1H a[i]!=\"1W\"){a[i]={}}2k(a[i],b[i])}O{a[i]=b[i]}}};2k(A.B,f);A.B.K=1R(A.B.K);A.B.1b=1R(A.B.1b);f=A.B;N=g=f.K;z(!f.Z){4m 4j(\"4g 4d 4b 46 44 41 3Z 3W W 1W!\");}t j=f.Z.5K?f.Z:2R.5v(f.Z),q=j.3u(\\'2d\\'),1i,1y,1A,14,17,u,1d;v2M(){j.19=f.19;j.1h=f.1h;1i=j.4s(M);1d=1i.3u(\\'2d\\');1y=j.19;1A=j.1h;14=1y/2;17=1A/2;u=14<17?14:17;1i.2J=U;1d.3P(14,17);1d.G();q.3P(14,17);q.G();}2M();A.4Z=v(a){2k(A.B,a);2M();A.1g();C A};t k={4q:v(p){C p},4e:v(p){C E.1L(p,2)},4c:v(p){C E.1L(p,5)},3v:v(p){C 1-E.10(E.5C(p))},5k:v(p){C 1-(v(p){Q(t a=0,b=1;1;a+=b,b/=2){z(p>=(7-4*a)/11){C-E.1L((11-6*a-11*p)/4,2)+E.1L(b,2)}})(1-p)},4S:v(p){C 1-(v(p){t x=1.5;C E.1L(2,10*(p-1))*E.1T(20*E.1a*x/3*p)}})(1-p)}};t l=2u;v 3S(d){t e=2v 3R;l=2x(v){t a=2v 3R-e,1M=a/d.2m;z(1M>1){1M=1}t b=1H d.2g=\"v\"?d.2g:k[d.2g];t c=b(1M);d.3Q(c);z(1M==1){2b(1)}},d.2j||10);v 3l(){l&&2b(1);t b=(1S-N),1n=N,29=f.1c;3S({2j:29.2j,
```

2m:29.2m,2g:29.20,3Q:v(a){N=1R(1n)+b*a;1p.1g()}});q.5l="30";A.1g=v(
){z(!1i.2J){1d.3M(-14,-17,1y,1A);1d.G();t a=q;q=1d;3L();3K();3I();
3H();3F();3D();3z();1i.2J=M;q=a;5Ga}q.3M(-14,-17,1y,1A);q.G();q.4a
(1i,-14,-17,1y,1A);z(!W.28){t b=2x(v){z(!W.28){C}2b(b);2K();2L();
z(!1G){1p.13&&1p.13();1G=M}},10)}O{2K();2L();z(!1G){1p.13&&1p.13();1
G=M}}C A};v S(a){C a*E.1a/4J};v 1l(a,b,c){t
d=q.4Y(0,0,0,c);d.1V(0,a);d.1V(1,b);C d};v 3L(){t a=u/D*5g,3x=u-
a,2a=u/D*5q,5u=u-2a,1f=u/D*5z,5A=u 1f;3t=u/D*5F;q.G();z(f.2H)
{q.2o=3x;q.2n='1e(0,0,0, 0.5)'}q.P();q.16(0,0,a,0,E.1a*2,M);q.L=11
(\'#42\','\'#43\','a);q.T();q.R();q.P();q.16(0,0,2a,0,E.1a*2,M);q.L=11
(\'#49\','\'#36\','2a);q.T();q.P();q.16(0,0,1f,0,E.1a*2,M);q.L=11(\'#3
j\','\'#3s\','1f);q.T();q.P();q.16(0,0,3t,0,E.1a*2,M);q.L=f.J.2P;q.T()
;q.G());v 3H(){t r=u/D*2T;q.2e=2;q.2U=f.J.V;q.G();Q(t
i=0;i<f.V.H;++i){ta=45+i*(1U/(f.V.H-1));q.1z(S(a));q.P();q.1K(0,r);
q.F(0,r-u/D*15);q.1X();q.R();q.G()}z(f.2C){q.1z(S(2X));q.P();q.16
(0,0,r,S(45),S(4N),U);q.1X();q.R();q.G()}};v 3I(){t r=u/D*2T;
q.2e=1;q.2U=f.J.2r;q.G();t b=f.2r*(f.V.H-1);Q(t i=0;i<b;++i){t
a=45+i*(1U/b);q.1z(S(a));q.P();q.1K(0,r);q.F(0,r-u/D*7.5);q.1X();
q.R();q.G()}};v 3F(){t r=u/D*55;Q(t i=0;i<f.V.H;++i){t
a=45+i*(1U/(f.V.H-1)),p=1w(r,S(a));q.1x=20*(u/1q)+"2i
2Y";q.L=f.J.3e;q.2e=0;q.2h="2f";q.27(f.V[i],p.x,p.y+3)}};v
3D(){z(!f.1J){C}q.G();q.1x=24*(u/1q)+"2i
2Y";q.L=f.J.1J;q.2h="2f";q.27(f.1J,0,-u/4.25);q.R()};v
3z(){z(!f.1E){C}q.G();q.1x=22*(u/1q)+"2i
2Y";q.L=f.J.1E;q.2h="2f";q.27(f.1E,0,u/3.25);q.R()};v 32(a){t
b=f.2q.2E,34=f.2q.2D;a=1R(a);t
n=(a<0);a=E.35(a);z(b>0){a=a.5t(b).2V().1j(\'.\');Q(t i=0,s=34-
a[0].H;i<s;++i){a[0]='0'+a[0]}a=(n?\'-
\':\'\'+a[0]+\'.\'+a[1])}O{a=E.30(a).2V();Q(t i=0,s=34-
a.H;i<s;++i){a='0'+a}a=(n?\'-\':\'\'+a)C a};v 1w(r,a){t
x=0,y=r,10=E.10(a),1T=E.1T(a),X=x*1T-y*10,Y=x*10+y*1T;C{x:X,y:Y}};v
3K(){q.G();t a=u/D*2T;t b=a-u/D*15;Q(t i=0,s=f.1o.H;i<s;i++)}{t
c=f.1o[i],39=(f.1b-f.K)/1U,1P=S(45+(c.1n-f.K)/39),1N=S(45+(c.1D-
f.K)/39);q.P();q.1z(S(2X));q.16(0,0,a,1P,1N,U);q.R();q.G();t
d=1w(b,1P),3a=1w(a,1P);q.1K(d.x,d.y);q.F(3a.x,3a.y);t
e=1w(a,1N),3b=1w(b,1N);q.F(e.x,e.y);q.F(3b.x,3b.y);q.F(d.x,d.y);q.1C
();q.L=c.1F;q.T();q.P();q.1z(S(2X));q.16(0,0,b,1P-
0.2,1N+0.2,U);q.R();q.1C();q.L=f.J.2P;q.T();q.G()}};v 2L(){t
a=u/D*12,1f=u/D*8,1u=u/D*3X,1r=u/D*20,2l=u/D*4,1B=u/D*2,38=v(){q.3f=
2;q.3g=2;q.2o=10;q.2n='1e(5L, 3h, 3h,
0.45)'};38();q.G();z(N<0){N=E.35(f.K-N)}O z(f.K>0){N=
f.K}O{N=E.35(f.K)+N}q.1z(S(45+N/(f.1b-f.K)/1U));q.P();q.1K(-1B,-
1r);q.F(-2l,0);q.F(-1,1u);q.F(1,1u);q.F(2l,0);q.F(1B,-
1r);q.1C();q.L=11(f.J.1t.2s,f.J.1t.3c,1u-1r);q.T();q.P();q.F(-
0.5,1u);q.F(-1,1u);q.F(-2l,0);q.F(-1B,-1r);q.F(1B/2-2,-
1r);q.1C();q.L='1e(1Y, 1Y, 1Y,
0.2)'};q.T();q.R();38();q.P();q.16(0,0,a,0,E.1a*2,M);q.L=11(\'#3s\','
\'#36\','a);q.T();q.R();q.P();q.16(0,0,1f,0,E.1a*2,M);q.L=11("#47","
#48",1f);q.T()};v 3i(x,y,w,h,r){q.P();q.1K(x+r,y);q.F(x+w-
r,y);q.23(x+w,y,x+w,y+r);q.F(x+w,y+h-r);q.23(x+w,y+h,x+w-
r,y+h);q.F(x+r,y+h);q.23(x,y,x+r,y);q.1C()};v
2K(){q.G();q.1x=40*(u/1q)+"2i 30";t a=32(g),2Z=q.4f(\'-
\'+32(0)).19,y=u-u/D*33,x=0,2W=0.12*u;q.G();3i(-2Z/2-0.21*u,y-2W-
0.4i*u,2Z+0.3n*u,2W+0.4k*u,0.21*u);t b=q.4l(x,y-0.12*u-


```
|end|128|numbers|shadowOffsetX|shadowOffsetY|143|roundRect|eee|666|a
nimate|setRawValue|05|045|012|004|getElementsByTagName|f0f0f0|r3|get
Context|cycle|createElement|d0|250|drawUnits|cssText|styleSheets|bod
y|drawTitle|domReady|drawNumbers|addEventListener|drawMajorTicks|dra
wMinorTicks|trim|drawHighlights|drawPlate|clearRect|attributes|round
|translate|step|Date|_animate|push|eval|value|the|77|240|creating|w
hen|ddd|aaa|specified||not|e8e8e8|f5f5f5|fafafa|drawImage|was|quint|
element|quad|measureText|Canvas|fff|04|Error|07|createRadialGradient
|throw|Array|babab2|call|linear|head|cloneNode|navigator|userAgent|t
oLocaleLowerCase|indexOf|msie|prototype|http|smart|ip|net|styles|fon
ts|digital|mono|eot|ttf|180|face|family|src|315|Object|in|css|getVal
ue|elastic|clear|try|createTextNode|catch|sheet|createLinearGradient
|updateConfig|div|fontFamily|position|absolute|overflow|hidden|inne
rHTML|setTimeout|parentNode|removeChild|get|string|setValue|id|numbe
r|93|DOMContentLoaded|attachEvent|onload|bounce|lineCap|toUpperCase|
999|replace|canvas|91|canv|gauge|toFixed|d1|getElementById|item|node
Name|toLowerCase|88|d2|switch|acos|shift|122|85|delete|test|160|defa
ult|tagName|188'.split('|'),0,{}))
```

```
function GetArduinoInputs()
```

```
{
```

```
    nocache = "&nocache=" + Math.random() * 1000000;
```

```
    var request = new XMLHttpRequest();
```

```
    request.open("GET", "ajax_inputs" + nocache, true);
```

```
    request.send(null);
```

```
    setTimeout('GetArduinoInputs()', 200);
```

```
}
```

```
function GetArduinoIO()
```

```
{
```

```
    nocache = "&nocache=" + Math.random() * 1000000;
```

```
    var request = new XMLHttpRequest();
```

```
    request.onreadystatechange = function()
```

```
    {
```

```
        if (this.readyState == 4) {
```

```
            if (this.status == 200) {
```

```
                if (this.responseXML != null) {
```

```
                    light_ambient =
```

```
this.responseXML.getElementsByTagName('analog')[0].childNodes[0].nod
eValue;
```

```
                    raindrop =
```

```
this.responseXML.getElementsByTagName('analog')[1].childNodes[0].nod
eValue;
```

```
                    moisture =
```

```
this.responseXML.getElementsByTagName('analog')[2].childNodes[0].nod
eValue;
```

```
                    temperature =
```

```
this.responseXML.getElementsByTagName('analog')[3].childNodes[0].nod
eValue;
```

```
                    humidity =
```

```
this.responseXML.getElementsByTagName('analog')[4].childNodes[0].nod
eValue;
```

```
                    tank_height =
```

```
this.responseXML.getElementsByTagName('analog')[5].childNodes[0].nod
eValue;
```



```

        setTimeout('GetArduinoIO()', 1000);
        autoFAN = "";
        autoSOLENOID = "";
        FAN = "";
        SOLENOID = "";
    }
    function GetCheck()
    {
        if (DEVICE_form.AFAN.checked) {
            autoFAN = "&AFAN=1";
        }
        else {
            autoFAN = "&AFAN=0";
        }
        if (DEVICE_form.ASOLENOID.checked) {
            autoSOLENOID = "&ASOLENOID=1";
        }
        else {
            autoSOLENOID = "&ASOLENOID=0";
        }
    }
    function GetButton1()
    {
        if (FAN_state === 1) {
            FAN_state = 0;
            FAN = "&MFAN=0";
        }
        else {
            FAN_state = 1;
            FAN = "&MFAN=1";
        }
    }
    function GetButton2()
    {
        if (SOLENOID_state === 1) {
            SOLENOID_state = 0;
            SOLENOID = "&MSOLENOID=0";
        }
        else {
            SOLENOID_state = 1;
            SOLENOID = "&MSOLENOID=1";
        }
    }
}
</script>

<style>
    body {
        color:#000000;
        background-color:#228B22;
    }
    h2 {
        margin-top:10px;
        margin-right:40px;
        margin-bottom:20px;
        margin-left:40px;
    }

```

```

    font-weight: bold;
    font-style:normal;
    text-decoration: none;
    color:white;
    text-shadow: 2px 2px 4px #000000;
}
h3 {
    height:40px;
    margin-top:0px;
    margin-right:40px;
    margin-bottom:20px;
    margin-left:40px;
    font-weight: bold;
    font-style:normal;
    text-decoration: none;
    font-size: 4pt
    font-family: Arial ;
    color: white;
    font-weight: bold;
    font-style:normal;
    text-decoration: none;
    font-size: 20pt;
    color:white;
    text-shadow: 2px 2px 4px #000000;
}
div1a {
    height: 34px;
    width: 800px;
    border: 4px solid #32CD32;
    border-radius: 30px;
    background-color: #32CD32;
    margin-top:20px;
    margin-right:20px;
    margin-bottom:20px;
    margin-left:40px;
    padding:20px 40px 40px 40px;
    font-size:30px;
    color:white;
    font-family: Arial;
    position: fixed;
    text-align: ;
}
div1b {
    height: 34px;
    width: 660px;
    border: 4px solid #32CD32;
    border-radius: 30px;
    background-color: #32CD32;
    margin-top:20px;
    margin-right:40px;
    margin-bottom:20px;
    margin-left:800px;
    padding:20px 40px 40px 40px;
    color:white;
    font-family: Arial;

```

```

        position: fixed;
        text-align: ;
    }
    div1c {
        height: 33px;
        width: 1420px;
        border: 4px solid #32CD32;
        border-radius: 30px;
        background-color: #32CD32;
        margin-top:660px;
        margin-right:0px;
        margin-bottom:0px;
        margin-left:40px;
        padding:20px 40px 40px 40px;
        color:white;
        font-family: Arial;
        position: fixed;
        text-align: ;
    }
    div2b {
        height: 380px;
        width: 620px;
        border: 4px solid #228B22;
        border-radius: 30px;
        background-color: #228B22;
        margin-top:230px;
        margin-right:200px;
        margin-bottom:20px;
        margin-left:40px;
        padding:20px 40px 40px 40px;
        color:#228B22;
        font-family: Arial;
        font-color: black;
        position: fixed;
        text-align: center;
    }
    div3a {
        height: 8px;
        width: 620px;
        border: 4px solid #32CD32;
        border-radius: 30px;
        background-color: #32CD32;
        margin-top:150px;
        margin-right:0px;
        margin-bottom:0px;
        margin-left:840px;
        padding:20px 40px 40px 40px;
        color:white;
        font-family: Arial;
        position: fixed;
        text-align:center ;
    }
    div3b {
        height: 50px;
        width: 620px;

```

```

border: 4px solid white;
border-radius: 30px;
background-color: white;
margin-top:258px;
margin-right:0px;
margin-bottom:0px;
margin-left:840px;
padding:20px 40px 40px 40px;
color:white;
font-family: Arial;
position: fixed;
text-align:center ;
}
div4a {
height: 8px;
width: 620px;
border: 4px solid #32CD32;
border-radius: 30px;
background-color: #32CD32;
margin-top:410px;
margin-right:20px;
margin-bottom:20px;
margin-left:840px;
padding:20px 40px 40px 40px;
color:white;
font-family: Arial;
position: fixed;
text-align: center;
}
div4b {
height: 50px;
width: 620px;
border: 4px solid white;
border-radius: 30px;
background-color: white;
margin-top:520px;
margin-right:20px;
margin-bottom:20px;
margin-left:840px;
padding:20px 40px 40px 40px;
color:white;
font-family: Arial;
position: fixed;
text-align: center;
}
div2a {
height: 8px;
width: 620px;
border: 4px solid #32CD32;
border-radius: 30px;
background-color: #32CD32;
margin-top:150px;
margin-right:0px;
margin-bottom:0px;
margin-left:40px;

```

```

padding:20px 40px 40px 40px;
color:white;
font-family: Arial;
position: fixed;
text-align:center ;
}
button {
font-size: 120%;
color: black;
font-weight: bold;
border-radius: 10px;
}
form {
font-size: 130%;
color: black;
font-weight: bold;
border-radius: 10px;
}
</style>
</head>
<body onload="GetArduinoIO()">
<div1a>
<h2>Welcome to GreenHouse</h2>
</div1a>
<div1b></div1b>
<div2a><h3>Supervision Panel</h3></div2a>
<div2b>
<canvas id="an_gauge_1" width="200" height="200" data-
title="Ambient light"
data-units="%" data-highlights="0 30 #f33, 30 50 #cc5,50 70
#25B8D9, 70 100 #4D89F2"
data-major-ticks="0 10 20 30 40 50 60 70 80 90 100" data-
type="canv-gauge"
data-min-value="0" data-max-value="100"
data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_1').setValue(light_ambient);},
200);"></canvas>
<canvas id="an_gauge_2" width="200" height="200" data-
title="Raindrop" data-units="%" data-highlights="0 50 #4D89F2,
50 80 #25B8D9, 80 100 #cc5" data-major-ticks="0 10 20 30 40 50
60 70 80 90 100" data-type="canv-gauge" data-min-value="0" data-
max-value="100" data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_2').setValue(raindrop);},
200);"></canvas>
<canvas id="an_gauge_3" width="200" height="200" data-
title="Moisture" data-units="%" data-highlights="0 40 #4D89F2,
40 70 #25B8D9, 70 80 #cc5, 80 100 #f33" data-major-ticks="0 10
20 30 40 50 60 70 80 90 100" data-type="canv-gauge" data-min-
value="0" data-max-value="100" data-onready="setInterval( function()
{ Gauge.Collection.get('an_gauge_3').setValue(moisture);},
200);"></canvas>
<canvas id="an_gauge_4" width="200" height="200" data-
title="Temperature" data-units="°C" data-highlights="0 10
#4D89F2, 10 20 #25B8D9,20 35 #cc5, 35 50 #f33" data-major-

```



```

ticks="0 5 10 15 20 25 30 35 40 45 50" data-type="canv-gauge" data-
min-value="0" data-max-value="50" data-onready="setInterval(
function() {
Gauge.Collection.get('an_gauge_4').setValue(temperature);},
200);"></canvas>
  <canvas id="an_gauge_5" width="200" height="200" data-
title="Humidity" data-units="%" data-highlights="0 20 #f33, 20
50 #cc5,50 80 #25B8D9, 80 100 #4D89F2" data-major-ticks="0 10 20
30 40 50 60 70 80 90 100" data-type="canv-gauge" data-min-value="0"
data-max-value="100" data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_5').setValue(humidity);},
200);"></canvas>
  <canvas id="an_gauge_6" width="200" height="200" data-
title="Tank Height" data-units="cm" data-highlights="0 100 #f33,
100 200 #cc5, 200 300 #25B8D9, 300 450 #4D89F2" data-major-ticks="0
50 100 150 200 250 300 350 400 450" data-type="canv-gauge" data-
min-value="0" data-max-value="450" data-onready="setInterval(
function() {
Gauge.Collection.get('an_gauge_6').setValue(tank_height);},
200);"></canvas>
  </div2b>
  <div3a>
  <h3>Automated Scenarios</h3></div3a>
  <div3b><form id="check_DEVICES" name="DEVICE_form">
  <input type="checkbox" name="AFAN" value="0"
onclick="GetCheck()" />Automatic Airing<br /><br />
  <input type="checkbox" name="ASOLENOID" value="0"
onclick="GetCheck()" />Automatic Watering<br /><br />
  </form>
  </div3b>
  <div4a><h3>Control Panel</h3></div4a>
  <div4b>
  <button type="button" id="MFAN" onclick="GetButton1()">Airing is
OFF</button><br /><br />
  <button type="button" id="MSOLENOID"
onclick="GetButton2()">Watering is OFF</button><br /><br />
  </div4b>
  </body>
</html>

```

Παράρτημα Γ

Κώδικας ιστοσελίδας απεικόνισης μετρήσεων αισθητήρα φωτεινότητας:

```
<!DOCTYPE html>
<html>
  <style>
    h1 {
      width:1400;
      margin-top:30px;
      margin-right:85px;
      margin-bottom:40px;
      margin-left:85px;
      font-weight: bold;
      font-style:normal;
      text-decoration: none;
      background-color:#32CD32;
      color:white;
      text-shadow: 2px 2px 4px #000000;
      border-radius: 30px;
      padding-top:20px ;
      padding-bottom:20px ;
      font-family: Arial;
      text-align:center;
    }
    body {
      color:#000000;
      background-color:#228B22;
      text-align: center;
    }
  </style>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Ambient light</title>
  </head>
  <body>
    <h1>
      <a href="tank.html">
        
      </a>
      <a href="../home.html">
        
      </a>Ambient light
      <a href="raindrop.html">
        
      </a>
      <a href="../review_data.php">
        
      </a>
    </h1>
  </body>
</html>
```

```

</h1>
<link rel="stylesheet" href="style.css" type="text/css">
<script src="../amcharts/amcharts.js"
type="text/javascript"></script>
<script src="../amcharts/serial.js"
type="text/javascript"></script>
<script>
    AmCharts.loadJSON = function(url) {
        // create the request
        if (window.XMLHttpRequest) {
            // IE7+, Firefox, Chrome, Opera, Safari
            var request = new XMLHttpRequest();
        }
        else {
            // code for IE6, IE5
            var request = new
ActiveXObject('Microsoft.XMLHTTP');
        }
        // load it
        // the last "false" parameter ensures that our code
will wait before the
        // data is loaded
        request.open('GET', url, false);
        request.send();
        // parse and return the output
        return eval(request.responseText);
    };
</script>
<div id="chartdiv" style="width: 1400px; height: 600px;
background-color:white;margin: 0px 200px 85px 85px;border-radius:
30px; "></div>
<script>
    var chart;
    // create chart
    AmCharts.ready(function() {
        // load the data
        var chartData = AmCharts.loadJSON('data_all.php');
        // SERIAL CHART
        chart = new AmCharts.AmSerialChart();
        chart.dataProvider = chartData;
        chart.categoryField = "event";
        chart.dataDateFormat = "JJ:NN";
        chart.path = "../amcharts/";
        var categoryAxis = chart.categoryAxis;
        chart.dataDateFormat = "JJ:NN";
        var valueAxis = new AmCharts.ValueAxis();
        valueAxis.inside = true;
        valueAxis.tickLength = 0;
        valueAxis.axisAlpha = 0;
        valueAxis.minimum = 0;
        valueAxis.maximum = 100;
        chart.addValueAxis(valueAxis);
        valueAxis.title = "Light % ( 0% light , 100% dark)";
        // GRAPHS
        var graph = new AmCharts.AmGraph();

```

```

graph.valueField = "light";
graph.bullet = "round";
graph.bulletBorderColor = "#32CD32";
graph.bulletBorderThickness = 2;
graph.lineThickness = 2;
graph.lineAlpha = 0.5;
graph.lineColor = "#32CD32";
graph.valueunit = "cm";
chart.addGraph(graph);
// SCROLLBAR
var chartScrollbar = new AmCharts.ChartScrollbar();
chartScrollbar.graph = graph;
chartScrollbar.scrollbarHeight = 30;
chart.addChartScrollbar(chartScrollbar);
chart.creditsPosition = "bottom-right";
// CATEGORY AXIS
chart.categoryAxis.dateFormats = true;
// CURSOR
var chartCursor = new AmCharts.ChartCursor();
chartCursor.cursorAlpha = 0;
chartCursor.zoomable = true;
chartCursor.categoryBalloonEnabled = true;
chartCursor.valueLineEnabled = true;
chartCursor.valueLineBalloonEnabled = true;
chartCursor.valueLineAlpha = 0;
chart.addChartCursor(chartCursor);
chartCursor.color = "#FFFFFF";
chartCursor.cursorPosition = "mouse";
chart.creditsPosition = "top-right";
var balloon = chart.balloon;
// set properties
balloon.adjustBorderColor = true;
balloon.color = "#000000";
balloon.cornerRadius = 5;
balloon.fillColor = "#FFFFFF";
// WRITE
chart.write("chartdiv");
});
</script>
</body>
</html>

```

Κώδικας ιστοσελίδας απεικόνισης μετρήσεων αισθητήρα βροχής:

```

<!DOCTYPE>
<html>
<style>
h1 {
width:1400;
margin-top:30px;
margin-right:85px;
margin-bottom:40px;
margin-left:85px;
font-weight: bold;
font-style:normal;

```

```

        text-decoration: none;
        background-color:#32CD32;
        color:white;
        text-shadow: 2px 2px 4px #000000;
        border-radius: 30px;
        padding-top:20px ;
        padding-bottom:20px ;
        font-family: Arial;
        text-align:center;
    }
    body {
        color:#000000;
        background-color:#228B22;
        text-align: center;
    }
</style>
<head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Raindrop chart</title>
</head>
<body>
    <h1>
        <a href="light.html">
            
        </a>
        <a href="../home.html">
            
        </a>Raindrop chart
        <a href="moisture.html">
            
        </a>
        <a href="../review_data.php">
            
        </a>
    </h1>
    <link rel="stylesheet" href="style.css" type="text/css">
    <script src="../amcharts/amcharts.js"
type="text/javascript"></script>
    <script src="../amcharts/serial.js"
type="text/javascript"></script>
    <script>
        AmCharts.loadJSON = function(url) {
            // create the request
            if (window.XMLHttpRequest) {
                // IE7+, Firefox, Chrome, Opera, Safari
                var request = new XMLHttpRequest();
            }
            else {
                // code for IE6, IE5

```

```

        var request = new
ActiveXObject('Microsoft.XMLHTTP');
    }
    // load it
    // the last "false" parameter ensures that our code
will wait before the
    // data is loaded
request.open('GET', url, false);
request.send();
    // parse and return the output
    return eval(request.responseText);
};
</script>
<div id="chartdiv" style="width: 1400px; height: 600px;
background-color:white;margin: 0px 200px 85px 85px;border-radius:
30px; "></div>
<script>
    var chart;
    AmCharts.ready(function () {
        var chartData = AmCharts.loadJSON('data_all.php');
        // SERIAL CHART
        chart = new AmCharts.AmSerialChart();
        chart.path = "../amcharts/";
        chart.dataProvider = chartData;
        chart.categoryField = "event";
        chart.dataDateFormat = "JJ:NN";
        // AXES
        // category
        var categoryAxis = chart.categoryAxis;
        categoryAxis.dashLength = 1;
        categoryAxis.gridAlpha = 0.15;
        categoryAxis.axisColor = "#DADADA";
        // value
        var valueAxis = new AmCharts.ValueAxis();
        valueAxis.axisColor = "#DADADA";
        valueAxis.dashLength = 1;
        valueAxis.minimum = 0;
        valueAxis.maximum = 100;
        valueAxis.title = "Raindrop % (0% Rain , 100% Dry)";
        chart.addValueAxis(valueAxis);
        // GRAPH
        var graph = new AmCharts.AmGraph();
        graph.type = "smoothedLine";
        graph.bullet = "round";
        graph.bulletColor = "#FFFFFF";
        graph.useLineColorForBulletBorder = true;
        graph.bulletBorderAlpha = 1;
        graph.bulletBorderThickness = 2;
        graph.bulletSize = 7;
        graph.title = "raindrop";
        graph.valueField = "raindrop";
        graph.lineThickness = 2;
        graph.lineColor = "#00BBCC";
        chart.addGraph(graph);
        // CURSOR

```

```

        var chartCursor = new AmCharts.ChartCursor();
        chartCursor.cursorPosition = "mouse";
        chart.addChartCursor(chartCursor);
        // SCROLLBAR
        var chartScrollbar = new AmCharts.ChartScrollbar();
        chartScrollbar.graph = graph;
        chartScrollbar.scrollbarHeight = 30;
        chart.addChartScrollbar(chartScrollbar);
        chart.creditsPosition = "bottom-right";
        // WRITE
        chart.write("chartdiv");
    });
</script>
</body>
</html>

```

Κώδικας ιστοσελίδας απεικόνισης μετρήσεων αισθητήρα υγρασίας χώματος:

```

<!DOCTYPE>
<html>
  <style>
    h1 {
      width:1400;
      margin-top:30px;
      margin-right:85px;
      margin-bottom:40px;
      margin-left:85px;
      font-weight: bold;
      font-style:normal;
      text-decoration: none;
      background-color:#32CD32;
      color:white;
      text-shadow: 2px 2px 4px #000000;
      border-radius: 30px;
      padding-top:20px ;
      padding-bottom:20px ;
      font-family: Arial;
      text-align:center;
    }
    body {
      color:#000000;
      background-color:#228B22;
      text-align: center;
    }
  </style>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Soil moisture chart</title>
  </head>
  <body>
    <h1>
      <a href="raindrop.html">
        

```



```

    </a>
    <a href="../home.html">
        
    </a>Soil moisture chart
    <a href="dht.html">
        
    </a>
    <a href="../review_data.php">
        
    </a>
</h1>
<link rel="stylesheet" href="style.css" type="text/css">
<script src="../amcharts/amcharts.js"
type="text/javascript"></script>
<script src="../amcharts/serial.js"
type="text/javascript"></script>
<script>
    AmCharts.loadJSON = function(url) {
        // create the request
        if (window.XMLHttpRequest) {
            // IE7+, Firefox, Chrome, Opera, Safari
            var request = new XMLHttpRequest();
        }
        else {
            // code for IE6, IE5
            var request = new
ActiveXObject('Microsoft.XMLHTTP');
        }
        // load it
        // the last "false" parameter ensures that our code
will wait before the
        // data is loaded
        request.open('GET', url, false);
        request.send();
        // parse and return the output
        return eval(request.responseText);
    };
</script>
<div id="chartdiv" style="width: 1400px; height: 600px;
background-color:white;margin: 0px 200px 85px 85px;border-radius:
30px; "></div>
<script>
    var chart;
    AmCharts.ready(function () {
        var chartData = AmCharts.loadJSON('data_all.php');
        // SERIAL CHART
        chart = new AmCharts.AmSerialChart();
        chart.path = "../amcharts/";
        chart.marginLeft = 0;
        chart.marginRight = 0;
        chart.marginTop = 0;
        chart.dataProvider = chartData;
    });
</script>

```

```

chart.categoryField = "event";
// AXES
// category
var categoryAxis = chart.categoryAxis;
chart.dataDateFormat = "JJ:NN";
var valueAxis = new AmCharts.ValueAxis();
valueAxis.inside = true;
valueAxis.tickLength = 0;
valueAxis.axisAlpha = 0;
valueAxis.minimum = 0;
valueAxis.maximum = 100;
chart.addValueAxis(valueAxis);
valueAxis.title = "Moisture % ( 0% Wet , 100% Dry)";
// GRAPH
var graph = new AmCharts.AmGraph();
graph.dashLength = 3;
graph.lineColor = "#663300";
graph.valueField = "moisture";
graph.dashLength = 3;
graph.bullet = "round";
chart.addGraph(graph);
// CURSOR
var chartCursor = new AmCharts.ChartCursor();
chartCursor.cursorAlpha = 0;
chart.addChartCursor(chartCursor);
// GUIDES are used to create horizontal range fills
var guide = new AmCharts.Guide();
guide.value = 0;
guide.toValue = 10;
guide.fillColor = "#0066FF";
guide.fillAlpha = 0.4;
guide.lineAlpha = 0;
valueAxis.addGuide(guide);
var guide = new AmCharts.Guide();
guide.value = 10;
guide.toValue = 20;
guide.fillColor = "#0066FF";
guide.fillAlpha = 0.35;
guide.lineAlpha = 0;
valueAxis.addGuide(guide);
var guide = new AmCharts.Guide();
guide.value = 20;
guide.toValue = 30;
guide.fillColor = "#0066FF";
guide.fillAlpha = 0.3;
guide.lineAlpha = 0;
valueAxis.addGuide(guide);
var guide = new AmCharts.Guide();
guide.value = 30;
guide.toValue = 40;
guide.fillColor = "#0066FF";
guide.fillAlpha = 0.25;
guide.lineAlpha = 0;
valueAxis.addGuide(guide);
var guide = new AmCharts.Guide();

```

```

        guide.value = 40;
        guide.toValue = 50;
        guide.fillColor = "#66FFCC";
        guide.fillAlpha = 0.2;
        guide.lineAlpha = 0;
        valueAxis.addGuide(guide);
        var guide = new AmCharts.Guide();
        guide.value = 50;
        guide.toValue = 60;
        guide.fillColor = "#66FFCC";
        guide.fillAlpha = 0.1;
        guide.lineAlpha = 0;
        valueAxis.addGuide(guide);
        var guide = new AmCharts.Guide();
        guide.value = 60;
        guide.toValue = 70;
        guide.fillColor = "#CCFF33";
        guide.fillAlpha = 0.15;
        guide.lineAlpha = 0;
        valueAxis.addGuide(guide);
        var guide = new AmCharts.Guide();
        guide.value = 70;
        guide.toValue = 80;
        guide.fillColor = "#FF9900";
        guide.fillAlpha = 0.12;
        guide.lineAlpha = 0;
        valueAxis.addGuide(guide);
        var guide = new AmCharts.Guide();
        guide.value = 80;
        guide.toValue = 90;
        guide.fillColor = "#996633";
        guide.fillAlpha = 0.3;
        guide.lineAlpha = 0;
        valueAxis.addGuide(guide);
        var guide = new AmCharts.Guide();
        guide.value = 90;
        guide.toValue = 100;
        guide.fillColor = "#996633";
        guide.fillAlpha = 0.4;
        guide.lineAlpha = 0;
        valueAxis.addGuide(guide);
        // SCROLLBAR
        var chartScrollbar = new AmCharts.ChartScrollbar();
        chartScrollbar.graph = graph;
        chartScrollbar.scrollbarHeight = 30;
        chart.addChartScrollbar(chartScrollbar);
        chart.creditsPosition = "bottom-right";
        // WRITE
        chart.write("chartdiv");
    });
</script>
</body>
</html>

```

Κώδικας ιστοσελίδας απεικόνισης μετρήσεων αισθητήρα θερμοκρασίας – υγρασίας:

```
<!DOCTYPE>
<html>
  <style>
    h1 {
      width:1400;
      margin-top:30px;
      margin-right:85px;
      margin-bottom:40px;
      margin-left:85px;
      font-weight: bold;
      font-style:normal;
      text-decoration: none;
      background-color:#32CD32;
      color:white;
      text-shadow: 2px 2px 4px #000000;
      border-radius: 30px;
      padding-top:20px ;
      padding-bottom:20px ;
      font-family: Arial;
      text-align:center;
    }
    body {
      color:#000000;
      background-color:#228B22;
      text-align: center;
    }
  </style>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>amCharts examples</title>
  </head>
  <body>
    <h1>
      <a href="moisture.html">
        
      </a>
      <a href=" ../home.html">
        
      </a>Temperature and humidity chart
      <a href="tank.html">
        
      </a>
      <a href=" ../review_data.php">
        
      </a>
    </h1>
    <link rel="stylesheet" href="style.css" type="text/css">
  </body>
</html>
```

```

        <script src="../../amcharts/amcharts.js"
type="text/javascript"></script>
        <script src="../../amcharts/serial.js"
type="text/javascript"></script>
        <script>
            AmCharts.loadJSON = function(url) {
                // create the request
                if (window.XMLHttpRequest) {
                    // IE7+, Firefox, Chrome, Opera, Safari
                    var request = new XMLHttpRequest();
                }
                else {
                    // code for IE6, IE5
                    var request = new
ActiveXObject('Microsoft.XMLHTTP');
                }
                // load it
                // the last "false" parameter ensures that our code
will wait before the
                // data is loaded
                request.open('GET', url, false);
                request.send();
                // parse and return the output
                return eval(request.responseText);
            };
        </script>
        <div id="chartdiv" style="width: 1400px; height: 600px;
background-color:white;margin: 0px 200px 85px 85px;border-radius:
30px; "></div>
        <script>
            var chart;
            AmCharts.ready(function () {
                // SERIAL CHART
                var chartData = AmCharts.loadJSON('data_all.php');
                // SERIAL CHART
                chart = new AmCharts.AmSerialChart();
                chart.dataProvider = chartData;
                chart.path = "../../amcharts/";
                chart.categoryField = "event";
                chart.dataDateFormat = "JJ:NN:SS";
                chart.marginTop = 0;
                // AXES
                // category axis
                chart.categoryAxis.dateFormats = true;
                // as we have data of different units, we create two
different value axes
                // Duration value axis
                var durationAxis = new AmCharts.ValueAxis();
                durationAxis.title = "Humidity";
                durationAxis.gridAlpha = 0;
                durationAxis.axisAlpha = 0;
                durationAxis.inside = false;
                // the following line makes this value axis to
convert values to duration
            });
        </script>

```

```

        // it tells the axis what duration unit it should
use. mm - minute, hh - hour...
        chart.addValueAxis(durationAxis);
        // Distance value axis
        var distanceAxis = new AmCharts.ValueAxis();
        distanceAxis.title = "Temperature";
        distanceAxis.gridAlpha = 0;
        distanceAxis.axisAlpha = 0;
        distanceAxis.position = "right";
        distanceAxis.inside = false;
        distanceAxis.unit = "C";
        chart.addValueAxis(distanceAxis);
        // GRAPHS
        // duration graph
        var durationGraph = new AmCharts.AmGraph();
        durationGraph.title = "Humidity";
        durationGraph.valueField = "humidity";
        durationGraph.type = "line";
        durationGraph.valueAxis = durationAxis; // indicate
which axis should be used
        durationGraph.lineColor = "#0066FF";
        durationGraph.balloonText = "[[value]]";
        durationGraph.lineThickness = 1;
        durationGraph.legendValueText = "[[value]]";
        durationGraph.bullet = "round";
        durationGraph.bulletBorderColor = "#0066FF";
        durationGraph.bulletBorderAlpha = 1;
        durationGraph.bulletBorderThickness = 1;
        chart.addGraph(durationGraph);
        // distance graph
        var distanceGraph = new AmCharts.AmGraph();
        distanceGraph.valueField = "temperature";
        distanceGraph.title = "temperature";
        distanceGraph.type = "column";
        distanceGraph.fillAlphas = 0.4;
        distanceGraph.valueAxis = distanceAxis;
        durationAxis.unit = "%";
        distanceGraph.lineColor = "#009933";
        distanceGraph.lineAlpha = 0.2;
        chart.addGraph(distanceGraph);
        // CURSOR
        var chartCursor = new AmCharts.ChartCursor();
        chartCursor.zoomable = true;
        chartCursor.categoryBalloonDateFormat = "JJ";
        chartCursor.cursorAlpha = 0;
        chartCursor.valueLineEnabled = true;
        chartCursor.valueLineBalloonEnabled = true;
        chartCursor.valueLineAxis = distanceAxis;
        chart.addChartCursor(chartCursor);
        var chartScrollbar = new AmCharts.ChartScrollbar();
        chartScrollbar.scrollbarHeight = 30;
        chartScrollbar.offset = 10;
        chart.addChartScrollbar(chartScrollbar);
        // LEGEND
        var legend = new AmCharts.AmLegend();

```

```

        legend.bulletType = "round";
        legend.equalWidths = false;
        legend.valueWidth = 120;
        legend.color = "#000000";
        legend.useGraphSettings = true;
        chart.addLegend(legend);
        // WRITE
        chart.write("chartdiv");
    });
</script>
</body>
</html>

```

Κώδικας ιστοσελίδας απεικόνισης μετρήσεων αισθητήρα στάθμης δεξαμενής:

```

<!DOCTYPE HTML" >
<html>
  <style>
    h1 {
      width:1400;
      margin-top:30px;
      margin-right:85px;
      margin-bottom:40px;
      margin-left:85px;
      font-weight: bold;
      font-style:normal;
      text-decoration: none;
      background-color:#32CD32;
      color:white;
      text-shadow: 2px 2px 4px #000000;
      border-radius: 30px;
      padding-top:20px ;
      padding-bottom:20px ;
      font-family: Arial;
      text-align:center;
    }
    body {
      color:#000000;
      background-color:#228B22;
      text-align: center;
    }
  </style>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Water tank height chart</title>
  </head>
  <body>
    <h1>
      <a href="dht.html">
        
      </a>
      <a href="../home.html">

```



```

        
    </a>Water tank height chart
    <a href="light.html">
        
    </a>
    <a href="../review_data.php">
        
    </a>
</h1>
<link rel="stylesheet" href="style.css" type="text/css">
<script src="../amcharts/amcharts.js"
type="text/javascript"></script>
<script src="../amcharts/serial.js"
type="text/javascript"></script>
<script>
    AmCharts.loadJSON = function(url) {
        // create the request
        if (window.XMLHttpRequest) {
            // IE7+, Firefox, Chrome, Opera, Safari
            var request = new XMLHttpRequest();
        }
        else {
            // code for IE6, IE5
            var request = new
ActiveXObject('Microsoft.XMLHTTP');
        }
        // load it
        // the last "false" parameter ensures that our code
will wait before the
        // data is loaded
        request.open('GET', url, false);
        request.send();
        // parse and return the output
        return eval(request.responseText);
    };
</script>
<div id="chartdiv" style="width: 1400px; height: 600px;
background-color:white;margin: 0px 200px 85px 85px;border-radius:
30px; "></div>
<script>
    var chart;
    AmCharts.ready(function () {
        var chartData = AmCharts.loadJSON('data_all.php');
        // SERIAL CHART
        chart = new AmCharts.AmSerialChart();
        chart.dataProvider = chartData;
        chart.startDuration = 1;
        chart.depth3D = 50;
        chart.angle = 30;
        chart.path = "../amcharts/";
        chart.dataProvider = chartData;
        chart.categoryField = "event";
    });
</script>

```

```

// AXES
// category
var categoryAxis = chart.categoryAxis;
categoryAxis.gridAlpha = 0;
categoryAxis.axisAlpha = 0;
categoryAxis.gridPosition = "start";
// value
var valueAxis = new AmCharts.ValueAxis();
valueAxis.axisAlpha = 0;
valueAxis.gridAlpha = 0;
chart.addValueAxis(valueAxis);
// GRAPH
var graph = new AmCharts.AmGraph();
graph.valueField = "tank";
valueAxis.unit = "cm";
graph.balloonText = "<b>[[event]]: [[tank]]</b>";
graph.type = "column";
graph.lineColor = "#0066FF";
graph.lineAlpha = 0.5;
graph.topRadius = 1;
graph.fillAlphas = 0.8;
chart.addGraph(graph);
// CURSOR
var chartCursor = new AmCharts.ChartCursor();
chartCursor.cursorAlpha = 0;
chartCursor.zoomable = true;
chartCursor.categoryBalloonEnabled = true;
chartCursor.valueLineEnabled = true;
chartCursor.valueLineBalloonEnabled = true;
chartCursor.valueLineAlpha = 0;
chart.addChartCursor(chartCursor);
chartCursor.color = "#FFFFFF";
chartCursor.cursorPosition = "mouse";
chart.creditsPosition = "top-right";
// SCROLLBAR
var chartScrollbar = new AmCharts.ChartScrollbar();
chartScrollbar.graph = graph;
chartScrollbar.scrollbarHeight = 30;
chart.addChartScrollbar(chartScrollbar);
// WRITE
chart.write("chartdiv");
});
</script>
</body>
</html>

```