



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ  
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

“Σχεδίαση και υλοποίηση θερμοκηπίου με συστήματα εποπτείας και ελέγχου μέσω διαδικτύου και δικτύου ZigBee βασισμένο στην πλατφόρμα Arduino.”

Νεκτάριος Σαλιάϊ (ΑΜ: 7409)

ΕΠΙΒΛΕΠΩΝ:

Χαδέλλης Λουκάς

Τοπάλης Ευάγγελος

ΠΑΤΡΑ, 2022

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, 2022

#### ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ονοματεπώνυμο, Υπογραφή
2. Ονοματεπώνυμο, Υπογραφή
3. Ονοματεπώνυμο, Υπογραφή

#### **Υπεύθυνη Δήλωση Φοιτητή**

Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη εργασία.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Νεκτάριου Σαλιάϊ που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο Πανεπιστήμιο Πελοποννήσου, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.

## Περίληψη

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η υλοποίηση ενός Internet Of Things (IoT) θερμοκηπίου, που θα έχει ως σκοπό τον έλεγχο και την παρακολούθηση του. Στο θερμοκήπιο που θα υλοποιηθεί θα εκτελούνται διάφορα αυτοματισμού για την καλύτερη δυνατή ανάπτυξη των φυτών. Η λειτουργία του θερμοκηπίου θα βασιστεί στην πλατφόρμα Arduino, στην χρήση συσκευών Xbee αλλά και στην δημιουργία ιστοσελίδας ελέγχου και παρακολούθησης. Στο Arduino θα συνδεθούν αισθητήρες, συσκευές εξόδου, Ethernet Shield, αλλά και μια συσκευή Xbee. Με την χρήση των Xbee το Arduino θα επικοινωνεί ασύρματα με το πρωτόκολλο ZigBee. Επομένως συνδέοντας μια συσκευή Xbee (Coordinator) με το Arduino θα επικοινωνεί ασύρματα με την δεύτερη συσκευή Xbee (Router) έτσι θα διαβάζει ασύρματα τα δεδομένα τριών αισθητήρων, αλλά και θα ελέγχει τα τρία LED που θα είναι συνδεδεμένα στο Xbee. Όλα αυτά θα ενσωματωθούν σε μια κατασκευή που θα απεικονίζει ένα θερμοκήπιο ώστε να παρουσιαστεί με το καλύτερο δυνατό τρόπο η εργασία αυτή.

## Abstract

The subject of this thesis is the implementation of an IoT greenhouse that will have the purpose of controlling and monitoring it. In the greenhouse that we will implement, various automation scenarios will be executed for the best possible growth of the plants. This diploma will be based on the Arduino platform, the use of Xbee devices and the creation of a control and monitoring website. Sensors, output devices, Ethernet Shield, but also an Xbee device will be connected to the Arduino. By using Xbee the Arduino will communicate wirelessly with the ZigBee protocol. Next by connecting an Xbee device (Coordinator) to the Arduino it will communicate wirelessly with the second Xbee device (Router) so it will wirelessly read the data of three sensors and also control the three LEDs that will be connected to the Xbee. All this will be integrated into a model that will depict a greenhouse in order to present this work in the best possible way.

## Ευχαριστίες

Οφείλω να ευχαριστήσω τους καθηγητές μου Χαδέλλη Λουκά και Τοπάλη Ευάγγελο για την πολύτιμη βοήθεια τους στην εκπόνηση της διπλωματικής μου εργασίας. Επίσης θα ήθελα να ευχαριστήσω την οικογένειά μου που με στήριξε όλο αυτό το διάστημα.

## Περιγραφή

Στην διπλωματική αυτή θα υλοποιηθεί ένα IoT θερμοκήπιο που θα έχει ως στόχο τον έλεγχο και την παρακολούθηση των κλιματικών συνθηκών για την καλύτερη δυνατή ανάπτυξη των φυτών του θερμοκηπίου αλλά και την εξοικονόμηση των πόρων.

Αρχικά πρέπει να αναφερθεί ότι το λειτουργικό κομμάτι του θερμοκηπίου βασίζεται στην πλατφόρμα Arduino και στις διάφορες συσκευές που θα συνδεθούν σε αυτόν όπως είναι ο αισθητήρας θερμοκρασίας και υγρασίας, ο αισθητήρας φωτοαντίστασης, ο αισθητήρας απόστασης, οι συσκευές εξόδου, οι συσκευές Xbee, το Ethernet shield και άλλα.

Επιπλέον, χρησιμοποιώντας τα δύο Xbee γίνεται χρήση ασύρματης επικοινωνίας ZigBee. Επομένως η μια συσκευή Xbee λειτουργεί ως Coordinator και η άλλη συσκευή λειτουργεί ως Router. Συνεπώς το Arduino θα επικοινωνεί απευθείας μέσω του πρωτοκόλλου UART με το Xbee coordinator ενώ η δεύτερη συσκευή Xbee (Router) θα στέλνει και θα λαμβάνει δεδομένα από Xbee Coordinator μέσω του πρωτοκόλλου Zigbee. Στο Xbee Router θα συνδεθούν ο αισθητήρας υγρασίας χώματος, ο αισθητήρας αερίου, ο δεύτερος αισθητήρας φωτοαντίστασης και τρία LED.

Επιπρόσθετα, θα υπάρχει η δυνατότητα απομακρυσμένου ελέγχου αλλά και παρακολούθησης μέσω του τοπικού δικτύου από την ιστοσελίδα που θα υλοποιηθεί. Για να είναι εφικτό αυτό χρειάζεται η σύνδεση ενός Ethernet Shield με το Arduino ώστε να λειτουργεί ως Web Server και η δημιουργία μιας βάσης δεδομένων. Επομένως για να γίνει η οπτικοποίηση των δεδομένων από τους αισθητήρες στην ιστοσελίδα μας χρειάζεται η δημιουργία της βάσης δεδομένων και από εκεί να γίνει η οπτικοποίηση τους μέσω ιστοσελίδων με την βοήθεια της php και των google charts.

Επίσης, μέσω της ιστοσελίδας θα υπάρχει η επιλογή της λειτουργίας του συστήματος θερμοκηπίου σε δύο καταστάσεις, είτε στην χειροκίνητη λειτουργία, είτε στην αυτόματη λειτουργία.

Στην χειροκίνητη λειτουργία ο χρήστης έχει την δυνατότητα να ενεργοποιεί και να απενεργοποιεί κάθε συσκευή εξόδου ξεχωριστά. Για να γίνει ο έλεγχος χειροκίνητα μέσω της ιστοσελίδας χρειάζεται να χρησιμοποιήσουμε της μεθόδους HTTP POST και GET έτσι θα μπορούμε να ελέγχουμε το πότισμα, τον εξαερισμό, την θέρμανση, την ψύξη, τον φωτισμό αλλά θα μπορούμε και να ελέγχουμε τα τρία LED που είναι συνδεδεμένα στην συσκευή Xbee 2 που εξομοιώνουν το σύστημα ψεκασμού ομίχλης από νερό, το σύστημα σκίασης και το σύστημα ειδοποίησης.



Στην αυτόματη λειτουργία θα εκτελούνται τα σενάρια αυτοματισμού για την διευκόλυνση των εργασιών που απαιτούνται σε ένα θερμοκήπιο όπως είναι ο έλεγχος θερμοκρασίας, υγρασίας, ποτίσματος, φωτισμού, ηλιακής ακτινοβολίας και ο έλεγχος της ασφάλειας του θερμοκηπίου.

Όλα αυτά θα ενσωματωθούν σε μια κατασκευή που θα απεικονίζει ένα θερμοκήπιο με το σύστημα ποτίσματος και 2 μικρότερες κατασκευές η πρώτη κατασκευή θα περιέχει το Arduino mega, το Xbee coordinator και τις συσκευές εισόδου εξόδου που συνδέονται σε αυτό και στη δεύτερη κατασκευή θα περιέχει το Xbee Router και τις συσκευές εισόδου εξόδου που συνδέονται σε αυτό.

## *Πίνακας περιεχομένων*

Περίληψη .....	3
Abstract.....	3
Ευχαριστίες.....	3
Περιγραφή.....	4
Ευρετήριο Εικόνων.....	10
Ευρετήριο κυκλωμάτων.....	16
Ευρετήριο σχημάτων .....	17
Ευρετήριο πινάκων .....	18
1 Εισαγωγή.....	19
1.1 Πρόλογος.....	19
1.2 Σκοπός και στόχοι .....	19
1.3 Δομικό μέρος εργασίας .....	21
2 Γενικά για το θερμοκήπιο.....	22
2.1 Τι είναι το θερμοκήπιο .....	22
2.2 Κλιματικές συνθήκες ενός θερμοκηπίου .....	22
2.3 Εξέλιξη θερμοκηπίου .....	24
2.4 Τι είναι το IoT και τα θερμοκήπια IoT.....	24
2.5 Σε τι διαφέρει η προτεινόμενη υλοποίηση από τα <i>αλλά</i> IoT θερμοκήπια .....	25
3 Τεχνολογίες, πλατφόρμες και εργαλεία που εφαρμόστηκαν.....	26
3.1 Πρωτόκολλα επικοινωνίας που χρησιμοποιήθηκαν .....	26
3.1.1 Πρόλογος .....	26
3.1.2 Universal Asynchronous Receiver/Transmitter (UART) .....	26
3.1.3 Serial Peripheral Interface (SPI) .....	27
3.1.4 Inter-Integrated Circuit (I2C).....	27
3.1.5 Ethernet.....	28
3.1.6 Hyper Text Transfer Protocol (HTTP) .....	28

3.1.7	IEEE 802.15.4.....	28
3.1.8	ZigBee.....	29
3.2	Γλώσσες προγραμματισμού που χρησιμοποιήθηκαν.....	33
3.2.1	Λογισμικό Arduino.....	33
3.2.2	Λογισμικό XCTU.....	37
3.2.3	Γλώσσες προγραμματισμού διαδικτύου.....	34
3.3	Εργαλεία και πλατφόρμες που χρησιμοποιήθηκαν για την ιστοσελίδα.....	37
3.3.1	Visual Studio Code.....	37
3.3.2	Λογισμικό XAMPP.....	39
3.3.3	Apache.....	39
3.3.4	Google Charts.....	39
3.3.5	IFTTT.....	40
3.3.6	ThingSpeak.....	40
4	Υλικά που χρησιμοποιήθηκαν.....	41
4.1	Arduino.....	41
4.1.1	Τι είναι το Arduino.....	41
4.1.2	Arduino MEGA.....	42
4.1.3	Arduino Ethernet Shield.....	44
4.2	Δίοδος.....	45
4.3	Light Emitting Diode (LED).....	45
4.4	Τρανζίστορ πεδίου ημιαγωγών οξειδίου μετάλλου (MOSFET).....	46
4.5	Ρελέ ή Ηλεκτρονόμος.....	46
4.6	Αισθητήρας θερμοκρασίας και υγρασίας DHT11.....	47
4.7	Αισθητήρας υπέρυθρων Ultrasonic HC-SR04.....	49
4.8	Αισθητήρας φωτοαντίστασης LDR.....	52
4.9	Αισθητήρας αερίου MQ-07.....	54
4.10	Αισθητήρας υγρασίας εδάφους Soil Hygrometer.....	56

4.11	Ρολόι πραγματικού χρόνου - Real Time Clock (RTC DS3231).....	58
4.12	Οθόνη υγρών κρυστάλλων LCD.....	60
4.13	Ανεμιστήρας.....	64
4.14	Λαμπτήρας πυράκτωσης.....	64
4.15	Ηλεκτρική αντλία 12 Volt.....	65
4.16	Ηλεκτρική βαλβίδα.....	65
4.17	Xbee.....	66
4.17.1	Τι είναι το Xbee.....	66
4.17.2	AT and API mode.....	68
4.17.3	Τύποι Xbee.....	71
4.17.4	Επικοινωνία Xbee με τον υπολογιστή και Arduino.....	73
4.17.5	Διαμόρφωση Xbee μέσω XCTU.....	74
4.17.6	Οι διαμόρφωση του Xbee Router μέσω του Arduino.....	89
4.18	Υλικά κατασκευής.....	91
5	Περιγραφή σεναρίων αυτοματισμού θερμοκηπίου.....	108
5.1	Πρόλογος.....	108
5.2	Περιγραφή σεναρίου αυτοματισμού ελέγχου θερμοκρασίας.....	108
5.3	Περιγραφή σεναρίου αυτοματισμού ελέγχου ποτίσματος.....	113
5.4	Περιγραφή σεναρίου αυτοματισμού ελέγχου της ηλιακής ακτινοβολίας.....	119
5.5	Περιγραφή σεναρίου αυτοματισμού επέκτασης φωτισμού.....	124
5.6	Περιγραφή σεναρίου αυτοματισμού ελέγχου εξαερισμού.....	131
5.7	Περιγραφή σεναρίου αυτοματισμού ελέγχου ψεκασμού ομίχλης.....	136
5.8	Περιγραφή σεναρίου αυτοματισμού ελέγχου συναγερμού.....	142
5.9	Το συνολικό κύκλωμα του IoT Θερμοκηπίου.....	149
6	Δημιουργία ιστοσελίδας απομακρυσμένου ελέγχου και παρακολούθησης.....	154
6.1	Πρόλογος.....	154
6.2	Δημιουργία βάσης δεδομένων.....	154

6.3	Σύνδεση Arduino με την βάση δεδομένων .....	155
6.4	Δημιουργία αρχείου Index για την πλοήγηση στην ιστοσελίδα .....	162
6.5	Οπτικοποίηση δεδομένων σε μορφή πίνακα.....	164
6.6	Οπτικοποίηση δεδομένων σε μορφή γραφικών πραγματικού χρόνου.....	166
6.7	Οπτικοποίηση στατιστικών .....	172
6.8	Οπτικοποίηση δεδομένων σε μορφή μετρητών Gauge.....	185
6.9	ThingSpeak.....	190
6.9.1	Αποστολή δεδομένων στο ThingSpeak .....	190
6.9.2	Σύνδεση ThingSpeak με IFTTT.....	197
6.9.3	Οπτικοποίηση δεδομένων μέσω του ThingSpeak .....	210
6.10	Έλεγχος συσκευών εξόδου.....	221
7	Συμπεράσματα και προτάσεις περαιτέρω ανάπτυξης.....	227
7.1	Συμπεράσματα και ανακεφαλαιώσεαι.....	227
7.2	Προτάσεις περαιτέρω ανάπτυξης.....	227
	Παράρτημα.....	229
	Τελικός κώδικας Arduino .....	229
	Βιβλιογραφία .....	246

## Ευρετήριο Εικόνων

Εικόνα 2.1 Απεικόνιση εξωτερικής πλευράς ενός θερμοκηπίου. Πηγή .....	22
Εικόνα 3.1 Απεικόνιση λειτουργίας του Universal Asynchronous Receiver/Transmitter. Πηγή.....	26
Εικόνα 3.2 Απεικόνιση λειτουργίας Serial Peripheral Interface. Πηγή.....	27
Εικόνα 3.3 Απεικόνιση λειτουργίας Inter-Integrated Circuit. Πηγή .....	27
Εικόνα 3.4 Λογότυπο ZigBee. Πηγή .....	29
Εικόνα 3.5 Αναπαράσταση τοπολογίας αστέρα. ....	31
Εικόνα 3.6 Αναπαράσταση Τοπολογίας δέντρου .....	32
Εικόνα 3.7 Αναπαράσταση Τοπολογίας πλέγματος. ....	32
Εικόνα 3.8 Στιγμιότυπο οθόνης του Arduino IDE. ....	33
Εικόνα 3.9 Το περιβάλλον του λογισμικού XCTU .....	37
Εικόνα 3.10 Logo Hyper Text Markup Language. Πηγή .....	34
Εικόνα 3.11 Δομή εγγράφου HTML. Πηγή .....	34
Εικόνα 3.12 Λογότυπο CSS. Πηγή.....	35
Εικόνα 3.13 Λογότυπο JavaScript. Πηγή .....	35
Εικόνα 3.14 Logo Personal Home Pag. Πηγή .....	36
Εικόνα 3.15 Το Logo της MySQL. Πηγή.....	37
Εικόνα 3.16 Το λογότυπο του Visual Studio Code. Πηγή.....	38
Εικόνα 3.17 Το λογότυπο της XAMPP. Πηγή .....	39
Εικόνα 3.18 Λογότυπο Google Charts.Πηγή.....	39
Εικόνα 3.19 Λογότυπο IFTTT. Πηγή .....	40
Εικόνα 3.20 Το κανάλι ThingSpeak που χρησιμοποιούμε για την απεικόνιση των γραφικών μας.....	40
Εικόνα 4.1 Το λογότυπο του Arduino. Πηγή.....	41
Εικόνα 4.2 Πλακέτα Arduino Uno μπροστινή όψη. ....	41
Εικόνα 4.3 Πλακέτα Arduino Uno πίσω όψη .....	42
Εικόνα 4.4 Η πλακέτα Arduino mega με τα μέρη που αποτελείται.Πηγή.....	43
Εικόνα 4.5 Ασπίδα Arduino Ethernet Shield. Πηγή .....	44
Εικόνα 4.6 Δίοδοι 1N4007.....	45
Εικόνα 4.7 Απεικόνιση Light-emitting diode. Πηγή .....	45
Εικόνα 4.8 Τρανζίστορ MOSFET IRF540N. ....	46
Εικόνα 4.9 Ρελέ 5 Volt DC .....	46
Εικόνα 4.10 Μπροστινή όψη αισθητήρας θερμοκρασίας DHT11. ....	47

Εικόνα 4.11 Απεικόνιση αισθητήρα υπερήχων και σημάτων. Πηγή.....	49
Εικόνα 4.12 Διάγραμμα χρονισμού υπερήχων HC-SR04. Πηγή.....	49
Εικόνα 4.13 Αισθητήρας υπέρυθρων Ultrasonic HC-SR04. Πηγή .....	50
Εικόνα 4.14 Απεικόνιση αισθητήρα φωτοαντίστασης.....	52
Εικόνα 4.15 Απεικόνιση αισθητήρα αερίου. Πηγή .....	54
Εικόνα 4.16 Απεικόνιση αισθητήρα υγρασίας εδάφους. Πηγή.....	56
Εικόνα 4.17 Απεικόνιση συσκευής ρολογιού πραγματικού χρόνου DC3231. Πηγή .....	58
Εικόνα 4.18 Απεικόνιση Οθόνης υγρών κρυστάλλων.....	60
Εικόνα 4.19 Απεικόνιση σύνδεσης οθόνης LCD με το Chip PCF8574. Πηγή .....	60
Εικόνα 4.20 Παρουσίαση αποτελέσματος του κώδικα.....	63
Εικόνα 4.21 Ανεμιστήρας υπολογιστή 12 Volt. ....	64
Εικόνα 4.22 Τα μέρη από τα οποία αποτελείται ένας λαμπτήρας πυράκτωσης. Πηγή .....	64
Εικόνα 4.23 Ηλεκτρική αντλία 12 Volt .....	65
Εικόνα 4.24 Ηλεκτρική βαλβίδα 12 Volt .....	65
Εικόνα 4.25 Το Xbee S2. ....	66
Εικόνα 4.26 Σύνταξη μιας εντολής AT. ....	68
Εικόνα 4.27 Διαφορετικοί διαθέσιμοι τύποι κεραιών για τα Xbee.Πηγή.....	72
Εικόνα 4.28 Απεικονίζει Spark fun Xbee Explorer USB μπροστινή και πίσω όψη. Πηγή .	73
Εικόνα 4.29 Το λογισμικό XCTU.....	75
Εικόνα 4.30 Οι θύρες που είναι συνδεδεμένες οι συσκευές.....	75
Εικόνα 4.31 Η παράμετροι που έχουμε ορίσει για τις θύρες.....	76
Εικόνα 4.32 Η δύο συσκευές που είχαμε συνδέσει ο coordinator και το router .....	76
Εικόνα 4.33 Το πίσω μέρος του Xbee που φαίνεται κυκλωμένο το Product Family .....	77
Εικόνα 4.34 Το Firmware που απαιτητέ για το Coordinator. ....	78
Εικόνα 4.35 Το Firmware που απαιτητέ για το Router. ....	78
Εικόνα 4.36 Η ρυθμίσεις που έχουμε ορίσει για των Coordinator στην πρώτη μορφή εργασίας. ....	79
Εικόνα 4.37 Η ρυθμίσεις για το Xbee Router το Pan ID και το JV.....	80
Εικόνα 4.38 Η ρυθμισει NI Node Identifier. ....	81
Εικόνα 4.39 Το IR που έχει οριστεί ως 1388.....	81
Εικόνα 4.40 Εδώ φαίνεται η κονσόλα. ....	82
Εικόνα 4.41 Το εργαλείο στο οποίο δημιουργούμε τα Frame.....	83
Εικόνα 4.42 Το Frame type 0x10- Transmit Request.....	84
Εικόνα 4.43 Το Frame type 0x08 AT-Command. ....	85

Εικόνα 4.44 Το Frame type 0x10 AT-Command. ....	86
Εικόνα 4.45 Το frame που θα σταλθεί σε δεκαεξαδικό σύστημα.....	87
Εικόνα 4.46 Εδώ φαίνεται με μπλε χρώμα το πλαίσιο που στείλαμε.....	87
Εικόνα 4.47 Εδώ εμφανίζεται το πλαίσιο που έλαβε ο Coordinator. ....	88
Εικόνα 4.48 Απεικόνιση δικτύου Zigbee μέσω του τρίτου μέρους εργασίας XCTU. ....	89
Εικόνα 4.49 Απεικόνιση από τις ρυθμίσεις του XCTU.....	91
Εικόνα 4.50 Συνολική κατασκευή (μπροστινή όψη).....	91
Εικόνα 4.51 Συνολική κατασκευή (πίσω όψη).....	92
Εικόνα 4.52 Σχέδιο 3D του θερμοκηπίου με τις διαστάσεις του μέσω της πλατφόρμας Tinkercad. ....	92
Εικόνα 4.53 Αριστερή όψη θερμοκήπιο που απεικονίζει των ανεμιστήρα εξαερισμού.. ...	93
Εικόνα 4.54 Δεξιά όψη που απεικονίζει των ανεμιστήρα ψύξης και το εξωτερικό LDR. ...	94
Εικόνα 4.55 Το εξωτερικό μέρος από το κουτί που φιλόξενη το Arduino Mega.....	95
Εικόνα 4.56 Η πίσω όψη με την τροφοδοσίες και την σύνδεση Ethernet. ....	95
Εικόνα 4.57 Το εσωτερικό του κουτιού.....	96
Εικόνα 4.58 Το εσωτερικό από το κουτί που φιλοξενεί το Arduino Mega.....	96
Εικόνα 4.59 Το καπάκι του κουτιού με της τοποθετημένες συσκευές.....	97
Εικόνα 4.60 Το Arduino MEGA.....	97
Εικόνα 4.61 Πλακέτα ελέγχου των Fan και της LED ταινίας. ....	98
Εικόνα 4.62 Τα Relay για την ηλεκτρική βαλβίδα και την αντήλια.....	98
Εικόνα 4.63 Το Relay της λάμπας θερμότητας. ....	99
Εικόνα 4.64 Πίσω όψη του ρολογιού πραγματικού χρόνου. ....	99
Εικόνα 4.65 Μπροστινή όψη ρολογιού πραγματικού χρόνου. ....	99
Εικόνα 4.66 Μπροστινή όψη του κουτιού Xbee ....	100
Εικόνα 4.67 Η πάνω όψη από το κουτί που φιλοξενεί το Xbee Router. ....	101
Εικόνα 4.68 Πίσω όψη του κουτιού Xbee. ....	101
Εικόνα 4.69 Το εσωτερικό του Xbee Box. ....	102
Εικόνα 4.70 Το καπάκι από το Xbee box με της συσκευές εισόδου, εξόδου και το Xbee	102
Εικόνα 4.71 Η πίσω όψη του Xbee και του αισθητήρα MQ-07 στο καπάκι του Xbee Box. ....	103
Εικόνα 4.72 Ο διαιρέτης τάσης του αισθητήρα υγρασίας εδάφους στο καπάκι του Xbee Box. ....	103
Εικόνα 4.73 Η πίσω όψη της πλακέτας του εσωτερικού LDR στο καπάκι του Xbee Box. ....	104



Εικόνα 4.74 Η πλακέτα των συσκευών εξόδου στο καπάκι του Xbee Box. ....	104
Εικόνα 4.75 Η κλέμα διακλαδώσεις και το Power Bank που τροφοδοτεί το Xbee και της συσκευές. ....	105
Εικόνα 4.76 Ο αισθητήρας υγρασίας εδάφους και ο σωλήνας ποτίσματος .....	105
Εικόνα 4.77 Το συνολικό σύστημα ποτίσματος. ....	106
Εικόνα 4.78 Το δοχείο, η ηλεκτρική βαλβίδα, η αντλία και ο αισθητήρας απόστασης. ....	107
Εικόνα 5.1 Μέση διάρκεια ημέρας στην Αθήνα. Πηγή.....	126
Εικόνα 5.2 Εικόνα από το θερμοκήπιο που υλοποιήσαμε. ....	152
Εικόνα 5.3 Πάνο όψη θερμοκηπίου. ....	153
Εικόνα 5.4 Αριστερή όψη του θερμοκηπίου. ....	153
Εικόνα 6.1 Εκκίνηση του Apache και της MySQL από τον πίνακα ελέγχου.....	154
Εικόνα 6.2 Απεικονίσει της βάσεις δεδομένων MySQL. ....	155
Εικόνα 6.3 Πίνακας βάσεις δεδομένων .....	162
Εικόνα 6.4 Αρχική σελίδα της ιστοσελίδας. ....	164
Εικόνα 6.5 Στιγμιότυπο οθόνης ιστοσελίδας όπου απεικονίζεται ο πίνακας δεδομένων...166	166
Εικόνα 6.6 Γραφική παράσταση της θερμοκρασίας. ....	169
Εικόνα 6.7 Γραφική παράσταση υγρασίας. ....	170
Εικόνα 6.8 Γραφική παράσταση εξωτερικής φωτοαντίστασης. ....	170
Εικόνα 6.9 Γραφική παράσταση εσωτερικής φωτοαντίστασης.....	170
Εικόνα 6.10 Γραφική παράσταση επιπέδου νερού. ....	171
Εικόνα 6.11 Γραφική παράσταση υγρασίας εδάφους.....	171
Εικόνα 6.12 Γραφική παράσταση διοξειδίου του άνθρακα. ....	171
Εικόνα 6.13 Γράφημα εβδομαδιαίων στατιστικών θερμοκρασίας. ....	175
Εικόνα 6.14 Γράφημα εβδομαδιαίων στατιστικών υγρασίας. ....	176
Εικόνα 6.15 Γράφημα εβδομαδιαίων στατιστικών εξωτερικής φωτοαντίστασης.....	176
Εικόνα 6.16 Γράφημα εβδομαδιαίων στατιστικών επιπέδου νερού. ....	177
Εικόνα 6.17 Γράφημα εβδομαδιαίων στατιστικών υγρασίας εδάφους. ....	177
Εικόνα 6.18 Γράφημα εβδομαδιαίων στατιστικών εσωτερικής φωτοαντίστασης. ....	178
Εικόνα 6.19 Γράφημα εβδομαδιαίων στατιστικών διοξειδίου του άνθρακα.....	178
Εικόνα 6.20 Γράφημα μηνιαίων στατιστικών θερμοκρασίας.....	182
Εικόνα 6.21 Γράφημα μηνιαίων στατιστικών υγρασίας.....	182
Εικόνα 6.22 Γράφημα μηνιαίων στατιστικών εξωτερικής φωτοαντίστασης. ....	183
Εικόνα 6.23 Γράφημα μηνιαίων στατιστικών επιπέδου νερού.....	183
Εικόνα 6.24 Γράφημα μηνιαίων στατιστικών υγρασίας εδάφους. ....	184

Εικόνα 6.25 Γράφημα μηνιαίων στατιστικών εσωτερικής φωτοαντίστασης. ....	184
Εικόνα 6.26 Γράφημα μηνιαίων στατιστικών διοξειδίου του άνθρακα. ....	185
Εικόνα 6.27 Οι έξι μετρητές που απεικονίζουν της τιμές των αισθητήρων. ....	190
Εικόνα 6.28 Το κανάλι Arduino Sensor Data. ....	190
Εικόνα 6.29 Οι ρυθμίσεις στο ThingSpeak για την δημιουργία καναλιού. ....	191
Εικόνα 6.30 Τα κλειδιά ThingSpeak. ....	192
Εικόνα 6.31 Απεικόνιση της αρχικής σελίδας IFTTT. ....	198
Εικόνα 6.32 Επιλογή Create για δημιουργία Applet. ....	198
Εικόνα 6.33 Επιλογή του If This. ....	198
Εικόνα 6.34 Επιλογή υπηρεσίας Webhooks. ....	199
Εικόνα 6.35 Επιλογή του Receive a web request. ....	199
Εικόνα 6.36 Συμπλήρωση πεδίου Event Name. ....	200
Εικόνα 6.37 Επιλογή του Then That. ....	200
Εικόνα 6.38 Επιλογή υπηρεσίας Email. ....	200
Εικόνα 6.39 Επιλογή της δράσης Send me an email. ....	201
Εικόνα 6.40 Συντάξει του email που θα σταλθεί. ....	201
Εικόνα 6.41 Ολοκλήρωση του Applet. ....	202
Εικόνα 6.42 Ονομασία του Applet. ....	202
Εικόνα 6.43 Επιλογή του My services. ....	203
Εικόνα 6.44 Επιλογή του Webhooks. ....	203
Εικόνα 6.45 Εισαγωγή στα Documentation του Webhooks. ....	204
Εικόνα 6.46 Τοποθέτησε event name. ....	204
Εικόνα 6.47 Το URL για την αποστολή του Email. ....	205
Εικόνα 6.48 Η δοκιμή του URL. ....	205
Εικόνα 6.49 Επιλογή του Thing HTTP. ....	205
Εικόνα 6.50 Η σελίδα δημιουργίας Thing HTTP. ....	206
Εικόνα 6.51 Η απαραίτητες ρύθμισης του Thing HTTP. ....	206
Εικόνα 6.52 Επιλογή του React από το menu ThingSpeak. ....	206
Εικόνα 6.53 Επιλογή του New React. ....	207
Εικόνα 6.54 Η απαραίτητες ρυθμίσεις στο React ώστε να σταλθεί το email. ....	207
Εικόνα 6.55 Το email που στάλθηκε. ....	208
Εικόνα 6.56 Τα τρία Applets που δημιουργήθηκαν στο IFTTT. ....	208
Εικόνα 6.57 Τα τρία ThingHTTP που δημιουργήθηκαν στο ThingSpeak. ....	209
Εικόνα 6.58 Τα τρία React που δημιουργήθηκαν στο ThingSpeak. ....	209

Εικόνα 6.59 Προσθήκη των Public γραφικών στο ThingSpeak.....	210
Εικόνα 6.60 Στιγμιότυπο γραφικής παράστασης μέσω ανάλυσης MATLAB. ....	210
Εικόνα 6.61 Εισαγωγή κώδικα Visualize correlation between temperature and humidity στο MATLAB Code. ....	212
Εικόνα 6.62 Η έξοδος του κώδικα Visualize correlation between temperature and humidity .....	212
Εικόνα 6.63 Διακύμανσή της θερμοκρασίας για τις τελευταίες 10 ώρες.....	213
Εικόνα 6.64 Η θερμοκρασία για τρεις ξεχωριστές ημέρες σε ένα γράφημα. ....	215
Εικόνα 6.65 Προσθήκη γραφικών από το ThingSpeak στην ιστοσελίδα.....	215
Εικόνα 6.66 Εμφάνιση γραφικών από το ThingSpeak. ....	218
Εικόνα 6.67 Στιγμιότυπο οθόνης που φαίνεται η επιλογή Add Widgets. ....	218
Εικόνα 6.68 Στιγμιότυπο οθόνης που φαίνονται οι επιλογές των Widgets που προσφέρει το ThingSpeak. ....	219
Εικόνα 6.69 Σταθερές τιμές στο ThingSpeak στην Ιστοσελίδα. ....	221
Εικόνα 6.70 Απεικόνιση της ενεργοποίησης σε αυτόματη επιλογή.....	226
Εικόνα 6.71 Απεικόνιση σε χειροκίνητη επιλογή όλα τα Switch κλειστά. ....	226

## Ευρετήριο κυκλωμάτων

Κύκλωμα 4.1 Σύνδεση Arduino MEGA με τον αισθητήρα DHT11. ....	48
Κύκλωμα 4.2 Σύνδεση Arduino MEGA με αισθητήρα Ultrasonic HC-SR04 .....	51
Κύκλωμα 4.3 Σύνδεση Arduino MEGA με αισθητήρα LDR. ....	53
Κύκλωμα 4.4 Σύνδεση Arduino MEGA με αισθητήρα MQ-07. ....	55
Κύκλωμα 4.5 Σύνδεση Arduino MEGA με αισθητήρα Soil Hygrometer sensor. ....	57
Κύκλωμα 4.6 Σύνδεση Arduino MEGA με ρολόι RTC. ....	59
Κύκλωμα 4.7 Σύνδεση Arduino MEGA με οθόνη LCD, ρολόι RTC DS3231 και αισθητήρα DHT11. ....	61
Κύκλωμα 5.1 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου θερμοκρασίας. ...	109
Κύκλωμα 5.2 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου ποτίσματος. ....	114
Κύκλωμα 5.3 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου της ηλιακής ακτινοβολίας. ....	120
Κύκλωμα 5.4 Συνολική σύνδεση σεναρίου αυτοματισμού επέκτασης φωτισμού. ....	125
Κύκλωμα 5.5 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου εξαερισμού. ....	132
Κύκλωμα 5.6 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου ψεκασμού ομίχλης από νερό. ....	137
Κύκλωμα 5.7 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου συναγερμού. ....	143
Κύκλωμα 5.8 Το συνολικό κύκλωμα που συνδέεται στο Arduino mega. ....	149
Κύκλωμα 5.9 Το συνολικό κύκλωμα που συνδέετε στο Xbee Router. ....	150

## Ευρετήριο σχημάτων

Σχήμα 5.1 Το λογικο διαγραμμα του σεναρίου αυτοματισμού ελέγχου θερμοκρασίας.....	111
Σχήμα 5.2 Το λογικο διαγραμμα από το σύστημα ποτίσματος 1. ....	115
Σχήμα 5.3 Το λογικο διαγραμμα από το σύστημα ποτίσματος. ....	116
Σχήμα 5.4 Το λογικο διαγραμμα από του σεναρίου αυτοματισμού ελέγχου της ηλιακής ακτινοβολίας. ....	121
Σχήμα 5.5 Σενάριο αυτοματισμού επέκτασης φωτισμού. ....	127
Σχήμα 5.6 Σεναρίου αυτοματισμού ελέγχου εξαερισμού.....	133
Σχήμα 5.7 Σεναρίου αυτοματισμού ελέγχου ψεκασμού ομίχλης από νερό.....	138
Σχήμα 5.8 Σεναρίου αυτοματισμού ελέγχου συναγερμού.....	144

## Ευρετήριο πινάκων

Πίνακας 4.1 Σύγκριση των υποστηριζόμενων πρωτοκόλλων Arduino Mega. Πηγή.....	43
Πίνακας 4.2 Περιγραφή ακίδων Xbee. Πηγή .....	67
Πίνακας 4.3 Δομή δεδομένων σε λειτουργία API. Πηγή .....	69
Πίνακας 4.4 Οι υπόλοιποι τύποι πλαισίου λειτουργίας API (Byte 3). Πηγή.....	70
Πίνακας 4.5 Οι υπόλοιποι τύποι Receiver Option από το 14 Byte. Πηγή.....	71
Πίνακας 4.6 Η περίπτωση που έχει ενεργοποιηθεί το D3.Πηγή .....	71

# 1 Εισαγωγή

## 1.1 Πρόλογος

Η γεωργία είναι ένα σημαντικό κομμάτι της κοινωνίας μας, καθώς παρέχει τροφή στους ανθρώπους. Λόγω, όμως, της κλιματικής αλλαγής, της μείωσης των πόρων αλλά και την αύξηση του πληθυσμού οι αγρότες αντιμετωπίζουν προβλήματα, γιατί οι καιρικές συνθήκες έχουν γίνει πιο απρόβλεπτες και μερικές φορές γίνονται ακραίες, με αποτέλεσμα να καταστρέφονται καλλιέργειες και να χάνουν χρόνο και χρήμα με αποτέλεσμα να παλεύουν για μια υγιή επιχείρηση. Για αυτό το λόγο γίνεται χρήση των θερμοκηπίων όπου μπορούμε να ελέγχουμε τις κλιματικές συνθήκες του μικροκλίματος. Καλλιεργώντας, όμως, τα φυτά στο θερμοκήπιο υπάρχει η ανάγκη καταγραφής, χειρισμού διάφορων παραμέτρων, όπως της θέρμανσης, της ψύξης, της υγρασίας, του φωτισμού, του αερισμού, της σκίασης, του ποτίσματος και διάφορων άλλων αναγκών που έχει κάθε φυτό και όλα αυτά πρέπει να γίνονται από τον αγρότη χωρίς να παραμελούνται. Καθώς, όμως, η τεχνολογία αναπτύσσεται, οι αγρότες στρέφονται σε νέες τεχνολογίες όπως το Internet Of Things (IoT) που κάνουν αυτοματοποιημένες διεργασίες, απομακρυσμένο έλεγχο και παρακολούθηση αυξάνοντας την ποιότητα της παραγωγής και την ανθεκτικότητα των καλλιεργειών εξοικονομώντας χρήματα, ενέργεια, νερό και όλα αυτά χωρίς την απαίτηση να βρίσκεται στο θερμοκήπιο ο αγρότης.

## 1.2 Σκοπός και στόχοι

Το αντικείμενο της διπλωματικής εργασίας είναι η υλοποίηση ενός Internet Of Things (IoT) θερμοκηπίου που θα έχει ως σκοπό τον έλεγχο και την παρακολούθηση των κλιματικών συνθηκών για την καλύτερη δυνατή ανάπτυξη των φυτών αλλά και την εξοικονόμηση των πόρων.

Η στόχοι της διπλωματικής αυτής είναι η κατανόηση των μικροελεγκτών, του πρωτοκόλλου Zigbee και άλλων πρωτοκόλλων επικοινωνίας, των βάσεων δεδομένων, την γνωριμία με τις πλατφόρμες του ThingSpeak, του IFTTT και την ανάπτυξη δεξιοτήτων σε γλώσσες προγραμματισμού όπως C++, HTML, CSS, JS, PHP, SQL.

Η διπλωματική αυτή θα βασιστεί στην πλατφόρμα Arduino, στην χρήση συσκευών Xbee για ασύρματη επικοινωνία ZigBee αλλά και στην δημιουργία ιστοσελίδας ελέγχου και παρακολούθησης. Στο Arduino θα συνδεθούν διάφοροι αισθητήρες, συσκευές εξόδου, Ethernet Shield, αλλά και συσκευές Xbee που θα χρησιμοποιηθούν για την ασύρματη επικοινωνία ZigBee.

Επίσης, χρησιμοποιώντας δύο Xbee και διαμορφώνοντας με τον κατάλληλο τρόπο που θα αναφερθεί στα επόμενα κεφάλαια γίνεται χρήση ασύρματης επικοινωνίας ZigBee. Επομένως, η μια συσκευή Xbee λειτουργεί ως coordinator και η άλλη συσκευή λειτουργεί ως Router. Συνεπώς το Arduino θα επικοινωνεί απευθείας μέσω του πρωτοκόλλου UART με το Xbee coordinator, ενώ η δεύτερη συσκευή Xbee Router θα στέλνει και θα λαμβάνει δεδομένα από Xbee coordinator μέσω του πρωτοκόλλου ZigBee. Το Xbee Router θα στέλνει τα δεδομένα από τον αισθητήρα υγρασίας χώματος, τον αισθητήρα αερίου και από τον δεύτερο αισθητήρα φωτοαντίστασης αλλά και το Arduino μέσω του Xbee coordinator θα ελέγχει το σύστημα σκίασης, το σύστημα ειδοποίησης ακραίου συμβάντος και το σύστημα ψεκασμού.

Πιο συγκεκριμένα, στο σύστημα του θερμοκηπίου θα εκτελούνται διάφορα σενάρια αυτοματισμού για την καλύτερη δυνατή ανάπτυξη των φυτών. Ο σκοπός του συστήματος είναι να ελέγχονται οι κλιματικές αλλαγές για την βελτίωση της καλλιέργειας των φυτών με την παράλληλη εξοικονόμηση ενέργειας.

Το σύστημα του θερμοκηπίου θα αποτελείται από 7 σενάρια αυτοματισμού τα οποία είναι:

1. Το σενάριο αυτοματισμού ελέγχου θερμοκρασίας που θα ελέγχει την εσωτερική θερμοκρασία του θερμοκηπίου.
2. Το σενάριο αυτοματισμού ελέγχου εξαερισμού συμβάλει στον εμπλουτισμό του οξυγόνου, στην μείωση της υγρασίας και στον αερισμό του χώρου σε περίπτωση φωτιάς.
3. Το σενάριο αυτοματισμού ελέγχου συναγερμού που θα ενεργοποιείται σε περίπτωση φωτιάς ή υπερχειλίσης του νερού από την δεξαμενή.
4. Το σενάριο αυτοματισμού ελέγχου ψεκασμού ομίχλης από νερό που είναι για την αύξηση της υγρασίας περιβάλλοντος αλλά και για την πυρόσβεση.
5. Το σενάριο αυτοματισμού ελέγχου ποτίσματος που ανάλογα με την υγρασία του χώματος θα γίνει το πότισμα των φυτών.
6. Το σενάριο αυτοματισμού του σκίαστρου που χρησιμοποιείται για την μείωση της ηλιακής ακτινοβολίας.
7. Το σενάριο αυτοματισμού ελέγχου επέκτασης φωτισμού που είναι για την επέκταση του φωτός τους χειμερινούς μήνες αλλά και τις βροχερές μέρες.

Το Arduino θα λαμβάνει τα δεδομένα από μερικούς αισθητήρες ασύρματα μέσω δικτύου ZigBee χρησιμοποιώντας τα module Xbee και μερικά δεδομένα ενσύρματα από τους αισθητήρες που με την βοήθεια ενός Ethernet Shield θα στέλνονται σε μια βάση δεδομένων.



Με συνδυασμό της HTML/CSS/PHP/MySQL θα διαμορφωθεί μια ιστοσελίδα που θα υπάρχει η δυνατότητα ελέγχου και παρακολούθησης του θερμοκηπίου. Επίσης τρέχοντα και στατιστικά δεδομένα θα παρουσιαστούν σε ιστοσελίδες με την βοήθεια των google charts και του ThingSpeak.

Όλα αυτά θα ενσωματωθούν σε μια κατασκευή που θα απεικονίζει ένα θερμοκήπιο, η οποία θα περιέχει 2 μικρότερες κατασκευές: η πρώτη κατασκευή θα περιέχει το Arduino mega, το Xbee 1 (Coordinator) και τις συσκευές εισόδου/εξόδου και η δεύτερη κατασκευή θα περιέχει το Xbee 2 (Router) μαζί με τις συσκευές εισόδου/εξόδου και την τροφοδοσία του. Με την υλοποίηση αυτής της κατασκευής η διπλωματική θα παρουσιαστεί με το καλύτερο δυνατό τρόπο.

### 1.3 Δομικό μέρος εργασίας

Αρχικά στο δεύτερο κεφάλαιο, θα αναπτυχθεί η έννοια του θερμοκηπίου και πως έχει εξελιχθεί αυτή μέχρι και σήμερα. Επίσης θα γίνει αναφορά στις ιδανικές συνθήκες που χρειάζονται τα φυτά σε ένα θερμοκήπιο.

Στο τρίτο κεφάλαιο αναλύονται τα πρωτόκολλα επικοινωνίας, οι γλώσσες προγραμματισμού, οι πλατφόρμες αλλά και οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της διπλωματικής.

Στο τέταρτο κεφάλαιο παρουσιάζεται το υλικό μέρος που επιλέχθηκε.

Στο πέμπτο κεφάλαιο αναλύονται τα σενάρια αυτοματισμού ελέγχου που υλοποιήθηκαν.

Το έκτο κεφάλαιο αφορά την υλοποίηση της ιστοσελίδας για τον έλεγχο και την παρακολούθηση του θερμοκηπίου.

Τέλος, στο έβδομο κεφάλαιο περιγράφονται τα συμπεράσματα της διπλωματικής εργασίας αλλά και προτάσεις για περαιτέρω ανάπτυξη.

## 2 Γενικά για το θερμοκήπιο

### 2.1 Τι είναι το θερμοκήπιο



Εικόνα 2.1 Απεικόνιση εξωτερικής πλευράς ενός θερμοκηπίου. [Πηγή](#)

Το θερμοκήπιο είναι ένας κλειστός χώρος κατασκευασμένος συνήθως από γυαλί ή από κάποιο άλλο διαφανές υλικό, ώστε να εγκλωβίζεται η θερμική ενέργεια του ηλίου, για να αυξηθεί η θερμοκρασία μέσα στο θερμοκήπιο. Χρησιμοποιώντας το θερμοκήπιο μας δίνεται η δυνατότητα, λόγω του ότι είναι κλειστός χώρος, να ελέγχονται οι κλιματικές συνθήκες εντός του θερμοκηπίου, ανάλογα με τις απαιτήσεις της καλλιέργειας. [1] [2]

Η θερμοκρασία αυξάνεται στο θερμοκήπιο από την ηλιακή ενέργεια λόγω του ότι τα διαφανή υλικά από τα οποία αποτελείται το θερμοκήπιο επιτρέπουν την ηλιακή ακτινοβολία να εισέρχεται μέσα σε αυτό. Αυτή η ηλιακή ενέργεια απορροφάται από το έδαφος και από τα αντικείμενα που βρίσκονται μέσα στο θερμοκήπιο, με αποτέλεσμα να αυξάνεται η θερμοκρασία του περιβάλλοντος. Αφού το θερμοκήπιο είναι μία κλειστή κατασκευή, δεν επιτρέπει τον αέρα, που έχει γίνει θερμότερος λόγω των αντικειμένων και του εδάφους, να βγει έξω από αυτό και έτσι αυξάνεται η θερμοκρασία στο εσωτερικό του θερμοκηπίου. [2]

### 2.2 Κλιματικές συνθήκες ενός θερμοκηπίου

Για την σωστή ανάπτυξη των φυτών απαιτούνται συγκεκριμένες κλιματικές συνθήκες. Ωστόσο πολλές χώρες σε όλο τον κόσμο δεν έχουν αυτές τις ιδανικές κλιματικές συνθήκες που θα αναφέρουμε παρακάτω. Κάνοντας έτσι χρήση ενός θερμοκηπίου μπορούμε να ρυθμίζουμε τις κλιματικές συνθήκες που απαιτούνται για κάθε φυτό.

Οι κλιματικές συνθήκες που είναι απαραίτητες να ρυθμιστούν συνήθως σε ένα θερμοκήπιο, ώστε να αποφευχθούν προβλήματα στην ανάπτυξη των φυτών και να βοηθηθεί η φωτοσύνθεση, αλλά και η αναπνοή των φυτών είναι: η θερμοκρασία, η υγρασία εδάφους, η

υγρασία του αέρα, η ηλιακή ακτινοβολία, η φωτεινότητα του εσωτερικού χώρου και ο εμπλουτισμός του οξυγόνου. [3]

Όταν σε ένα θερμοκήπιο υπάρχουν υψηλές θερμοκρασίες, τα φυτά έχουν μειωμένη βιοσύνθεση χλωροφύλλης και ως εκ τούτου μειωμένη φωτοσύνθεση. Επίσης όταν σε ένα θερμοκήπιο υπάρχει θερμοκρασία ψύξης από 1°C έως 5°C, για 24 έως 36 ώρες είναι σημαντικά επικίνδυνη ή ακόμη και θανατηφόρα για κάποια είδη φυτών. Εφαρμόζοντας τα αυτοματοποιημένα συστήματα στο θερμοκήπιο προσεγγίζουμε τις ιδανικές θερμοκρασίες και προστατεύουμε τα φυτά από τις ακτίνες του ηλίου. Η ιδανική θερμοκρασία σε ένα θερμοκήπιο για τα περισσότερα φυτά κυμαίνεται από 26 έως 30 βαθμούς Κελσίου. Την θερμοκρασία μπορούμε να την αυξήσουμε τους κρύους μήνες ή την νύχτα μέσω θερμοπομπών ή μέσω της γεωθερμίας και μπορούμε να την μειώσουμε στις περιόδους που υπάρχει υψηλή θερμοκρασία ανοίγοντας παράθυρα είτε με κάποιο σύστημα ψύξης (Air Condition) είτε συνδυασμό αυτών. Επιπρόσθετα, τα θερμοκήπια χρειάζονται σκίαστρα για τους καλοκαιρινούς μήνες, ώστε να περιορίζεται η ηλιακή ακτινοβολία που είναι η αιτία για διάφορες ασθένειες των φυτών. Επιπλέον, βοηθάει στην μείωση της θερμοκρασίας του θερμοκηπίου. [4] [5]

Η αναπνοή των φυτών βασίζεται σε μια σταθερή υγρασία του εδάφους αλλά και του περιβάλλοντος για τη μέγιστη ανάπτυξη. Όταν έχουμε ξηρές συνθήκες για τα φυτά έχουμε μειωμένη ανάπτυξη των φυτών, ανάπτυξη παράσιτων και ανάπτυξη ασθενειών. Οπότε κρατώντας κάποιες ιδανικές τιμές, οι οποίες αναφέρονται παρακάτω, υπάρχει η καλύτερη δυνατή ανάπτυξη των φυτών. Η ιδανική υγρασία στο περιβάλλον του θερμοκηπίου για τα περισσότερα φυτά κυμαίνεται από 70% έως 90%. Την υγρασία την ρυθμίζουμε με την χρήση συστήματος ψεκασμού ομίχλης από νερό για αύξηση και μέσω ανεμιστήρων για μείωση. Επίσης, η υγρασία του εδάφους είναι ένας από τους κυριότερους παράγοντες που χρειάζεται ένα φυτό, για να αναπτυχθεί. Η αύξηση υγρασίας του εδάφους γίνεται μέσω ποτίσματος, ενώ θα γίνεται μόνο όταν η υγρασία του χώματος δεν είναι αρκετή για το φυτό, οπότε έτσι θα γίνεται εξοικονόμηση ενέργειας και νερού. [4] [5]

Τέλος, ένας άλλος σημαντικός παράγοντας σε οριζόμενα θερμοκήπια είναι η επέκταση του φωτισμού για να αυξηθεί η απόδοση στις καλλιέργειες. Τα περισσότερα φυτά χρειάζονται για την φωτοσύνθεση 6 έως 16 ώρες φως ανάλογα το φυτό. Επίσης, θα πρέπει να αναφερθεί ότι τα φυτά δεν μπορούν να εκτεθούν σε φως ολόκληρο το εικοσιτετράωρο, γιατί αυτό οδηγεί στην εξάντληση και παύση της ανάπτυξης τους. [6]

## 2.3 Εξέλιξη θερμοκηπίου

Όταν πρωτοεμφανίστηκαν τα θερμοκήπια, οι κλιματικές συνθήκες άλλαζαν με χειροκίνητες ενέργειες του αγρότη. Αυτό είχε ως αποτέλεσμα τη σπατάλη ενέργειας και πόρων, τη μη γρήγορη απόκριση του αγρότη στις κλιματικές αλλαγές, κ.λ.π.. Επίσης, η παρουσία του αγρότη στο θερμοκήπιο ήταν απαραίτητη ώστε να κάνει μετρήσεις αλλά και τυχόν ενέργειες που απαιτούνταν.

Με την πάροδο του χρόνου και την εξέλιξη των τεχνολογιών τα θερμοκήπια άρχισαν να έχουν αυτοματισμούς για τον έλεγχο των κλιματικών συνθηκών. Οι αυτοματισμοί στα θερμοκήπια διευκόλυναν τους αγρότες και μείωσαν την εργασία που χρειαζόταν να κάνουν στα κλασικά θερμοκήπια. Έτσι, με την πάροδο του χρόνου ήρθε η ανάγκη της απομακρυσμένης παρακολούθησης και ελέγχου. [7]

Για αυτό σήμερα στα θερμοκήπια γίνεται χρήση IoT συστημάτων, τα οποία προσφέρουν την δυνατότητα συλλογής δεδομένων από τους αισθητήρες, απομακρυσμένης παρακολούθησης και ελέγχου σε εικοσιτετράωρη βάση σε πραγματικό χρόνο, ενισχύοντας την αποδοτικότητα, την ποιότητα και την ανθεκτικότητα της καλλιέργειας, εξοικονομώντας παράλληλα ενέργεια και πόρους. [8]

## 2.4 Τι είναι το IoT και τα θερμοκήπια IoT

Ο ορός Διαδίκτυο των Πραγμάτων (IoT-Internet of Things) είναι ένα δίκτυο στο οποίο συνδέονται συσκευές που έχουν αισθητήρες, λογισμικό και δυνατότητα σύνδεσης στο διαδίκτυο, έτσι ώστε να μπορούν να ανταλλάξουν δεδομένα και πληροφορίες μεταξύ τους. Στόχος του IoT είναι η βελτίωση της παραγωγικότητας, της αποτελεσματικότητας, της εξοικονόμησης πόρων, της εξοικονόμησης ενέργειας και άλλα. [9]

Η χρήση του IoT έχει αυξηθεί με εκθετικούς ρυθμούς και αξιοποιείται σε πληθώρα από συσκευές είτε είναι οικιακές είτε βιομηχανικές. Λόγω αυτής της εξέλιξης του IoT και της μεγάλης αποτελεσματικότητάς του στον έλεγχο και παρακολούθηση συσκευών το IoT έχει εισαχθεί και στα θερμοκήπια.

Ένα θερμοκήπιο IoT έχει διάφορους αισθητήρες, οι οποίοι στέλνουν τα δεδομένα τους σε μια πλατφόρμα που ενεργεί και εντοπίζει τυχόν κίνδυνο ασθενειών και μόλυνσης των καλλιεργειών. Με αυτό τον τρόπο οι εργασίες των αγροτών ελαχιστοποιούνται, βελτιώνεται η καλλιέργεια και η απόδοση. Υπάρχει ανάπτυξη πόρων και οι αγρότες έχουν πρόσβαση σε πολλές πληροφορίες του θερμοκηπίου. Επίσης, με την χρήση IoT συστήματος στο θερμοκήπιο μπορούμε να έχουμε ιδανικές συνθήκες για τα φυτά, και αυτόματα συστήματα

ύδρευσης. Με αυτόν τον τρόπο να διασφαλίζονται οι καλύτερες συνθήκες καλλιέργειας, ενώ υπάρχει η δυνατότητα ελέγχου και αποφυγής ασθενειών. Τέλος, έχουμε την δυνατότητα ενσωμάτωσης συστημάτων ασφάλειας. [10] [11]

## 2.5 Σε τι διαφέρει η προτεινόμενη υλοποίηση από τα αλλά IoT θερμοκήπια

Με βάση τα παραπάνω θα υλοποιηθεί ένα IoT θερμοκήπιο το οποίο δεν είναι το μοναδικό έργο που υφίσταται και είναι εξέλιξη των υφιστάμενων έργων.

Μια υλοποίηση τέτοιου έργου είναι η πτυχιακή εργασία με τίτλο «Μονάδα ελέγχου περιβάλλοντος θερμοκηπίου» του Δημήτρη Τσελίκα η οποία έχει ως σκοπό την προσομοίωση ενός ελεγχόμενου θερμαινόμενου θερμοκηπίου. [12]

Ακόμα ένα παρόμοιο έργο είναι το έργο του ViDes με τίτλο «Automated Greenhouse». Το έργο αυτό είναι ένα θερμοκήπιο που έχει ως στόχο την ρύθμιση της θερμοκρασίας, την ρύθμιση της υγρασίας του εδάφους, αλλά και την ρύθμιση της φωτεινότητας. Επίσης δημιούργησε μια ιστοσελίδα για τον έλεγχο του θερμοκηπίου εξ αποστάσεως. [13]

Άλλη μια υλοποίηση είναι η μεταπτυχιακή διπλωματική εργασία του Βαρελά Παναγιώτη με τίτλο «Εξυπνο Θερμοκήπιο» η οποία έχει ως σκοπό την υλοποίηση μιας πλήρους λειτουργικής μονάδας παρακολούθησης και ελέγχου των περιβαλλοντικών συνθηκών ενός θερμοκηπίου. [14]

Παρόμοια προσπάθεια είναι τα προϊόντα της Sensaphone με το προϊόν Sentinel Monitoring System που έχει ως σκοπό την παρακολούθηση του θερμοκηπίου. [15]

Υπάρχουν και αλλά παρόμοια έργα με αυτό η διαφορά όμως του IoT θερμοκηπίου από τα υπόλοιπα που υπάρχουν είναι ότι συνδυάζει την ασύρματη επικοινωνία ZigBee που είναι ένα πρωτόκολλο δεδομένων που προσφέρει χαμηλή κατανάλωση ενέργειας, εύκολη επέκταση με νέα εξαρτήματα, έλεγχο και μετάδοση δεδομένων, εύρος μετάδοσης από 10 έως 100 μέτρα και άλλα πολλά πλεονεκτήματα. Το ZigBee χρησιμοποιείτε για αμφίδρομη επικοινωνία, δηλαδή για την αποστολή δεδομένων για έλεγχο συσκευών από ένα μικροελεγκτή για παράδειγμα αλλά και την λήψη δεδομένων για παράδειγμα από αισθητήρες που είναι συνδεδεμένοι σε έναν μικροελεγκτή. Επίσης, διαφέρει από τα αλλά έργα στο ότι γίνεται χρήση συστήματος ειδοποίησης σε περίπτωση φωτιάς και υπερχειλίσης. Τέλος, υπάρχει έλεγχος επέκτασης φωτισμού και σύστημα αύξησης υγρασίας και όλα αυτά να ελέγχονται και παρακολουθούνται μέσω της ιστοσελίδας.

### 3 Τεχνολογίες, πλατφόρμες και εργαλεία που εφαρμόστηκαν

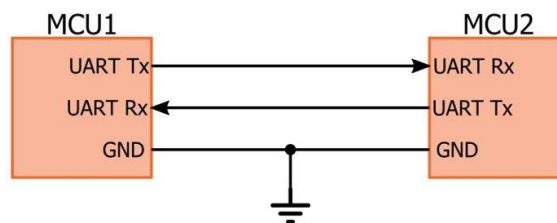
#### 3.1 Πρωτόκολλα επικοινωνίας που χρησιμοποιήθηκαν

##### 3.1.1 Πρόλογος

Η διασύνδεση υπολογιστών, κόμβων, ή υπολογιστικών δικτύων περιέχει μια πληθώρα από πρωτόκολλα επικοινωνίας για τις επικοινωνίες των διαφόρων κόμβων. Υπάρχουν διάφοροι τρόποι επικοινωνίας μεταξύ 2 ή περισσότερων υπολογιστών ή κόμβων. Για να γίνει η ανταλλαγή δεδομένων, χρειάζεται ένα σύνολο κανόνων που περιγράφουν πώς θα γίνει η επικοινωνία τους και αυτό ονομάζεται πρωτόκολλο επικοινωνίας. Οι κανόνες αυτοί περιγράφουν τον τεμαχισμό της πληροφορίας που είναι προς μετάδοση, ώστε να μπορεί να μεταφερθεί σε πακέτα, την διαδρομή που θα ακολουθήσει μέσα στο δίκτυο το πακέτο, για να φθάσει στον προορισμό, μεθόδους επαλήθευσης λήψης της πληροφορίας στον δέκτη και μεθόδους ανάκτησης ή επανάληψη της αποστολής μέρους της πληροφορίας που δεν έφτασε ή καταστράφηκε. Αυτό συμβαίνει, επειδή αφενός οι εφαρμογές και οι τοπολογίες διαφέρουν και αφετέρου υπάρχουν διαφορετικές απαιτήσεις επικοινωνίας που δεν μπορεί να καλύψει μόνο ένα πρωτόκολλο. Τα πρωτόκολλα επικοινωνίας εφαρμόζονται σε υλικό και λογισμικό. [16] [17]

Παρακάτω θα μιλήσουμε για τα πρωτόκολλο UART, το πρωτόκολλο SPI, το πρωτόκολλο I2C, το πρωτόκολλο Ethernet, το πρωτόκολλο HTTP, το πρωτόκολλο IEEE 802.15.4 και το πρωτόκολλο Zigbee.

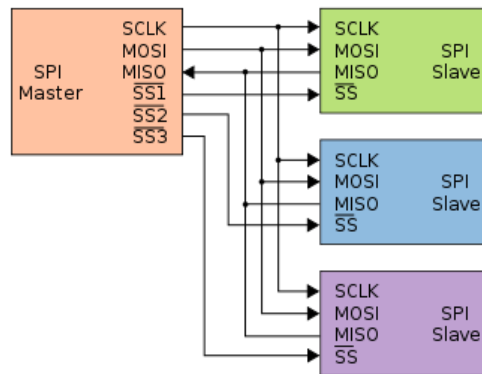
##### 3.1.2 Universal Asynchronous Receiver/Transmitter (UART)



Εικόνα 3.1 Απεικόνιση λειτουργίας του Universal Asynchronous Receiver/Transmitter. [Πηγή](#)

Το UART (Universal Asynchronous Receiver/Transmitter) είναι ένα ολοκληρωμένο κύκλωμα που κάνει μετάδοση και λήψη σειριακών δεδομένων ασύγχρονα σε μορφή πακέτων μέσω δύο καλωδίων του RX για την λήψη δεδομένων και του TX για την αποστολή δεδομένων. [18]

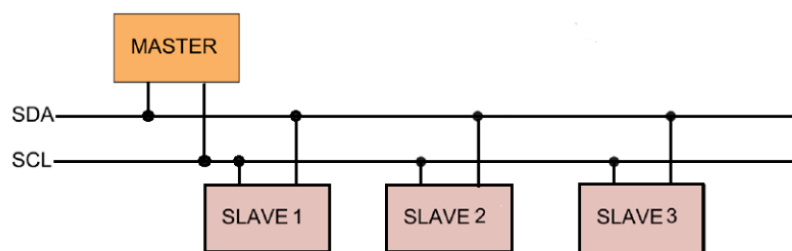
### 3.1.3 Serial Peripheral Interface (SPI)



Εικόνα 3.2 Απεικόνισή λειτουργίας Serial Peripheral Interface. [Πηγή](#)

Το SPI (Serial Peripheral Interface) είναι ένα σύγχρονο σειριακό πρωτόκολλο επικοινωνίας Full Duplex που χρησιμοποιείται συχνά σε ενσωματωμένα συστήματα για την μεταφορά δεδομένων μεταξύ ενός μικροελεγκτή (Master) και περιφερειακών (Slave) σε κοντινή απόσταση μέσω ενός διαύλου που έχει έως και τέσσερις αγωγούς: το SCK που είναι το ρολόι, το MOSI που είναι για αποστολή δεδομένων, το MISO που είναι για την λήψη δεδομένων και το SS για την επιλογή συσκευής με την οποία θέλουμε να επικοινωνήσουμε. Επίσης είναι ένα πρωτόκολλο με υψηλή ταχύτητα μετάδοσης δεδομένων, τα οποία δεν διακόπτονται, αλλά έχουν συνεχή ροή. Τέλος, το SPI έχει την δυνατότητα σύνδεσης πολλαπλών συσκευών στον ίδιο δίαυλο δεδομένων. [18]

### 3.1.4 Inter-Integrated Circuit (I2C)



Εικόνα 3.3 Απεικόνισή λειτουργίας Inter-Integrated Circuit. [Πηγή](#)

Το Inter-Integrated Circuit (I2C) είναι ένα σύγχρονο σειριακό πρωτόκολλο επικοινωνίας που χρησιμοποιείται συχνά σε ενσωματωμένα συστήματα για την μεταφορά δεδομένων χαμηλής ταχύτητας μεταξύ ενός ή περισσότερων Μικροελεγκτών (Master) και ενός ή περισσότερων περιφερειακών (Slave) μέσω ενός αμφίδρομου δίαυλου που έχει δύο αγωγούς που το ένα χρησιμοποιείται για το σειριακό ρολόι (SCL) και το άλλο για τα σειριακά δεδομένα (SDA).



Τα δεδομένα στέλνονται από έναν μικροελεγκτή σε μια περιφερειακή συσκευή σε μορφή πακέτων, για να επιτευχθεί αυτό, η κάθε περιφερειακή συσκευή έχει μια μοναδική διεύθυνση. [18]

### 3.1.5 Ethernet

Το Ethernet είναι μια ενσύρματη τεχνολογία δικτύωσης υπολογιστών που χρησιμοποιείται συνήθως σε τοπικά δίκτυα (LAN), δίκτυα μητροπολιτικής περιοχής (MAN) και δίκτυα ευρείας περιοχής (WAN). Το IEEE καθορίζει στην οικογένεια προτύπων που ονομάζεται IEEE 802.3. Το πρωτόκολλο Ethernet αγγίζει τόσο το επίπεδο 1 (φυσικό επίπεδο) όσο και το επίπεδο 2 (επίπεδο σύνδεσης δεδομένων) στο μοντέλο διασύνδεσης ανοικτών συστημάτων (OSI). [19]

### 3.1.6 Hyper Text Transfer Protocol (HTTP)

Το HTTP είναι ένα πρωτόκολλο πελάτη/διακομιστή προσανατολισμένο στις συναλλαγές πληροφοριών. Μια από τις πιο συνηθισμένες συναλλαγές είναι τα έγγραφα HTML ανάμεσα σε ένα φυλλομετρητή και σε έναν διακομιστή. Οι πληροφορίες που μεταφέρονται είναι απλό κείμενο, υπερκείμενο, ήχος, εικόνες, βίντεο, βάσεις δεδομένων ή οποιαδήποτε άλλη πληροφορία που μεταφέρεται μέσω του διαδικτύου. Η επικοινωνία μεταξύ του πελάτη και του διακομιστή επιτυγχάνεται με την ανταλλαγή μηνυμάτων. Τα μηνύματα που στέλνει ο πελάτης, που συνήθως είναι ένα πρόγραμμα περιήγησης ιστού, είναι αιτήματα και τα μηνύματα που στέλνει ο διακομιστής είναι απαντήσεις. [20]

Το πρωτόκολλο HTTP υποστηρίζει πολλές μεθόδους αιτημάτων για την μεταφορά δεδομένων από διακομιστή. Οι πιο δημοφιλείς μέθοδοι αιτημάτων είναι η GET, η POST, η put, η delete, η patch, η copy και άλλες. [21] Η μέθοδος GET είναι αίτημα ανάκτησης δεδομένων από ένα διακομιστή, ενώ η μέθοδος POST είναι για την αποστολή δεδομένων στον διακομιστή. [22] [23]

### 3.1.7 IEEE 802.15.4

#### 3.1.7.1 Ορισμός

Το IEEE 802.15.4 είναι ένα τεχνικό πρότυπο για ασύρματη επικοινωνία που εκδόθηκε από το IEEE, και καθορίζει την λειτουργία ενός χαμηλού επιπέδου ασύρματου προσωπικού δικτύου (LR-WPAN) και σκοπεύει να προσφέρει επικοινωνία χαμηλού κόστους και χαμηλής ταχύτητας μεταξύ συσκευών. [24]



### 3.1.7.2 Τύποι συσκευών που ορίζει το προδιαγραφή IEEE 802.15.4

Η προδιαγραφή IEEE 802.15.4 ορίζει δύο τύπους συσκευών συσκευές πλήρους λειτουργίας και συσκευές μειωμένης λειτουργίας. Οι συσκευές πλήρους λειτουργίας έχουν όλες τις δυνατότητες που προσφέρει η προδιαγραφή IEEE 802.15.4, όπως το να μπορούν να επικοινωνούν με οποιαδήποτε συσκευή στο δίκτυο. Αρκεί να είναι ενεργές, για να ακούνε τα πάντα στο δίκτυο. Οι συσκευές μειωμένης λειτουργίας έχουν περιορισμένες δυνατότητες, όπως το ότι μπορούν να επικοινωνήσουν μόνο με μια συσκευή πλήρους λειτουργίας. [25]

### 3.1.8 ZigBee

#### 3.1.8.1 Τι είναι το ZigBee



*Εικόνα 3.4 Λογότυπο ZigBee. [Πηγή](#)*

Το ZigBee ορίστηκε από το IEEE (Institute of Electrical and Electronics Engineers) το 1998. Το δίκτυο ZigBee είναι ένα προσωπικό δίκτυο περιοχής Wireless Personal Area Network (WPAN) που είναι κατάλληλο για εφαρμογές ασύρματης δικτύωσης, για έλεγχο και παρακολούθηση με ραδιοσυχνότητες (RF) που απαιτούν χαμηλό ρυθμό δεδομένων χαμηλής κατανάλωσης ενέργειας, υψηλής απόδοσης (35 kbps), χαμηλό λανθάνοντα χρόνο, προβλέψιμο χρόνο επικοινωνίας και ασφαλή δικτύωση χαμηλού κόστους. Επίσης, υποστηρίζει ταχύτητες δεδομένων έως και 250 kbps.

Το ZigBee χρησιμοποιεί για να επικοινωνήσει, ραδιοσυχνότητες (RF). Υπάρχουν 3 διαθέσιμες συχνότητες:

1. 2,4 GHz –η οποία χρησιμοποιείται παγκόσμια και μοιράζεται με το Bluetooth και Wi-Fi
2. 915 MHz – η οποία χρησιμοποιείται στην Αμερική.
3. 868 MHz – η οποία χρησιμοποιείται στην Ευρώπη, αλλά απαγορεύεται σε μερικές χώρες.

Το ZigBee βασίζεται στο πρότυπο δικτύωσης IEEE 802.15.4 για γρήγορη δικτύωση από σημείο σε σημείο ή από ένα σημείο σε πολλαπλά σημεία. Στόχος του ZigBee είναι να έχει χαμηλότερο ρυθμό μετάδοσης δεδομένων σε σύγκριση με άλλα δίκτυα WPAN. Επίσης,

στοχεύει σε εφαρμογές χαμηλότερης ισχύος που θα προσφέρουν χρόνια λειτουργία σε μη επαναφορτιζόμενη μπαταρία και μπορεί να συνδεθεί με περισσότερους κόμβους ( $2^{16} =$  διεύθυνση 16-bit  $> 65.000$ ) ενώ για παράδειγμα το Bluetooth προσφέρει σύνδεση μόνο 8 συσκευών. Το ZigBee έχει σχεδιαστεί για έλεγχο και παρακολούθηση, ενώ το Bluetooth για αντικατάσταση καλωδίου. Για να μπορούν να επικοινωνούν όλες οι συσκευές μεταξύ τους, το δίκτυο ορίζει κάθε συσκευή με ένα μοναδικό αναγνωριστικό PAN (PAN ID). Αυτό το αναγνωριστικό είναι κοινό μεταξύ όλων των συσκευών του ίδιου δικτύου. [26] [27]

### 3.1.8.2 Τύποι συσκευών ZigBee

Όπως η προδιαγραφή IEEE 802.15.4 έχει ορίσει τους τύπους συσκευών έτσι και το ZigBee ορίζει αντίστοιχα τρεις διαφορετικούς τύπους συσκευών : **συντονιστής**, **δρομολογητής** και **τερματικές συσκευές**.

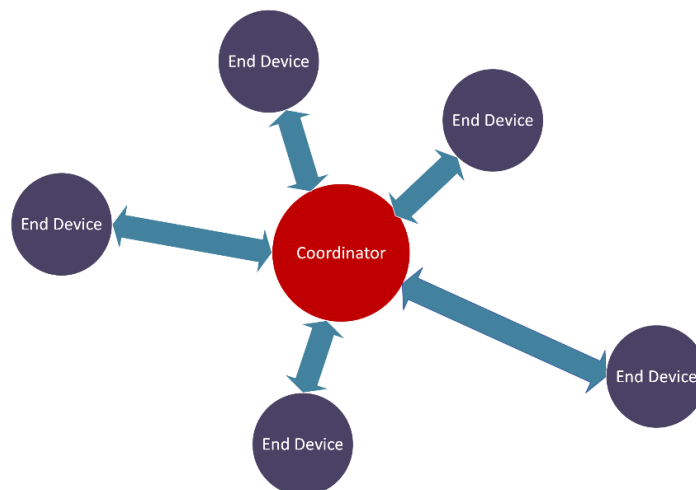
- i. Ένας **συντονιστής** (Coordinator) έχει τα ακόλουθα χαρακτηριστικά:
  - Επιλέγει ένα κανάλι και ένα αναγνωριστικό PAN (τόσο 64-bit όσο και 16-bit), για να ξεκινήσει το δίκτυο. Μετά από αυτό, συμπεριφέρεται ουσιαστικά σαν δρομολογητής.
  - Μπορεί να επιτρέψει στους δρομολογητές και τις τελικές συσκευές να συνδεθούν στο δίκτυο.
  - Μπορεί να βοηθήσει στη δρομολόγηση δεδομένων.
  - Δεν μπορεί να κοιμηθεί και θα πρέπει να τροφοδοτείται από το δίκτυο. [28] [29]
- ii. Ένας **δρομολογητής** (Router) έχει τα ακόλουθα χαρακτηριστικά:
  - Πρέπει να συνδεθεί σε ένα ZigBee PAN για να μπορέσει να μεταδώσει, να λάβει ή να δρομολογήσει δεδομένα.
  - Μετά τη σύνδεση, μπορεί να επιτρέψει στους δρομολογητές και τις τελικές συσκευές να συνδεθούν στο δίκτυο.
  - Μετά την ένταξη, μπορεί να βοηθήσει στη δρομολόγηση δεδομένων.
  - Δεν μπορεί να κοιμηθεί και θα πρέπει να τροφοδοτείται από το δίκτυο. [28] [29]
- iii. Μια **τελική συσκευή** (End Devices) είναι απαραίτητη, μόνο εάν χρειάζεται να γίνεται εξοικονόμηση ενέργειας στο έργο, αφού το τελικό Xbee θα λειτουργεί μόνο όταν ζητούνται δεδομένα από αυτό. Μια τελική συσκευή έχει τα ακόλουθα χαρακτηριστικά:

- Πρέπει να συνδεθεί με έναν δρομολογητή ή συντονιστή σε ένα ZigBee PAN για να μπορέσει να μεταδώσει ή να λάβει δεδομένα.
- Δεν επιτρέπεται η σύνδεση συσκευών στο δίκτυο.
- Πρέπει πάντα να μεταδίδει και να λαμβάνει δεδομένα RF μέσω αυτού του δρομολογητή ή συντονιστή. Δεν είναι δυνατή η δρομολόγηση δεδομένων. Ο δρομολογητής ή ο συντονιστής που επέτρεψε τη σύνδεση μιας τελικής συσκευής γίνεται ο "γονέας" της τελικής συσκευής.
- Δεδομένου ότι η τελική συσκευή μπορεί να αδρανεί, ο γονέας πρέπει να μπορεί να αποθηκεύσει ή να διατηρήσει εισερχόμενα πακέτα δεδομένων που προορίζονται για την τελική συσκευή έως ότου η τελική συσκευή μπορέσει να αφυπνιστεί και να λάβει τα δεδομένα.
- Μπορεί να εισάγει λειτουργίες χαμηλής κατανάλωσης για εξοικονόμηση ενέργειας και μπορεί να τροφοδοτείται με μπαταρία. [28] [29]

### 3.1.8.3 Τοπολογίες ZigBee

Οι τοπολογίες που μπορούν να γίνουν σε ένα δίκτυο ZigBee είναι :

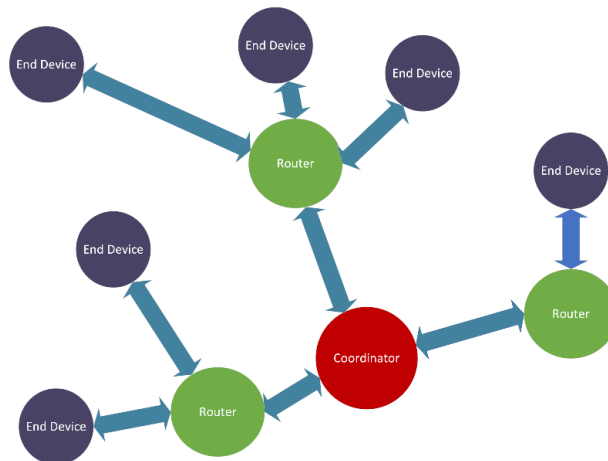
1. Η Τοπολογία **αστέρας** είναι μια διαμόρφωση δικτύου που η επικοινωνία μιας συσκευής ως συντονιστής (Coordinator) με οποιαδήποτε αριθμό από δρομολογητές (Router) ή τερματικές συσκευές (End Device), που επικοινωνούν μεταξύ τους χωρίς να υπάρχει άμεση επικοινωνία των τελικών συσκευών (End Devices). [30] [25]



Εικόνα 3.5 Αναπαράσταση τοπολογίας αστέρα.

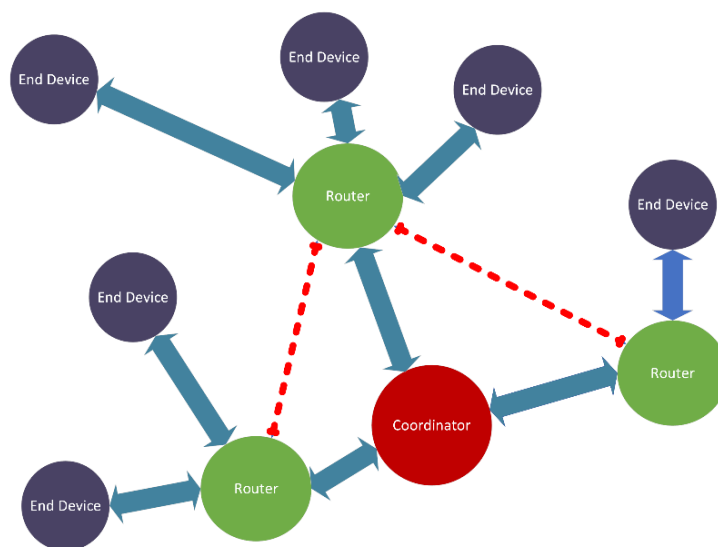
2. Η Τοπολογία **δέντρου** είναι ένας τύπος τοπολογίας Peer to Peer στην οποία η επικοινωνία γίνεται μεταξύ μιας συσκευής που λειτουργεί ως συντονιστής (Coordinator) και με οποιοδήποτε αριθμό από δρομολογητές (Router) ή τελικές συσκευές (End Device). Επιπρόσθετα, οι δρομολογητές μπορούν συνδεθούν με τις

τελικές συσκευές σε μια ιεραρχία από την υψηλότερη που είναι ο συντονιστής προς την χαμηλότερη που είναι η τελική συσκευή. Επίσης, οι τελικές συσκευές δεν μπορούν να επικοινωνήσουν άμεσα μεταξύ τους αλλά μέσω των συντονιστών ή των δρομολογητών. Στην τοπολογία αυτή υπάρχουν τουλάχιστον τρία διαφορετικά επίπεδα όπου δύο συνδεδεμένοι κόμβοι μεταξύ τους μπορούν να έχουν μόνο μία αμοιβαία σύνδεση, επομένως μπορεί να υπάρχει μόνο ένας σύνδεσμος μεταξύ τους. [31] [25]



Εικόνα 3.6 Αναπαράσταση τοπολογίας δέντρο

3. Στην Τοπολογία **πλέγματος** όλες οι συσκευές διασυνδέονται μεταξύ τους. Αυτή η σύνδεση χρησιμοποιείται συνήθως σε εφαρμογές όπου η εμβέλεια μεταξύ δύο σημείων μπορεί να είναι μεγαλύτερη από την εμβέλεια των δύο συσκευών, αλλά, εάν υπάρχουν ενδιάμεσες συσκευές, μπορούν να προωθηθούν οποιαδήποτε πακέτα προς και από τις επιθυμητές συσκευές. [32] [25]



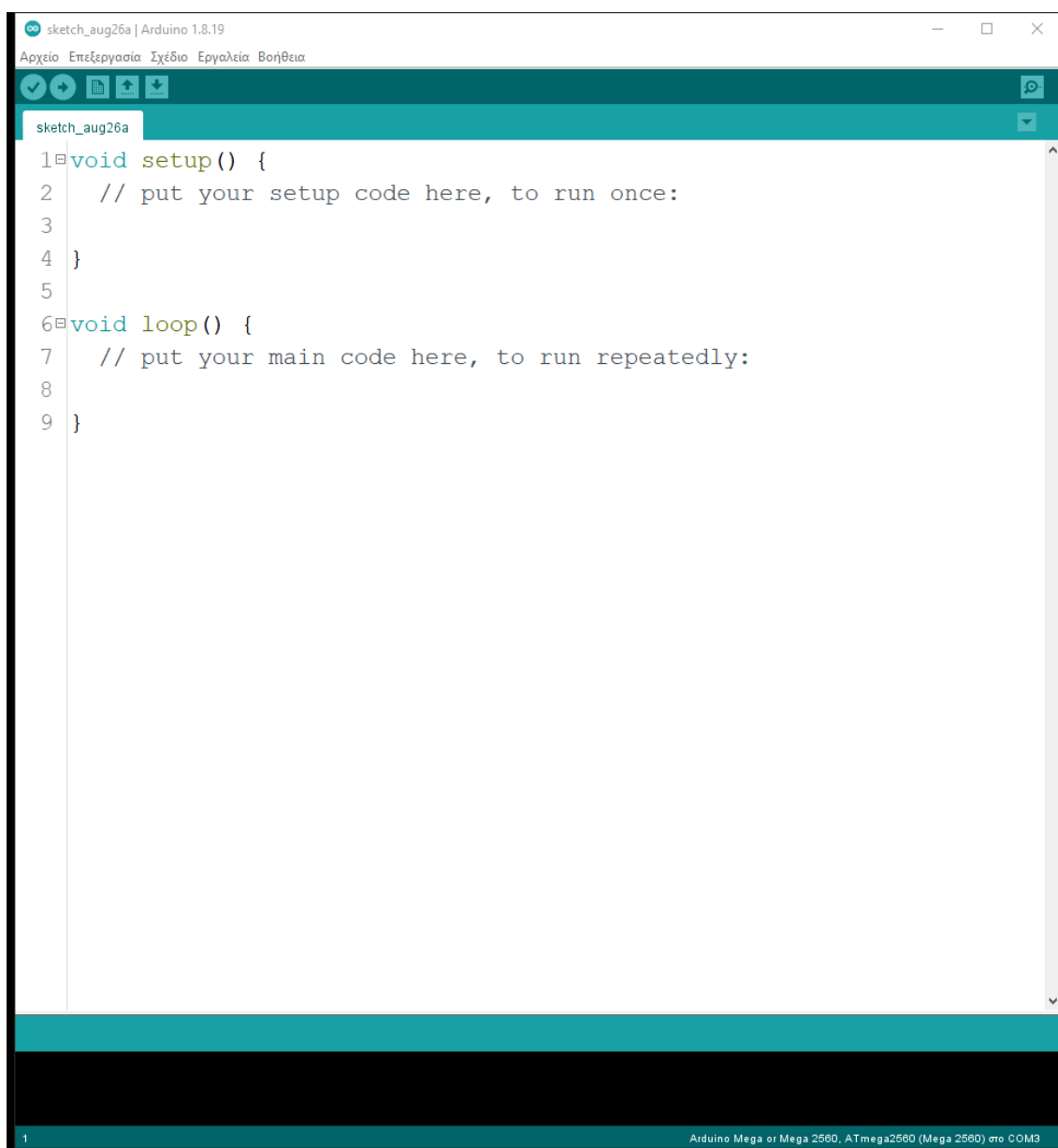
Εικόνα 3.7 Αναπαράσταση τοπολογίας πλέγματος.

## 3.2 Γλώσσες προγραμματισμού που χρησιμοποιήθηκαν

### 3.2.1 Λογισμικό Arduino

Το λογισμικό Arduino IDE (Integrated Development environment) είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης ανοιχτού κώδικα. Το Arduino IDE δίνει την δυνατότητα σύνταξης του κώδικα, της μεταγλώττισης και βοηθάει στον εντοπισμό σφαλμάτων. Επίσης, περιέχει μια περιοχή μηνυμάτων, μια σειριακή οθόνη, μια γραμμή με κουμπιά που έχουν διάφορες λειτουργίες, όπως κουμπί ελέγχου κώδικα, κουμπί μεταγλώττισης, κουμπί για νέο σκίτσο και το κουμπί για την σειριακή οθόνη.

Το λογισμικό έχει εύχρηστο περιβάλλον και εγκατάσταση στον υπολογιστή διευκολύνει την σύνταξη του κώδικα και τη μεταγλώττιση στην πλακέτα Arduino. [33]



```
sketch_aug26a | Arduino 1.8.19
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια
sketch_aug26a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
1
Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) στο COM3
```

Εικόνα 3.8 Στιγμιότυπο οθόνης του Arduino IDE.

Ο κώδικας Arduino είναι αυτός που γράφεται μέσα στο Arduino IDE και είναι αυτός που φορτώνεται στον μικροελεγκτή. Ο κώδικας Arduino βασίζεται στην γλώσσα C/C++ με προσθήκη βιβλιοθηκών, ειδικών μεθόδων και συναρτήσεων. [34]

## 3.2.2 Γλώσσες προγραμματισμού διαδικτύου

### 3.2.2.1 Hyper Text Markup Language (HTML)



Εικόνα 3.9 Λογότυπο Hyper Text Markup Language. [Πηγή](#)

Η Hyper Text Markup Language (HTML) είναι μια γλώσσα σήμανσης, δηλαδή περιέχει οδηγίες που απευθύνονται στο φυλλομετρητή για το πώς να προβληθεί μια σελίδα. Το αρχείο κώδικα HTML διαβάζεται από έναν φυλλομετρητή και αναλόγως παρουσιάζεται το κείμενο, οι εικόνες και οι σύνδεσμοι. Έτσι γράφοντας σε έναν επεξεργαστή κειμένου με συγκεκριμένο τρόπο και δομή, ο φυλλομετρητής βλέποντας την δομή αυτή γνωρίζει που και πως θα παρουσιάσει το κείμενο, τις εικόνες και που θα πάει, όταν κάποιος κάνει κλικ σε ένα σύνδεσμο. Κάθε αρχείο HTML για να αναγνωριστεί από το φυλλομετρητή πρέπει να αναγραφεί ως επέκταση αρχείων .html. [35]

```
<html>
<κεφάλι>
  <title>Τίτλος σελίδας</title>
</head>
<σώμα>
  <h1>Αυτή είναι μια επικεφαλίδα</h1>
  <p>Αυτή είναι μια παράγραφος.</p>
  <p>Αυτή είναι μια άλλη παράγραφος.</p>
</body>
</html>
```

Εικόνα 3.10 Δομή εγγράφου HTML. [Πηγή](#)

### 3.2.2.2 Cascading Style Sheets (CSS)



Εικόνα 3.11 Λογότυπο CSS. [Πηγή](#)

Το Cascading Style Sheets (CSS) είναι η γλώσσα που χρησιμοποιείται ώστε να οριστεί πως θα εμφανιστεί το στυλ ενός εγγράφου HTML. Κάθε αρχείο CSS για να αναγνωριστεί από το φυλλομετρητή πρέπει να αναγραφεί ως επέκταση αρχείων .css. [36]

### 3.2.2.3 JavaScript (JS)



Εικόνα 3.12 Λογότυπο JavaScript. [Πηγή](#)

Η JavaScript είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται σε προγράμματα περιήγησης ιστού. Επίσης παρέχει την αλληλεπίδραση του προγράμματος περιήγησης με τον χρήστη, δημιουργώντας έτσι δυναμικές ιστοσελίδες. Δηλαδή γίνονται στην ιστοσελίδα αλλαγές, χωρίς να απαιτείται η μη αυτόματη φόρτωση της ιστοσελίδας. Ακόμα κάποιες πλατφόρμες όπως είναι το Node JS επιτρέπουν να χρησιμοποιείται JavaScript από την πλευρά του διακομιστή. Τέλος, η γλώσσα προγραμματισμού JavaScript χρησιμοποιείται μαζί με την HTML και την CSS, χρησιμοποιώντας τα tag `<script></script>` ή δημιουργώντας ένα αρχείο .js. [37]

Ακολουθεί παράδειγμα κώδικα από το κομμάτι του κώδικα `soilgraph.php` για την εμφάνιση των τιμών σε γραφική παράσταση μέσω των `google charts`.

```
<script type="text/javascript"> //open tag js
  google.charts.load('current', {'packages':['corechart']});
  google.charts.setOnLoadCallback(drawChart);
  function drawChart() {
    var data = google.visualization.arrayToDataTable([
```

```

        ['event', 'soilhygro'], //Data from data base
    <?php
        while($row = mysqli_fetch_array($result)){
            // εμφάνιση των δεδομένων
            echo "[".$row["event"]."', ".$row["soilhygro"]."',";
        }
    ?>
    });
    var options = {
        title: 'Soil hygrometers sensor' // τίτλος γραφικής
    };
    var chart = new
    google.visualization.LineChart(document.getElementById('curve_chart'));
    chart.draw(data, options);
}
</script> // close tag js

```

#### 3.2.2.4 Personal Home Page (php)



Εικόνα 3.13 Λογότυπο Personal Home Pag. [Πηγή](#)

Η php είναι μια ευρέως χρησιμοποιούμενη ανοιχτού κώδικα γλώσσα δέσμης ενεργειών από την πλευρά του διακομιστή, αλλά και γλώσσα γενικής χρήσης που χρησιμοποιούν πολλοί προγραμματιστές για την ανάπτυξη Ιστού και μπορεί να ενσωματωθεί στην HTML. Η PHP επιτρέπει στους προγραμματιστές ιστού να δημιουργούν ένα δυναμικό περιεχόμενο που αλληλοεπιδρά με βάσεις δεδομένων και ένα ισχυρό εργαλείο για τη δημιουργία δυναμικών και διαδραστικών ιστοσελίδων. Κάθε αρχείο php για να αναγνωριστεί από τον φυλλομετρητή πρέπει να αναγραφεί ως επέκταση αρχείων .php. [38]

#### 3.2.2.5 Structed Query Language (SQL)

Η SQL είναι μια γλώσσα για την εξαγωγή και την οργάνωση δεδομένων που είναι αποθηκευμένα σε μια σχεσιακή βάση δεδομένων. Μια βάση δεδομένων είναι ένας πίνακας με στήλες και γραμμές στον οποίο είναι αποθηκευμένα σε αυτόν δεδομένα. [39]



### 3.2.2.6 MySQL

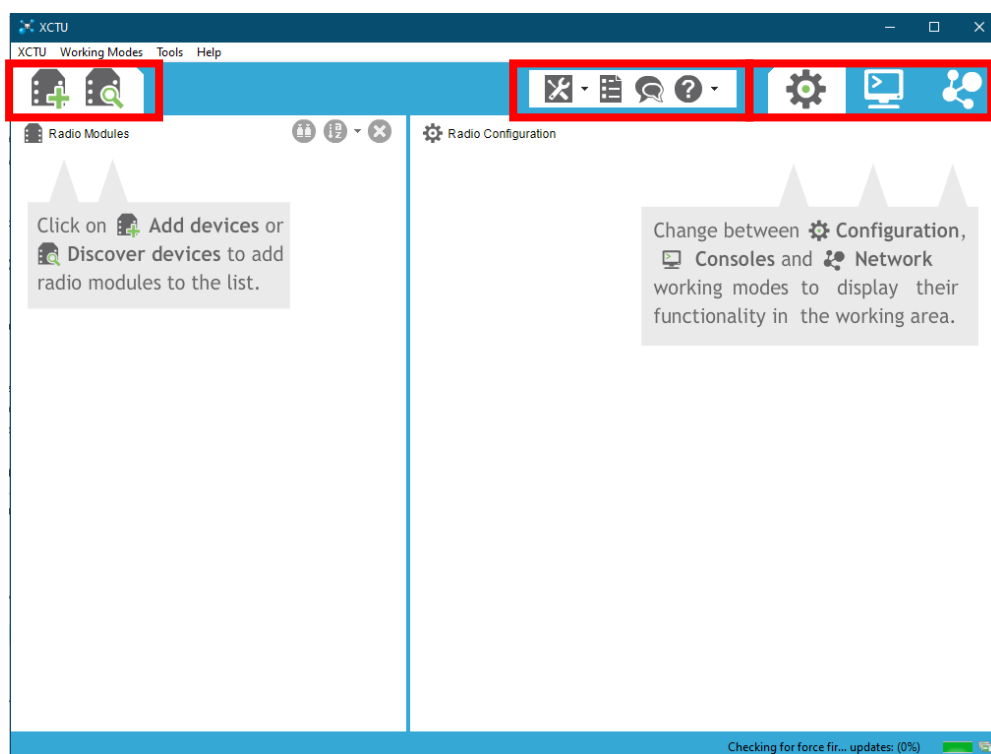


Εικόνα 3.14 Το Λογότυπο της MySQL. [Πηγή](#)

Η MySQL είναι ένα δωρεάν και ανοιχτού κώδικα, ευρέως χρησιμοποιούμενο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που αναπτύχθηκε από τη ORACLE και βασίζεται στην δομημένη γλώσσα SQL. [40]

## 3.3 Εργαλεία και πλατφόρμες που χρησιμοποιήθηκαν

### 3.3.1 Λογισμικό XCTU



Εικόνα 3.15 Το περιβάλλον του λογισμικού XCTU

Το XCTU είναι μια δωρεάν εφαρμογή που προσφέρει η DIGI, έχει σχεδιαστεί, για να επιτρέπει την ρύθμιση, την διαμόρφωση και τη δοκιμή μονάδων RF Xbee μέσω μιας απλής στη χρήση γραφικής διεπαφής. Τα Xbee μπορούν να συνδεθούν με το XCTU μέσω μιας

σειριακής μονάδας όπως το SparkFun Xbee Explorer USB και υστέρτα συνδέοντας μέσω ενός καλωδίου mini USB με τον υπολογιστή.

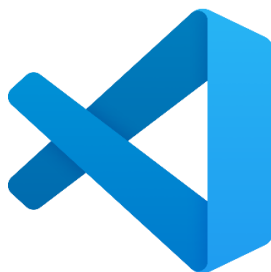
Με το X-CTU ρυθμίζονται τα Xbees, ώστε να δημιουργηθούν δικτυωτά δίκτυα με δεκάδες Xbee ή απλώς ένα ζεύγος δεδομένων εναλλαγής. Επίσης, υπάρχει η δυνατότητα ενημερώσεων στο υλικό-λογισμικό και τροποποίησης παραμέτρων. [41]

Στην αρχική επιφάνεια του λογισμικού περιέχει επιλογές αναζήτησης και προσθήκης των συσκευών Xbee. Επίσης περιέχει ένα κουμπί για εργαλεία, για της ρύθμισης του λογισμικού, για τα Feedback και για την βοήθεια στο λογισμικό.

Ακόμα περιέχει τρεις διαφορετικές μορφές εργασίας:

- η πρώτη μορφή εργασίας είναι Configuration mode που είναι για την διαμόρφωση των Xbee,
- η δεύτερη μορφή εργασίας Console mode που επιτρέπει την επικοινωνία, την δημιουργία πλαισίων και των έλεγχο της επικοινωνία με της συσκευές Xbee
- η τρίτη μορφή εργασίας Network mode που είναι για την οπτικοποίηση της τοπολογίας και της διασύνδεσης του δικτύου που έχει δημιουργηθεί. [42]

### 3.3.2 Visual Studio Code



*Εικόνα 3.16 Το λογότυπο του Visual Studio Code. [Πηγή](#)*

Για την σύνταξη του κώδικα για της ιστοσελίδας γίνεται χρήση του Visual Studio Code. Το Visual Studio Code είναι ένας δωρεάν επεξεργαστής κώδικα με διάφορες επεκτάσεις, ώστε να καλύπτουν κάθε ανάγκη του προγραμματιστή. Επίσης βοηθάει των προγραμματιστή γραφή του κώδικα, βοηθάει στον εντοπισμό σφαλμάτων και διορθώνει τον κώδικα χρησιμοποιώντας το εργαλείο intelli-sense είναι ένα εργαλείο συμπλήρωσης κώδικα. [43] [44]

### 3.3.3 Λογισμικό XAMPP



Εικόνα 3.17 Το λογότυπο της XAMPP. [Πηγή](#)

Για να στήσουμε την ιστοσελίδα, θα κάνουμε χρήση της php, MySQL και Apache οπότε θα χρησιμοποιήσουμε το XAMPP που τα περιλαμβάνει όλα αυτά μαζί. Το XAMPP είναι διακομιστής ιστού που είναι δωρεάν και ανοιχτού κώδικα. Επίσης το XAMPP έχει διάφορα Services και για αυτό το όνομα του έχει προέλθει από τα αρχικά γράμματα αυτών των Services το “X” σημαίνει Cross-platform, το “A” διακομιστής Apache, το “M” MySQL, το “P” PHP και το “P” Perl. Το XAMPP είναι ένα πρόγραμμα που μπορεί να εγκατασταθεί σε έναν υπολογιστή, ώστε να τον μετατρέψει σε έναν Web Server.. [45]

### 3.3.4 Apache

Το Apache είναι ένα λογισμικό διακομιστή ιστού ανοιχτού κώδικα που εκτελείται σε έναν Web Server και είναι υπεύθυνο για την αποδοχή αιτημάτων HTTP και αποστολή πίσω τις ζητούμενες πληροφοριών με τη μορφή ιστοσελίδων. Ο Apache δίνει τα περιεχόμενα μιας ιστοσελίδας στο φυλλομετρητή [46] [47]

### 3.3.5 Google Charts



Εικόνα 3.18 Λογότυπο Google Charts. [Πηγή](#)

Τα Google charts είναι δωρεάν βιβλιοθήκες που είναι βασισμένες σε JavaScript που προορίζονται, για να εφαρμοστεί σε έργα ιστού για την οπτικοποίηση δυναμικών δεδομένων. Υποστηρίζει διάφορα διαγράμματα όπως Column Chart, Line Chart, Pie Chart, Donut Chart, Gauge και άλλα. Επίσης τα google charts χρησιμοποιούν τις τεχνολογίες HTML5/SVG, για να υποστηρίζουν μια ποικιλία από συσκευές και πρόγραμμα περιήγησης. [48] [49]

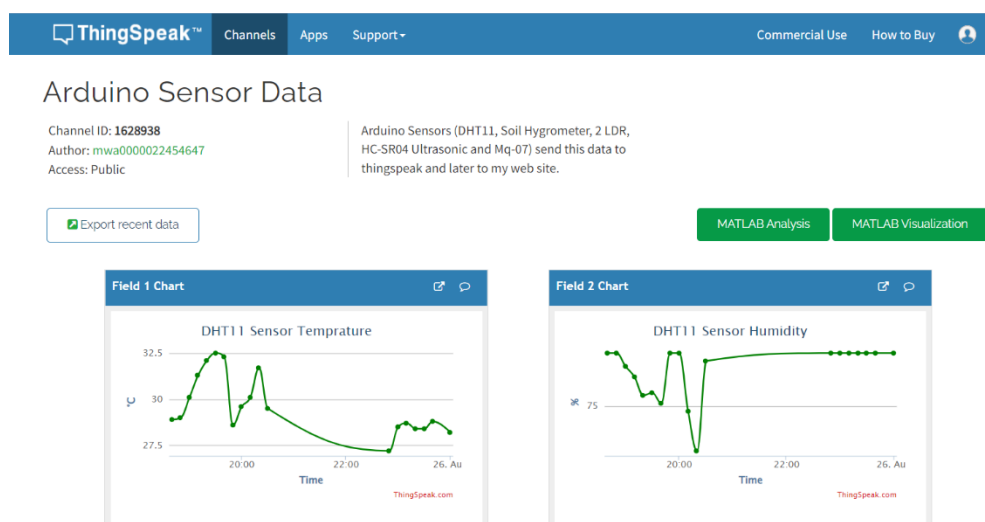
### 3.3.6 IFTTT



Εικόνα 3.19 Λογότυπο IFTTT. [Πηγή](#)

Το IFTTT (If This, Then That) είναι μια υπηρεσία ιστού που συνδέει εφαρμογές, συσκευές και υπηρεσίες ιστού μαζί όπως Gmail, google assistance, twitter και αλλά ώστε να συνδέσει της διάφορες αυτές υπηρεσίες με αυτοματισμούς. [50]

### 3.3.7 ThingSpeak



Εικόνα 3.20 Το κανάλι ThingSpeak που χρησιμοποιούμε για την απεικόνιση των γραφικών μας.

Το ThingSpeak είναι μια IoT πλατφόρμα που παρέχει διάφορες υπηρεσίες για την επεξεργασία και την οπτικοποίηση δεδομένων που λαμβάνουμε και στέλνουμε από τις συσκευές μας στο cloud σε εφαρμογές Internet of Things (IoT).

Το ThingSpeak δίνει την δυνατότητα να οπτικοποιήσουμε τα δεδομένα μας μέσω γραφικών παραστάσεων, μετρητών και σταθερών αριθμών, επίσης μας δίνει την δυνατότητα σύνδεσης με διάφορες υπηρεσίες, όπως είναι το Twitter για αποστολή αυτοματοποιημένων μηνυμάτων ή το MATLAB για επεξεργασία δεδομένων, σύνδεση με οποιαδήποτε ιστοσελίδα χρησιμοποιώντας το API Key και αλλά. [51] [52]

## 4 Υλικά που χρησιμοποιήθηκαν

### 4.1 Arduino

#### 4.1.1 Τι είναι το Arduino



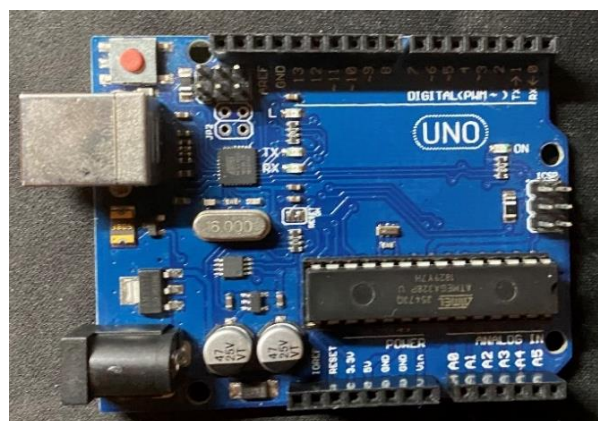
Εικόνα 4.1 Το λογότυπο του Arduino. [Πηγή](#)

Το Arduino είναι μια πλατφόρμα ανοιχτού κώδικα που αποτελείται από την πλακέτα (Hardware) και από το λογισμικό (Software). [53]

Η πλακέτα Arduino αποτελείται από μια μητρική πλακέτα με ενσωματωμένο έναν μικροελεγκτή με άλλα ηλεκτρονικά εξαρτήματα. Ο μικροελεγκτής περιέχει επεξεργαστή, μνήμη και προγραμματιζόμενες εισόδους και εξόδους, οι οποίες μπορούν να είναι είτε ψηφιακές, είτε αναλογικές και όλα αυτά είναι ενσωματωμένα σε ένα μόνο Chip. [53]

Το Arduino είναι μια αξιόπιστη πλατφόρμα, πολύ οικονομική και διαθέτει πολύ μεγάλες δυνατότητες, καθώς μπορεί να συνδεθεί με συσκευές εισόδου όπως: μπουτόν, ροοστάτες, αισθητήρες, συσκευές εξόδου όπως λυχνίες LED, LCD οθόνη, κ.λ.π.. Επίσης υπάρχει η δυνατότητα επέκτασης του ανάλογα με το Project με τα κατάλληλα Shields. Τέλος, το Arduino προγραμματίζεται σε μια πολύ ισχυρή γλώσσα προγραμματισμού την wiring που είναι συνδυασμός C και C++. [54]

Τα παραπάνω καθιστούν το Arduino ιδανικό για εφαρμογές αυτοματισμού, ρομποτικής και Internet of Things (IoT) όπου και χρησιμοποιείται ευρέως με δισεκατομμύρια ενεργές συσκευές παγκοσμίως. [55]



Εικόνα 4.2 Πλακέτα Arduino Uno μπροστινή όψη.



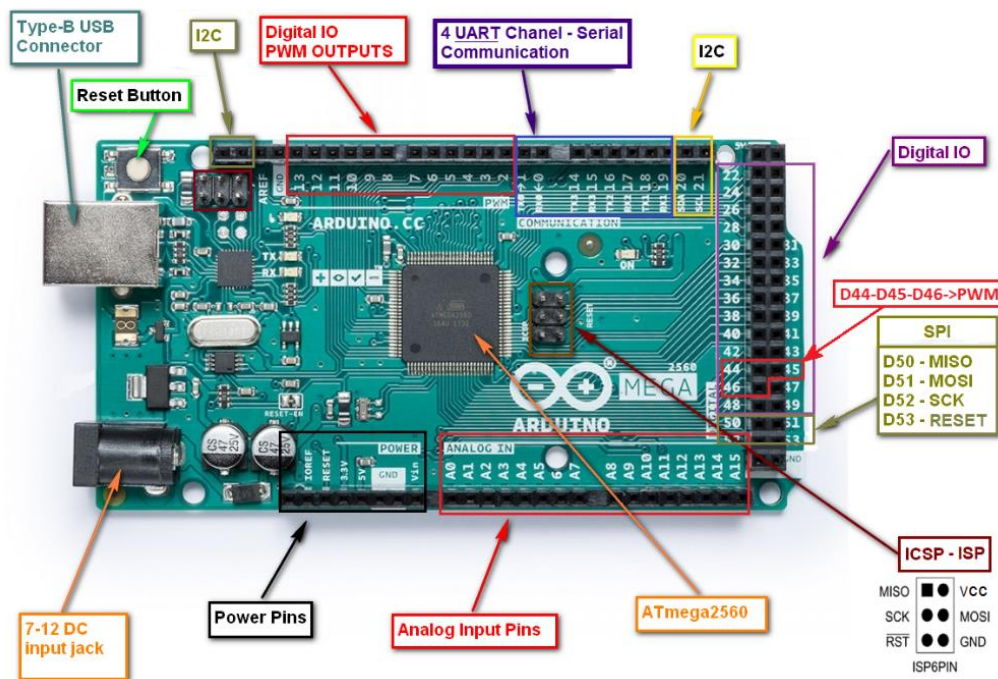
*Εικόνα 4.3 Πλακέτα Arduino Uno πίσω όψη*

Η πλατφόρμα Arduino αποτελείται από πολλές διαφορετικές πλακέτες, για να ταιριάζει για κάθε ανάγκη. Η πιο δημοφιλής πλακέτα είναι το Arduino UNO, η οποία έχει δεκαπέντε ψηφιακές ακίδες εισόδου/εξόδου, 5 αναλογικές, Flash Memory 32 KB, SRAM 2KB και EEPROM 1 KB. Οι δυνατότητες που προσφέρει το UNO δεν καλύπτουν της ανάγκες της παρούσας πτυχιακής συνεπώς η επόμενη επιλογή είναι το Arduino MEGA ή το Arduino DUE που είναι εκδόσεις με περισσότερες εισόδους/εξόδους και περισσότερη μνήμη. Η πλακέτα DUE έχει περισσότερες δυνατότητες από την πλακέτα MEGA, ωστόσο το Arduino DUE έχει πολύ κακή συμβατότητα με τα Shields. Επιλέχθηκε το Arduino MEGA 2560 (R3) Board που βασίζεται στο AT mega2560, λόγω της χρήσης του Ethernet Shield. [56] [57]

#### 4.1.2 Arduino MEGA

Το Arduino MEGA διαθέτει 54 ψηφιακές ακίδες εισόδου/εξόδου, από τις οποίες οι 15 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM, 16 αναλογικές, μια σύνδεση USB, μια κεφαλίδα ICSP και μια υποδοχή τροφοδοσίας 7-12V. Οι ψηφιακές ακίδες του Arduino δίνουν είτε 5 V (ορισμός HIGH), είτε 0 V (ορισμός LOW). Οι έξοδοι PWM σημαίνουν Pulse Width Modulation, από αυτές τις εξόδους διατίθεται αναλογική έξοδο σε τετραγωνικό σήμα κύματος, αρά γίνεται αλλαγή του χρόνου ON και OFF του σήματος χρησιμοποιώντας την εντολή analogWrite(byte) και αναγράφοντας μέσα σε παρένθεση τον Κύκλο λειτουργίας (Duty Cycle). Για 100% Duty Cycle αναγράφεται τα 255 Bytes, ενώ για 50% Duty Cycle αναγράφεται 127 bytes και ούτε καθεξής. Έτσι, με αυτόν τον τρόπο γίνεται έλεγχος για παράδειγμα της φωτεινότητας ενός LED ή της ταχύτητας ενός κινητήρα. Επιπλέον το Arduino mega έχει μνήμη Flash 256 KB, SRAM 8KB και EEPROM 4KB [58] [59]





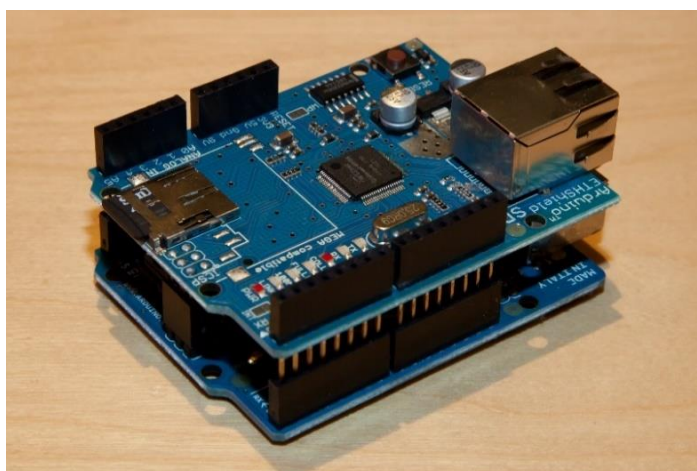
Εικόνα 4.4 Η πλακέτα Arduino mega με τα μέρη που αποτελείται. [Πηγή](#)

Το Arduino mega υποστηρίζει το πρωτόκολλο επικοινωνίας UART έχοντας 4 κανάλια επικοινωνίας, το πρωτόκολλο επικοινωνίας ISP έχοντας μια κεφαλίδα, και το πρωτόκολλο επικοινωνίας I2C έχοντας 4 ακίδες. [58]

Πρωτόκολλα	UART	I2C	SPI
<b>Δυσκολία</b>	Απλό	Εύκολη σύνδεση πολλαπλών συσκευών	Πολύπλοκο όσο αυξάνεται η συσκευή
<b>Ταχύτητα</b>	Πιο αργό	Πιο γρήγορα από το UART	Ο πιο γρήγορος
<b>Αριθμός συσκευών</b>	Έως 2 συσκευές	Έως 127, αλλά γίνεται πολύπλοκο	Πολλές, αλλά γίνεται πολύπλοκο
<b>Αριθμός καλωδίων</b>	2	2	4
<b>Duplex</b>	Full Duplex	Half Duplex	Full Duplex
<b>Αριθμός masters και slaves</b>	Single to Single	Multiple slaves and masters	1 master, Multiple slaves
<b>Data Distance</b>	15m	0,5m	0,1m
<b>Data rate</b>	20Kbps	1Mbps	25Mbps

Πίνακας 4.1 Σύγκριση των υποστηριζόμενων πρωτοκόλλων Arduino Mega. [Πηγή](#)

### 4.1.3 Arduino Ethernet Shield



Εικόνα 4.5 Ασπίδα Arduino Ethernet Shield. [Πηγή](#)

Τα Shields είναι πλακέτες που μπορούν να στοιβάζονται πάνω από τις πλακέτες Arduino αλλά και η μία με την άλλη, ώστε να αυξηθούν οι λειτουργίες του Arduino. Τα Shields είναι κατασκευασμένα στην ίδια μορφή με την πλακέτα Arduino. Τα Shields επικοινωνούν με την πλακέτα Arduino μέσω SPI ή I2C ή UART. Για τα περισσότερα Shields, οι κατασκευαστές έχουν δημιουργήσει βιβλιοθήκες και παραδείγματα, ώστε η χρήση τους να γίνει πιο εύκολη. Υπάρχουν πολλά Shields όπως: κινητήρα, Relay, Ethernet κ.λ.π.. [60]

Στην εργασία θα γίνει χρήση του Ethernet Shield, που είναι ένας ελεγκτής Ethernet, ώστε να γίνει σύνδεση του Arduino με το δίκτυο.

Ένα δίκτυο είναι μια ομάδα υπολογιστών, διακομιστών, συσκευών δικτύου, περιφερειακών ή άλλων συσκευών που είναι συνδεδεμένες μεταξύ τους, για να επιτρέπεται η κοινή χρήση δεδομένων. Για να υπάρχει ένα δίκτυο, θα πρέπει να υπάρχουν διάφοροι Servers, για να καλύπτουν τις ανάγκες του δικτύου, όπως ένας File Server που θα διαχειρίζεται τα αρχεία, το λειτουργικό σύστημα του δικτύου και την ροή των δεδομένων, ή ένας Communication Server που παρέχει την δυνατότητα αποστολής των email, όπως ένας **Web Server** για την φιλοξενία μιας ιστοσελίδας και όπως ένας Database Server, στον οποίο θα αποθηκεύονται οι όλες η πληροφορίες. Το κάθε δίκτυο αποτελείται από κόμβους. Ο κόμβος είναι κάθε συσκευή που συνδέεται στο δίκτυο. Για να είναι η δυνατή η σύνδεση ενός Server με ένα δίκτυο LAN χρειάζεται να έχουν ο καθένας τους έναν ελεγκτή Ethernet. [61] [62]

Το Arduino Ethernet Shield έρχεται με μια μονάδα Ethernet Wiznet W5500, με εσωτερικό buffer 32K και κάρτα SD, η οποία επιτρέπει στους χρήστες να έχουν πρόσβαση από το



διαδίκτυο και να αποθηκεύουν πληροφορίες σε μια κάρτα SD. Το Arduino λειτουργεί ως πελάτης ή διακομιστής ανάλογα με τις ρυθμίσεις που θα γίνουν. [63]

Κάνοντας χρήση του Ethernet Shield υπάρχει η δυνατότητα να αποσταλούν δεδομένα από τους αισθητήρες, οι οποίοι είναι συνδεδεμένοι στο Arduino, στο ThingSpeak, αλλά και στην βάση δεδομένων που θα δημιουργηθεί.

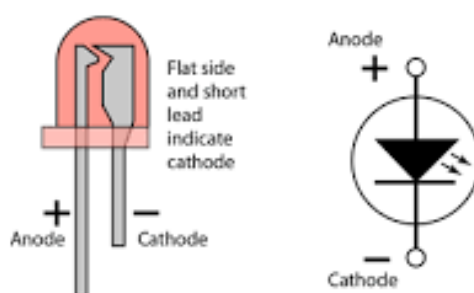
## 4.2 Δίοδος



Εικόνα 4.6 Δίοδοι 1N4007

Η δίοδος είναι ένα ημιαγωγό στοιχείο που επιτρέπει την ροή ρεύματος από την άνοδο προς την κάθοδο. Η δίοδος θα χρησιμοποιηθεί ως δίοδος Snubber που βοηθάει στην εξάλειψη των παροδικών τάσεων που προκαλούνται, όταν ένα μαγνητικό πηνίο (όπως αυτά που βρίσκονται σε κινητήρα, ρελέ ή ηλεκτρομαγνητική βαλβίδα) που χάνει ξαφνικά την ισχύ του. [64]

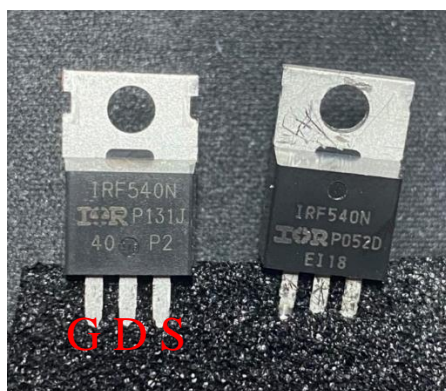
## 4.3 Light Emitting Diode (LED)



Εικόνα 4.7 Απεικόνιση Light-emitting diode. [Πηγή](#)

Τα LED (Light Emitting Diode) είναι συσκευές ημιαγωγών που αποτελούνται από μια δίοδο, που όταν πολωθεί ορθά και περάσει ρεύμα από αυτό, θα εκπέμπει φως. Τα LED υπάρχουν σε διάφορα μεγέθη και χρώματα. [65]

#### 4.4 Τρανζίστορ πεδίου ημιαγωγών οξειδίου μετάλλου (MOSFET)



Εικόνα 4.8 Τρανζίστορ MOSFET IRF540N.

Το MOSFET (Metal Oxide Semiconductor Field Effect Transistor) είναι τρανζίστορ με απομονωμένη την πύλη. Επίσης είναι διάταξη ελεγχόμενη από τάση σε αντίθεση με τα BJT που είναι ελεγχόμενου ρεύματος και για αυτό το ρεύμα της πύλης είναι πολύ μικρό. Τα MOSFET χρησιμοποιούνται σαν διακόπτες ή σαν ενίσχυση έχουν 3 ακροδέκτες Gate (G) Drain (D) και Source(S). Υπάρχουν 2 τύποι MOSFET n-channel και p-channel. Στους τύπους n-channel η πύλη συνδέεται με την  $V_{cc}$ , ενώ στους τύπους με p-channel η πύλη συνδέεται με το GND. [66]

Θα χρησιμοποιηθεί ως διακόπτης που θα ελέγχεται από το Arduino. Το Arduino μπορεί να ελέγχει συσκευές που είναι 5 Volt και για αυτό χρησιμοποιούμε το MOSFET IRF540N που είναι τύπου NPN ως διακόπτη για τον έλεγχο των ανεμιστήρων και των LED RGB που είναι 12 Volt.

#### 4.5 Ρελέ ή Ηλεκτρονόμος



Εικόνα 4.9 Ρελέ 5 Volt DC.

Το ρελέ ή ηλεκτρονόμος είναι ένας ηλεκτρομηχανικός διακόπτης που μπορεί να ανοίγει και να κλείνει της επαφές του υπό τον έλεγχο ενός αλλού ηλεκτρικού κυκλώματος μικρότερης ισχύος από το κύκλωμα εισόδου. Η λειτουργία του ρελέ ή ηλεκτρονόμου βασίζεται στο μαγνητισμό οπότε δίνοντας μια μικρή τάση για παράδειγμα 5 Volt στο πηνίο δημιουργείται μαγνητισμός και έλκει μια επαφή η οποία ανοίγει η κλίνει ανάλογα εάν είναι επαφή Normally close η Normally open, το κύκλωμα ισχύος που είναι έως 250 Volt. [67]

#### 4.6 Αισθητήρας θερμοκρασίας και υγρασίας DHT11



Εικόνα 4.10 Μπροστινή όψη αισθητήρας θερμοκρασίας DHT11.

Ο αισθητήρας DHT11 είναι ένας μικρός, οικονομικός ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας που αποτελείται από ένα θερμίστορ αρνητικού συντελεστή, στο οποίο η αντίσταση μειώνεται με την αύξηση της θερμοκρασίας, και έναν πυκνωτή ανίχνευσης υγρασίας που έχει δύο ηλεκτρόδια με ένα υπόστρωμα συγκράτησης υγρασίας ως διηλεκτρικό μεταξύ τους, ώστε να μπορεί να μετρηθεί η σχετική υγρασία. [68]

Η υγρασία είναι η συγκέντρωση των υδρατμών που υπάρχει στον αέρα, και η μέτρηση της γίνεται με τρεις τρόπους:

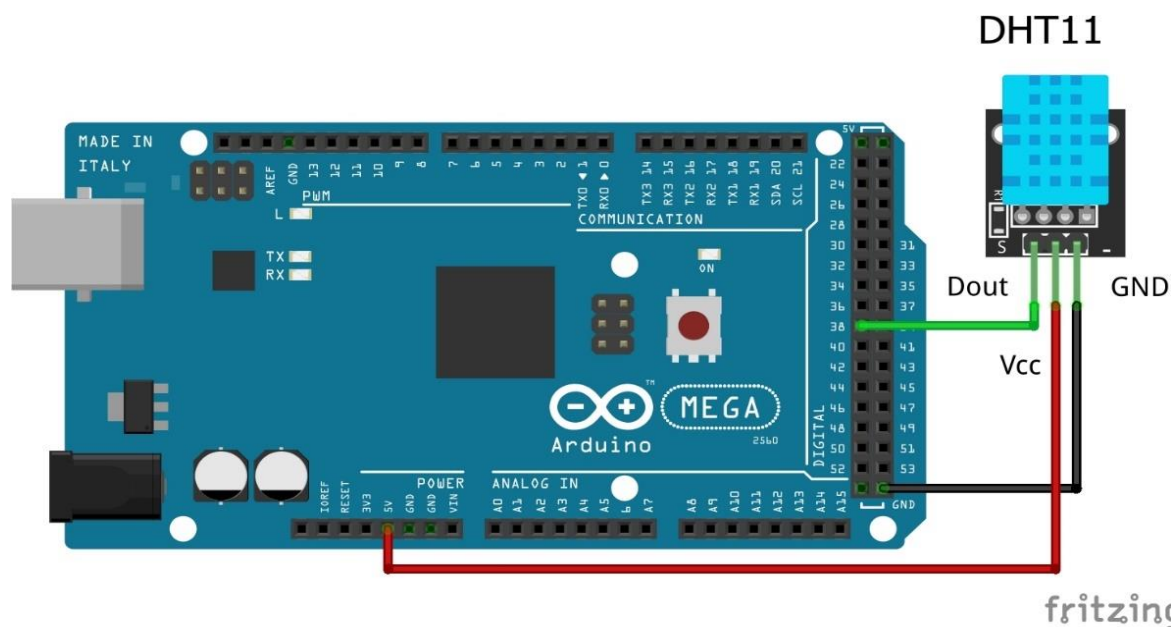
- A. από την απόλυτη υγρασία
- B. από την σχετική υγρασία και
- C. από την ειδική υγρασία.

Απόλυτη υγρασία είναι η πραγματική ποσότητα υδρατμών στον αέρα ανεξάρτητα από την θερμοκρασία. Η ειδική υγρασία είναι η αναλογία της μάζας των υδρατμών που περιέχονται στον αέρα προς τη μάζα του ξηρού αέρα για το ίδιο περιεχόμενο αέρα. Ενώ η σχετική

υγρασία που θα μετράμε με τον DHT11 είναι η πραγματική ποσότητα υδρατμών στον αέρα σε σύγκριση με την ποσότητα υγρασίας που μπορεί να υπάρχει στον αέρα στην ίδια θερμοκρασία. [69]

Ο αισθητήρας DHT11 όπως φαίνεται στην [Εικόνα 4.10](#) έχει τρεις ακίδες μια για την τάση 3,3 έως 5 Volt, μια ακίδα GND και μια ακίδα για το ψηφιακό σήμα εξόδου, το εύρος μέτρησης θερμοκρασίας είναι από 0 έως 50 βαθμούς Κελσίου και εύρος μέτρησης υγρασίας από 20 έως 100%. Επίσης, ο ρυθμός δειγματοληψίας αυτού του αισθητήρα είναι 1Hz και λειτουργεί με μέγιστο ρεύμα 2,5 mA. [68]

Στο [Κύκλωμα 4.1](#) παρουσιάζεται ο τρόπος διασύνδεσης του αισθητήρα DHT11 με το Arduino Mega.



*Κύκλωμα 4.1 Σύνδεση Arduino MEGA με τον αισθητήρα DHT11.*

Παρακάτω ακολουθεί ο κώδικας για την εμφάνιση της θερμοκρασίας και της υγρασίας στην σειριακή οθόνη.

Επίσης απαραίτητο για την λειτουργία του παρακάτω κώδικα είναι εισαγωγή της βιβλιοθήκης DHT\_sensor\_library-1.4.3, που χρησιμοποιείται για τους αισθητήρες Dht11, Dht22 κ.λ.π.

```
/*-----( Import needed libraries )-----*/
#include "DHT.h"//Βιβλιοθήκη για αισθητήρα Dht11
/*-----( Declare Constants and Pin Numbers )-----*/
#define SEN_Dht11 38 // Digital pin connected to the DHT sensor
/*-----( Declare Variables )-----*/
// defines variables
```

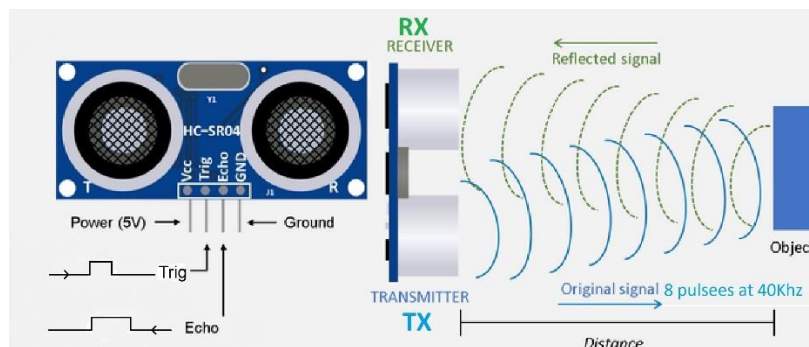
```

float Val_humi = 0; // Humidity variables
float Val_temp = 0; // Temperature variables
/*-----( Declare objects )-----*/
// select sensor type the DHT
#define DHTTYPE DHT11
DHT dht(SEN_Dht11, DHTTYPE);

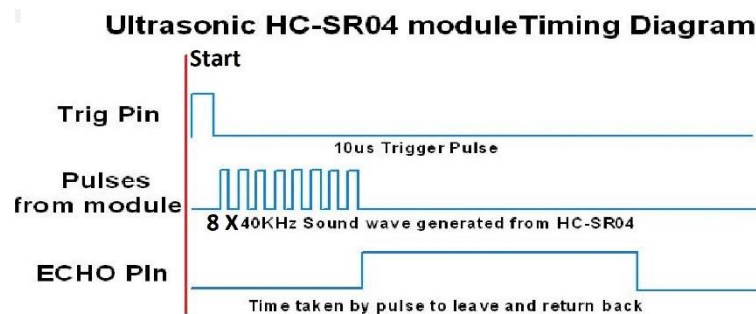
void setup() {
//Serial Communication is starting with 9600 of baud rate speed
Serial.begin(9600);
dht.begin();// Start the DHT
}
void loop() {
  Read_Sen_Dht11_Loop(); // read the data sensor
  Print_Serial_loop(); // print the data sensor to serial monitor
}.
// Read data Dht11 Data sensor
void Read_Sen_Dht11_Loop() {
// read the humidity
Val_humi = dht.readHumidity();
// Read temperature as Celsius (the default)
Val_temp = dht.readTemperature();
}
// Print to serial monitor Data sensor
void Print_Serial_loop(){
// Εμφάνιση υγρασίας
Serial.print(F("Humidity: "));
Serial.print(Val_humi);
//Εμφάνιση Θερμοκρασίας
Serial.print(F("% Temperature: "));
Serial.print(Val_temp);Serial.println(F("°C "));
}

```

#### 4.7 Αισθητήρας υπέρυθρων Ultrasonic HC-SR04



Εικόνα 4.11 Απεικόνιση αισθητήρα υπέρυθρων και σημάτων. [Πηγή](#)



Εικόνα 4.12 Διάγραμμα χρονισμού υπέρυθρων HC-SR04. [Πηγή](#)

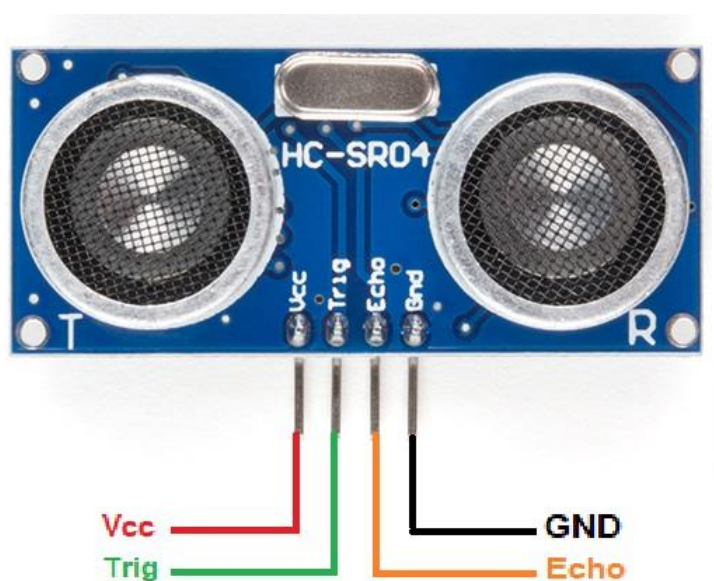


Ο αισθητήρας υπερήχων Ultrasonic HC-SR04 είναι ένας αισθητήρας που μετράει απόσταση μεταξύ αυτού και του αντικείμενου που βρίσκεται μπροστά του χρησιμοποιώντας υπερηχητικά κύματα. Για να γίνει εφικτό αυτό, έχει 2 μετατροπείς συχνότητας όπως φαίνεται στην [Εικόνα 4.11](#) οι οποίοι μετατρέπουν την ηλεκτρική ενέργεια σε υπερηχητικά κύματα και αντίστροφα. Το ένα λειτουργεί ως πομπός δηλαδή εκπέμπει ένα υπερηχητικό κύμα 40 KHz και όταν χτυπήσει ένα αντικείμενο αντανακλάται και το λαμβάνει ο δεύτερος μετατροπέας συχνότητας οπότε είναι ο δέκτης. Ανάλογα με την ταχύτητα του ήχου και τον χρόνο που θα κάνει το σήμα μεταξύ πομπού και δέκτη και με αυτό τον τρόπο υπολογίζει την απόσταση. [70]

Ο τύπος με τον οποίο υπολογίζεται η απόσταση είναι ( $L = 1/2 * Χρόνος * C$ ), όπου

- L είναι η απόσταση
- T είναι ο χρόνος μεταξύ της εκπομπής και της λήψης και
- C είναι η ηχητική ταχύτητα. (Η τιμή πολλαπλασιάζεται επί 1/2 επειδή T είναι ο χρόνος για την απόσταση μετάβασης και επιστροφής).

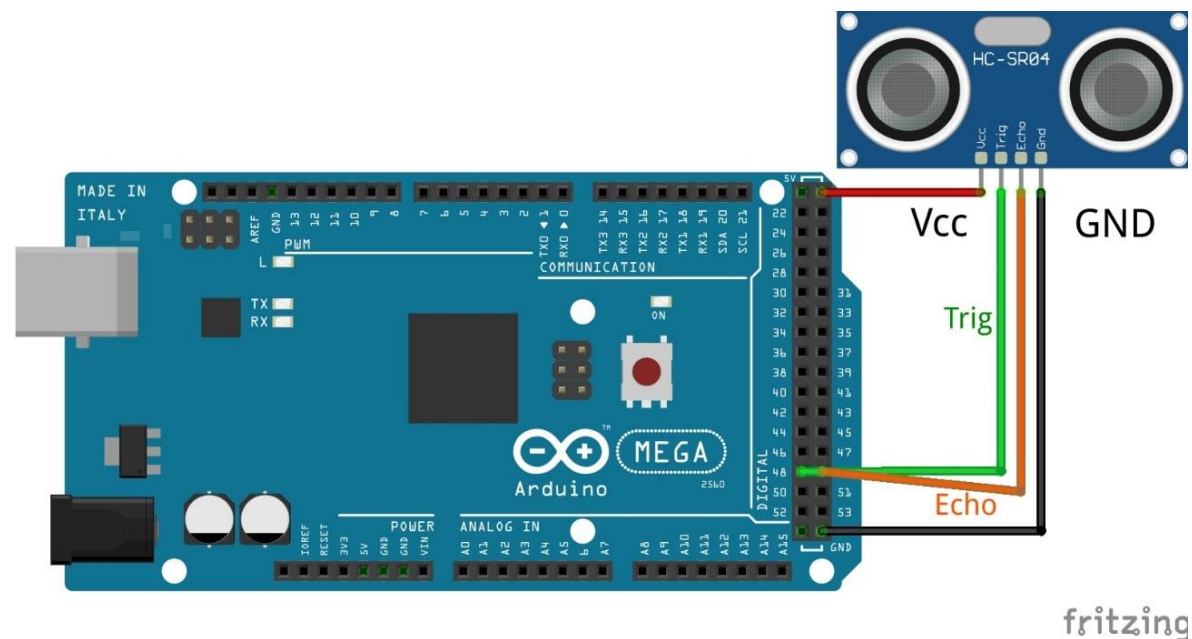
Όπως απεικονίζεται στην [Εικόνα 4.12](#) για να ξεκινήσει η μέτρηση, θα πρέπει ένας μικροελεγκτής να στείλει ένα σήμα ενεργοποίησης 10μs, μόλις ενεργοποιηθεί ο αισθητήρας Ultrasonic, στέλνει 8 παλμούς των 40 kHz και εκκινεί έναν χρονοδιακόπτη. Μόλις ληφθεί το απεσταλμένο σήμα ο χρονοδιακόπτης σταματάει και έτσι η έξοδος του αισθητήρα είναι ένας παλμός που έχει διάρκεια όσο η χρονική διάρκεια του χρονοδιακόπτη. Ο αισθητήρας υπερήχων Ultrasonic HC-SR04 έχει εύρος από 1cm έως 3m.



Εικόνα 4.13 Αισθητήρας υπέρυθρων Ultrasonic HC-SR04. [Πηγή](#)

Όπως φαίνεται στην [Εικόνα 4.13](#) ο αισθητήρας έχει 4 ακροδέκτες ο πρώτος ακροδέκτης είναι ο VCC που είναι η τροφοδοσία του (από 3,3 έως 5V), ο δεύτερος ακροδέκτης είναι ο TRIG που είναι αυτός που λαμβάνει το σήμα ενεργοποίησης από τον μικροελεγκτή, ο τρίτος ακροδέκτης ECHO είναι η έξοδος του αισθητήρα που είναι το χρονικό διάστημα που διένυσε το ηχητικό κύμα και τέταρτος ακροδέκτης είναι ο GND. [71]

Στο [Κύκλωμα 4.2](#) παρουσιάζεται ο τρόπος διασύνδεσης του αισθητήρα υπέρυθρων Ultrasonic HC-SR04 με το Arduino Mega.



*Κύκλωμα 4.2 Σύνδεση Arduino MEGA με αισθητήρα Ultrasonic HC-SR04.*

Ακολουθεί παράδειγμα κώδικα για εμφάνιση επιπέδου νερού σε εκατοστά και επί τοις εκατό στην σειριακή οθόνη.

Επίσης απαραίτητο για την λειτουργία του παρακάτω κώδικα είναι εισαγωγή της βιβλιοθήκης HCSR04\_ultrasonic\_sensor-2.0.3, που χρησιμοποιείται για των αισθητήρα Ultrasonic HC-SR04.

```

/*-----( Import needed libraries )-----*/
#include <HCSR04.h> //library Ultrasonic Sensor HC-SR04

/*-----( Declare Constants and Pin Numbers )-----*/
//attach pin D49 Arduino to pin Trig of HC-SR04
#define SEN_UltraSonicTrigPin 49
//attach pin D48 Arduino to pin Echo of HC-SR04
#define SEN_UltraSonicEchoPin 48

/*-----( Declare objects )-----*/
//initialisation class HCSR04 (trig pin , echo pin)
HCSR04 hc(SEN_UltraSonicTrigPin, SEN_UltraSonicEchoPin);

/*-----( Declare Variables )-----*/

```

```

int Val_distance; // variable for the distance measurement
int Val_distancePerCent; // Variable for distance per cent

void setup(){
//Serial Communication is starting with 9600 of baud rate speed
  Serial.begin(9600);

// print some text in Serial Monitor
  Serial.println(F("Ultrasonic Sensor HC-SR04 Test"));
  Serial.println(F("with Arduino UNO R3"));

pinMode(SEN_UltraSonicTrigPin, OUTPUT); // Sets the trig Pin as an OUTPUT
pinMode(SEN_UltraSonicEchoPin, INPUT); // Sets the echo Pin as an INPUT
}

void loop() {
  Print_Serial_loop();
  Read_Sen_UltraSonic_Loop();
}

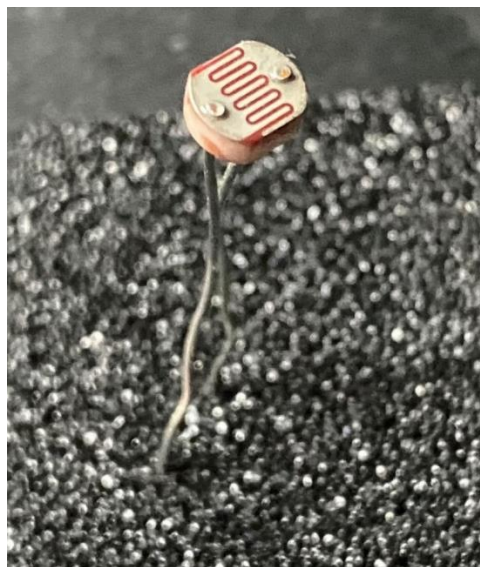
void Read_Sen_UltraSonic_Loop(){
Val_distance = hc.dist(); // Read Sensor ultrasonic data

//Κάνουμε αντιστοίχιση των τιμών του αισθητήρα UltraSonic σε
//εκατοστά σε εκατό τις εκατό για την καλύτεροί απεικόνιση της
//στάθμης νερού στο δοχείο. Το δοχείο είναι 17 εκατοστά αρά
//τότε θα έχει 0% νερό ενώ εάν είναι 0 εκ. τότε έχουμε 100%
//νερό μέσα στο δοχείο.
  Val_distancePerCent = map(Val_distance, 17, 0, 0, 99);
}

void Print_Serial_loop(){
  // Displays the distance on the Serial Monitor
  Serial.print(F("Distance percent: "));
  Serial.print(Val_distance);
  Serial.println(F("%"));
}

```

#### 4.8 Αισθητήρας φωτοαντίστασης LDR



Εικόνα 4.14 Απεικόνιση αισθητήρας φωτοαντίστασης.

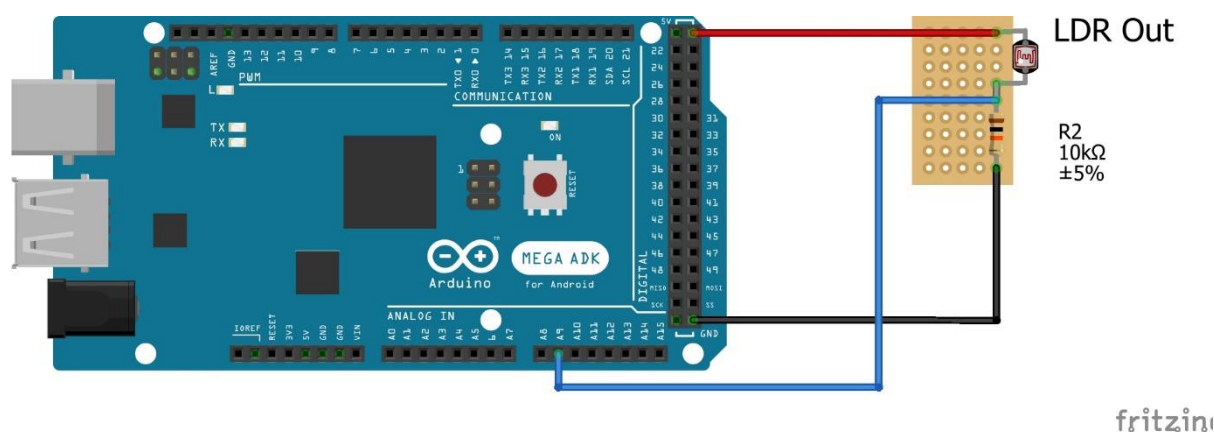


Ο αισθητήρας φωτοαντίστασης LDR (Light Dependent Resistor) είναι μια αναλογική συσκευή εισόδου που αποτελείται από μια αντίσταση κατασκευασμένη από ημιαγώγιμα υλικά. Η ειδική αντίσταση του μεταβάλλεται ανάλογα με την φωτεινότητα που προσπίπτει στην επιφάνεια του. Όταν η φωτεινότητα είναι χαμηλή ο αισθητήρας έχει μεγάλη αντίσταση ενώ όταν η φωτεινότητα είναι υψηλή έχει μικρή αντίσταση.

Οι φωτοαντιστάσεις λειτουργούν με βάση την αρχή της φωτοαγωγιμότητας. Η φωτοαγωγιμότητα είναι ένα οπτικό φαινόμενο κατά το οποίο η αγωγιμότητα του υλικού αυξάνεται όταν το φως απορροφάται από το υλικό. [72]

Όταν το φως προσπίπτει στην επιφάνεια του LDR, τα ηλεκτρόνια στη ζώνη σθένους του ημιαγώγιμου υλικού διεγείρονται στη ζώνη αγωγιμότητας. Αυτά τα φωτόνια του φωτός που προσπίπτουν στον LDR θα πρέπει να έχουν ενέργεια μεγαλύτερη από το διάκενο ζώνης του ημιαγωγού, για να κάνουν τα ηλεκτρόνια να πηδούν από τη ζώνη σθένους στη ζώνη αγωγιμότητας. Ως εκ τούτου, όταν το φως με αρκετή ενέργεια χτυπά στη συσκευή, όλο και περισσότερα ηλεκτρόνια διεγείρονται στη ζώνη αγωγιμότητας που οδηγεί σε μεγάλο αριθμό φορέων φορτίου. Το αποτέλεσμα αυτής της διαδικασίας είναι ότι όλο και περισσότερο ρεύμα αρχίζει να ρέει μέσω της συσκευής, όταν το κύκλωμα είναι κλειστό, και ως εκ τούτου λέγεται ότι η αντίσταση της συσκευής έχει μειωθεί. Το LDR έχει 2 ποδαράκια χωρίς πολικότητα και τροφοδοτείται με  $V_{cc} +5$  έως  $3.3V$  και με την γη  $GND$ . [73]

Στο [Κύκλωμα 4.3](#) παρουσιάζεται ο τρόπος διασύνδεσης του αισθητήρα LDR με το Arduino Mega.



Κύκλωμα 4.3 Σύνδεση Arduino MEGA με αισθητήρα LDR.

Ακολουθεί παράδειγμα κώδικα για εμφάνιση δεδομένων LDR στην σειριακή οθόνη.

```
/*-----( Declare Constants and Pin Numbers )-----*/  
// Analog pin A9 connected to the LDR Sensor  
int SEN_LdrOut = A9;
```

```

/*-----( Declare Variables )-----*/
//LDR Variable
int Val_OutLdr = 0;

void setup(){
  //Serial Communication is starting with 9600 of baud rate speed
  Serial.begin(9600);
}

void loop(){
  //Read the data sensor
  Read_Sen_OutLDR_Loop();
  // Print the data sensor to serial monitor
  Print_Serial_loop();
}

void Read_Sen_OutLDR_Loop() {
  // read the analog data 0-1023 byte
  Val_OutLdr = analogRead(SEN_LdrOut);
  //Αντιστοιχίσει των Bytes σε επί τοις εκατό
  Val_OutLdr = map(Val_OutLdr, 400, 1030, 0, 100);
}

//εμφάνιση τιμών στην σειριακής οθόνης
void Print_Serial_loop(){
  Serial.print(F("LDR Value is: "));
  Serial.println(Val_OutLdr);
  Serial.print(F(" %"));
}

```

#### 4.9 Αισθητήρας αερίου MQ-07



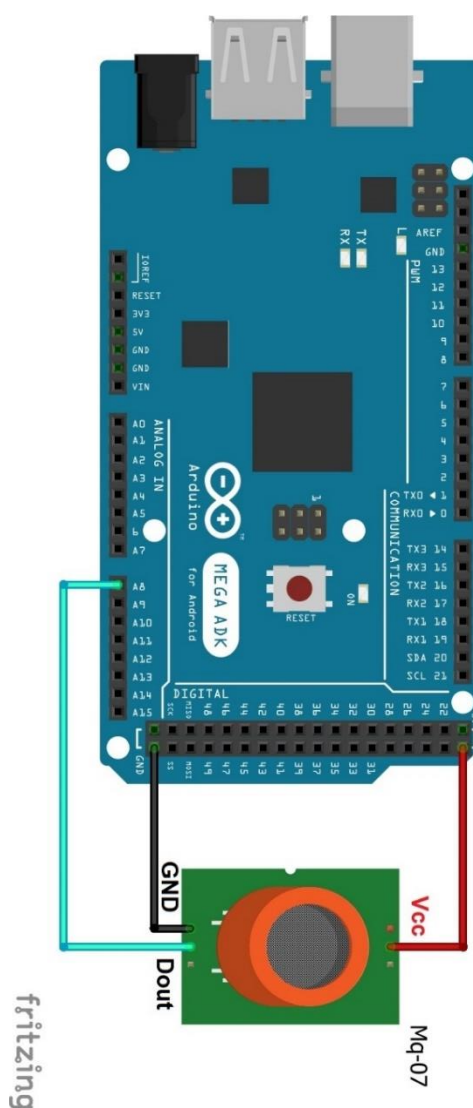
Εικόνα 4.15 Απεικόνιση αισθητήρα αερίου. [Πηγή](#)

Ένας αισθητήρας αερίου είναι μια συσκευή που χρησιμοποιείται για τη μέτρηση της συγκέντρωσης αερίων στο περιβάλλον. Στο εσωτερικό του περιέχει έναν χημικό αισθητήρα ο οποίος συνήθως περιέχει Οξείδιο του κασσίτερου (SNO<sub>2</sub>) που έχει χαμηλή αγωγιμότητα στον καθαρό αέρα. Όσον αφορά τον αισθητήρα, με βάση την συγκέντρωση του αερίου η τάση στην έξοδο του μεταβάλλεται, μεταβάλλοντας την αντίσταση του υλικού. Όταν

υπάρχει υψηλή συγκέντρωση αερίων, τότε η αντίσταση μειώνεται, ενώ εάν υπάρχει χαμηλή συγκέντρωση αερίων, τότε η αντίσταση αυξάνεται. [74] [75]

Όπως απεικονίζετε στην [Εικόνα 4.15](#) ο αισθητήρας έχει 4 ακίδες, μία για την τροφοδοσία Vcc από 3,3V έως 5V, μία για την γη GND, και 2 ακόμα για την έξοδο που μία είναι για την ψηφιακή έξοδο και το άλλο για αναλογική έξοδο.

Στο [Κύκλωμα 4.4](#) παρουσιάζεται ο τρόπος διασύνδεσης του αισθητήρα αερίου με το Arduino Mega.



Κύκλωμα 4.4 Σύνδεση Arduino MEGA με αισθητήρα MQ-07.

Ακολουθεί παράδειγμα κώδικα για εμφάνιση δεδομένων αισθητήρα αερίου MQ-07 στην σειριακή οθόνη.

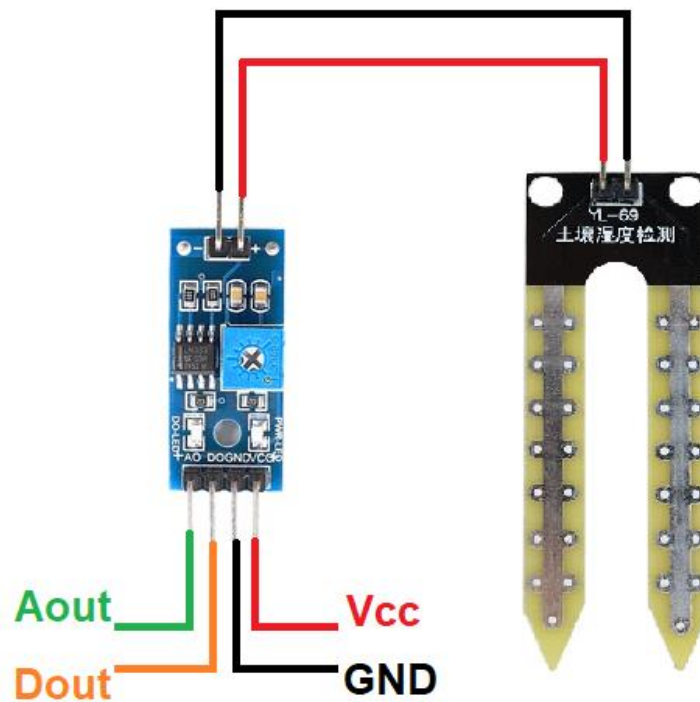
```
#define GASSENSOR A8 // Analog pin A8 connected to the Mq-07 Sensor
void setup() {
  //Start the serial screen on the 9600 baud
  Serial.begin(9600);
}
```

```

    pinMode(GASSENSOR, INPUT); //Set the pin mode
}
void loop() {
    GasSensor();
}
void GasSensor() {
    int gas = analogRead(GASSENSOR); //read the sensor analog data
    //print the Analog data 0-1023 to serial monitor
    Serial.print("Gas is: "); Serial.println(gas);
}

```

#### 4.10 Αισθητήρας υγρασίας εδάφους Soil Hygrometer

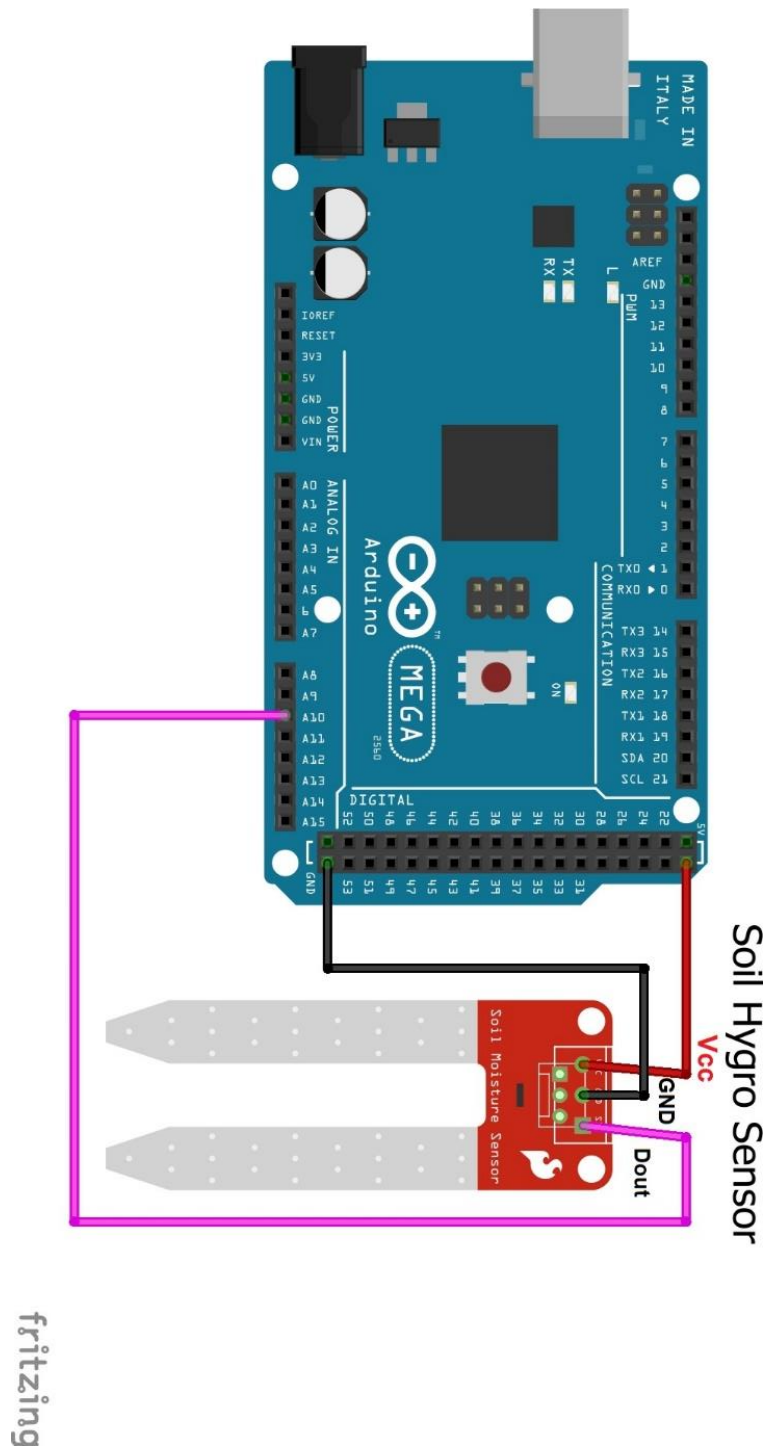


Εικόνα 4.16 Απεικόνιση αισθητήρα υγρασίας εδάφους. [Πηγή](#)

Ο αισθητήρας υγρασίας εδάφους Soil Moisture Sensor and Hygrometer Module είναι μια συσκευή που χρησιμοποιείται για την ανίχνευση και την μέτρηση της υγρασίας στο έδαφος. Ο αισθητήρας μετράει την ογκομετρική περιεκτικότητα του νερού μέσα στο έδαφος. Ακόμα ο αισθητήρας έχει δύο ανιχνευτές που όταν τοποθετούνται στο έδαφος σχηματίζουν μια μεταβλητή αντίσταση η οποία μεταβάλλεται ανάλογα με την περιεκτικότητα νερού στο έδαφος. Η αντίσταση αυξάνεται όταν η υγρασία του εδάφους είναι χαμηλή διότι δεν υπάρχει καλή αγωγιμότητα ενώ όταν η υγρασία είναι υψηλή η αντίσταση μειώνεται διότι υπάρχει καλύτερη αγωγιμότητα μεταξύ των δύο ανιχνευτών. [76]

Όπως φαίνεται στην [Εικόνα 4.16](#) ο αισθητήρας έχει 4 ακίδες, οι 2 από αυτές είναι για την τροφοδοσία 1 ακίδα για GND και άλλη μια για την Vcc που είναι από 3,3V έως 5V η άλλες 2 ακίδες είναι η έξοδος του η μια είναι για χρήση ως αναλογική και η άλλη για ψηφιακή.

Στο [Κύκλωμα 4.5](#) παρουσιάζεται ο τρόπος διασύνδεσης του αισθητήρα υγρασίας εδάφους με το Arduino Mega.



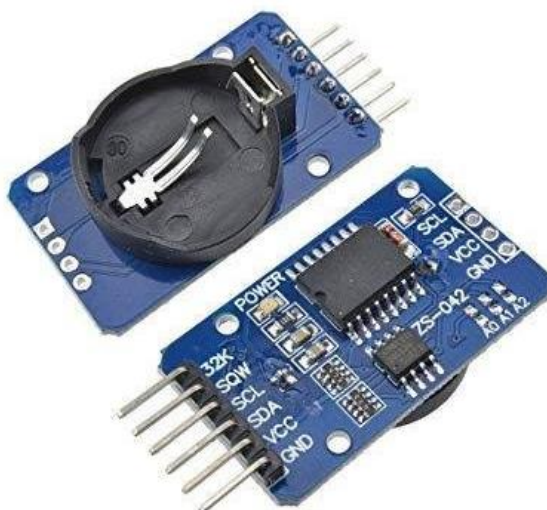
Κύκλωμα 4.5 Σύνδεση Arduino MEGA με αισθητήρα Soil Hygrometer sensor.

Ακολουθεί παράδειγμα κώδικα για εμφάνιση δεδομένων αισθητήρα Soil Hygrometer sensor στην σειριακή οθόνη.

```
#define HYGROPIN A10 // Analog pin A10 connected to the Soil Hygrometer
Sensor
void setup() {
// Serial Begin so we can see the data from the moisture sensor in our
//serial input window.
Serial.begin(9600);
}

void loop() {
int hygro = analogRead(HYGROPIN); //Read analog value Soil Hygrometer
//Εμφάνιση στην σειριακή οθόνη της αναλογικές τιμές του αισθητήρα
Serial.print("Soil Hygrometer= ");
Serial.print(hygro);
}
```

#### 4.11 Ρολόι πραγματικού χρόνου - Real Time Clock (RTC DS3231)

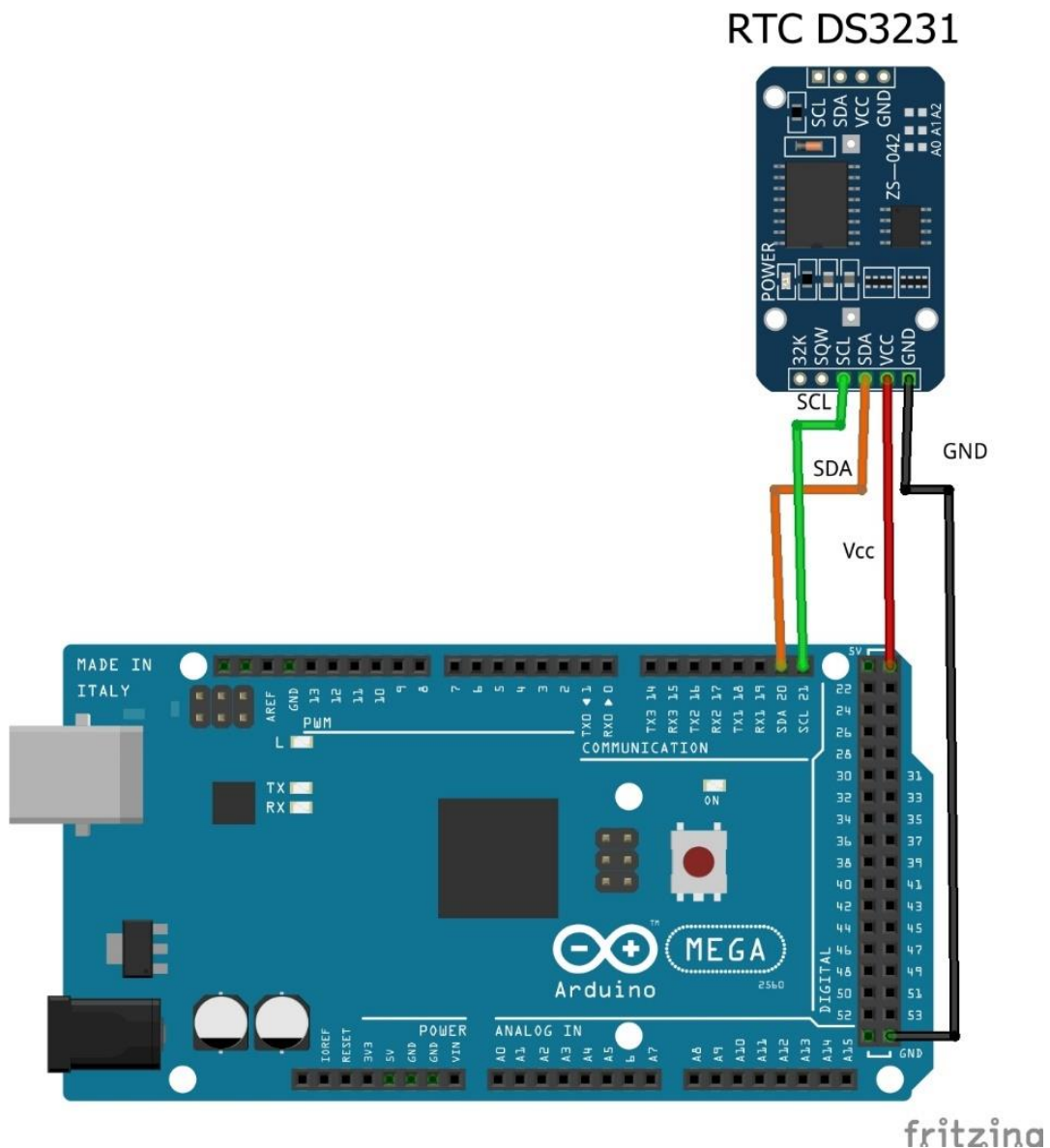


Εικόνα 4.17 Απεικόνιση συσκευής ρολογιού πραγματικού χρόνου DC3231. [Πηγή](#)

Το ρολόι πραγματικού χρόνου Real Time clock (RTC) DC3231 είναι ένα ρολόι χαμηλής κατανάλωσης, χαμηλού κόστους και υψηλής ακρίβειας που μετράει τον χρόνο σε ώρες, λεπτά, δευτερόλεπτα, ημέρες, μήνες και χρόνια. Επιπρόσθετα το RTC είναι ολοκληρωμένο chip με 8 ακίδες που μετράει τον χρόνο. Επίσης όπως φαίνεται στην [Εικόνα 4.17](#) έχει 6 ακίδες από της οποίες η πρώτη ακίδα είναι για το GND γη, η επόμενη είναι για την τάση που είναι από 3,3 έως 5 Volt. Παράλληλα με την τροφοδοσία του έχει και μια εφεδρική μπαταρία σε περίπτωση που χάσει την τροφοδοσία του. Στην επόμενη ακίδα είναι το SDA SCL για σύνδεση με μικροελεγκτή μέσω διαύλου SPI ή I2C. Ενώ η SQW ακίδα εξάγει ένα τετραγωνικό σήμα 1Hz, 4kHz, 8kHz ή 32kHz. Τέλος η ακίδα 32K είναι η έξοδος του ταλαντωτή 32 Khz. [77] [78]



Στο [Κύκλωμα 4.6](#) παρουσιάζεται ο τρόπος διασύνδεσης του ρολογιού με το Arduino Mega.



*Κύκλωμα 4.6 Σύνδεση Arduino MEGA με ρολόι RTC.*

Παρακάτω ακολουθεί ο κώδικας για την εμφάνιση της ημέρας, του μήνα, του έτους, της ώρας, του λεπτού και του δευτερολέπτου στην σειριακή οθόνη.

Επίσης απαραίτητο για την λειτουργία του παρακάτω κώδικα είναι εισαγωγή της βιβλιοθήκης Wire που επιτρέπει την επικοινωνία με συσκευές I2C και της βιβλιοθήκης DS3231 και έκδοση 1.1.0 που είναι για το ρολόι πραγματικού DS3231.

```
//Βιβλιοθήκη που επιτρέπει να επικοινωνείτε με συσκευές I2C
//Mega2560 20 (SDA), 21 (SCL)
#include <Wire.h>
#include <DS3231.h> //library for real time clock RTC ds3231
DS3231 rtc(SDA, SCL); // I2C communication
Time t;
void setup() {
```

```

Serial.begin(9600); //start serial to 9600 baud
Wire.begin(); //Start the I2C protocol
rtc.begin(); // Start RTC dc331 on Arduino
}
void loop() {
  t = rtc.getTime(); //read the time and date
  //print to serial the time and date
  Serial.print("Date : "); Serial.print(t.date); Serial.print("/");
  Serial.print(t.mon); Serial.print("/"); Serial.print(t.year);
  Serial.print("Hour : "); Serial.print(t.hour); Serial.print(":");
  Serial.print(t.min); Serial.print(":"); Serial.println(t.sec);
}

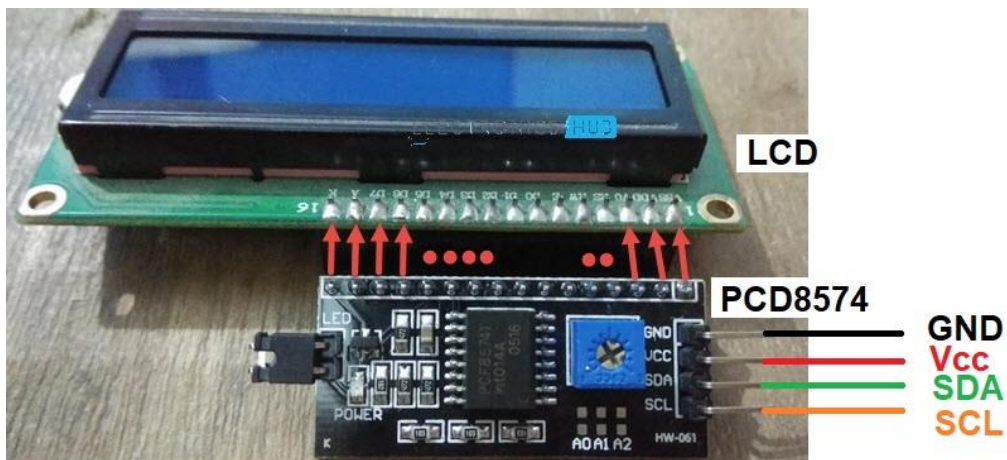
```

#### 4.12 Οθόνη υγρών κρυστάλλων LCD



Εικόνα 4.18 Απεικόνιση Οθόνης υγρών κρυστάλλων.

Η οθόνη υγρών κρυστάλλων LCD είναι μια οικονομική οθόνη κατάλληλη για απεικόνιση αριθμών και κειμένου. Οι οθόνες αυτές υπάρχουν σε διαφορά μεγέθη όπως 16x2, 20x4 και άλλα, για παράδειγμα η οθόνη 16x2 που χρησιμοποιείται στην εργασία έχει 16 στήλες και 2 σειρές. Στις 16 στήλες μπορούν να χρησιμοποιηθούν 16 χαρακτήρες, άρα συνολικά μπορούν να χρησιμοποιηθούν 32 χαρακτήρες και στις 2 σειρές.

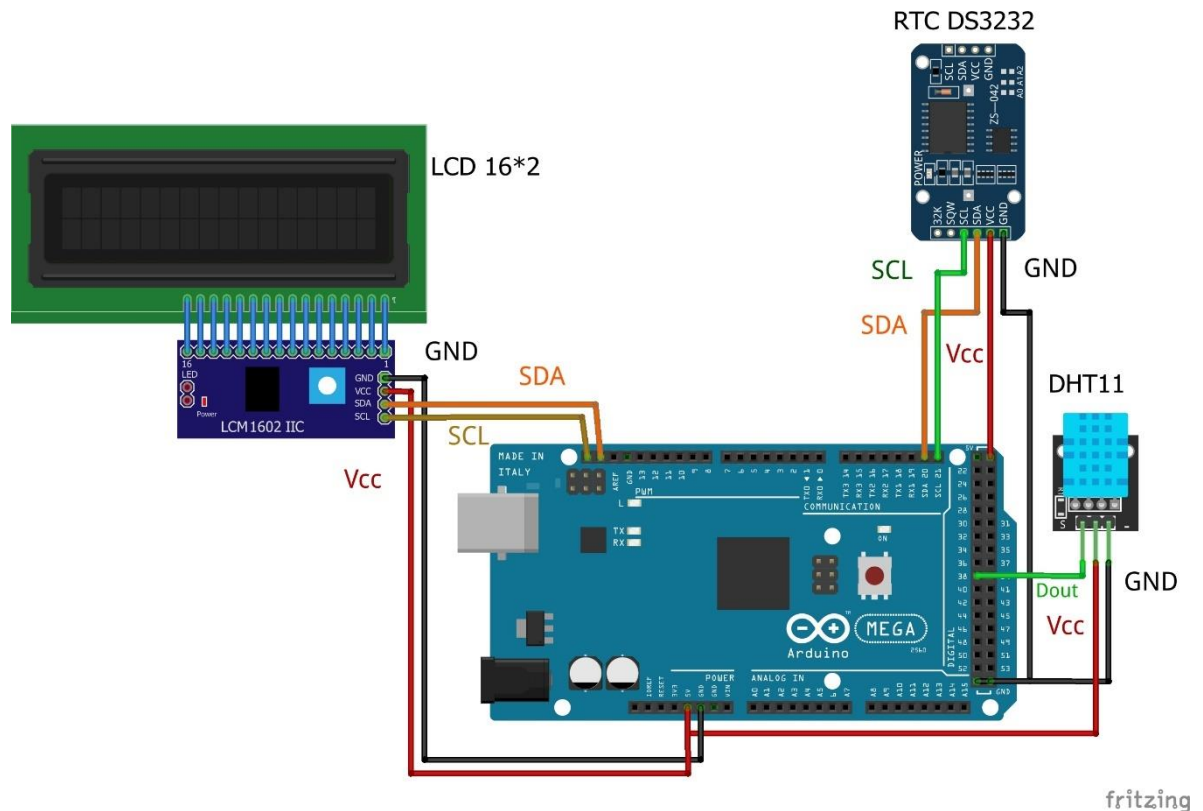


Εικόνα 4.19 Απεικόνιση σύνδεσης οθόνης LCD με το Chip PCF8574. [Πηγή](#)



Η οθόνη αυτή έχει 12 ακίδες που δυσκολεύει την σύνδεση με το Arduino, αλλά χρησιμοποιώντας το chip PCF8574 γίνεται χρήση της επικοινωνίας I2C και οι ακίδες θα μειωθούν σε 4 ακίδες, όπως φαίνεται στην [Εικόνα 4.19](#) όπου η ακίδα 1 είναι το GND, η ακίδα 2 είναι η Vcc +5V και οι επόμενες 2 ακίδες SDA και SCL είναι για την επικοινωνία I2C. [79] [80]

Στο [Κύκλωμα 4.7](#) παρουσιάζεται ο τρόπος διασύνδεσης της οθόνης με το Arduino Mega, σε συνδυασμό με το ρολόι και το αισθητήρα θερμοκρασίας.



Κύκλωμα 4.7 Σύνδεση Arduino MEGA με οθόνη LCD, ρολόι RTC DS3231 και αισθητήρα DHT11..

Παρακάτω ακολουθεί κώδικας χρήσης της οθόνης σε συνδυασμό με το ρολόι και τον αισθητήρα θερμοκρασίας.

Επίσης απαραίτητο για την λειτουργία του παρακάτω κώδικα είναι εισαγωγή της βιβλιοθήκης Wire που επιτρέπει την επικοινωνία με συσκευές I2C και της βιβλιοθήκης LiquidCrystal\_I2C-1.1.2 για την λειτουργία της οθόνης LCD.

```

/*-----( Import needed libraries )-----*/

#include "DHT.h" // library for Dht11
//Βιβλιοθήκη που επιτρέπει να γίνει χρήση επικοινωνίας με συσκευές I2C
//για το arduino Mega2560 η ακίδες I2C είναι η 20 για το SDA και η 21 για
SCL
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // library for LCD

```

```

#include <DS3231.h> //library for real time clock RTC ds3231
/*-----( Declare Constants and Pin Numbers )-----*/
#define SEN_Dht11 38 // Digital pin connected to the DHT sensor
/*-----( Declare Variables )-----*/
// defines variables
float Val_humi = 0; // Humidity variables
float Val_temp = 0; // Temperature variables
// LCD address and geometry and library initialization
const byte lcdAddr = 0x27; // Address of I2C backpack
const byte lcdCols = 16; // Number of characters in a row
const byte lcdRows = 2; // Number of lines
/*-----( Declare objects )-----*/
#define DHTTYPE DHT11 // selected sensor DHT type the DHT 11
DHT dht(SEN_Dht11, DHTTYPE);
// set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(lcdAddr, lcdCols, lcdRows);
DS3231 rtc(SDA, SCL);
Time t;
//Η δημιουργία χαρακτήρα στην LCD οθόνη 16*2 είναι εφικτή
//διότι κάθε εάν από τα 16*2 κουτάκια που έχει οθόνη LCD
//το καθένα περιέχει 5&8 για να απεικονίσει
//κάθε χαρακτήρα οπότε ορίζοντας ποιο pixel θα ανάβει
//μπορούμε να δημιουργήσουμε χαρακτήρες.
byte thermometer [8] = { //icon for thermometer
    B00100,
    B01010,
    B01010,
    B01110,
    B01110,
    B11111,
    B11111,
    B01110
};
byte drop[8] = { //icon for water Drop
    B00100,
    B00100,
    B01010,
    B01010,
    B10001,
    B10001,
    B10001,
    B01110,
};
void setup(){
    Serial.begin(9600); //start serial to 9600 baud
    dht.begin();// Start the dht11
    rtc.begin();// Start RTC dc331 on Arduino
    Wire.begin(); //Start the I2C protocol
    lcd.begin(16, 2); //Να ξεκινήσει η οθόνη για 16 στήλες και 2 γραμμές
    lcd.init(); // initialize the lcd
    lcd.backlight(); //Turns the backlight on
    lcd.createChar(1, thermometer); // Δημιουργία χαρακτήρα θερμομέτρου
    lcd.createChar(2, drop); // Δημιουργία χαρακτήρα σταγόνας
    lcd.clear(); //Καθαρισμός οθόνης
}

void loop(){
    //Το Loop για την εμφάνιση των δεδομένων στην οθόνη
    Print_Lcd_loop();
    // print the data sensor to serial monitor
    Print_Serial_loop();
}

```

```

//----Read the value of Real Time Clock-----
void Read_Rtc_Loop() {
  t = rtc.getTime();
}
void Read_Sen_Dht11_Loop() {
  Val_humi = dht.readHumidity(); // read the humidity
  Val_temp = dht.readTemperature(); // Read Temperature as Celsius (the
  default)
}
// Displays the value on the Serial Monitor
void Print_Lcd_loop() {
  Read_Rtc_Loop(); // διάβασε τον χρόνο
  Read_Sen_Dht11_Loop(); //διάβασε την θερμοκρασία και υγρασία
  // start to print at the first row the temperature
  lcd.setCursor(0, 0); lcd.write(1); lcd.print(Val_temp);
  lcd.print((char)223); lcd.print("C ");
  //Print the humidity
  lcd.setCursor(9, 0); lcd.write(2); lcd.print(Val_humi);
  lcd.print("%");
  //print the time
  lcd.setCursor(1, 1); lcd.print(" Time: "); lcd.print(t.hour);
  lcd.print(":"); lcd.print(t.min); lcd.print(":");
  lcd.println(t.sec);
}
// Print to serial monitor Data sensor
void Print_Serial_loop(){
  Serial.println(F("*****"));
  Serial.print(F("Humidity: ")); Serial.print(Val_humi);
  Serial.print(F("% "));
  Serial.print(F("Temperature: "));
  Serial.print(Val_temp);Serial.println(F("°C "));
}

```

Το αποτέλεσμα από των παραπάνω κώδικα εμφανίζεται στην [Εικόνα 4.19](#).



*Εικόνα 4.20 Παρουσίαση αποτελέσματος του κώδικα.*

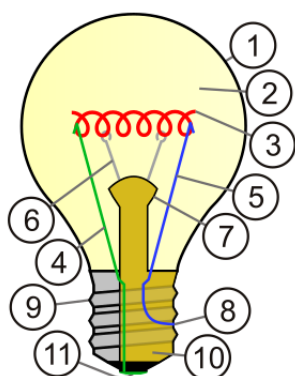
### 4.13 Ανεμιστήρας.



Εικόνα 4.21 Ανεμιστήρας υπολογιστή 12 Volt.

Ο ανεμιστήρας 12 Volt είναι ένας κινητήρας DC Brushless που έχει συνδεδεμένη μια φτερωτή πάνω στον άξονα του. Οι κινητήρες DC Brushless σε αντίθεση με τους DC κινητήρες μόνιμου μαγνήτη δεν έχουν ψήκτρες, αλλά χρησιμοποιούν μια πλακέτα που λειτουργεί σαν διακόπτης με έναν αισθητήρα HALL για την εύρεση των πόλων. Συνοψίζοντας, όταν τροφοδοτηθεί με DC ρεύμα ο στάτης δημιουργεί ένα σταθερό μαγνητικό πεδίο και έχοντας το μαγνητικό πεδίο από τους μόνιμους μαγνήτες δημιουργείται κίνηση της φτερωτής. [81] [82]

### 4.14 Λαμπτήρας πυράκτωσης



1. Περίγραμμα γυάλινης λάμπας
2. Χαμηλής πίεσης αδρανές αέριο ( αργό , άζωτο , κρυπτό , ξένο )
3. Νήμα βολφραμίου
4. Σύρμα επαφής (μπαίνει στο στέλεχος)
5. Σύρμα επαφής (φεύγει από το στέλεχος)
6. Καλώδια στήριξης (το ένα άκρο είναι ενσωματωμένο στο στέλεχος, χωρίς ρεύμα)
7. Στέλεχος (γυάλινη βάση)
8. Σύρμα επαφής (φεύγει από το στέλεχος)
9. Καπάκι (μανίκι)
10. Μόνωση ( υαλώδες άλας )
11. Ηλεκτρική επαφή

Εικόνα 4.22 Τα μέρη από τα οποία αποτελείται ένας λαμπτήρας πυράκτωσης. [Πηγή](#)

Ο λαμπτήρας πυράκτωσης είναι ο παλιότερος τύπος ηλεκτρικής πηγής φωτός, ο οποίος κατασκευάστηκε από τον Τόμας Έντισον το 1891. Η λειτουργία του λαμπτήρα βασίζεται στο φαινόμενο της θέρμανσης ενός λεπτού μεταλλικού νήματος που είναι συνήθως κατασκευασμένο από βολφράμιο και τυλιγμένο σε σπείρες. Όπου όταν το νήμα διαρέεται

από ρεύμα φωτοβολεί από τη θέρμανσή του. Η κατασκευή αυτή περικλείεται από μια γυάλινη σφαίρα χαμηλής πίεσης αερίου. [83]

#### 4.15 Ηλεκτρική αντλία 12 Volt



*Εικόνα 4.23 Ηλεκτρική αντλία 12 Volt*

Η ηλεκτρική αντλία 12 Volt είναι ένα μηχάνημα που μετατρέπει την ηλεκτρική ενέργεια σε υδραυλική ενέργεια και χρησιμοποιείται για την μετακίνηση ή την ανύψωση υγρών από μια περιοχή χαμηλότερου υψομέτρου σε μια περιοχή υψηλότερου υψομέτρου. Επίσης αποτελείται από έναν κινητήρα και από τον στρόβιλο. [84]

#### 4.16 Ηλεκτρική βαλβίδα

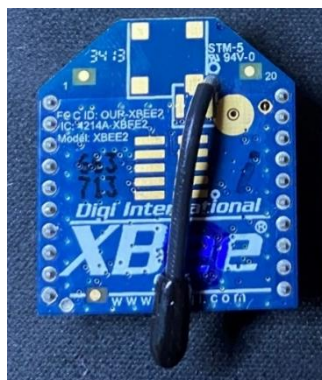


*Εικόνα 4.24 Ηλεκτρική βαλβίδα 12 Volt*

Η ηλεκτρική βαλβίδα είναι μια βαλβίδα που χρησιμοποιείται για τον απομακρυσμένο έλεγχο ρευστών που μπορεί να είναι υγρό ή αέριο. Οι βαλβίδες αποτελούνται από ένα πηνίο που στο εσωτερικό του έχει ένα έμβολο και το ελατήριο όταν εφαρμοστεί τάση, τότε στο πηνίο δημιουργείται ένα μαγνητικό πεδίο που έλκει το έμβολο με συνέπεια να επιτρέπεται η ροή του ρευστού. Τέλος υπάρχουν δύο είδη βαλβίδων, οι κανονικά ανοιχτοί (Normally Open) ή οι κανονικά κλειστοί (Normally Close), ανάλογα με τον τύπο του ελατηρίου το μαγνητικό πεδίο ασκούν αντίστροφες δυνάμεις

## 4.17 Xbee

### 4.17.1 Τι είναι το Xbee



Εικόνα 4.25 Το Xbee S2.

Τα Digi Xbee είναι κάρτες που μπορούν να επικοινωνούν ασύρματα μεταξύ τους σε εμβέλεια 40 μέτρων σε εσωτερικούς χώρους ή 90 μέτρων σε εξωτερικούς χώρους. Τα Xbee δεν διαθέτουν κανέναν μικροελεγκτή ή επεξεργαστή από μόνα τους, επομένως δεν μπορούν να διαχειριστούν τα δεδομένα που λαμβάνονται ή αποστέλλονται. Μπορούν απλώς να μεταφέρουν τις πληροφορίες που λαμβάνουν. Μπορούν να συνδεθούν με άλλους μικροελεγκτές και επεξεργαστές, όπως είναι το Arduino, το Raspberry Pi ή ένας υπολογιστής μέσω σειριακής διεπαφής. Έτσι, οι μονάδες Xbee έχουν δύο τύπους επικοινωνίας, την σειριακή και ασύρματη επικοινωνία. [85]

- a) Με σειριακή επικοινωνία (UART), οι μονάδες Xbee μπορούν να επικοινωνούν με μικροελεγκτές και επεξεργαστές. [85]
- b) Η ασύρματη επικοινωνία πραγματοποιείται μεταξύ 2 συσκευών Xbee τουλάχιστον μέσω ραδιοσυχνοτήτων (RF). [85]

Με αυτόν τον τρόπο, ο ελεγκτής μπορεί να παρακολουθεί και να ελέγχει απομακρυσμένες συσκευές μέσω των μονάδων Xbee. [85]

Γενικά, οι μονάδες Xbee παρέχουν ασύρματη συνδεσιμότητα τερματικού σημείου σε συσκευές και λειτουργούν εντός της ζώνης συχνοτήτων GSM 2,4 GHz. Αυτές οι μονάδες υποστηρίζουν το πρωτόκολλο IEEE 802.15.4 ή ZigBee mesh ή Digi Mesh ή ZigBee. Είναι σχεδιασμένα για εφαρμογές υψηλής απόδοσης που απαιτούν χαμηλό λανθάνοντα χρόνο και προβλέψιμο χρόνο επικοινωνίας. Οι μονάδες Xbee διαθέτουν δυνατότητα διεύθυνσης πηγής / προορισμού με υποστήριξη επικοινωνίας unicast και εκπομπής. Τα Xbee χρειάζονται τάση τροφοδοσίας 2,7 – 3,6V και διαθέτουν 20 ακίδες από τις οποίες οι 11 είναι ψηφιακές ακίδες I/O, οι 4 αναλογικές ακίδες εισόδου ADC (10-bit) ενώ έχει και ακίδες εξόδου PWM. Επίσης, διαθέτει σειριακές ακίδες UART για επικοινωνία με υπολογιστή και τον μικροελεγκτή. Ορισμένες μονάδες Xbee (π.χ. S2C) υποστηρίζουν επίσης τη διεπαφή SPI. [85]

Όλες οι ακίδες του Xbee αναφέρονται στον παρακάτω πίνακα.

Pin No.	Name	Direction	Description
1	VCC	-	3,3 Power Supply
2	DOUT	Output	UART Data out (TX)
3	DIN/CONFIG	Input	UART Data in (RX)
4	DIO12	Output	Digital I/O 12
5	RESET	Input	Module Reset (asserted low by bringing pin to ground-reset pulse $\geq 200\text{nS}$ )
6	PWM0/RSSI/DIO10	Output	PWM analog Output 0, Received Signal Strength Indicator, Digital I/O 10
7	PWM1/ DIO11	Output	PWM Output, Digital I/O 11
8	Reserved	-	Do not connect
9	DTR/SLEEP_RQ/DIO8	Input	Data terminal ready (Hardware handshkin signal), Pin Sleep Control (asserted low), Digital I/O 8
10	GND	-	Ground
11	AD4/DIO4	Input / Output	Analog Input 4/ Digital I/O 4
12	CTS/DIO7	Input / Output	Clear-To-Send Flow (hardwa handshaking) Control, Digital I/O 7
13	ON/SLEEP	Output	Module Status sleep Indicator (off when module is sleeping), High = ON, Low = SLEEP
14	VREF	Input	Reference Voltage for ADC (Not use in series 2)
15	ASSOC/AD5/DIO5	Input / Output	Associated Indicator: blinks if module is associated with a network, steady if not, Analog Input 5 or Digital I/O 5
16	RTS/AD6/DIO6	Input / Output	Request-To-Send Flow Control (hardware handshaking), Analog Input 6 or Digital I/O 6
17	AD3/DIO3	Input / Output	Analog Input 3 or Digital I/O 3
18	AD2/DIO2	Input / Output	Analog Input 2 or Digital I/O 2
19	AD1/DIO1	Input / Output	Analog Input 1 or Digital I/O 1
20	AD0/DIO0/commis	Input / Output	Analog Input 0 or Digital I/O 0, commissioning button

Πίνακας 4.2 Περιγραφή ακίδων Xbee. [Πηγή](#)

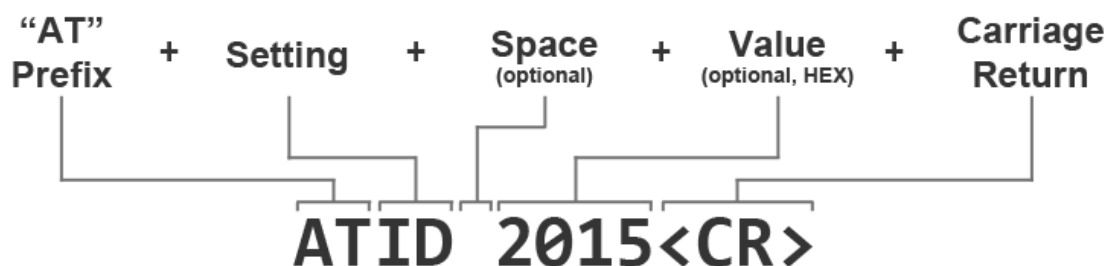


## 4.17.2 AT and API mode

Οι μονάδες Xbee επικοινωνούν μεταξύ τους με δύο τρόπους με διαφανή AT και λειτουργία API.

- Σε διαφανή λειτουργία AT Command (Transparent Mode) οι μονάδες Xbee λειτουργούν ως αντικατάσταση σειριακής γραμμής, δηλαδή από σημείο σε σημείο (peer to peer). Όλα τα δεδομένα που λαμβάνονται μέσω σειριακής εισόδου μεταδίδονται αμέσως μέσω της κεραίας στην απομακρυσμένη μονάδα Xbee που προσδιορίζεται από τη Διεύθυνση Προορισμού στη μνήμη. Δεν απαιτείται σχηματισμός πακέτων, αλλά απλή αποστολή σειριακών δεδομένων. Επίσης επειδή δεν δημιουργούνται πακέτα, η διεύθυνση προορισμού και ο τύπος είναι και τα δύο σταθερά. [86]

Τα AT commands είναι εντολές που χρησιμοποιούνται για τον έλεγχο ενός modem οι οποίες ξεκινούν με το AT. Τα AT Command έχουν σκοπό την διαμόρφωση των Xbee και συντάσσονται όπως φαίνεται στην [Εικόνα 4.39](#).



Εικόνα 4.26 Σύνταξη μιας εντολής AT.

- Ενώ η λειτουργία API είναι μία "Διεπαφή προγραμματισμού εφαρμογής", που τα δεδομένα μεταδίδονται σε καθορισμένο πλαίσιο με πληροφορίες προορισμού και ωφέλιμο φορτίο. Το πλαίσιο (Frame) ξεκινά με ένα Start Byte (0x7E), μετά το Frame Length, Frame Type, Frame Data και τελειώνει με το Checksum. Επίσης, υπάρχει η δυνατότητα επικοινωνίας με πολλούς κόμβους και η επιβεβαίωση παράδοσης πακέτου σε κάθε μεταδιδόμενο πακέτο. [29]



Παρακάτω παρουσιάζεται παράδειγμα πλαισίου σε πίνακα η δομή δεδομένων σε λειτουργία API από την αποστολή του πλαισίου Router Xbee στο Arduino:

BYTE	Frame HEX		Description
0	7E		Start BYTE
1	00	MSB	Frame Length (ο αριθμός μεταξύ του Length και του checksum που είναι 0x18=24)
2	18	LSB	
3	92		Frame Type
4	00		8 <sup>2</sup> =64-bit address of sender(Router). Set to 0xFFFFFFFFFFFFFFFF (unknown 64-bit address) if the sender 's 64-bit address is unknown.
5	13		
6	A2		
7	00		
8	40		
9	B5		
10	EB		
11	64		
12	23	MSB	
13	D5	LSB	
14	01		Receiver Option
15	01		Number of samples in payload. Always set to 1 due to XBEE limitations
16	04	MSB	Digital Channel Mask – Indicates which I/O pins are set to DIO
17	20	LSB	
18	0E		"Analog Channel Mask – Indicates which pins are set to ADC" ("Data Out Data in Xbee C Associate/DIO5 S2 R Xbee Roles E")
19	00	MSB	"There will be two bytes here for every pin set for ADC" ("Data Out Data in Xbee C Associate/DIO5 S2 R Xbee Roles E")
20	20	LSB	
21	03	MSB	Analog sample 1
22	D5	LSB	
23	00	MSB	Analog sample 2
24	98	LSB	
25	01	MSB	Analog sample 3
26	1F	LSB	
27	98		Checksum

Πίνακας 4.3 Δομή δεδομένων σε λειτουργία API. [Πηγή](#)

Παρακάτω στον [Πίνακα 4.4](#) εμφανίζονται τα υπόλοιπα Frame Type από το 3<sup>ο</sup> byte του [Πίνακα 4.3](#).

<b>0x08</b>	AT Command	Queries or sets parameters on the local XBee
<b>0x09</b>	AT Command Queue Parameter Value	Queries or sets parameters on the local XBee without applying changes
<b>0x10</b>	Transmit Request	Transmits wireless data to the specified destination
<b>0x11</b>	Explicit Addressing Command Frame	Allows Zigbee application layer fields (endpoint and cluster ID) to be specified for a wireless data transmission
<b>0x17</b>	Remote AT Command Request	Queries or sets parameters on the specified remote XBee module
<b>0x21</b>	Create Source Route	Creates a source route in the module
<b>0x24</b>	Register Joining Device	Registers a module with the Trust Center
<b>0x88</b>	AT Command Response	Displays the response to previous AT command frame
<b>0x8A</b>	Modem Status	Displays event notifications such as reset, association, disassociation, and so on.
<b>0x8B</b>	Transmit Status	Indicates wireless data transmission success or failure
<b>0x90</b>	Receive Packet	Sends wirelessly received data out the serial interface (AO = 0)
<b>0x91</b>	Explicit Rx Indicator	Sends wirelessly received data out the serial interface when explicit mode is enabled (AO 0)
<b>0x92</b>	IO Data Sample Rx Indicator	Sends wirelessly received IO data out the serial interface
<b>0x94</b>	XBee Sensor Read Indicator	Sends wirelessly received sensor sample (from a Digi 1-wire sensor adapter) out the serial interface
<b>0x95</b>	Node Identification Indicator	Displays received node identification message when explicit mode is disabled (AO = 0)
<b>0x97</b>	Remote AT Command Response	Displays the response to previous remote AT command requests
<b>0x98</b>	Extended Modem Status	Displays what is happening during the association when Verbose Join is enabled (DC10)
<b>0xA0</b>	Over-the-Air Firmware Update Status	Provides a status indication of a firmware update transmission attempt
<b>0xA1</b>	Router Record Indicator	Displays the multiple route hopes after a Zigbee route record command
<b>0xA3</b>	Many-to-One Route Request Indicator	Indicates a many-to-one route request is received
<b>0xA5</b>	Join Notification Status	Indicates a module attempts to join, rejoin, or leave the network

*Πίνακας 4.4 Οι υπόλοιποι τύποι πλαισίου λειτουργίας API (Byte 3). [Πηγή](#)*

Επίσης στο Πίνακα 5 εμφανίζονται τα υπόλοιπα Receiver Option από το δέκατο τέταρτο Byte στο Πίνακα 3.

0x01	Packet acknowledged.
0x02	Packet was a broadcast packet.
0x20	Packet encrypted with APS encryption.
0x40	Packet was sent from an end device (if known).

Πίνακας 4.5 Οι υπόλοιποι τύποι Receiver Option από το 14 Byte. [Πηγή](#)

Τέλος, στον [Πίνακα 4.6](#) επεξηγείτε η μάσκα για την αναλογική είσοδο/έξοδο που είναι ενεργοποιημένη στο απομακρυσμένο Xbee (Router) από τον [Πίνακα 4.3](#) το 18ο Byte του.

Στο Frame του [Πίνακα 4.3](#) το 18ο Byte είναι το E σε Δεκαεξαδικό σύστημα, εάν μετατραπεί σε δυαδική μορφή είναι ίσο με 1110 άρα συμπληρώνοντας τον [Πίνακα 4.6](#) διαπιστώνουμε ότι είναι ενεργοποιημένα το D1, D2, και το D3.

BYTE 16-17	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	X	X	X	D12	D11	D10	X	X	D7	D6	D5	D4	D3	D2	D1	D0

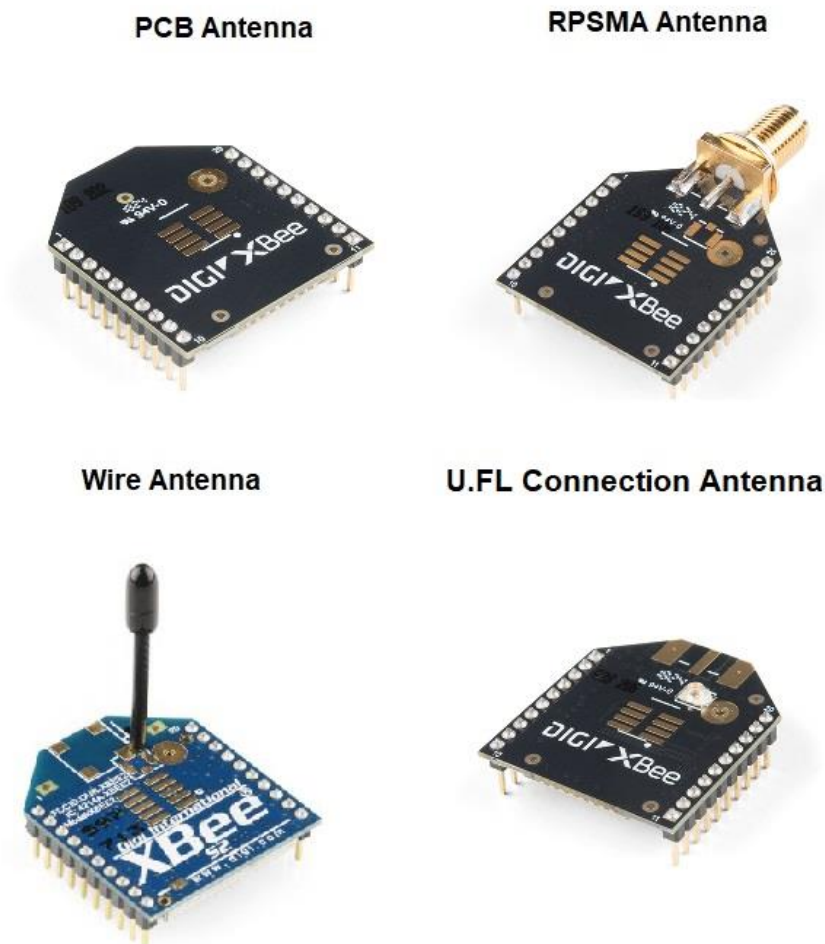
Πίνακας 4.6 Η περίπτωση που έχει ενεργοποιηθεί το D3. [Πηγή](#)

#### 4.17.3 Τύποι Xbee

Η Digi έχει διαθέσιμους διαφορετικούς τύπους μονάδων Xbee που το καθένα έχει διαφορετικές προδιαγραφές. Επισημαίνεται ότι παρόλο που όλες οι μονάδες ανεξάρτητα από τη σειρά ή τον τύπο έχουν 20 ακίδες που είναι ίδιες σε τοπολογία και λειτουργία σε όλες της σειρές, ενώ δεν υπάρχει η δυνατότητα να επικοινωνήσουν μεταξύ τους, εάν είναι άλλη σειρά ή άλλος τύπος. Μερικοί από τους πιο δημοφιλείς τύπους Xbee αναφέρονται παρακάτω. [87]

- Τα Xbee Series 1 επικοινωνούν με το πρωτόκολλο 802.15.4, έτσι είναι τα πιο εύκολα στη χρήση, γιατί δεν χρειάζεται να ρυθμιστούν. [87]
- Τα Xbee series 2 επικοινωνούν με το πρωτόκολλο ZigBee Mesh, έτσι επιτρέπουν να δημιουργούνται πολύπλοκα δίκτυα πλέγματος. Επίσης, μπορούν να επικοινωνούν σε AT ή API. [87]

- Τα Xbee series 2B επικοινωνούν με το πρωτόκολλο ZigBee Mesh, αλλά έχουν βελτιωμένο υλικό σε σχέση με τα Xbee S2 βελτιώνοντας πράγματα όπως η κατανάλωση ενέργειας. [87]
- Τα Xbee series 3 επικοινωνούν με το πρωτόκολλο 802.15.4 ή ZigBee ή DigiMesh. Επίσης, δίνουν την δυνατότητα επικοινωνίας μέσω UART ή SPI. [87]



Εικόνα 4.27 Διαφορετικοί διαθέσιμοι τύποι κεραιών για τα Xbee. [Πηγή](#)

Η Digi προσφέρει τις συσκευές Xbee με διαφορετικές κεραιές, με τα πλεονεκτήματα και μειονεκτήματά της ανάλογα με την υλοποίηση. Οι κεραιές που είναι διαθέσιμες είναι:

- Η Chip Antenna είναι ένα μικρό επίπεδο κεραμικό τσιπ που είναι στο ίδιο επίπεδο με το σώμα του Xbee. Οι κεραιές τσιπ έχουν καρδιοειδές μοτίβο ακτινοβολίας, που σημαίνει ότι το σήμα εξασθενεί σε πολλές κατευθύνσεις. Είναι ανθεκτικό, γρήγορο, εύκολο και φθινό. [87]
- Η Καλωδιακή (wire) κεραία είναι ένα μικρό καλώδιο που είναι κολλημένο στο σώμα του συσκευής Xbee. Είναι απλό και προσφέρει πανκατευθυντική ακτινοβολία,

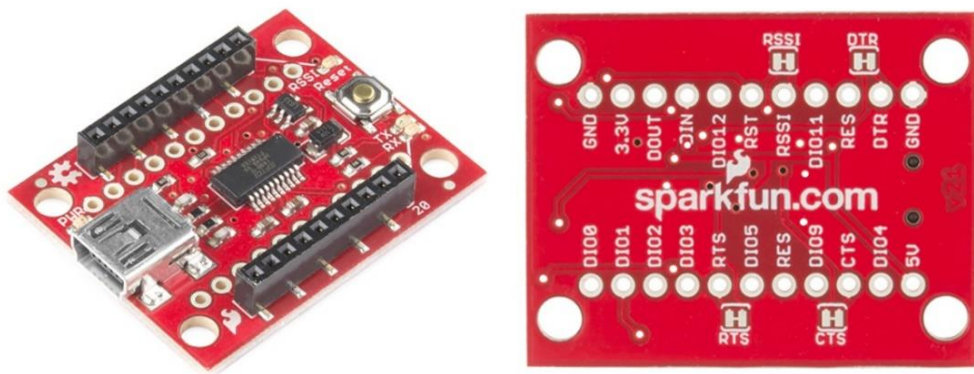
δηλαδή η μέγιστη απόσταση μετάδοσης είναι σχεδόν ίδια προς όλες τις κατευθύνσεις, όταν το σύρμα του είναι ίσιο και κάθετο στη μονάδα. [87]

- Η Κεραία u.FL που είναι ένας μικροσκοπικός σύνδεσμος που βγαίνει από μια καλωδιακή σύνδεση για τη σύνδεση της δικής μας κεραίας, αυτό εξυπηρετεί όταν θέλουμε να βάλουμε την κεραία έξω από το κουτί ώστε να μην περιορίζεται το σήμα από αυτό. [87]
- Η Κεραία RPSMA που είναι μια μεγαλύτερη υποδοχή σύνδεσης από την υποδοχή U.FL για τη σύνδεση της δικής μας κεραίας χωρίς καλώδιο αλλά απευθείας στο Xbee, για άλλη μια φορά εξαιρετική, αν το αντικείμενο σας βρίσκεται σε κουτί και θέλετε την κεραία σας έξω από το κουτί. [87]
- Η Trace Antenna που ονομάζεται επίσης κεραία PCB, αυτές είναι ενσωματωμένες απευθείας στη πλακέτα κυκλώματος του Xbee. Η κεραία PCB προσφέρει πολλά από τα ίδια πλεονεκτήματα (και μειονεκτήματα) όπως η κεραία τσιπ με πολύ χαμηλότερο κόστος κατασκευής και αποδίδουν περίπου το ίδιο με τις συρμάτινες κεραίες. [87]

Επίσης, όλοι οι τύποι Xbee είναι διαθέσιμοι σε εκδόσεις Regular και Pro. Τα Pro είναι λίγο μεγαλύτερα, χρησιμοποιούν περισσότερη ισχύ και κοστίζουν περισσότερα χρήματα. Η μεγαλύτερη ισχύς σημαίνει μεγαλύτερη εμβέλεια, δηλαδή από τα 1,2 χιλιόμετρα στα 3,2 χιλιόμετρα για τα Xbee S3 Pro και Regular. [87]

#### 4.17.4 Επικοινωνία Xbee με τον υπολογιστή και Arduino.

Η επικοινωνία των Xbee με το Arduino, αλλά και με τον υπολογιστή γίνεται μέσω του πρωτοκόλλου UART. Χρειάζεται να συνδεθούν τα Xbee με τον υπολογιστή για τον προγραμματισμό τους. Η Digi International προσφέρει ένα λογισμικό, το X-CTU, για τη διαμόρφωση του Xbee. Για να επιτευχθεί η επικοινωνία, χρησιμοποιούμε το Spark Fun Xbee Explorer USB. [88]



Το Spark Fun Xbee Explorer USB περιλαμβάνει:

- ένα τσιπ FTDI, το οποίο επιτρέπει στον χρήστη διασύνδεση με ένα καλώδιο mini USB, ώστε να γίνει η σειριακή επικοινωνία (UART) με τον υπολογιστή. Στην εργασία χρησιμοποιήθηκαν 2 SparkFun Xbee Explorer USB, ώστε να υπάρχει η δυνατότητα ταυτόχρονης διαμόρφωσης των δυο συσκευών Xbee στον υπολογιστή σε διαφορετικές σειριακές θύρες. [88]
- επίσης έρχεται με μεταφραστή λογικού επιπέδου (TTL) που μετατρέπει τα 5V του Arduino σε 3.3 V που χρειάζονται για την τροφοδοσία του Xbee. [88]
- Τέλος, οι ακίδες των Xbee δεν είναι φιλικές προς τα Breadboard, έτσι με την χρήση του Spark Fun Explorer δίνετε η δυνατότητα τα Xbee να συνδεθούν σε Breadboard. [88]

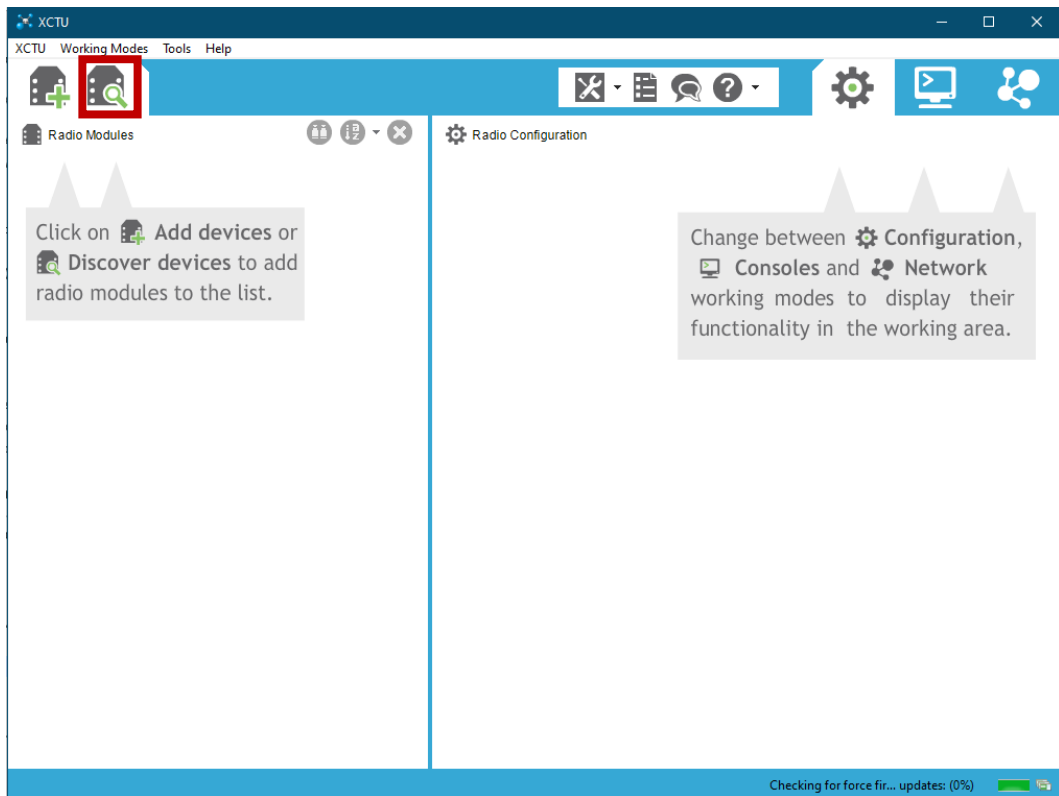
#### 4.17.5 Διαμόρφωση Xbee μέσω XCTU

##### 4.17.5.1 Πρόλογος

Παρακάτω θα γίνει διαμόρφωση δύο συσκευών Xbee για να επικοινωνούν μεταξύ τους μέσω του λογισμικού XCTU. Για να γίνει η επικοινωνία μέσω δύο συσκευών Xbee απαιτείται μια συσκευή Xbee να διαμορφωθεί ως Coordinator και η άλλη συσκευή Xbee ως Router. Επίσης η επικοινωνία μεταξύ τους θα είναι σε API mode, ώστε να ανταλλάσσουν μεταξύ τους πακέτα.

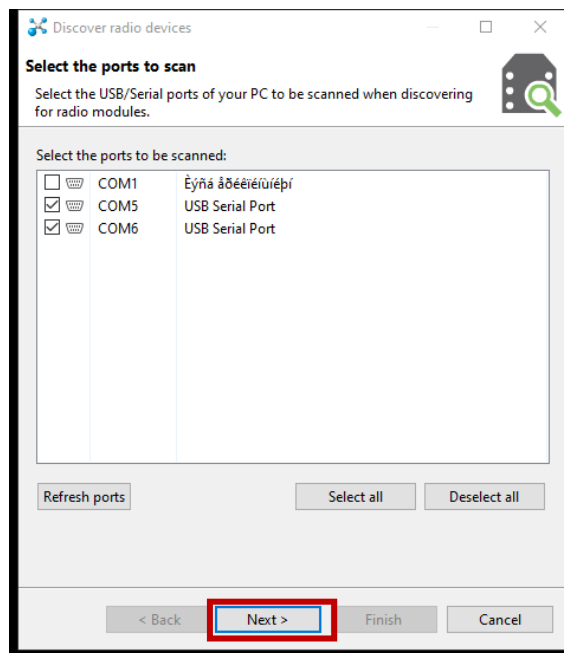
##### 4.17.5.2 Σύνδεση XCTU με Xbee

Αρχικά θα πρέπει τα Xbee να αναγνωριστούν από το XCTU συνδέοντας το Xbee πάνω στο Xbee Explorer και το Xbee Explorer μέσω καλωδίου με το υπολογιστή. Τότε μπορούμε να ανοίξουμε το XCTU που έχουμε κατεβάσει είδη από την ιστοσελίδα της DIGI. Έπειτα πατώντας το κουμπί (που δείχνεται μέσα στο κόκκινο πλαίσιο στην [Εικόνα 4.29](#)) κάνουμε αναζήτηση συνδεδεμένων Xbee.

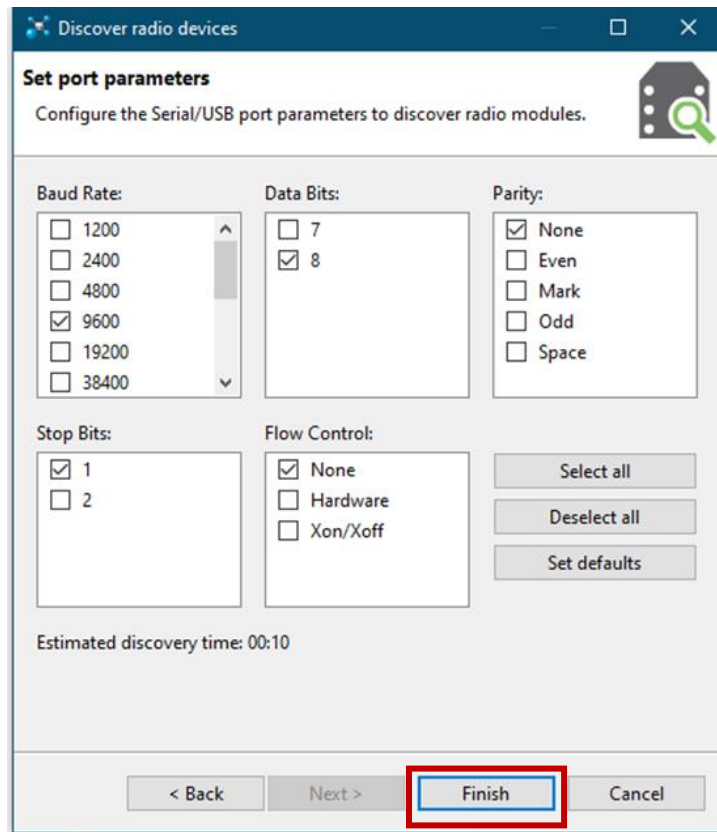


Εικόνα 4.29 Το λογισμικό XCTU

Υστέρα εμφανίζεται ένα παράθυρο [Εικόνα 4.30](#) ώστε να επιλέξουμε σε ποια θύρα COM έχουμε συνδέσει τα Xbee. Έπειτα πατώντας Next εμφανίζεται το παράθυρο [Εικόνα 4.31](#) όπου εκεί ορίζουμε της παραμέτρους της θύρας η οποίες είναι: Baud Rate:9600, Stop Bits:1, Data Bits:8, Flow Control: None και Parity: None όπως φαίνονται στην εικόνα.

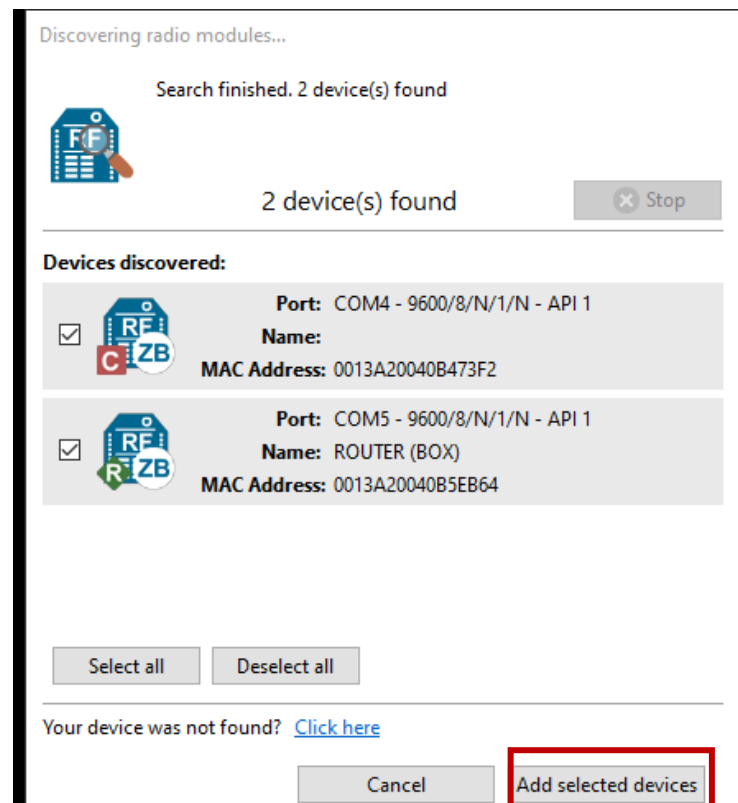


Εικόνα 4.30 Οι θύρες που είναι συνδεδεμένες οι συσκευές



Εικόνα 4.31 Η παράμετροι που έχουμε ορίσει για της θύρες

Έπειτα από την αναζήτηση εμφανίζονται οι συσκευές που βρέθηκαν.



Εικόνα 4.32 Η δύο συσκευές που είχαμε συνδέσει ο coordinator και το router



Όταν πατήσουμε το Add selected devices, εμφανίζεται η πρώτη μορφή εργασίας (Configuration mode) ([Εικόνα 4.36](#)) όπου μπορούμε να τα διαμορφώσουμε τις συσκευές ανάλογα με την υλοποίηση.

#### 4.17.5.3 Ενημέρωση του Firmware των Xbee

Πριν την διαμόρφωση των Xbee κάνουμε πρώτα Update στο Firmware. Για να κάνουμε το Firmware update απαιτείται η γνώση του Product Family (XB24-Z&) το οποίο αναφέρεται στο πίσω μέρος του Xbee ([Εικόνα 4.33](#)).

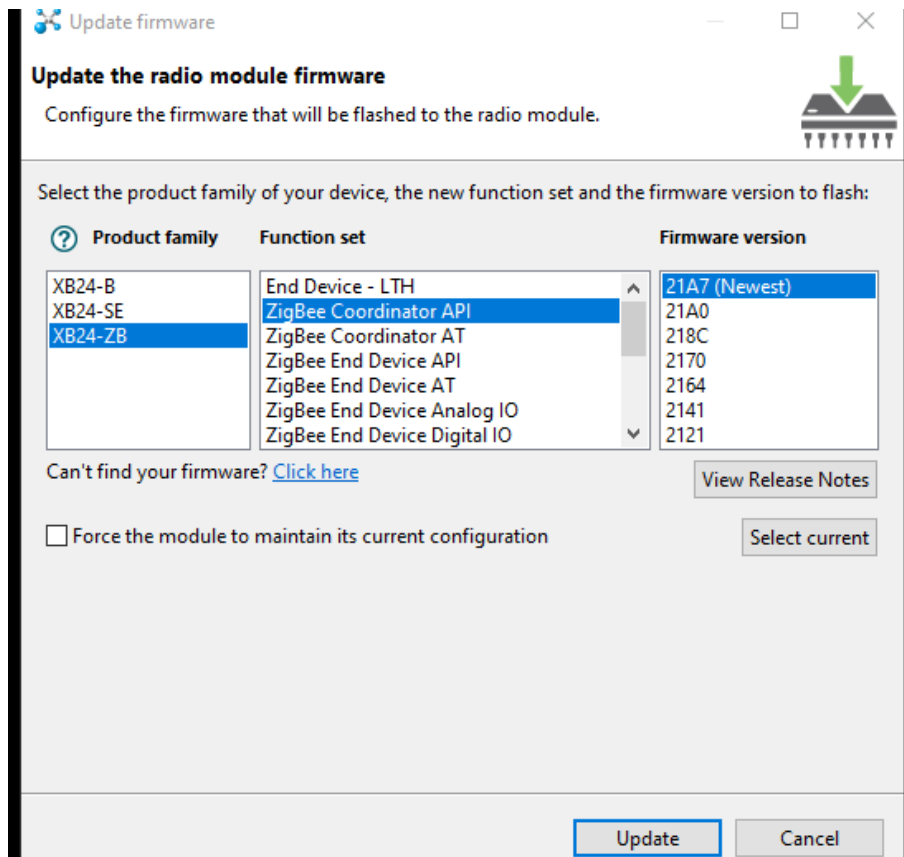


Εικόνα 4.33 Το πίσω μέρος του Xbee που φαίνεται κυκλωμένο το Product Family

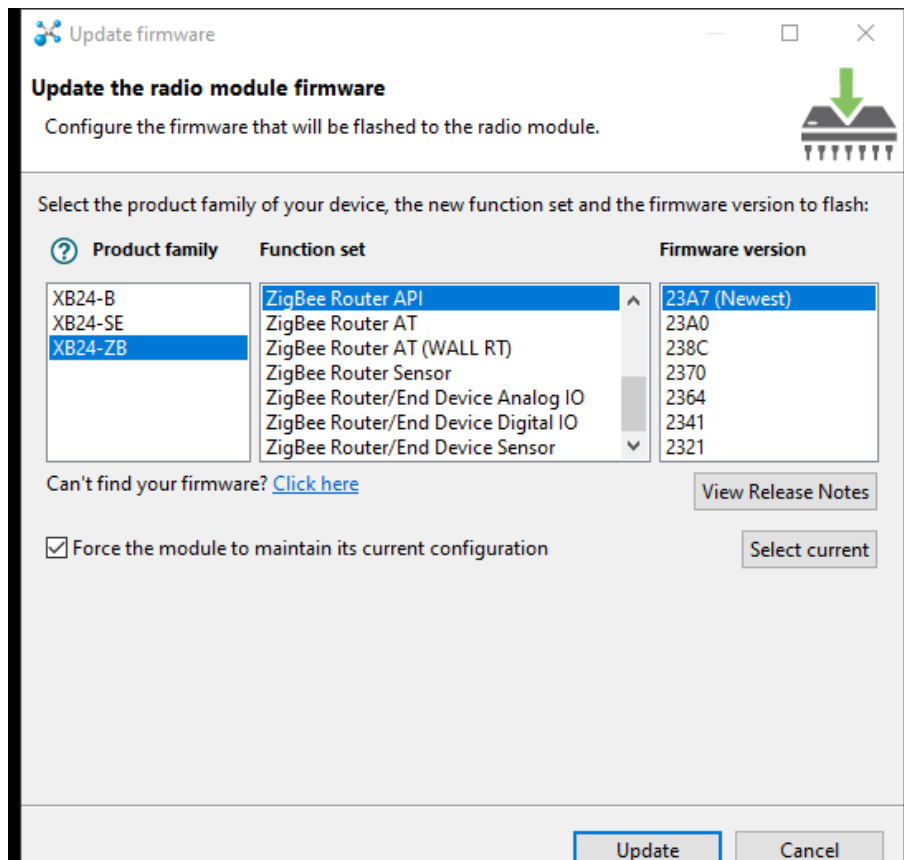
Έχοντας γνώση του Product Family και επειδή θα χρησιμοποιηθεί το Coordinator και το Router σε API mode, επιλέγονται οι παρακάτω ρυθμίσεις([Εικόνα 4.34](#)):

- A. Product Family: XB24-zb,
- B. Function Set: Zigbee Coordinator API,
- C. Firmware Version: 21A7(Newest)

και γίνεται το update.



Εικόνα 4.34 Το Firmware που απαιτείται για το Coordinator.

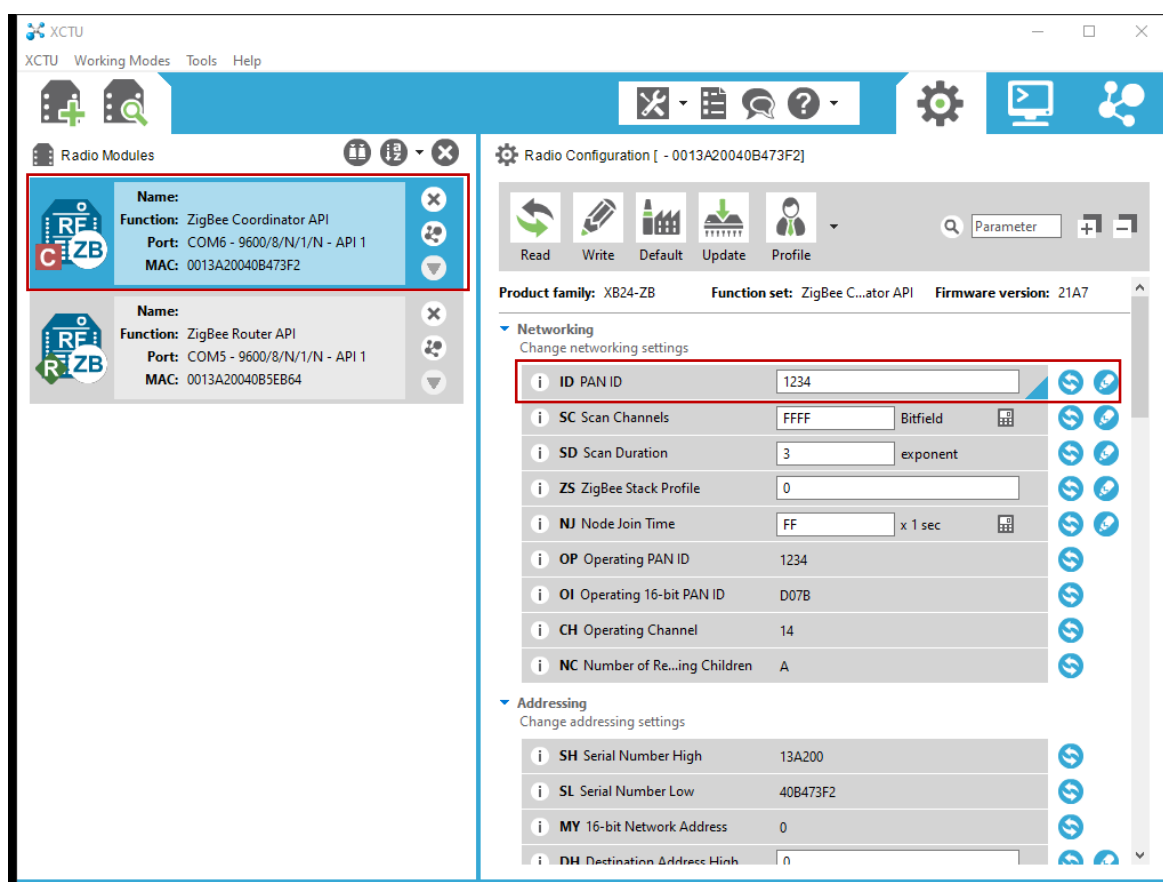


Εικόνα 4.35 Το Firmware που απαιτείται για το Router.

Στην συνέχεια του update θα πρέπει να γίνουν κάποιες ρυθμίσεις, οι οποίες αναφέρονται παρακάτω.

#### 4.17.5.4 Βασικές ρυθμίσεις Xbee

Η ρυθμίσει στο Configuration mode που πρέπει να γίνει στο Xbee Coordinator ([Εικόνα 4.36](#)) είναι να οριστεί το ID Pan ID, (Το Pan ID είναι ένα μοναδικό αναγνωριστικό που ορίζει κάθε δίκτυο και πρέπει να είναι ίδιο για κάθε συσκευή Xbee που χρειάζεται να επικοινωνήσουν μεταξύ τους).



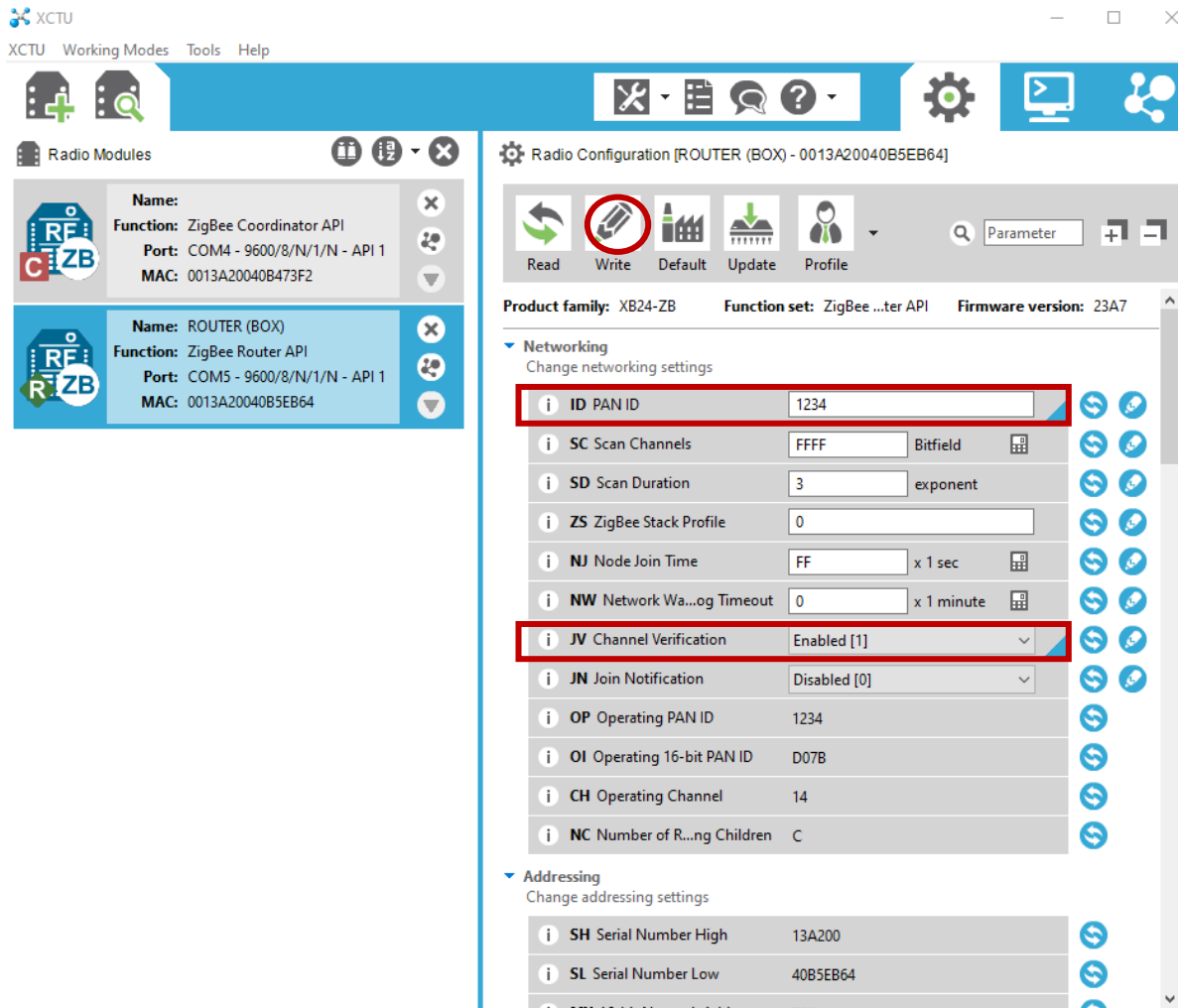
Εικόνα 4.36 Η ρυθμίσεις που έχουμε ορίσει για των Coordinator στην πρώτη μορφή εργασίας.

Οι ρυθμίσεις στο Configuration mode που απαιτούνται να γίνονται για την λειτουργία του Xbee ως Router ([Εικόνα 4.37](#), [Εικόνα 4.38](#), και [Εικόνα 4.39](#)) είναι:

- να οριστεί το ίδιο ID Pan ID με τον Coordinator (Ένα μοναδικό αναγνωριστικό)
- το JV Channel Certification σε Enabled [1] (επαληθεύει ότι έχει συνδεθεί με των συντονιστή (Cordinator))
- NI Node Identifier σε «ROUTER (BOX)»(ορίζουμε το όνομα που θέλουμε να εμφανίζεται στο Xbee, μπορεί να παραληφθεί),
- το IR IO Sampling Rate σε 1388 (είναι ο ρυθμός δειγματοληψίας των I/O)

Επίσης για να μπορεί το Xbee Router να διαβάζει αναλογικές τιμές από τους αισθητήρες που έχουν συνδεθεί στις ακίδες D1, D2, D3 απαιτείται να ορίσουμε της παρακάτω ρύθμισης (Εικόνα 4.39):

- D1 AD1/DIO1 Configuration σε ADC [2] (το pin διαβάζει αναλογικές τιμές)
- D2 AD2/DIO2 Configuration σε ADC [2] (το pin διαβάζει αναλογικές τιμές)
- D3 AD1/DIO3 Configuration σε ADC [2] (το pin διαβάζει αναλογικές τιμές)



Εικόνα 4.37 Η ρυθμίσεις για το Xbee Router το Pan ID και το JV.

▼ Addressing  
Change addressing settings

i	SH Serial Number High	13A200	
i	SL Serial Number Low	40B5EB64	
i	MY 16-bit Network Address	FFFE	
i	DH Destination Address High	0	
i	DL Destination Address Low	0	
i	NI Node Identifier	ROUTER (BOX)	
i	NH Maximum Hops	1E	
i	BH Broadcast Radius	0	
i	AR Many-to-One Route Broadcast Time	FF	x 10 sec
i	DD Device Type Identifier	30000	
i	NT Node Discovery Backoff	3C	x 100 ms
i	NO Node Discovery Options	0	
i	NP Maximum Number of API Transmission Bytes	FF	
i	CR PAN Conflict Threshold	3	

Εικόνα 4.38 Η ρυθμίσει NI Node Identifier.

▼ I/O Settings  
Modify DIO and ADC options

i	D0 AD0/DIO0 Configuration	Commissioning Button [1]	
i	D1 AD1/DIO1 Configuration	ADC [2]	
i	D2 AD2/DIO2 Configuration	ADC [2]	
i	D3 AD3/DIO3 Configuration	ADC [2]	
i	D4 DIO4 Configuration	Disabled [0]	
i	D5 DIO5/Assoc Configuration	Associated indicator [1]	
i	P0 DIO10 Configuration	RSSI PWM Output [1]	
i	P1 DIO11 Configuration	Disabled [0]	
i	P2 DIO12 Configuration	Disabled [0]	
i	PR Pull-up Resistor Enable	1FFF	
i	LT Associate LED Blink Time	0	x 10 ms
i	RP RSSI PWM Timer	28	x 100 ms
i	DO Device Options	1	Bitfield

▼ I/O Sampling  
Configure IO sampling parameters

i	IR IO Sampling Rate	1388	x 1 ms
i	IC Digital IO Change Detection	0	
i	V+ Supply Voltage High Threshold	0	

**Time calculator**

Time (ms):

Hex. value:

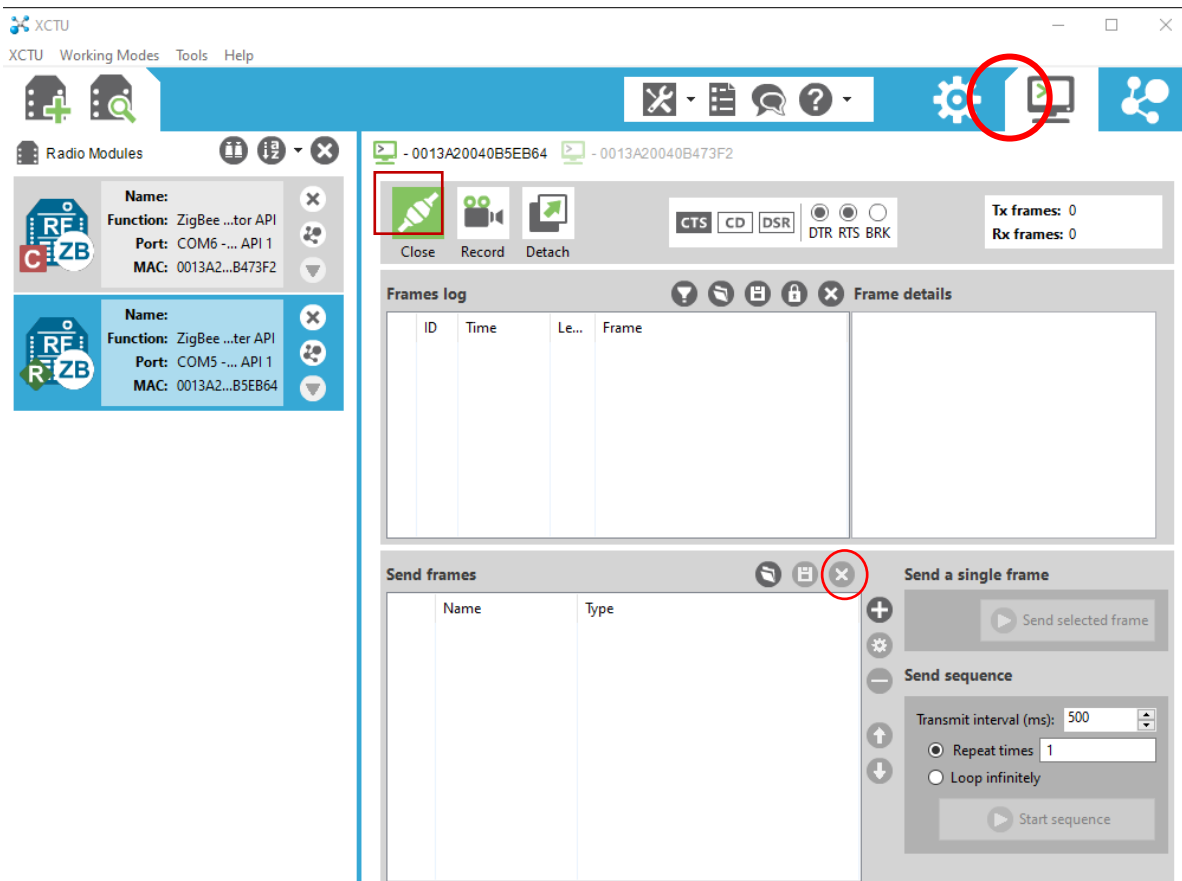
Εικόνα 4.39 Το IR που έχει οριστεί ως 1388.

Τέλος, πατώντας το write που φαίνεται στην [Εικόνα 4.36](#) κυκλωμένο οι ρυθμίσεις που κάναμε εγγράφονται στο Xbee.

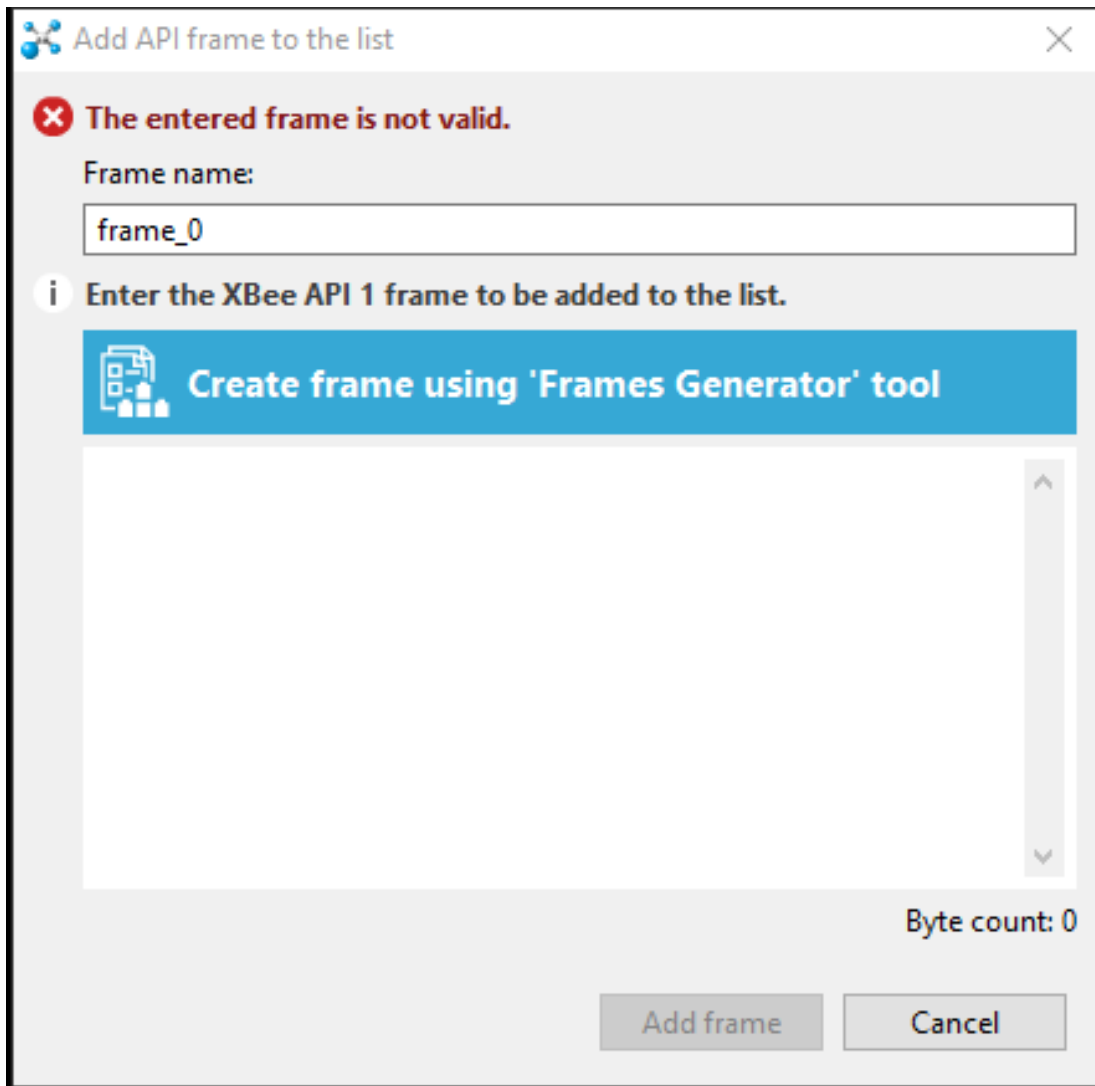
#### 4.17.5.5 Δημιουργία πλαισίων μέσω του Frame generator

Αφού έγιναν οι απαραίτητες ρυθμίσεις πατώντας αυτό που έχει κυκλωθεί στην [Εικόνα 4.39](#) μεταφερόμαστε στην δεύτερη μορφή εργασίας Console mode που επιτρέπει την δημιουργία πλαισίων και των έλεγχο της επικοινωνίας με της συσκευές Xbee. Τα πλαίσια είναι παρόμοια με αυτά που θα στέλνονται στο Arduino.

Πατώντας το κυκλωμένο σύν που εμφανίζεται κυκλωμένο στην [Εικόνα 4.39](#) και πατώντας το Create frame using 'Frames Generator' tool ([Εικόνα 4.39](#)) μπορούμε να δημιουργήσουμε ένα πλαίσιο για αποστολής ([Εικόνα 4.40](#)).



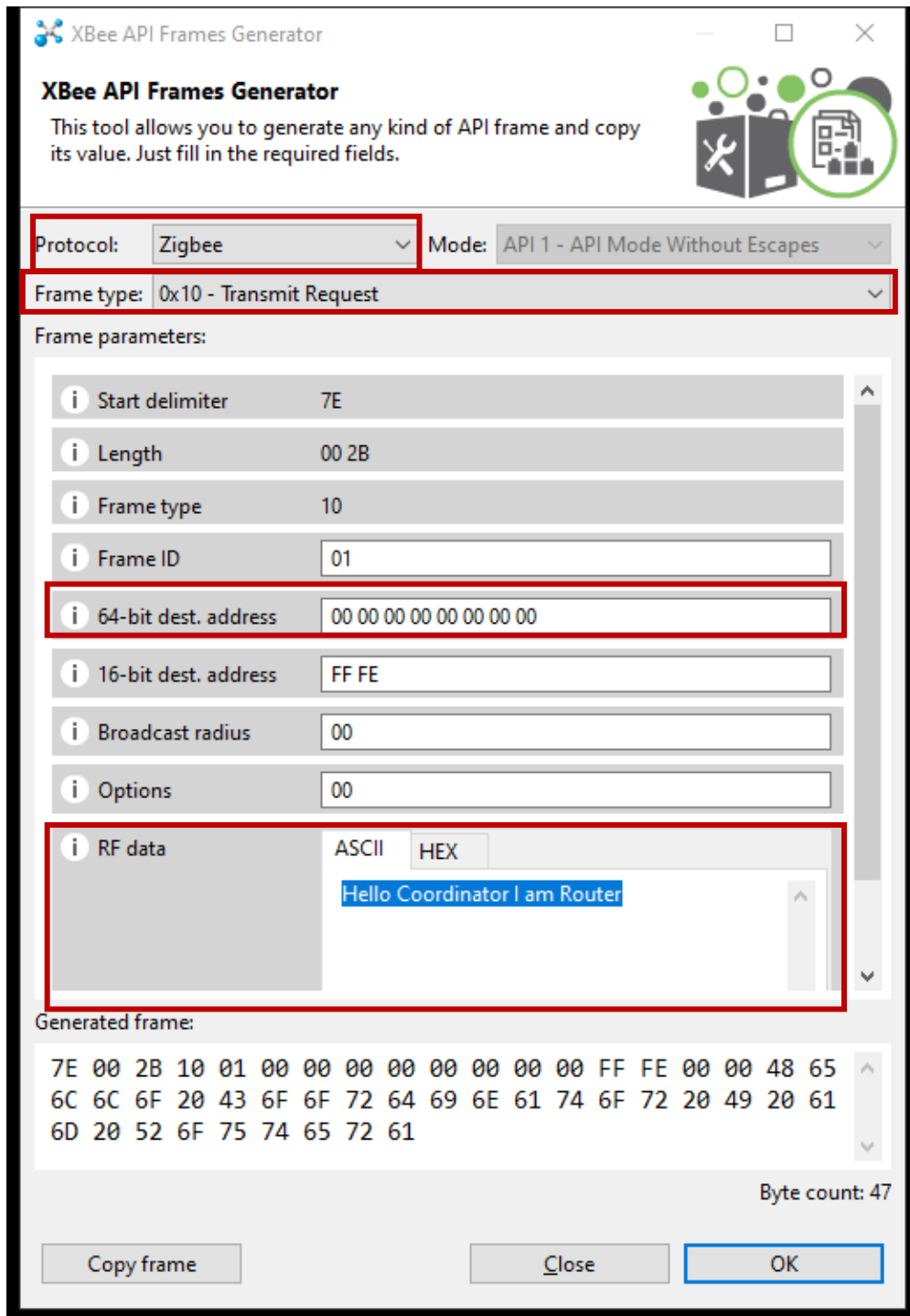
Εικόνα 4.40 Εδώ φαίνεται η κονσόλα.



Εικόνα 4.41 Το εργαλείο στο οποίο δημιουργούμε τα Frame.

Ανάλογα το Frame type που επιλέγεται υπάρχει η δυνατότητα δημιουργίας διαφορετικών πλαισίων. Στον [Πίνακας 4.4](#) αναφέρονται όλα τα Frame Type που μπορούν να χρησιμοποιηθούν στο πρωτόκολλο ZigBee. Παρακάτω ακολουθούν μερικά παραδείγματα από τρεις διαφορετικούς τύπους πλαισίων.

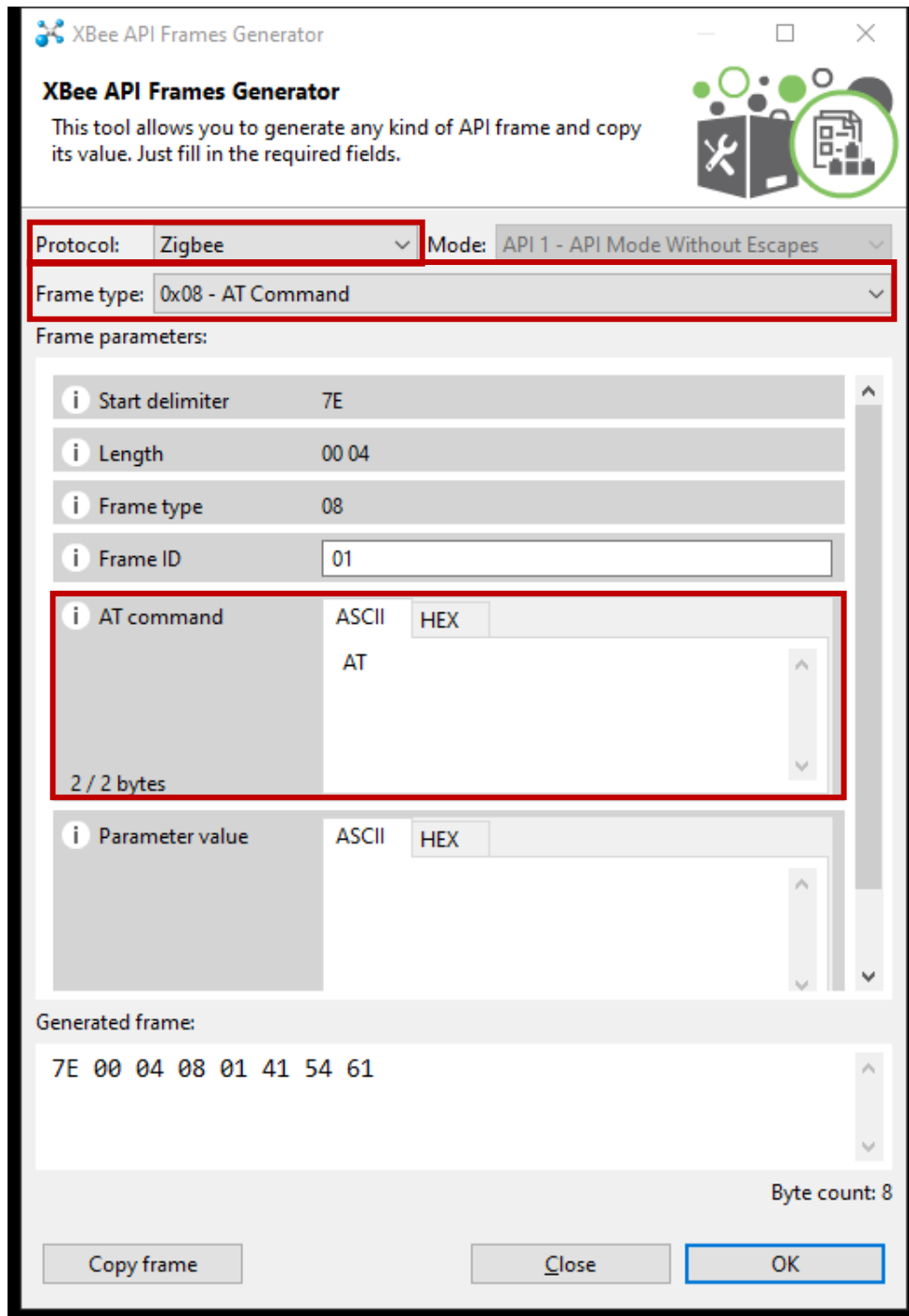
Ο πρώτος τύπος πλαισίων που θα αναφερθούμε δημιουργείται επιλέγοντας Protocol: Zigbee, Frame type: 0x10 Transmit Request και της παραμέτρους 64-bit dest. Address: 00 00 00 00 00 00 00 00 (64bit Address of Coordinator), RF data: Hello Coordinator I am Router. Συνεπώς γίνεται η αποστολή του κειμένου ASCII από το Xbee Router που είναι το επιλεγμένο στο Xbee Coordinator θα αποσταλθεί το κείμενο πατώντας το OK (Εικόνα 4.41).



Εικόνα 4.42 Το Frame type 0x10- Transmit Request.

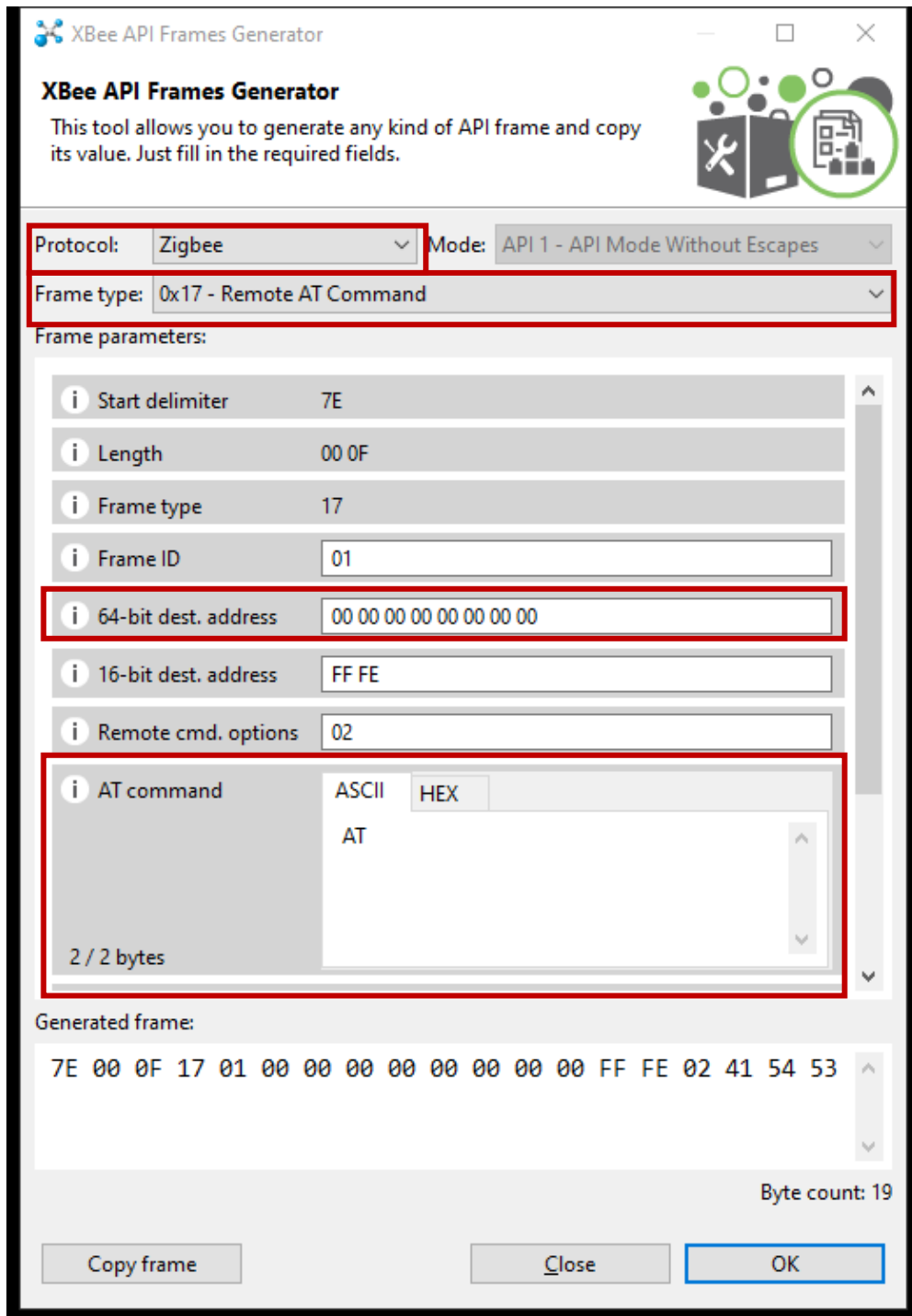
Επίσης άλλος ένας τύπος πλαισίων δημιουργείται επιλέγοντας το Protocol: Zigbee, Frame type: 0x08 AT-Command και γράφοντας στην παράμετρο AT command την εντολή που AT ώστε να αρχίσει να γίνεται διαμόρφωση του επιλεγμένου Xbee. Ακόμα μέσα στην παράμετρο μπορεί να γραφτεί μόνο μία εντολή για να δώσουμε και άλλη εντολή πρέπει να δημιουργήσουμε ξεχωριστό πλαίσιο. Πατώντας το OK γίνει η διαμόρφωση του Xbee (Εικόνα 4.42).





Εικόνα 4.43 To Frame type 0x08 AT-Command.

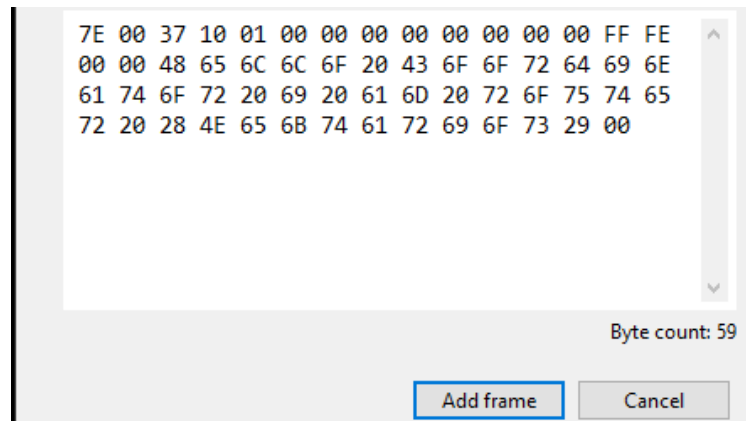
Ο τελευταίος τύπος πλαισίων που θα αναφερθούμε δημιουργείται επιλέγοντας το Protocol: Zigbee, Frame type: 0x10 AT-Command και γράφοντας στην παράμετρο 64-bit dest. Address: 00 00 00 00 00 00 00 00 (64bit Address of Coordinator), την παράμετρο AT command την εντολή που AT ώστε να αρχίσει να γίνεται διαμόρφωση του Coordinator Xbee. Επίσης μέσα στην παράμετρο μπορεί να γραφτεί μόνο μία εντολή για να δώσουμε και άλλη εντολή πρέπει να δημιουργήσουμε ξεχωριστό πλαίσιο. Πατώντας το OK θα αποσταλεί η εντολή στο Coordinator (Εικόνα 4.43).



Εικόνα 4.44 Το Frame type 0x10 AT-Command.

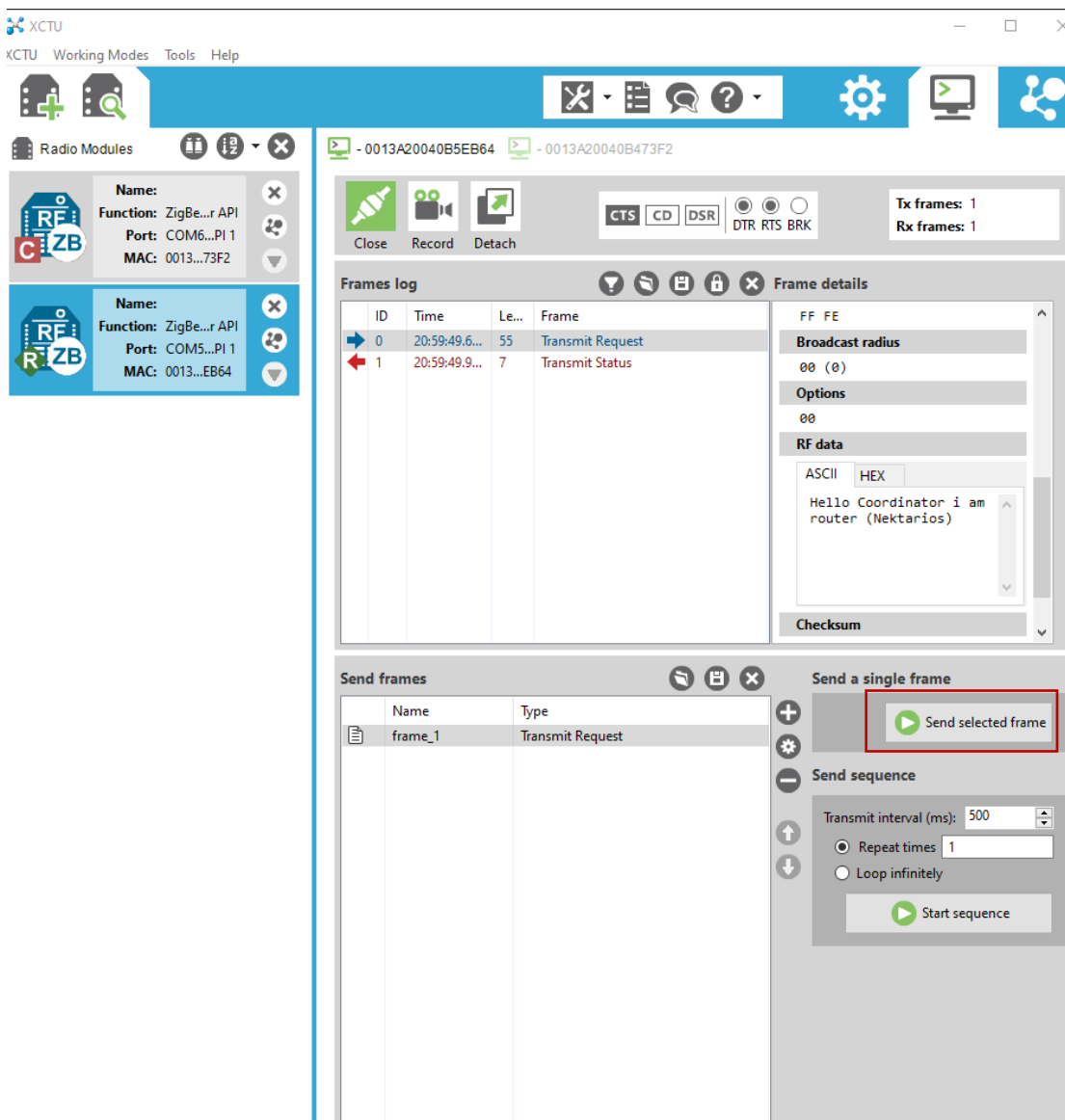
#### 4.17.5.6 Αποστολή πλαισίου του πρώτου παραδείγματος με τύπο 0x10

Η αποστολή ενός πλαισίου γίνεται με τον ίδιο τρόπο με αυτόν που περιγράφεται ανεξάρτητα από το Frame type. Συνεχίζεται ο πρώτος τύπος πλαισίου 0x10 Transmit Request που θα παρουσιάσει η αποστολή αυτού του πλαισίου. Από την στιγμή που πατήσαμε το OK εμφανίζεται η [Εικόνα 4.42](#) όπου φαίνεται σε Δεκαεξαδικό σύστημα το πλαίσιο που θα αποσταλεί στο Coordinator. Υστερα επιλέγεται το add frame.

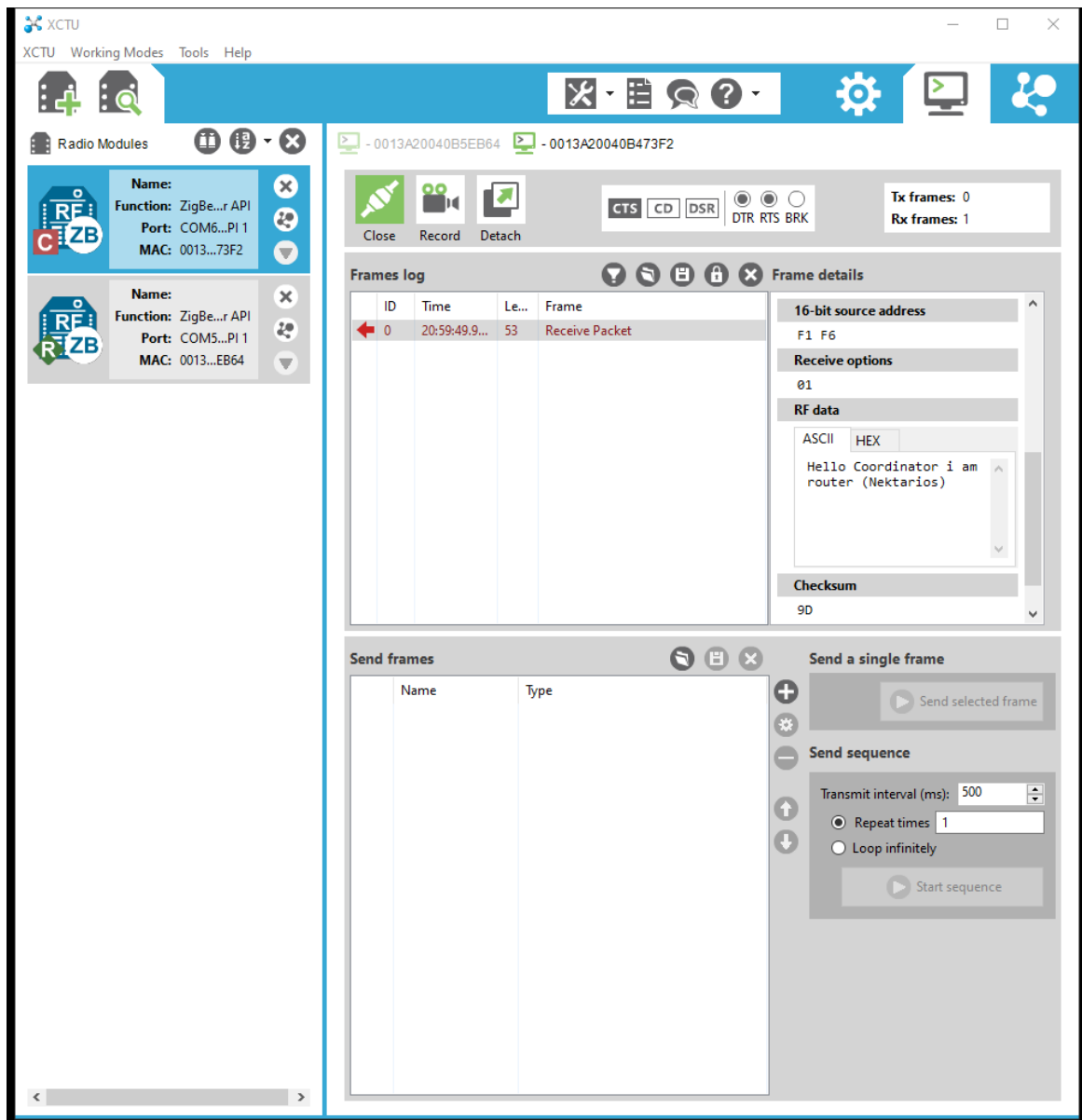


Εικόνα 4.45 Το frame που θα σταλθεί σε δεκαεξαδικό σύστημα.

Πατώντας το Send Selected frame αποστέλλεται το πλαίσιο από το Router στο Coordinator, όπως φαίνεται παρακάτω (Εικόνα 4.41 και Εικόνα 4.42).

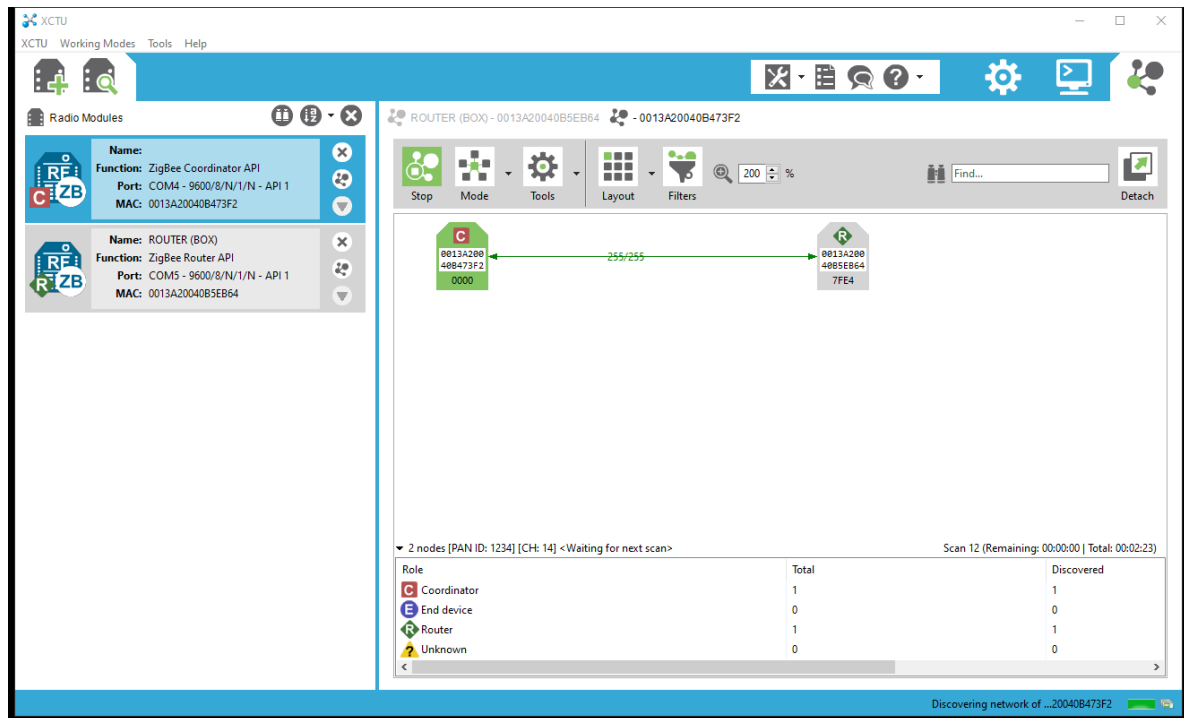


Εικόνα 4.46 Εδώ φαίνεται με μπλε χρώμα το πλαίσιο που στείλαμε.



Εικόνα 4.47 Εδώ εμφανίζεται το πλαίσιο που έλαβε ο Coordinator.

Τέλος, στην τρίτη μορφή εργασίας Network mode που είναι για την οπτικοποίηση της τοπολογίας και της διασύνδεσης του δικτύου που έχει δημιουργηθεί (Εικόνα 4.47) φαίνεται η σύνδεση των μονάδων Xbee και μπορεί να μετρηθεί και η απόσταση επικοινωνίας τους.



Εικόνα 4.48 Απεικόνιση δικτύου Zigbee μέσω του τρίτου μέρους εργασίας XCTU.

#### 4.17.6 Οι διαμόρφωση του Xbee Router μέσω του Arduino

Η διαμόρφωση των Xbee μπορεί να γίνει μέσω των AT Commands αρκεί να δημιουργηθεί ένα πλαίσιο με την κατάλληλη δομή δεδομένων που περιγράφεται στο [Κεφάλαιο 4.17.2](#). Επίσης χρειάζεται να έχει γίνει ενημέρωση Firmware στα Xbee ([Κεφάλαιο 4.17.5.3](#)). Ακόμα απαιτείται να γίνουν η βασικές ρυθμίσεις στα Xbee ([Κεφάλαιο 4.17.5.4](#)).

Συνεπώς για να γίνει διαμόρφωση του Xbee Router Remotely μέσω του Arduino απαιτείται το Xbee Coordinator που είναι συνδεδεμένο στο Arduino να στείλει ένα Frame που περιέχει AT Command που θα δημιουργηθεί μέσω του Arduino και να αποσταλθεί στο Xbee Router.

Το πλαίσιο που δημιουργείται μέσω του Arduino για να μπορεί να διαμορφώσει τα Xbee Router πρέπει να περιέχει των παρακάτω κώδικα. Ο παρακάτω κώδικας αλλάζει την κατάσταση της ακίδες P1.

```
void setRem_LED_Yellow_MOV_Sunshade_DIO11(char value) {
    Serial.write(0x7E); //start byte
    Serial.write(byte(0x0)); // high part of length (always zero)
    //low part of length (the number of bytes that follow, not including
    //check
    Serial.write(0x10);
    Serial.write(0x17); //Frame type 0x17 is a remote AT command
    Serial.write(byte(0x0)); //frame id set to zero for no reply
    //Id of recipient, or use 0xFF for broadcast
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
}
```

```

Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(0xFF); // 0xFF for broadcast
Serial.write(0xFF); // 0xFF for broadcast
//16 bit of recipient or 0xFFFFE
Serial.write(0xFF);
Serial.write(0xFE);
Serial.write(0x02); // 0x02 to apply changes immediately on remote
//AT command name in ASCII character
Serial.write('P');
Serial.write('1');
//AT command data in as many bytes as needed
Serial.write(value);
//checksum is all bytes after length bytes
long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '1' + value;
Serial.write(0xFF - (sum & 0xFF)); // calculate the proper checksum
}

```

Η κατάσταση που μπορούμε να αλλάξουμε σε ένα Pin είναι

- Disable[0]
- Commissioning Button [1] (Μόνο για Pin D1)
- Associated Indicator [1] (Μόνο για Pin D5)
- RSSI PWM Output [1] (Μόνο για Pin P0)
- ADC [2] (Υπάρχει η δυνατότητα μόνο για τα Pin D0 έως D3 που υποτρίζουν αναλογικές τιμές)
- Digital Input [3] (Ισχύει για όλα τα Pin D0 έως D5 και από P1 έως P2)
- Digital Out, LOW [4] (Ισχύει για όλα τα Pin D0 έως D5 και από P1 έως P2)
- Digital Out, HIGH [5] (Ισχύει για όλα τα Pin D0 έως D5 και από P1 έως P2)

Συνεπώς η τιμή Value στον κώδικα πρέπει να περιχέει μια από της παραπάνω καταστάσεις.

Η καταστάσεις αυτές εφαρμόζονται με την εντολή:

```
setRem_LED_Yellow_MOV_Sunshade_DIO11 (Value);
```

όπου Value = 0x00 ή 0x01 ή 0x02 ή 0x03 ή 0x04 ή 0x05

Ύστερα στο XCTU φαίνεται το PIN code PIN που θέλουμε να ελέγξουμε κάθε φορά [Εικόνα 5.2](#) είναι το P1 οπότε ορίζουμε το P1 σε 0x4 για LOW ή σε 0x5 για HIGH μέσω της μεταβλητής Value. Με αυτό τον τρόπο ελέγχονται τα pin του Router από το Arduino.

Στην περίπτωση του ελέγχου των ρυθμίσεων από ένα δεύτερο Router το μόνο που θα άλλαζε είναι ότι δεν θα έμπαινε ως destination address (δηλαδή από το 4 έως 12 BIT) η τιμή broadcast 0x0000000000000000FFFF αλλά η διεύθυνση του προς έλεγχο Router. Στην περίπτωση μας έχουμε μονό ένα Router οπότε δεν χρειάζεται να μπει destination address και

γίνεται broadcast δηλαδή σε όλες τις συσκευές Router (που στη δική μας περίπτωση είναι μόνο μία).

▼ I/O Settings  
Modify DIO and ADC options

<b>D0</b> AD0/DIO0 Configuration	Commissioning Button [1]	
<b>D1</b> AD1/DIO1 Configuration	Disabled [0]	
<b>D2</b> AD2/DIO2 Configuration	Disabled [0]	
<b>D3</b> AD3/DIO3 Configuration	Disabled [0]	
<b>D4</b> DIO4 Configuration	Disabled [0]	
<b>D5</b> DIO5/Assoc Configuration	Associated indicator [1]	
<b>P0</b> DIO10/PWM0 Configuration	RSSI PWM Output [1]	
<b>P1</b> DIO11 Configuration	Disabled [0]	
<b>P2</b> DIO12 Configuration	Disabled [0]	
<b>PR</b> Pull-up Resistor Enable	NA [1] NA [2]	
<b>LT</b> Associate LED Blink Time	Digital Input [3] Digital Out, Low [4] Digital Out, High [5]	
<b>RP</b> RSSI PWM Timer	28	x 100 ms
<b>DO</b> Device Options	1	Bitfield

Εικόνα 4.49 Απεικόνισή από τις ρυθμίσεις του XCTU.

#### 4.18 Υλικά κατασκευής

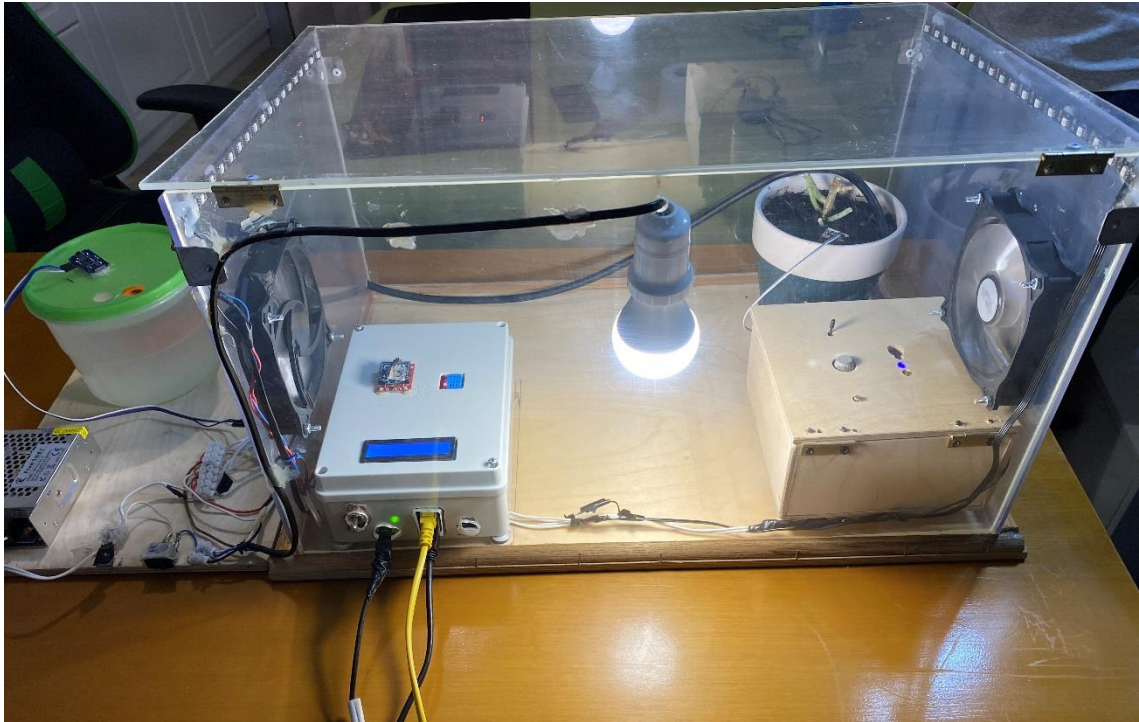
Για την καλύτερη κατανόηση της λειτουργίας του IoT θερμοκηπίου θα υλοποιηθεί μια κατασκευή από plexiglass και ξύλο.

Παρακάτω υπάρχουν ενδεικτικές εικόνες της συνολικής κατασκευής.

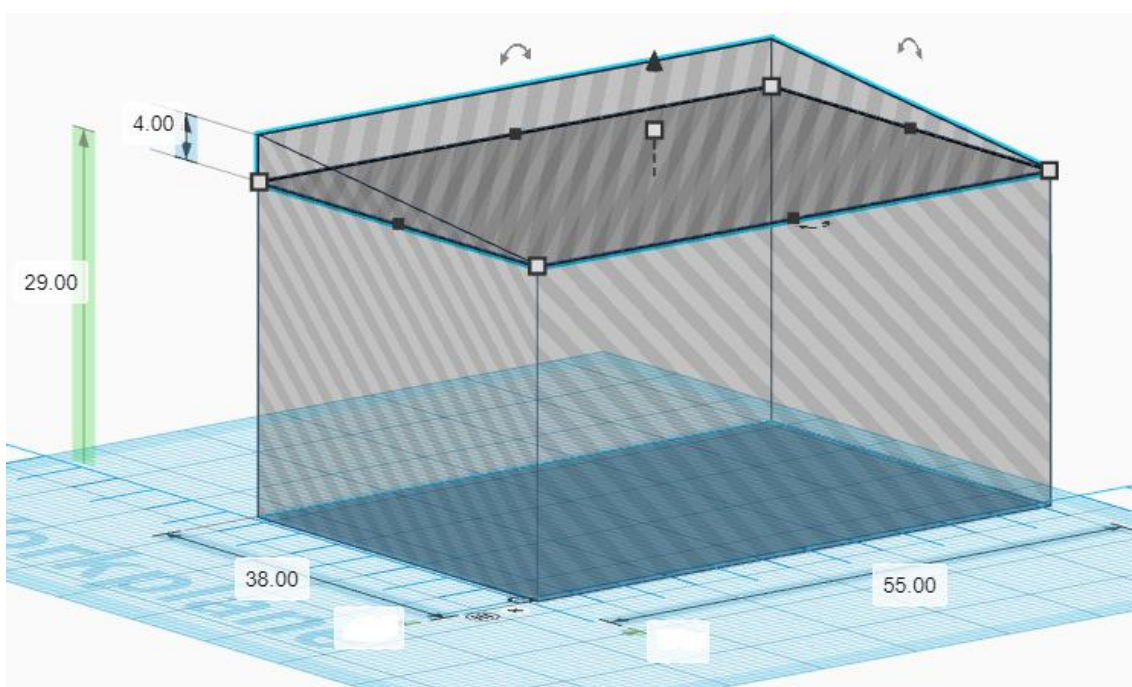


Εικόνα 4.50 Συνολική κατασκευή (μπροστινή όψη).





Εικόνα 4.51 Συνολική κατασκευή (πίσω όψη).



Εικόνα 4.52 Σχέδιο 3D του θερμοκηπίου με τις διαστάσεις του μέσω της πλατφόρμας Tinkercad.

Για την κατασκευή του θερμοκηπίου χρησιμοποιηθήκαν τα παρακάτω υλικά:

- 2 κομμάτια plexiglass 33\*38 cm
- 1 κομμάτι plexiglass 55\*29 cm
- 1 κομμάτι plexiglass 55\*39 cm
- 1 κομμάτι plexiglass 38\*56 cm



- 2 μεντεσέδες για να ανοίγει το πάνω μέρος από το θερμοκήπιο
- 4 μεταλλικές γωνιακές βάσεις
- 14 πριτσίνια για την σύνδεση των μεντεσέδων και των μεταλλικών γωνιακών βάσεων
- Ξύλινη βάση με διαστάσεις 58\*42 cm
- Ξύλινη κορνιζά με διαστάσεις 58\*42 cm για την στήριξη του plexiglass
- Σιλικόνη για την σύνδεση των plexiglass
- Ξυλόκολλά για την σύνδεση της βάσης με την κορνιζά.

Τέλος, έγιναν 2 ανοίγματα 10\*10 στα πλαϊνά plexiglass για την τοποθέτηση των 2 ανεμιστήρων οι οποίοι στηρίχτηκαν στο plexiglass με 4 βίδες και παξιμάδια για κάθε Fan.

Παρακάτω υπάρχουν ενδεικτικές εικόνες της κατασκευής.



*Εικόνα 4.53 Αριστερή όψη θερμοκήπιο που απεικονίζει των ανεμιστήρα εξαερισμού..*



*Εικόνα 4.54 Δεξιά όψη που απεικονίζει τον ανεμιστήρα ψύξης και το εξωτερικό LDR.*

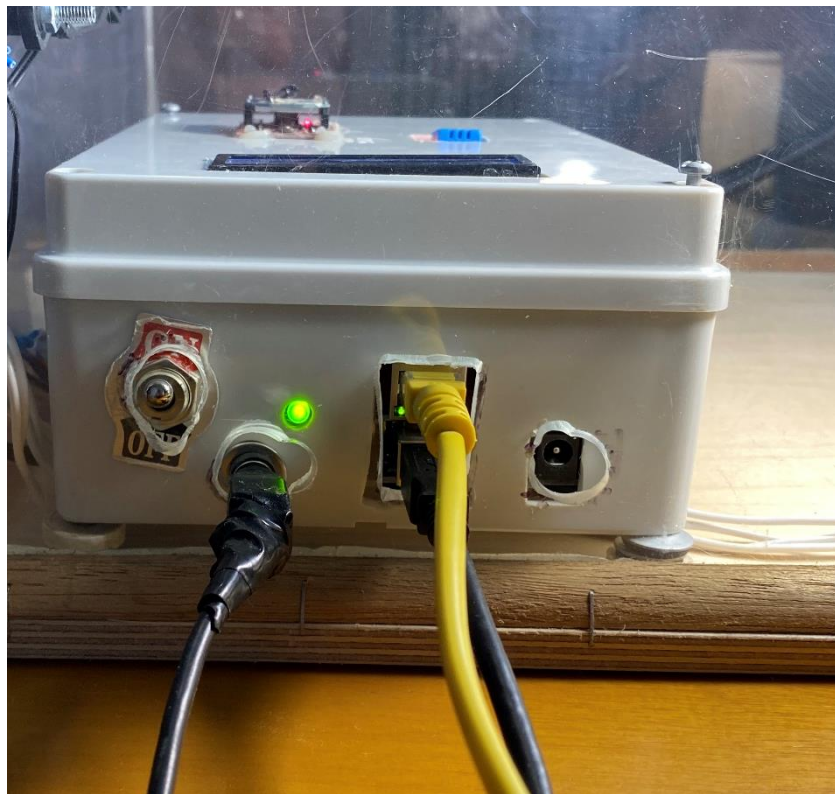
Για την φιλοξενία του Arduino mega, του ethernet shield, του Xbee S2 Coordinator, του DHT11, της οθόνης LCD, των relay, του RTC, της πλακέτας ελέγχου των Fan και της LED ταινίας χρησιμοποιήθηκε ένα στεγανό κουτί διακλάδωσης πλαστικό 20\*15\*7 cm.

Παρακάτω υπάρχουν ενδεικτικές εικόνες του κουτιού διακλαδώσεις (Arduino Box) με της συσκευές εισόδου και εξόδου.

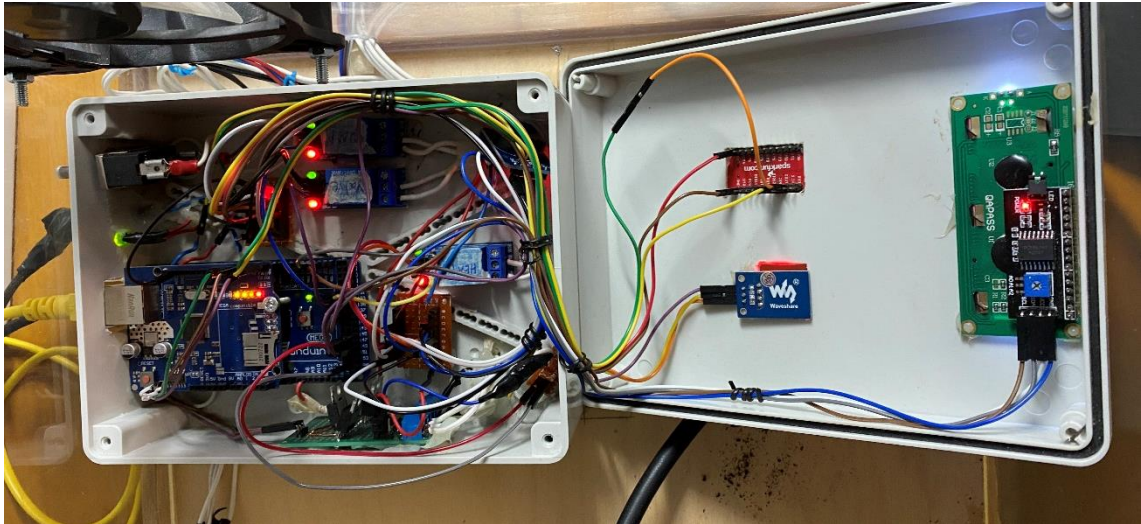




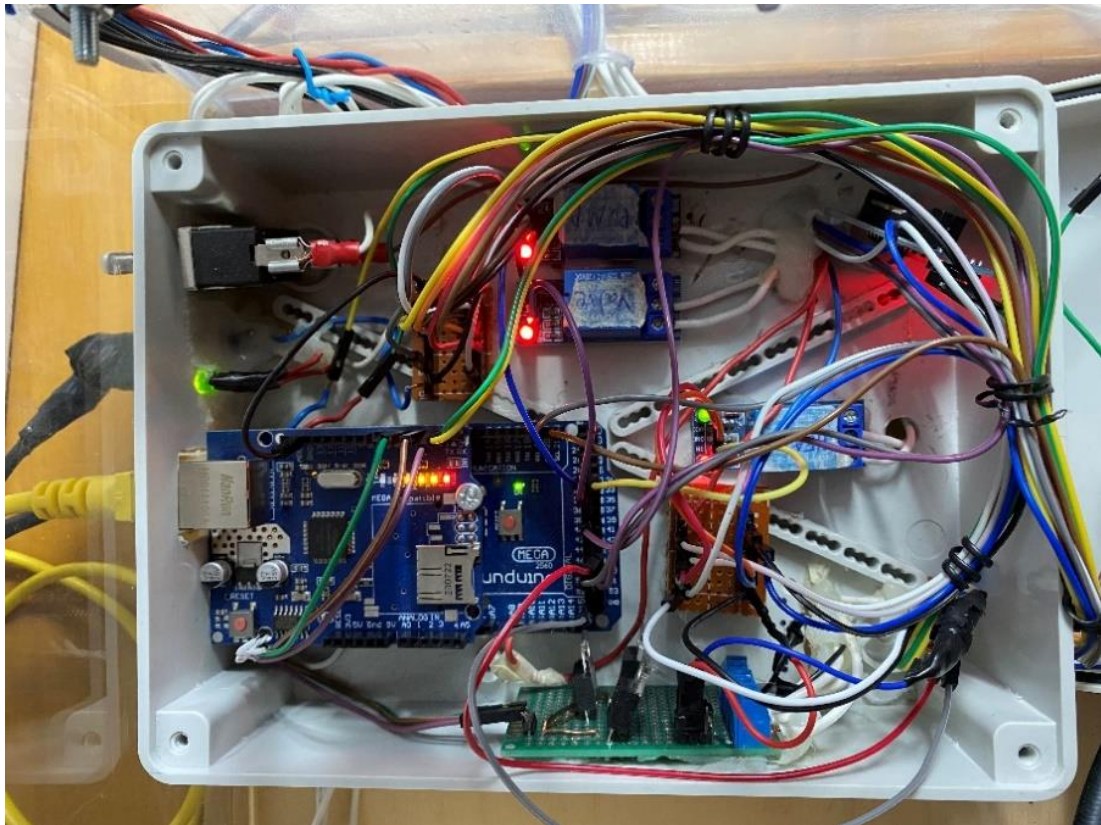
*Εικόνα 4.55 Το εξωτερικό μέρος από το κουτί που φιλόξενη το Arduino Mega.*



*Εικόνα 4.56 Η πίσω όψη με την τροφοδοσίες και την σύνδεση Ethernet.*

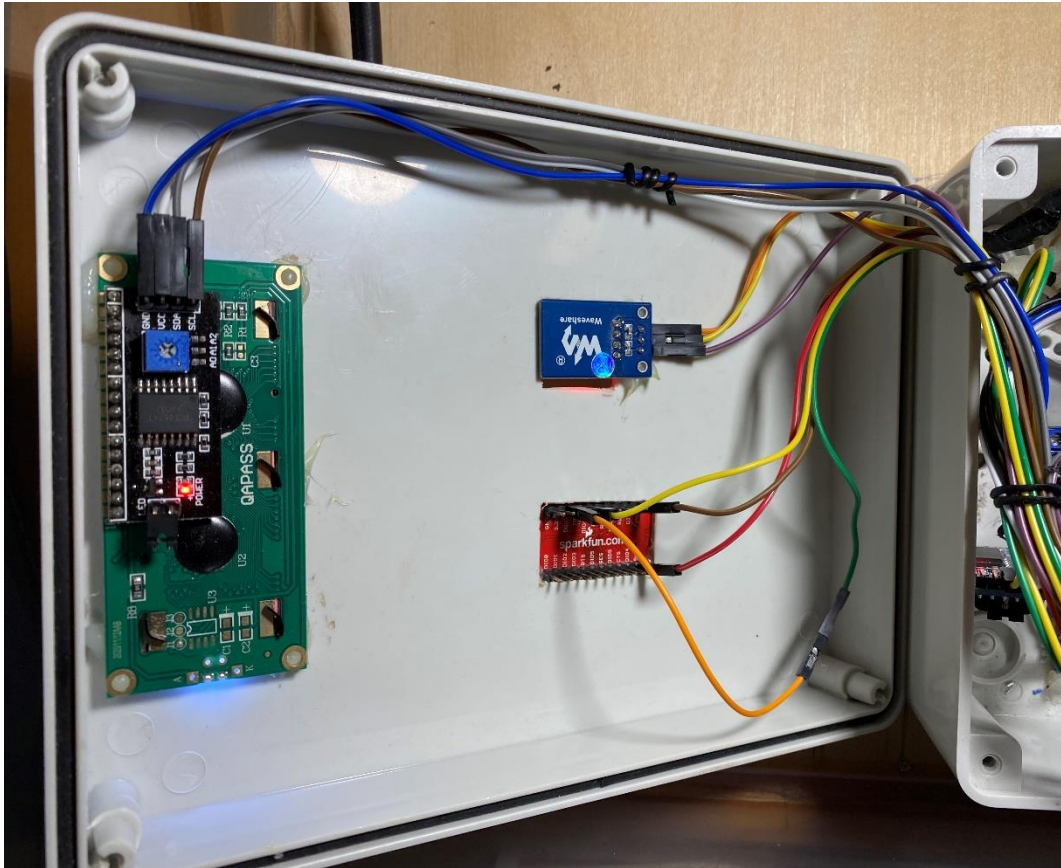


*Εικόνα 4.57 Το εσωτερικό του κουτιού.*

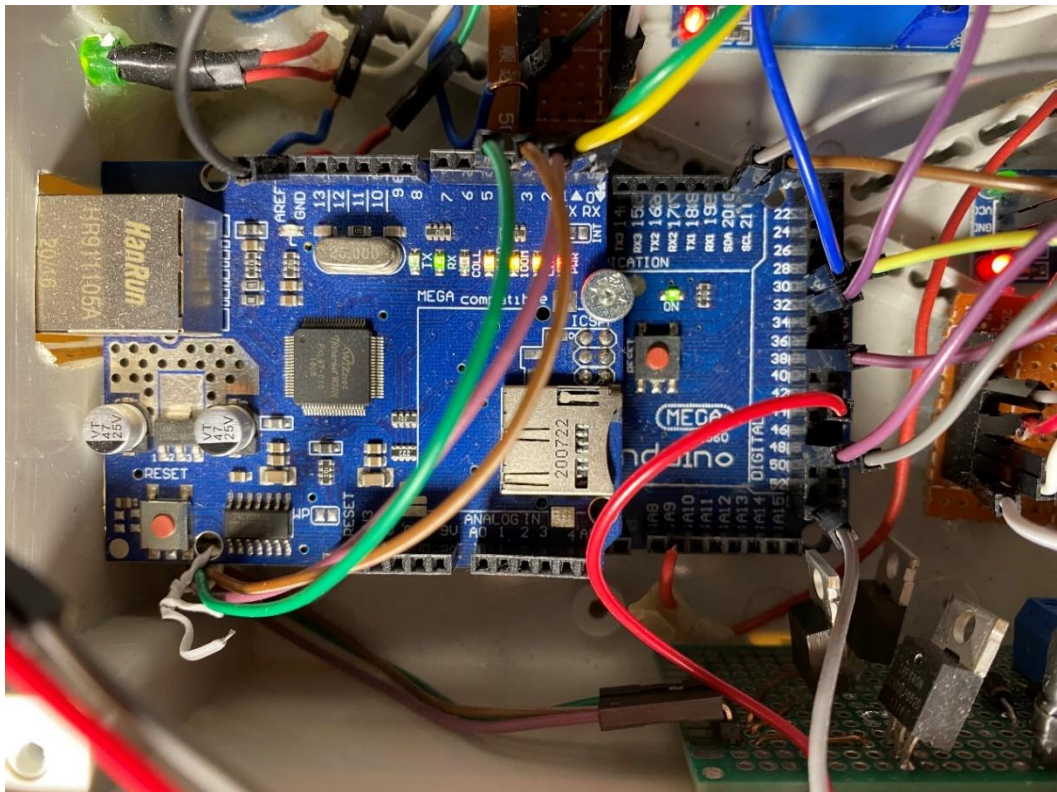


*Εικόνα 4.58 Το εσωτερικό από το κουτί που φιλοξενεί το Arduino Mega*



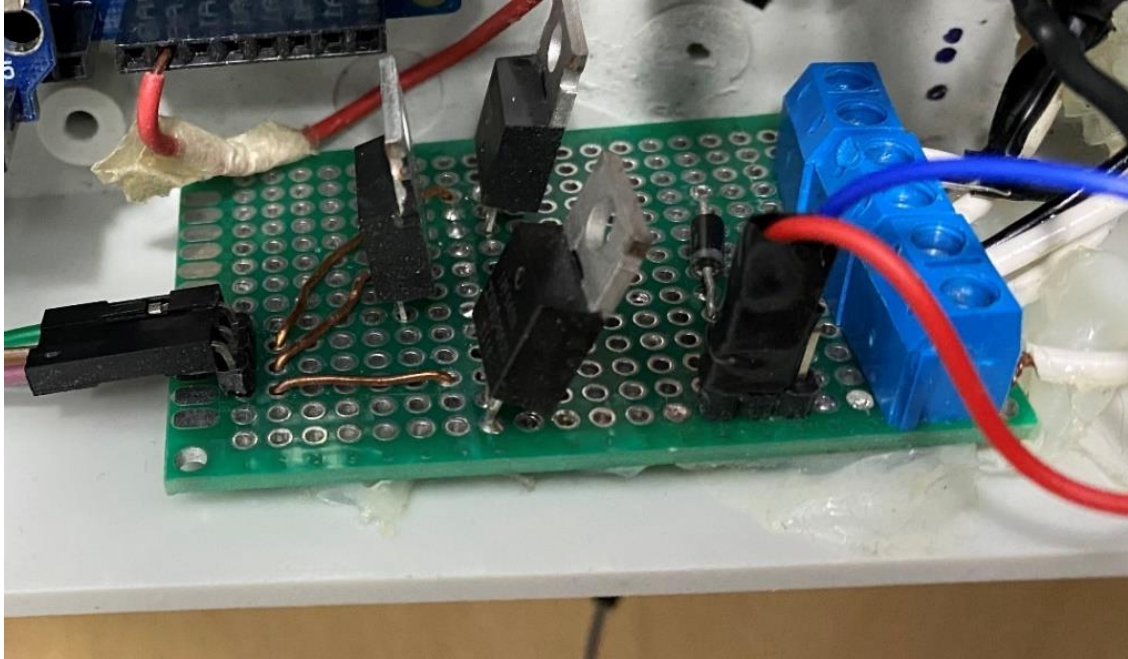


Εικόνα 4.59 Το καπάκι του κουτιού με της τοποθετημένες συσκευές.

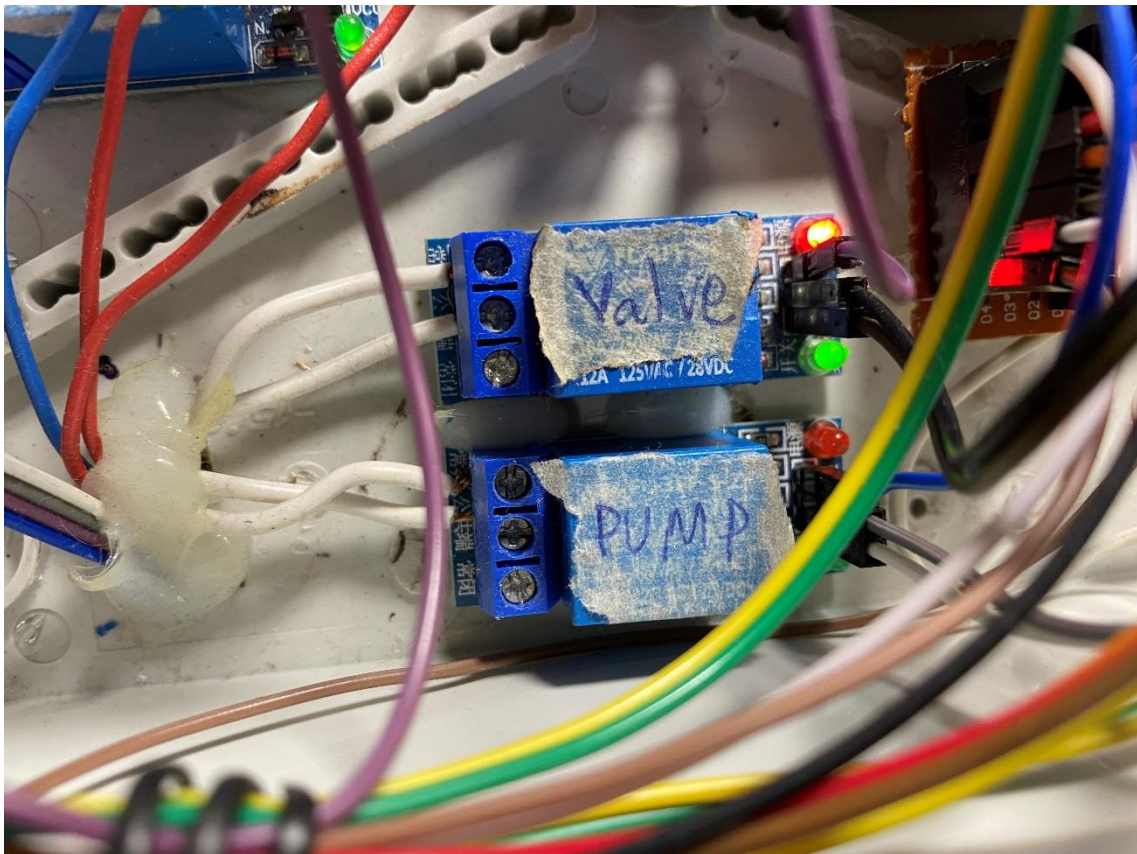


Εικόνα 4.60 Το Arduino MEGA





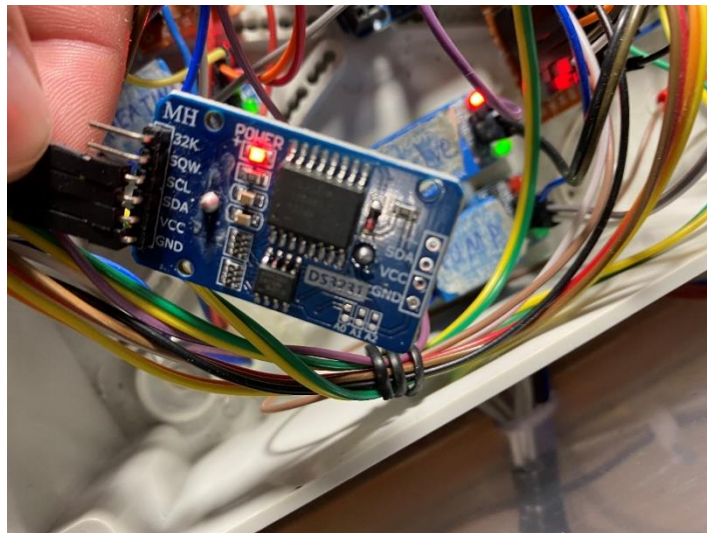
Εικόνα 4.61 Πλακέτα ελέγχου των Fan και της LED ταινίας.



Εικόνα 4.62 Τα Relay για την ηλεκτρική βαλβίδα και την αντήλια.



Εικόνα 4.63 Το Relay της λάμπας θερμότητας.



Εικόνα 4.64 Πίσω όψη του ρολογιού πραγματικού χρόνου.



Εικόνα 4.65 Μπροστινή όψη ρολογιού πραγματικού χρόνου.



Για την κατασκευή του ξύλινου κουτιού για την φιλοξενία του Xbee S2 Router, του power bank για την τροφοδοσία των συσκευών, του αισθητήρα αερίου, του LDR, του αισθητήρα υγρασίας χρώματος, των 3 LEDES και του Buzzer χρησιμοποιήθηκαν τα παρακάτω υλικά.

- 4 ξύλα με διαστάσεις 8\*17 cm
- 2 ξύλα με διαστάσεις 17\*17 cm
- 2 μεντεσέδες για το καπάκι
- βίδες
- ξυλόκολλά

Παρακάτω υπάρχουν ενδεικτικές εικόνες κουτιού Xbee (Xbee Box).

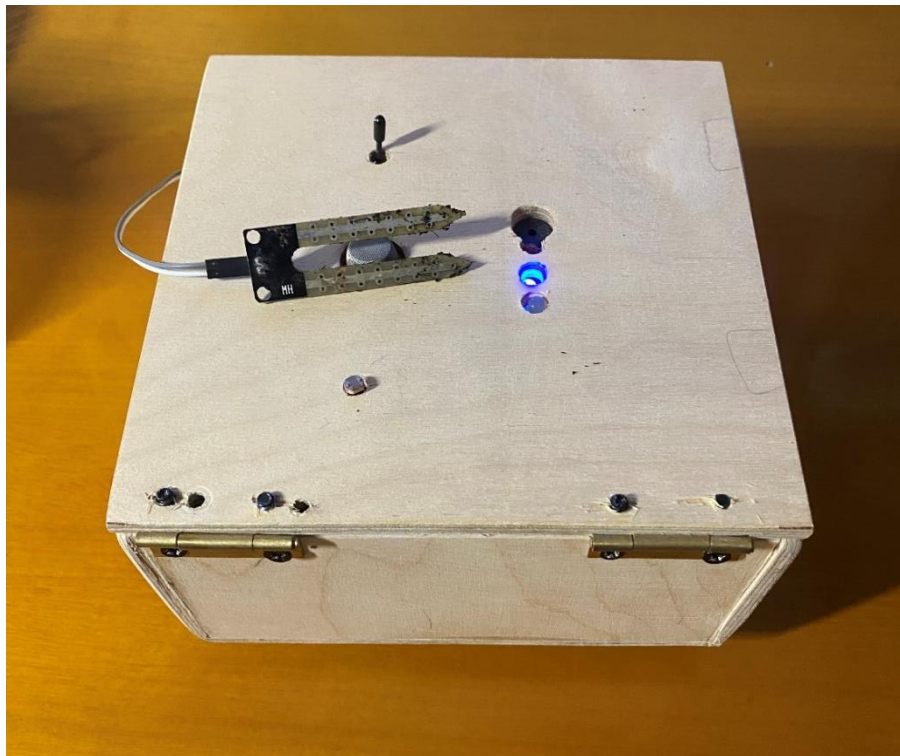


*Εικόνα 4.66 Μπροστινή όψη του κουτιού Xbee*

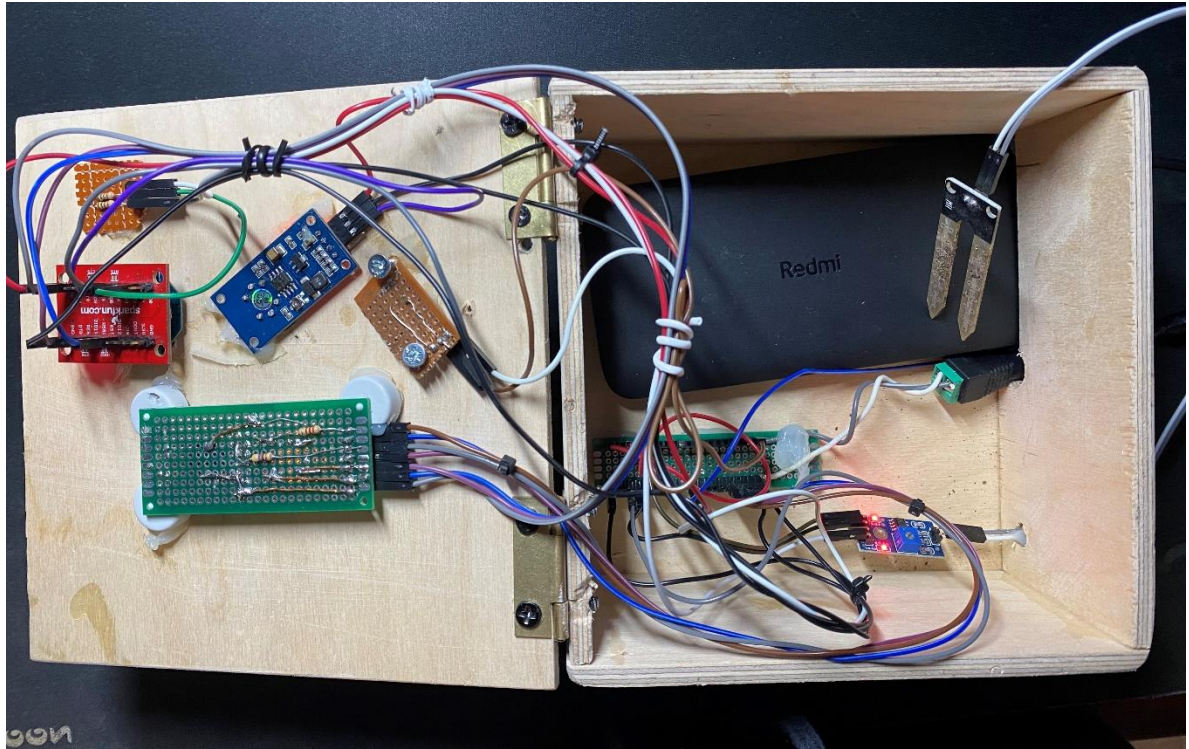




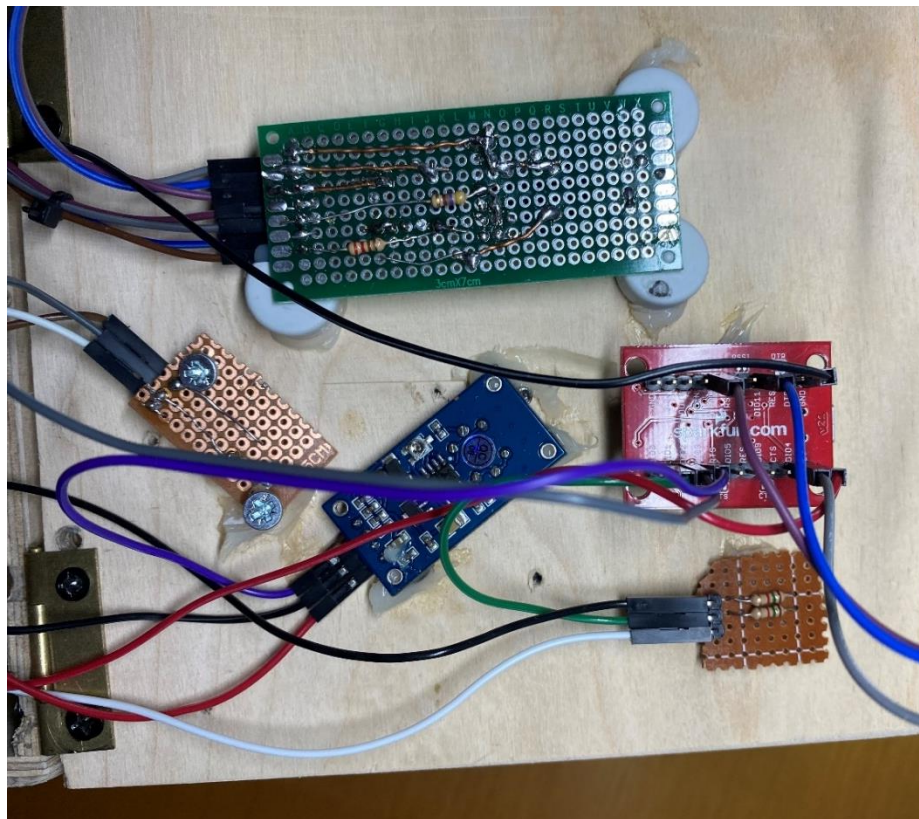
*Εικόνα 4.67 Η πάνω όψη από το κουτί που φιλοξενεί το Xbee Router.*



*Εικόνα 4.68 Πίσω όψη του κουτιού Xbee.*

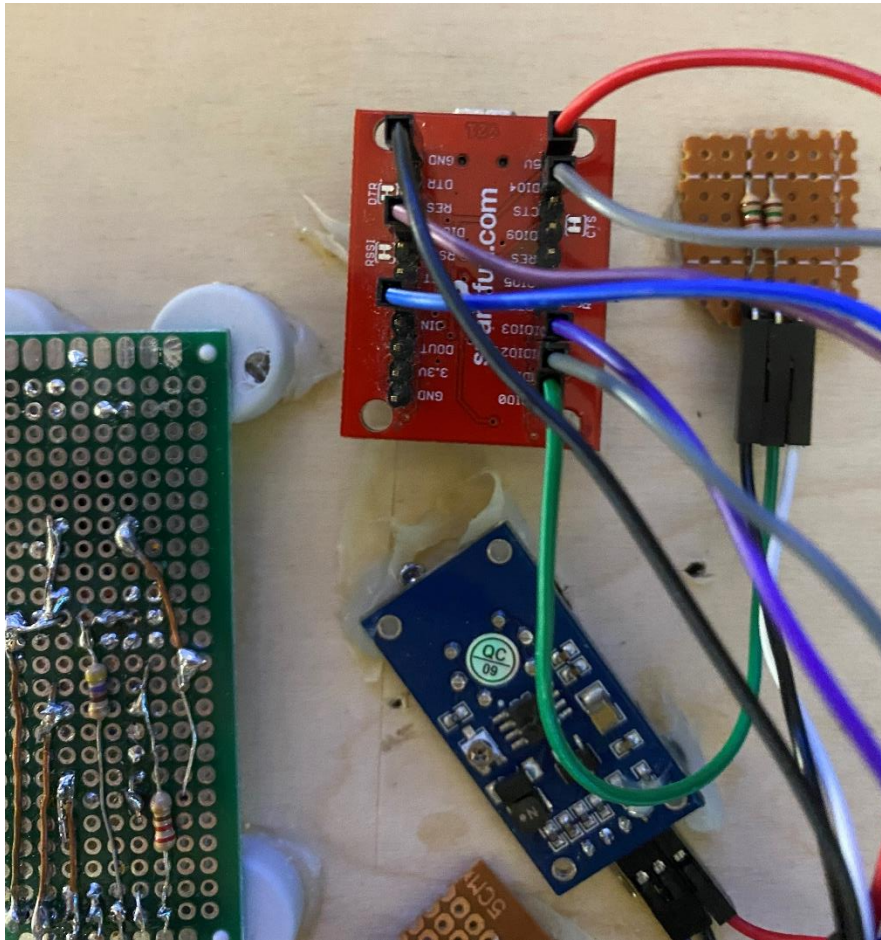


Εικόνα 4.69 Το εσωτερικό του Xbee Box.

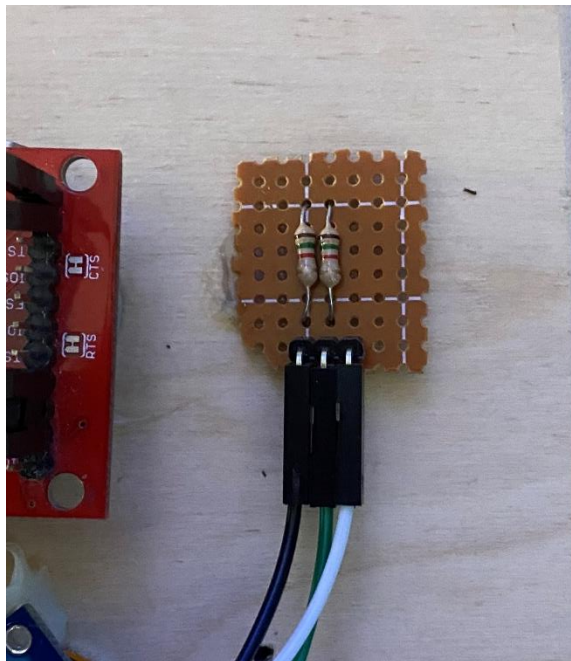


Εικόνα 4.70 Το καπάκι από το Xbee box με της συσκευές εισόδου, εξόδου και το Xbee

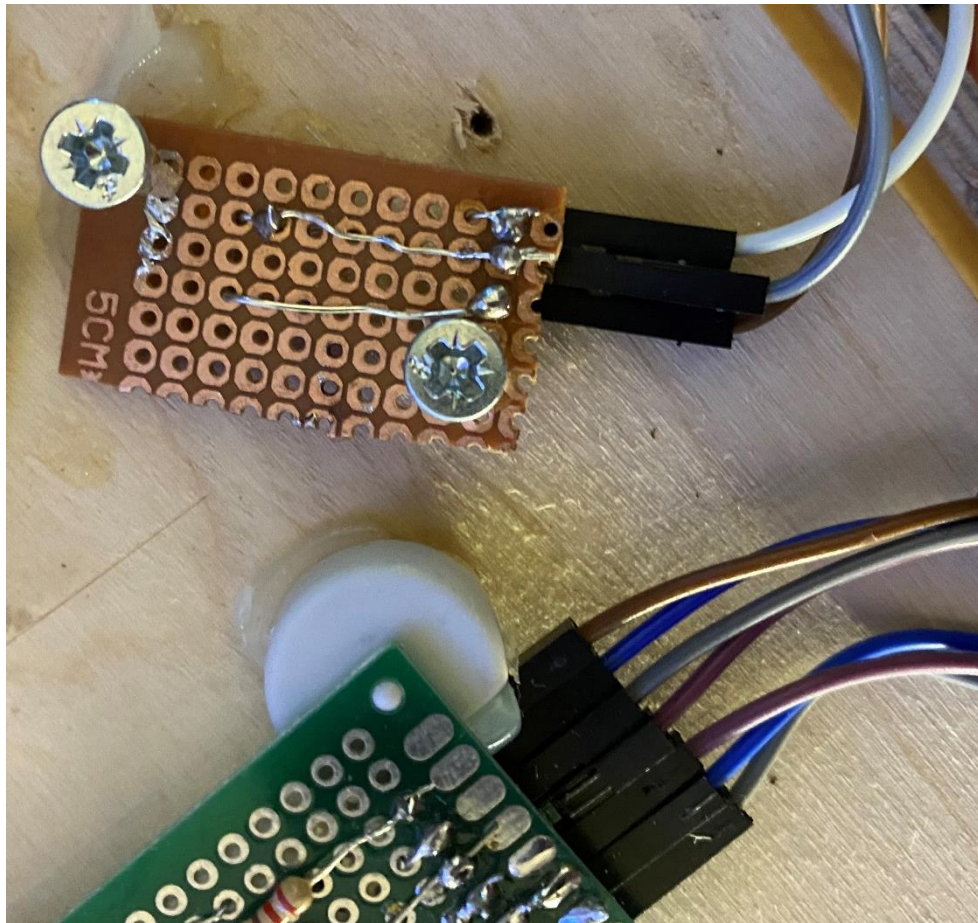




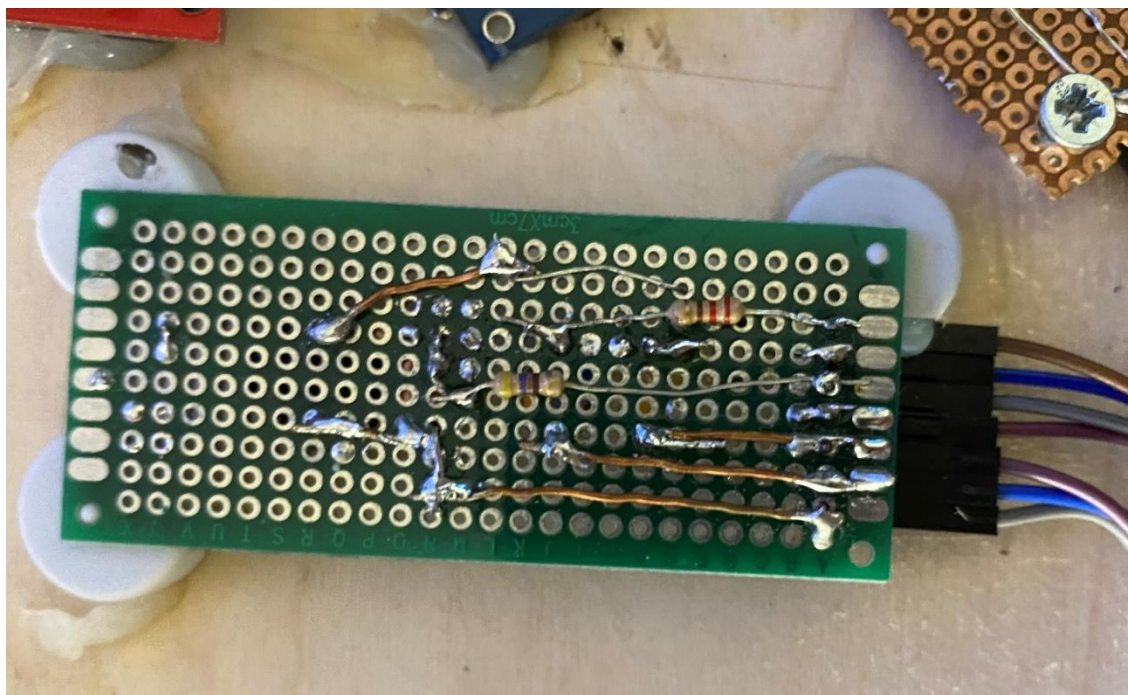
Εικόνα 4.71 Η πίσω όψη του Xbee και του αισθητήρα MQ-07 στο καπάκι του Xbee Box.



Εικόνα 4.72 Ο διαιρέτης τάσης του αισθητήρα υγρασίας εδάφους στο καπάκι του Xbee Box.

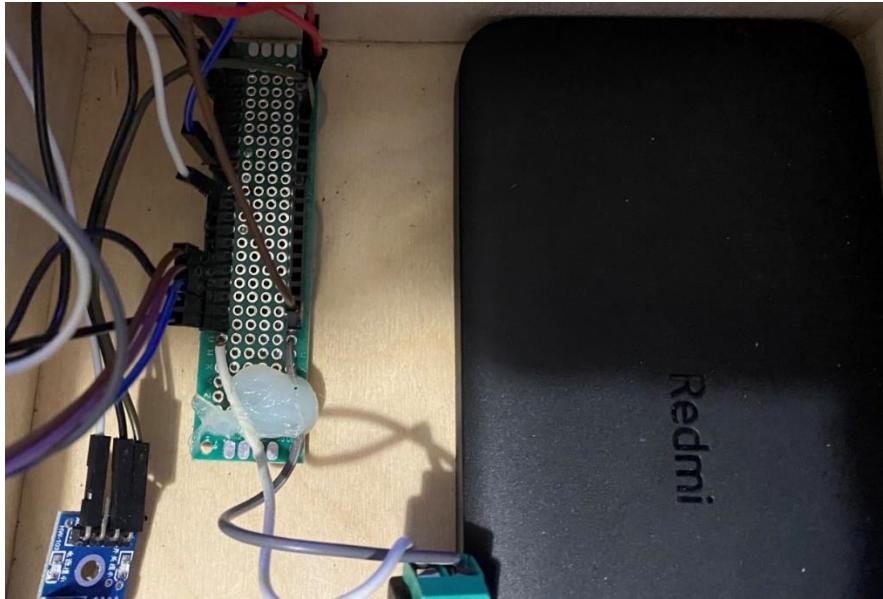


*Εικόνα 4.73 Η πίσω όψη της πλακέτας του εσωτερικού LDR στο καπάκι του Xbee Box.*



*Εικόνα 4.74 Η πλακέτα των συσκευών εξόδου στο καπάκι του Xbee Box.*



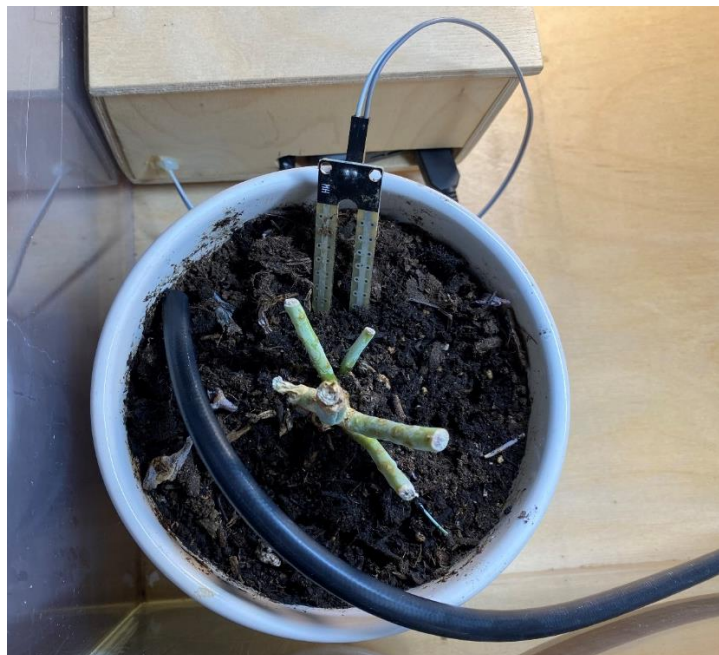


Εικόνα 4.75 Η κλέμα διακλαδώσεις και το Power Bank που τροφοδοτεί το Xbee και της συσκευές.

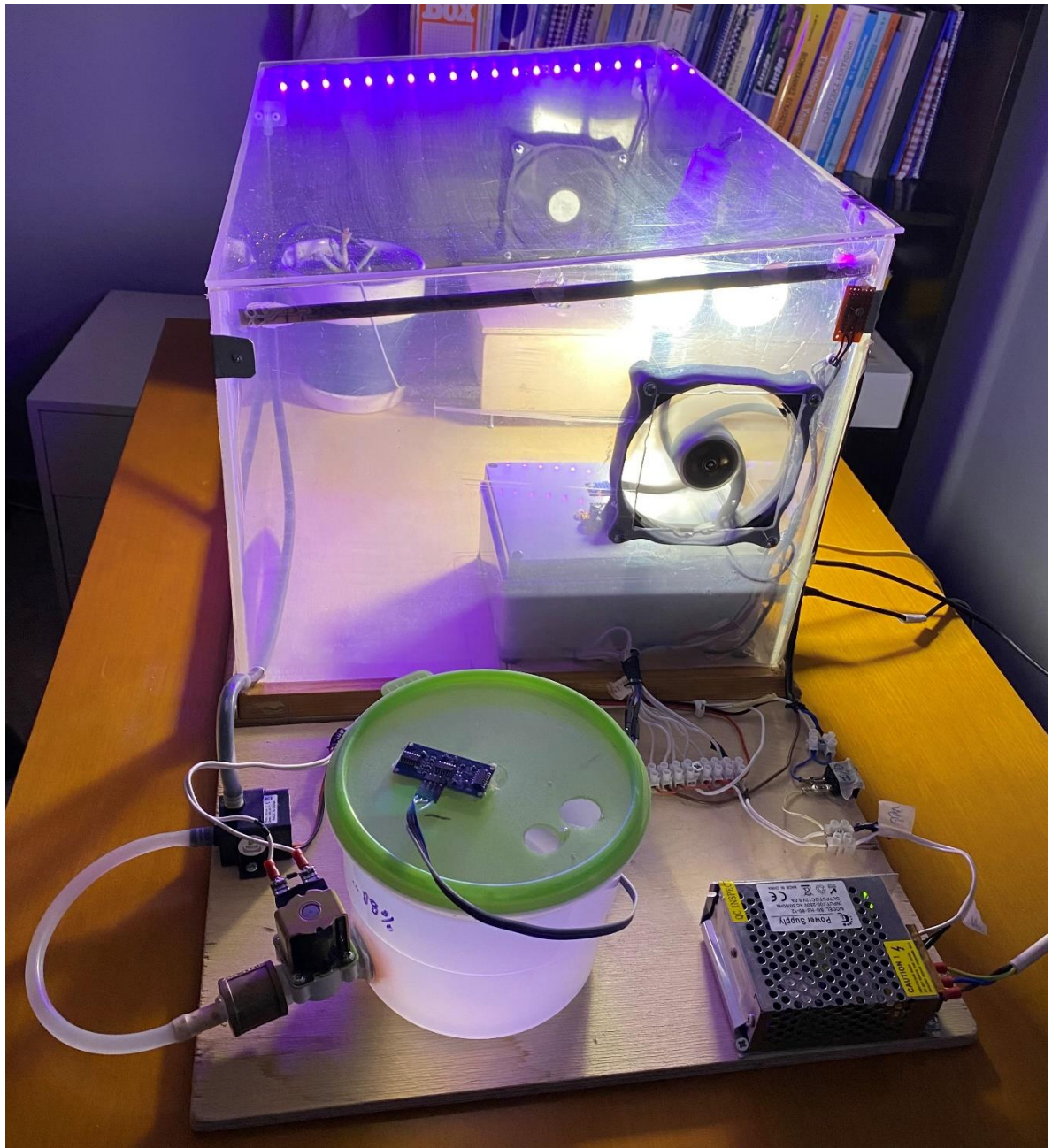
Για την κατασκευή του συστήματος ποτίσματος χρησιμοποιήθηκαν:

- ένα πλαστικό δοχείο
- σωληνάκια,
- διάφοροι connectors για την σύνδεσης της αντλίας με την βάνα και το δοχείου,
- και μια ξύλινη βάση

Παρακάτω υπάρχουν ενδεικτικές εικόνες του συστήματος ποτίσματος

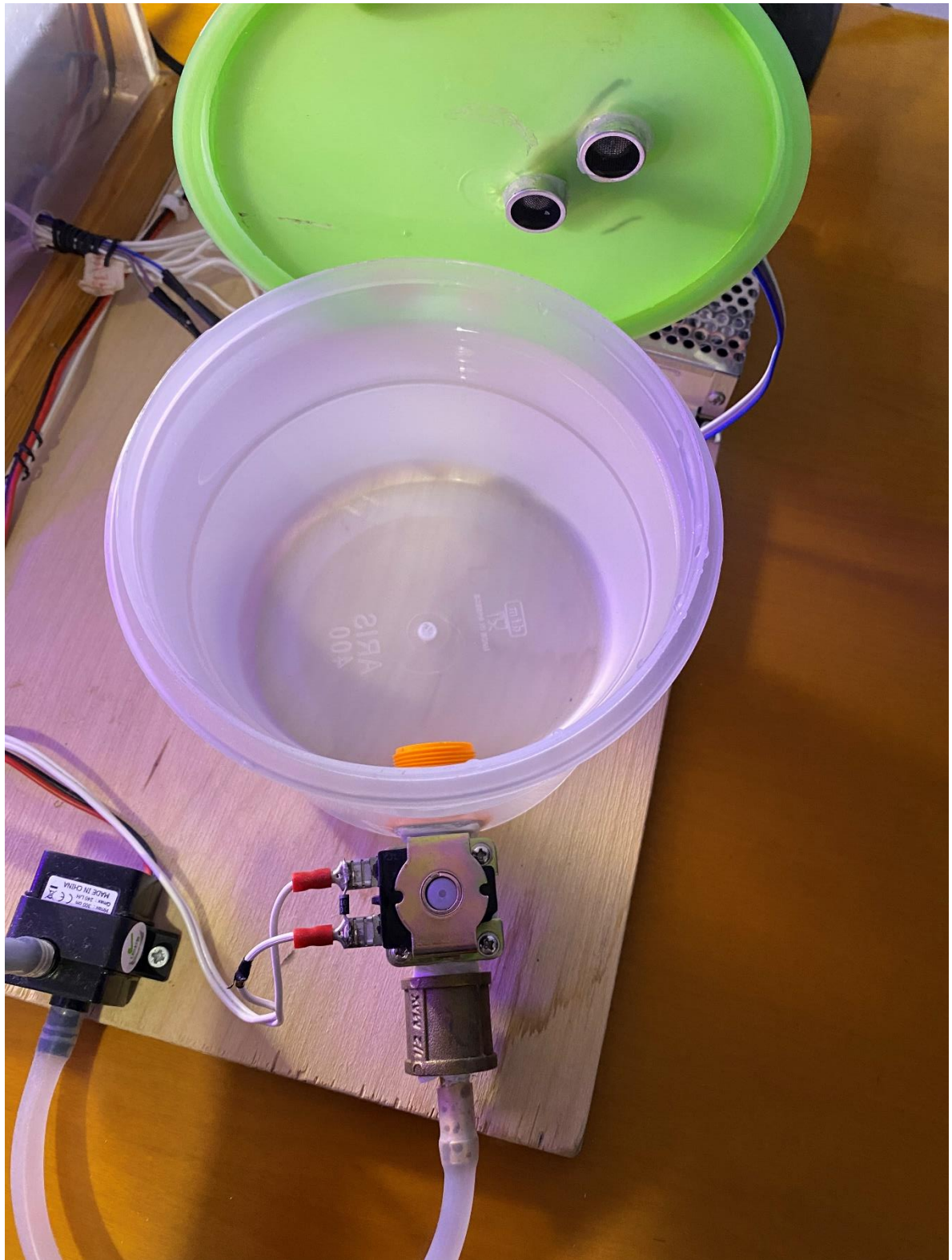


Εικόνα 4.76 Ο αισθητήρας υγρασίας εδάφους και ο σωλήνας ποτίσματος



*Εικόνα 4.77 Το συνολικό σύστημα ποτίσματος.*





Εικόνα 4.78 Το δοχείο, η ηλεκτρική βαλβίδα, η αντλία και ο αισθητήρας απόστασης.

## 5 Περιγραφή σεναρίων αυτοματισμού θερμοκηπίου

### 5.1 Πρόλογος

Στο κεφάλαιο αυτό θα παρουσιαστούν τα σενάρια αυτοματισμού του συστήματος θερμοκηπίου που θα έχουν ως στόχο τον έλεγχο του μικροκλίματος, τον έλεγχο του ποτίσματος των φυτών αλλά και την ειδοποίηση σε περίπτωση φωτιάς ή υπερχειλίσης του δοχείου ποτίσματος.

Το σύστημα του θερμοκηπίου αποτελείται από 7 σενάρια αυτοματισμού τα οποία είναι:

1. Το σενάριο αυτοματισμού ελέγχου θερμοκρασίας που θα ελέγχει την εσωτερική θερμοκρασία του θερμοκηπίου.
2. Το σενάριο αυτοματισμού ελέγχου εξαερισμού συμβάλει στον εμπλουτισμό του οξυγόνου, στην μείωση της υγρασίας και στον αερισμό του χώρου σε περίπτωση φωτιάς.
3. Το σενάριο αυτοματισμού ελέγχου συναγερμού που θα ενεργοποιείται σε περίπτωση φωτιάς ή υπερχειλίσης του νερού από το δοχείο.
4. Το σενάριο αυτοματισμού ελέγχου ψεκασμού ομίχλης από νερό που είναι για την αύξηση της υγρασίας περιβάλλοντος αλλά και για την πυρόσβεση.
5. Το σενάριο αυτοματισμού ελέγχου ποτίσματος που ανάλογα με την υγρασία του χώματος θα γίνει το πότισμα των φυτών.
6. Το σενάριο αυτοματισμού του σκίαστρου που χρησιμοποιείται για την μείωση της ηλιακής ακτινοβολίας.
7. Το σενάριο αυτοματισμού ελέγχου επέκτασης φωτισμού που είναι για την επέκταση του φωτός τους χειμερινούς μήνες αλλά και τις βροχερές μέρες.

### 5.2 Περιγραφή σεναρίου αυτοματισμού ελέγχου θερμοκρασίας

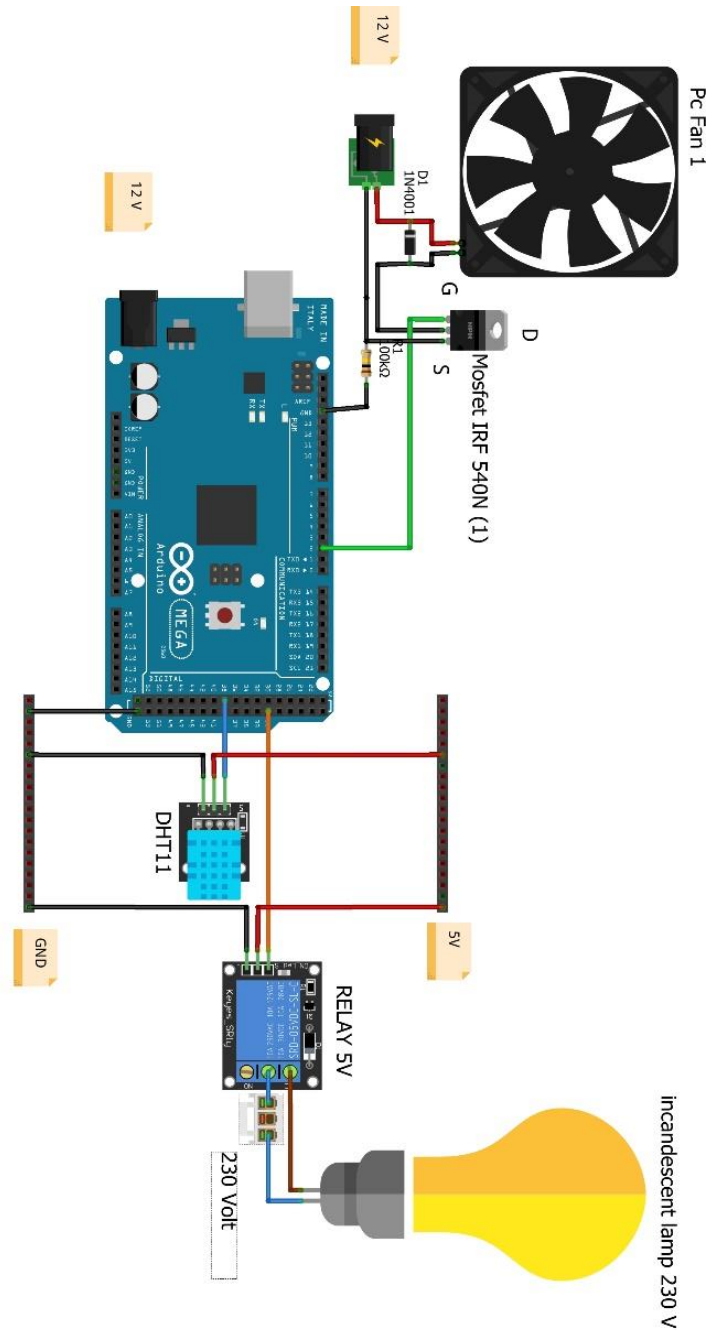
Το Σενάριο αυτοματισμού ελέγχου θερμοκρασίας αποτελείται από:

- 1 Xbee ως Coordinator του οποίου οι ακίδες του συνδέονται με το Arduino ως εξής: ακίδα D1-TX του Arduino με το DIN του Xbee και η ακίδα D0-RX του Arduino με το DOUT του Xbee.
- 1 Mosfet N- Channel 33A IRF540N-(1) που συνδέεται στην ακίδα D2 του Arduino mega.
- 1 δίοδο (D1) 1N4001 συνδέεται όπως φαίνεται στο παρακάτω Κύκλωμα 1.
- 1 αντίσταση 100 KΩ



- 1 λάμπα πυρακτώσεως 230 volt ως εξομοίωση ενός θερμαντικού στοιχείου που συνδέεται στο Heating Relay 5V.
- 1 Heating Relay 5V που συνδέεται στην ακίδα D31 του Arduino mega.

Στο [Κύκλωμα 5.1](#) παρουσιάζεται ο τρόπος διασύνδεσης των εξαρτημάτων για την υλοποίηση του σεναρίου.



fritzing

Κύκλωμα 5.1 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου θερμοκρασίας.

Περιγραφή σεναρίου:

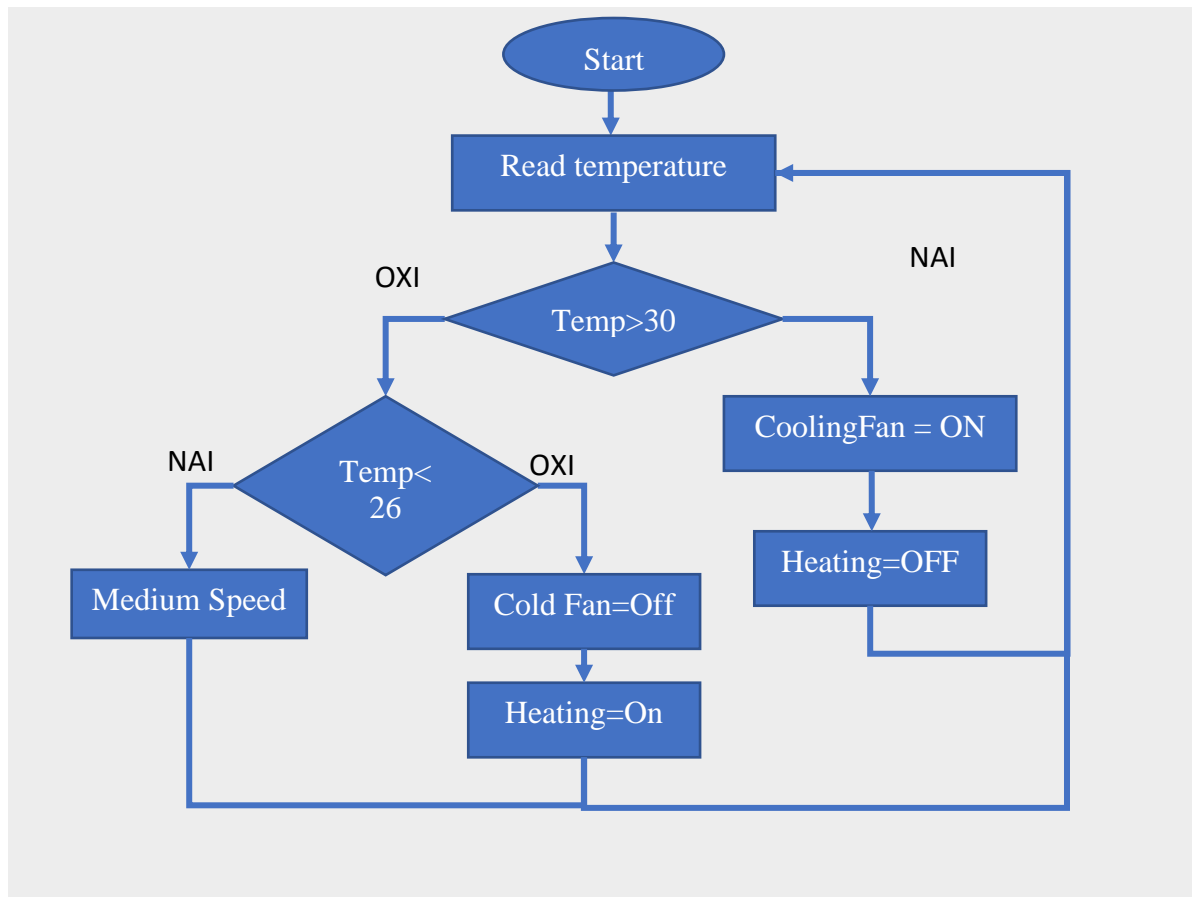
Η λειτουργία του σεναρίου αυτοματισμού ελέγχου της θερμοκρασίας είναι να διατήρηση της θερμοκρασίας από 26 έως 30 βαθμούς Κελσίου,

- Όταν θα χρειάζεται αύξηση θερμοκρασίας στο περιβάλλον του θερμοκηπίου, το Arduino θα ενεργοποιεί τον ρελέ 5V και ταυτόχρονα η λάμπα πυρακτώσεως θα ανάβει,
- ενώ για την μείωση της θερμοκρασίας στο περιβάλλον του θερμοκηπίου το Arduino θα ενεργοποιηθεί η πύλη (Gate) του Mosfet N-Channel 33A - IRF540N που με την σειρά της θα ενεργοποιήσει των 12 volt ανεμιστήρα, παράλληλα με τον ανεμιστήρα θα τοποθετηθεί μια δίοδο snubber. Η άνοδος συνδέεται στο πλιν του ανεμιστήρα και η κάθοδος στο συν του ανεμιστήρα, ώστε, όταν σταματάει ο ανεμιστήρας να μην γυρνάνε τα ρεύματα προς το Arduino.

Αρχικά, το Arduino θα διαβάζει την εσωτερική θερμοκρασία του θερμοκηπίου από τον αισθητήρα DHT11

- εάν η θερμοκρασία είναι πάνω από 30 βαθμούς Celsius τότε το Arduino θα έχει στο pin D3 Duty Cycle 100% και ο ανεμιστήρας θα δουλεύει στις Full στροφές,
- εάν η θερμοκρασία είναι κάτω από 30 βαθμούς Celsius, τότε το Arduino θα έχει στο pin D3 Duty Cycle 50% και ο ανεμιστήρας θα δουλεύει στις μισές στροφές και
- εάν είναι κάτω από 26 βαθμούς Celsius, τότε το Arduino θα έχει στο pin D3 Duty Cycle 0% και ο ανεμιστήρας θα είναι κλειστός επίσης τότε θα ανάψει η λάμπα πυρακτώσεως που εξομοιώνει το θερμαντικό στοιχείο, ώστε να αυξηθεί η θερμοκρασία.

Παρακάτω ακολουθεί το λογικό διάγραμμα ροής του σεναρίου ([Σχήμα 5.1](#)).



Σχήμα 5.1 Το λογικό διαγράμμα του σεναρίου αυτοματισμού ελέγχου θερμοκρασίας.

Ακολουθεί ο κώδικας Arduino για το σενάριο αυτοματισμού ελέγχου θερμοκρασίας.

```

/*-----( Import needed libraries )-----*/
//Βιβλιοθήκη για αισθητήρα DHT11
#include "DHT.h"

// Digital pin connected to the DHT sensor
#define SEN_Dht11 38
//Relay for heating the area
int RELAY_HeatingLight = 31;
//Pc fan for cooling area Gate Mosfet Irf540N 2 attach to d5 pin Arduino
const int MOS_CoolingFan = 3;

// Select type Dht11
DHT dht(SEN_Dht11, DHT11);

/*-----( Declare Variables )-----*/
// Temperature variables
float Val_temp = 0;

// Μεγίστη τιμή θερμοκρασίας σε βαθμούς celsius
int My_Val_Max_temp=30;
// Ελάχιστη τιμή θερμοκρασίας σε βαθμούς Celsius
int My_Val_Min_temp=26;

void setup() {

```

```

//start serial to 9600 baud
Serial.begin(9600);
// start the dht11
dht.begin();
// set the pin mode to input
pinMode(SEN_Dht11, INPUT);
// set the pin mode to OUTPUT
pinMode(RELAY_HeatingLight, OUTPUT);
// set the pin mode to OUTPUT
pinMode(MOS_CoolingFan, OUTPUT);

// set the pin to LOW
digitalWrite(RELAY_HeatingLight, LOW);
// set the pin to LOW
digitalWrite(MOS_CoolingFan, LOW);
}

void loop() {
  Read_Sen_Dht11_Loop();
  Print_Serial_loop();
  System_Temperature_Loop();
}

// Read Temperature as Celsius (the default)
void Read_Sen_Dht11_Loop() {
  Val_temp = dht.readTemperature();
}

// Control Temperature
void System_Temperature_Loop() {
  //εάν η θερμοκρασία είναι πάνω από My_Val_Max_temp=30 τότε
  if (Val_temp > My_Val_Max_temp) {
    //ο ανεμιστήρας να δουλέψει στις μέγιστες στροφές και
    digitalWrite(MOS_CoolingFan, HIGH);
    //η λάμπα θερμοκρασίας να είναι σβηστή
    digitalWrite(RELAY_HeatingLight, HIGH);
    //εμφάνισε στην σειριακή οθόνη την κατάσταση
    Serial.println(F("Full rpm Cool Fan temp is over 30 and Heating=OFF"));
  }
  //εάν η θερμοκρασία είναι πάνω από My_Val_Min_temp=26 τότε
  else if (Val_temp >= My_Val_Min_temp) {
    //ο ανεμιστήρας να δουλέψει στις μισές στροφές μέσω της PWM και
    analogWrite(MOS_CoolingFan, 127);
    // η λάμπα να είναι σβηστή
    digitalWrite(RELAY_HeatingLight, HIGH);
    // εμφάνισε στην σειριακή οθόνη την κατάσταση
    Serial.println(F("Medium rpm Cool Fan and Heating=OFF"));

    // εάν η θερμοκρασία δεν είναι πάνω από 30 αλλά ούτε πάνω από 26 τότε
  }else {
    // ο ανεμιστήρας να είναι σβηστός και
    digitalWrite(MOS_CoolingFan, LOW);
    //η λάμπα αναμμένη
    digitalWrite(RELAY_HeatingLight, LOW);
    // εμφάνισε στην σειριακή οθόνη την κατάσταση
    Serial.println(F("Zero rpm Cool Fan temp is under 26 and Heating=ON"));
  }
}

// Displays the value on the Serial Monitor
void Print_Serial_loop() {
  Serial.println(F("*****"));
}

```

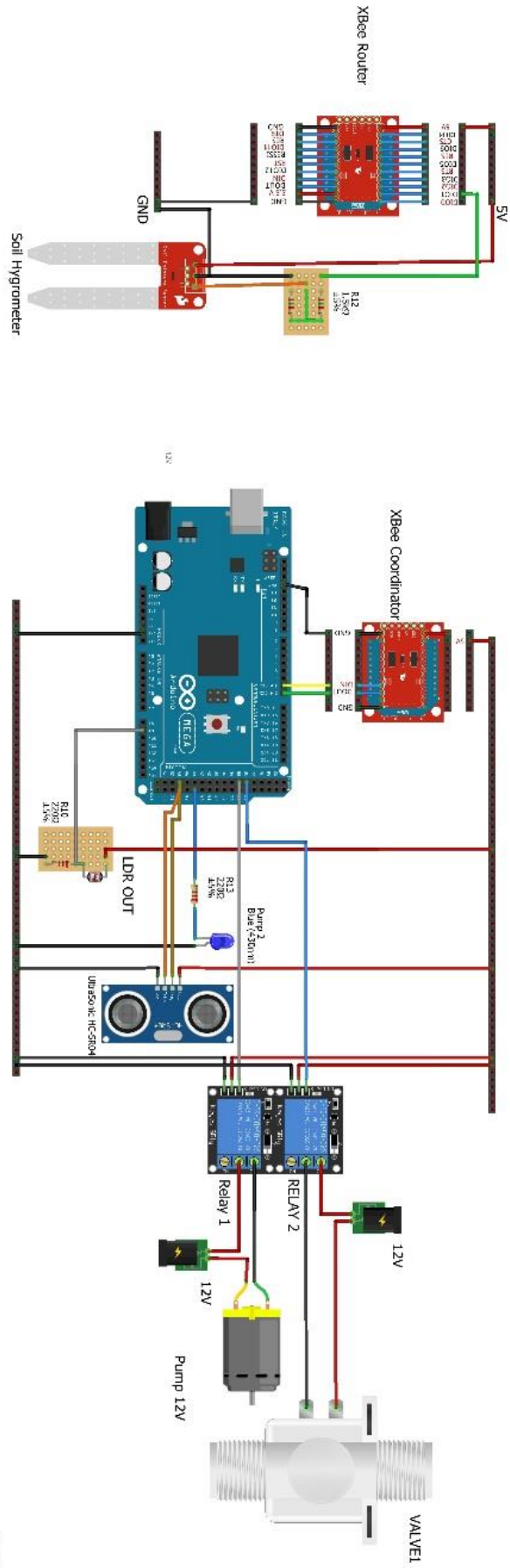
```
Serial.print(F("Temperature: ")); Serial.print(Val_temp);  
Serial.println(F("°C"));  
}
```

### 5.3 Περιγραφή σεναρίου αυτοματισμού ελέγχου ποτίσματος

Το σενάριο ελέγχου ποτίσματος αποτελείται από:

- 1 Xbee ως Coordinator του οποίου οι ακίδες του συνδέονται με το Arduino ως εξής: η ακίδα D1-TX του Arduino με το DIN του Xbee και η ακίδα D0-RX του Arduino με το DOUT του Xbee.
- 1 Xbee που χρησιμοποιείτε ως Router στον οποίο συνδέεται ο αισθητήρας υγρασίας.
- 1 αισθητήρας υγρασίας χώματος Soil Hygrometer που συνδέεται στην ακίδα DI01 του Xbee Router
- 1 αισθητήρας απόστασης Distance HC-SR04 Ultrasonic που συνδέεται στις ακίδες D48 το Trig και το D49 το Echo του Arduino mega.
- 1 Αισθητήρας Φωτοαντίστασης LDR στο Εξωτερικό του θερμοκηπίου που συνδέεται στην ακίδα A9 του Arduino mega.
- 1 μπλε led που εξομοιώνει μια δεύτερη αντλία (pump2), το οποίο συνδέεται στην ακίδα D45 του Arduino mega.
- 1 ηλεκτρονική βαλβίδα που συνδέεται σε ένα Valve Relay 5V.
- 1 αντλία 12 volt (pump1) που συνδέεται στο Pump Relay 5V
- 1 Valve Relay 5V που συνδέεται στο ακίδα D30 του Arduino mega.
- 1 Pump Relay 5V που συνδέεται στο ακίδα D32 του Arduino mega

Στο Κύκλωμα 5.2 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου ποτίσματος. [Κύκλωμα 5.2](#) παρουσιάζεται ο τρόπος διασύνδεσης των εξαρτημάτων για την υλοποίηση του σεναρίου.



fritzing

Κύκλωμα 5.2 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου ποτίματος.

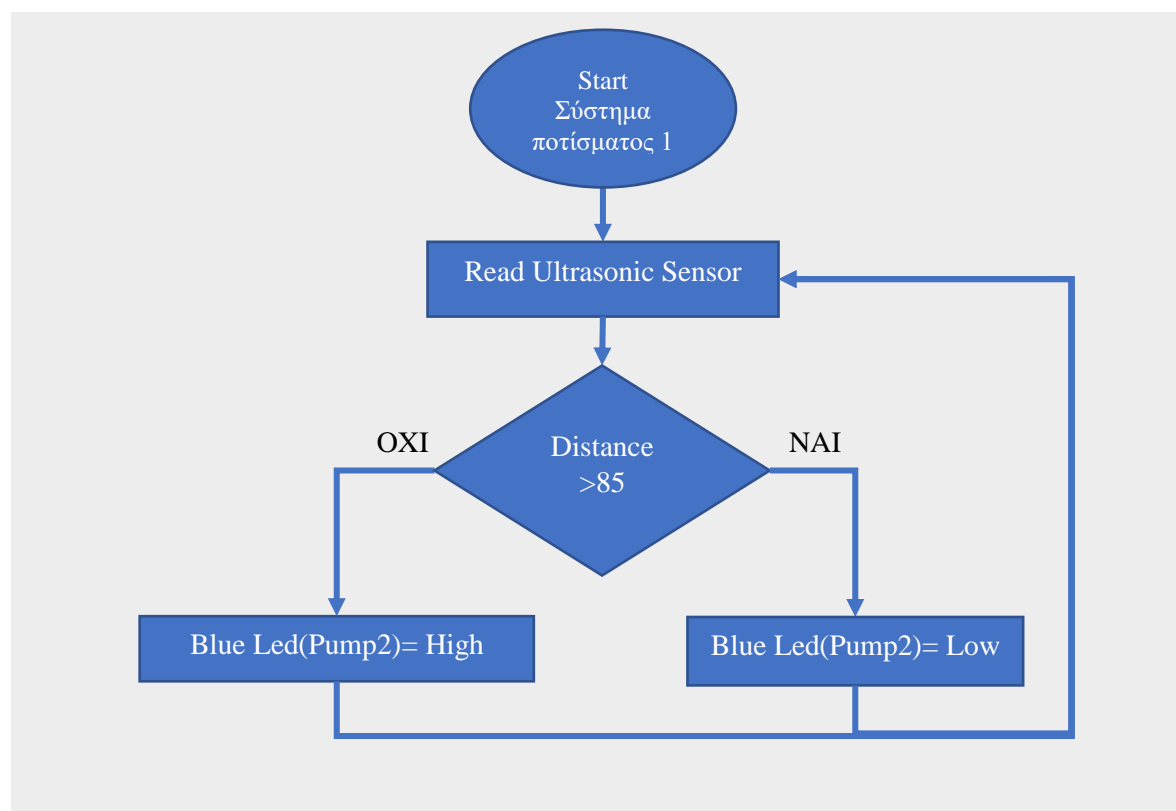
Περιγραφή σεναρίου:

Το σύστημα ποτίσματος αποτελείται από 2 μέρη. Το πρώτο μέρος του συστήματος (Σύστημα Ποτίσματος 1) ελέγχει την στάθμη του νερού στο δοχείο στα επιθυμητά επίπεδα, ενώ το δεύτερο μέρος του συστήματος (Σύστημα Ποτίσματος 2) είναι υπεύθυνο για το πότισμα των φυτών. Το πότισμα γίνεται μόνο εάν είναι μέρα [89], και εάν το επίπεδο νερού στο δοχείο είναι στην κατάλληλη στάθμη, ώστε να μην τραβάει αέρα η αντλία και εάν η υγρασία του χώματος δεν επαρκεί για το φυτό.

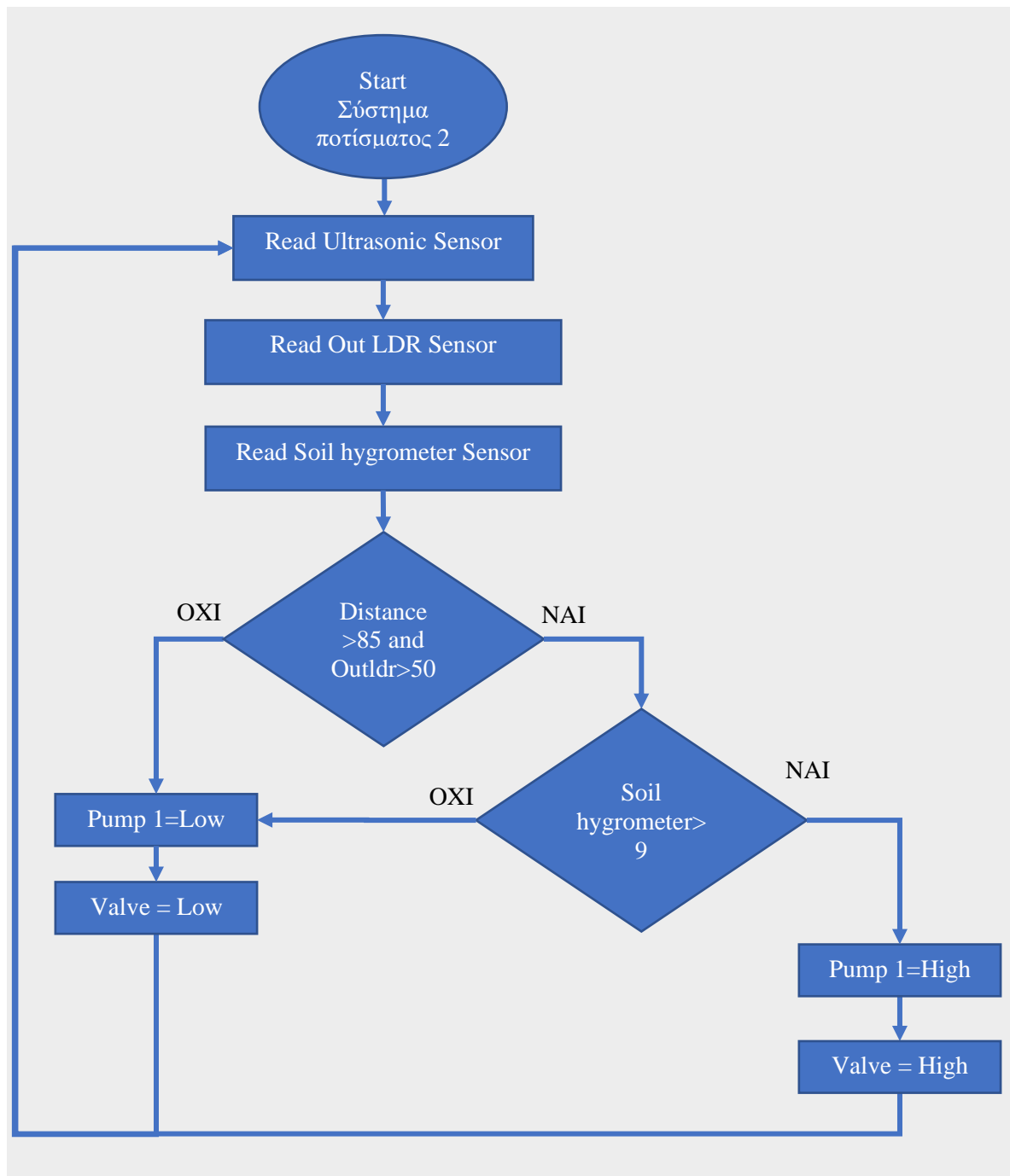
Αρχικά το σύστημα ποτίσματος 1 διαβάζει το επίπεδο του νερού από τον αισθητήρα HC-SR04 Ultrasonic Module Distance και εάν το ποσοστό νερού στο δοχείο είναι κάτω από 70% ανοίγει η αντλία (Pump2 (LED Blue)) που γεμίζει την δεξαμενή με νερό.

Υστέρα το σύστημα ποτίσματος 2 διαβάζει την εξωτερική φωτεινότητα με τον αισθητήρα LDR, το επίπεδο νερού του δοχείου, την υγρασία του χώματος και εάν το ποσοστό φωτός είναι πάνω από 50%, εάν το επίπεδο νερού στο δοχείο είναι πάνω από το 60% και εάν είναι κάτω από 9% η υγρασία του χώματος ενεργοποιείται η αντλία (pump1) και ηλεκτρονική βαλβίδα ώστε να ξεκινάει το πότισμα.

Παρακάτω ακολουθεί το λογικό διάγραμμα ροής του σεναρίου ([Σχήμα 5.2](#) και [Σχήμα 5.3](#)).



Σχήμα 5.2 Το λογικό διαγραμμα από το σύστημα ποτίσματος 1.



Σχήμα 5.3 Το λογικο διαγραμμα από το σύστημα ποτίσματος.

Ακολουθεί ο κώδικας Arduino για το σενάριο αυτοματισμού ελέγχου ποτίσματος.

```

/*-----( Import needed libraries )-----*/
//library Ultrasonic Sensor HC-SR04
#include <HCSR04.h>
//attach pin D48 Arduino to pin Trig of HC-SR04
#define SEN_UltraSonicTrigPin 48
// attach pin D49 Arduino to pin Echo of HC-SR04
#define SEN_UltraSonicEchoPin 49
//pin for Out LDR Sensor
  
```



```

unsigned int SEN_LdrOut = A9;
int RELAY_Pump1 = 30; //Relay of real Pump 1 for watering plant
int RELAY_ValveWater = 32; //Relay Electric valve for watering
int LED_Pump2 = 45; //led blue pump for filling a tank
//initialization class HCSR04 (trig pin , echo pin)
HCSR04 hc(SEN_UltraSonicTrigPin, SEN_UltraSonicEchoPin);
/*-----( Declare Variables )-----*/
int Val_distance; // variable for the distance measurement
// variable for the distance per cent measurement
int Val_distancePerCent;
int Val_OutLdr = 0; // variable for the out LDR data
// Xbee sensor variables for frame
unsigned int analogMSB1 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB1 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_SoilHygro = 0; // variable for the soil hygrometer data
unsigned int analogMSB2 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB2 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_InLdr = 0; // variable for the LDR in data
unsigned int analogMSB3 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB3 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_Mq07Gas = 0; // variable for the Mq-07 data
//-----Set the preference Constants
int My_Val_Max_distancePerCent=90; // μέγιστη τιμή νερού στο δοχείο
int My_Val_Min_distancePerCent=60; // ελάχιστη τιμή νερού στο δοχείο
int My_Val_Max_OutLdr=90; // μέγιστη τιμή εξωτερικής φωτεινότητας
int My_Val_Min_OutLdr=50; // ελάχιστη τιμή εξωτερικής φωτεινότητας
//he have WATER Give the desired soil moisture value
int My_Val_Max_SoilHygro=20;
// don't have WATER Give the desired soil moisture value
int My_Val_Min_SoilHygro=9;
void setup() {
  Serial.begin(9600); //start serial to 9600 baud
  //print some text in Serial Monitor
  Serial.println(F("Ultrasonic Sensor HC-SR04 Test!!"));
  pinMode(RELAY_ValveWater, OUTPUT); // set the pin mode to OUTPUT
  pinMode(RELAY_Pump1, OUTPUT); // set the pin mode to OUTPUT
  pinMode(LED_Pump2, OUTPUT); // set the pin mode to OUTPUT

  pinMode(SEN_UltraSonicEchoPin, INPUT); //Sets the echoPin as an INPUT
  pinMode(SEN_UltraSonicTrigPin, OUTPUT); //Sets the trigPin as an OUTPUT
  digitalWrite(LED_Pump2, LOW ); // set the pin to LOW
  digitalWrite(RELAY_Pump1, LOW ); // set the pin to LOW
  digitalWrite(RELAY_ValveWater, LOW ); // set the pin to LOW
}
void loop() {
  Read_Sen_UltraSonic_Loop();
  Read_Sen_OutLDR_Loop();
  Read_RemoteXbee_Loop();

  System_Water_Loop();
  Print_Serial_loop();
}
//----Read the distance at tank-----
void Read_Sen_UltraSonic_Loop() {
  Val_distance = hc.dist();
  //Mapping των αισθητήρα ultrasonic για την στάθμη νερού στο δοχείο.
  //Το δοχείο είναι 11 εκατοστά αρά τότε θα έχει 0% νερό
  //ενώ εάν είναι 1 εκ. τότε έχουμε 100% νερό μέσα στο δοχείο.
  Val_distancePerCent = map(Val_distance, 11, 1, 0, 100);
}
//----Read the value of ldr out of greenhouse
void Read_Sen_OutLDR_Loop() {

```

```

//Read the value of LDR out of greenhouse
Val_OutLdr = analogRead(SEN_LdrOut);
// αντιστοίχιση των Byte σε επί τοις εκατό
Val_OutLdr = map(Val_OutLdr, 400, 1030, 0, 100);
}
//----Read the value of Router Xbee-----
void Read_RemoteXbee_Loop() {
//checks for Xbee
// έλεγχος εάν το frame έχει τουλάχιστον 29 bits
if (Serial.available() >= 29) {
if (Serial.read() == 0x7E) { // έλεγχος εάν έχει start byte 7E
// Απόρριψη των πρώτων 21 byte
for (int i = 1; i < 21; i++) {
byte discardbyte = Serial.read();
}
//τα Πρώτα byte είναι η τιμή του Soil hygrometer
//Read the most significant bit of the analog data
analogMSB1 = Serial.read();
analogLSB1 = Serial.read(); // Read least significant bit
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα byte
Val_SoilHygro = analogLSB1 + (analogMSB1 * 256);
// αντιστοίχιση των bytes σε επί τοις εκατό
Val_SoilHygro = map(Val_SoilHygro, 0, 1030, 100, 0);
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα byte
//τα Επόμενα bytes είναι η τιμή του LDR
// Read the most significant bit of the analog data
analogMSB2 = Serial.read();
analogLSB2 = Serial.read();// Read least significant bit
Val_InLdr = analogLSB2 + (analogMSB2 * 256);
// αντιστοίχιση των bytes σε επί τοις εκατό
Val_InLdr = map(Val_InLdr, 0, 1030, 0, 100);
//τα τελευταία bytes είναι η τιμή το Gas Sensor Mq07
// Read the most significant bit of the analog data
analogMSB3 = Serial.read();
analogLSB3 = Serial.read();// Read least significant bit
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα byte
Val_Mq07Gas = analogLSB3 + (analogMSB3 * 256);
// αντιστοίχιση των bytes σε επί τοις εκατό
Val_Mq07Gas = map(Val_Mq07Gas, 0, 1030, 0, 100);
} }}
// Control the water system
void System_Water_Loop(){
// εάν το επίπεδο νερού είναι χαμηλότερη από 80 τότε
if (Val_distancePerCent < 80 ) {
// να ανάψει η αντλία 2 δηλαδή το LED για να γεμίσει το δοχείο με νερό
digitalWrite(LED_Pump2,HIGH );
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("PUMP2=ON deksmeni den exei arketo nero"));
}else{ // εάν το επίπεδο είναι υψηλότερο από 80 τότε
// η αντλία που εξομοιώνετε με το μπλε led θα είναι σβηστη
digitalWrite(LED_Pump2, LOW);
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("PUMP2=OFF h deksameni einai gemati"));
}
// εάν το επίπεδο του νερού είναι υψηλότερο από 60 και η τιμή της
//εξωτερική φωτεινότητας είναι υψηλότερη από 50 δηλαδή είναι μέρα τότε
//έλεγχε εάν χεριάζετε νερό το φυτό
if(Val_distancePerCent > My_Val_Min_distancePerCent &&
Val_OutLdr > My_Val_Min_OutLdr){
Serial.println("He have water | the out light is up 50->einai mera");
//εάν η τιμή του αισθητήρα soil hygrometer είναι μεγαλύτερη από 9 τότε
//Το φυτό χρειάζεται νερό και

```

```

if (Val_SoilHygro < My_Val_Min_SoilHygro){
  digitalWrite(RELAY_Pump1, LOW); // η αντλία είναι on και
  digitalWrite(RELAY_ValveWater, LOW); // η βάννα είναι on
  //Εμφάνιση της κατάστασης στην σειριακή οθόνη
  Serial.println("Den exei nero pump1=on");
}else{ // εάν είναι μικρότερη του 9 τότε το φυτό έχει νερό και
  digitalWrite(RELAY_Pump1, HIGH); // η αντλία είναι off και
  digitalWrite(RELAY_ValveWater, HIGH); // η βάννα είναι OFF
  //Εμφάνιση της κατάστασης στην σειριακή οθόνη
  Serial.println("Exei nero pump1=off");
}
//εάν το επίπεδο του νερού είναι χαμηλότερη από 60 και η τιμή της
//εξωτερικής φωτεινότητας είναι χαμηλότερη από 50 δηλαδή είναι βραδύ
//τότε
}else{
  digitalWrite(RELAY_Pump1, HIGH); // η αντλία είναι κλειστέ OFF και
  // η βάννα κλειστέ
  digitalWrite(RELAY_ValveWater, HIGH);
  //Εμφάνιση της κατάστασης στην σειριακή οθόνη
  Serial.println("Einai nixta pump1=off");}
// Displays the value on the Serial Monitor
void Print_Serial_loop() {
  Serial.print(F("LDR OUT Value: "));
  Serial.print(Val_OutLdr);
  Serial.println(F("%"));

  Serial.print(F("Distance: "));
  Serial.print(Val_distancePerCent);
  Serial.println(F("%"));

  Serial.print(F("Distance: "));
  Serial.print(Val_distance);
  Serial.println(F("cm"));

  Serial.print(F("Soil moisture: "));
  Serial.print(Val_SoilHygro);
  Serial.println(F("%"));}

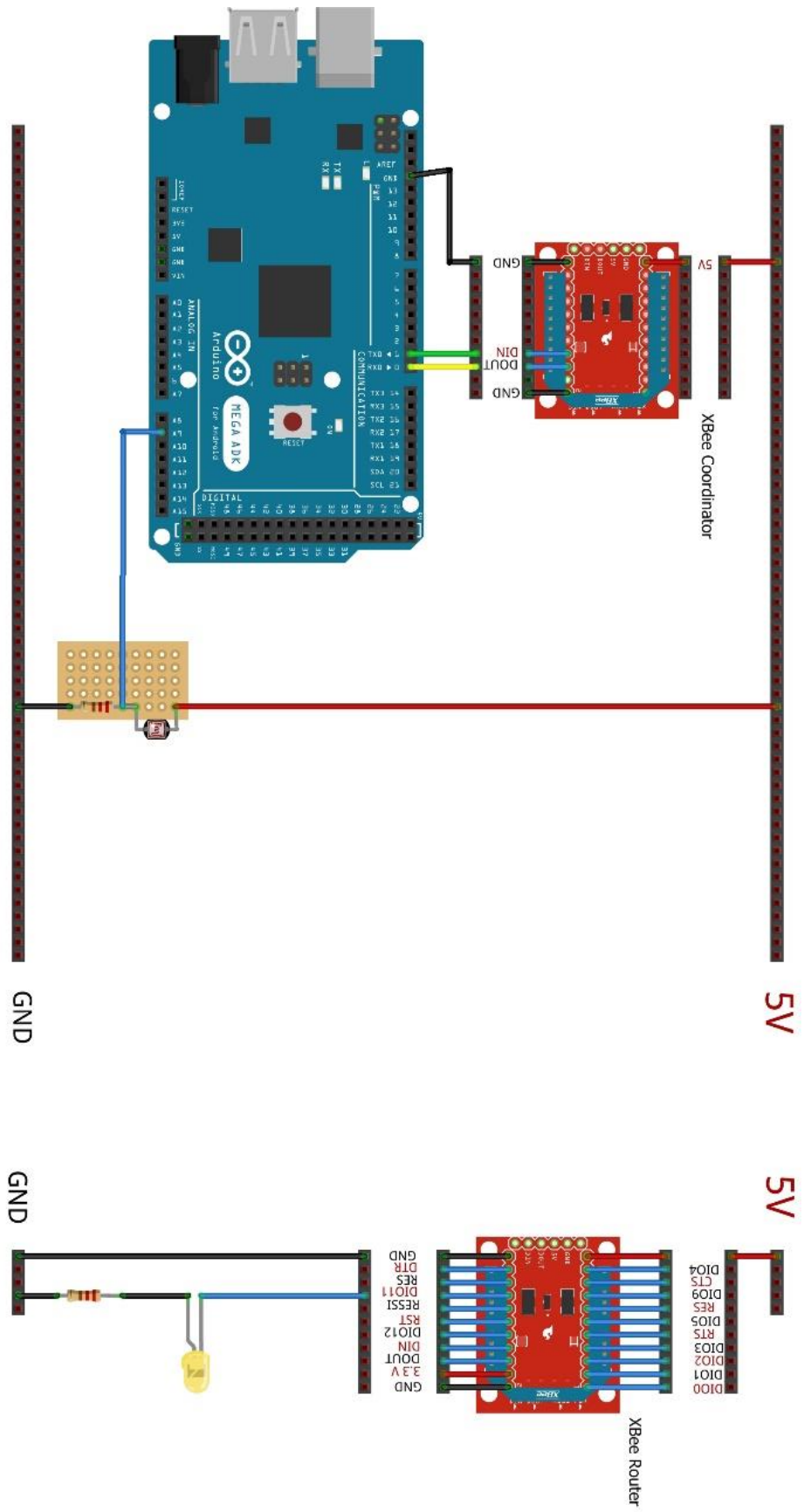
```

## 5.4 Περιγραφή σεναρίου αυτοματισμού ελέγχου της ηλιακής ακτινοβολίας

Το σενάριο ελέγχου της ηλιακής ακτινοβολίας αποτελείται από:

- 1 Xbee ως Coordinator του οποίου οι ακίδες του θα συνδέονται με το Arduino ως εξής ακίδα D1-TX του Arduino με το DIN του Xbee και η ακίδα D0-RX του Arduino με το DOUT του Xbee.
- 1 Xbee ως Router στον οποίο συνδέεται το κίτρινο LED που εξομοιώνει το μηχανισμό σκίασης του θερμοκηπίου.
- 1 κίτρινο LED που εξομοιώνει ένα μηχανισμό σκίασης του θερμοκηπίου που θα συνδέεται στην ακίδα DIO11 του Xbee Router.
- 1 Αισθητήρα Φωτοαντίστασης LDR στο Εξωτερικό του θερμοκηπίου που θα συνδέεται στην ακίδα A9 του Arduino mega.

Στο [Κύκλωμα 5.3](#) παρουσιάζεται ο τρόπος διασύνδεσης των εξαρτημάτων για την υλοποίηση του σεναρίου.



fritzing

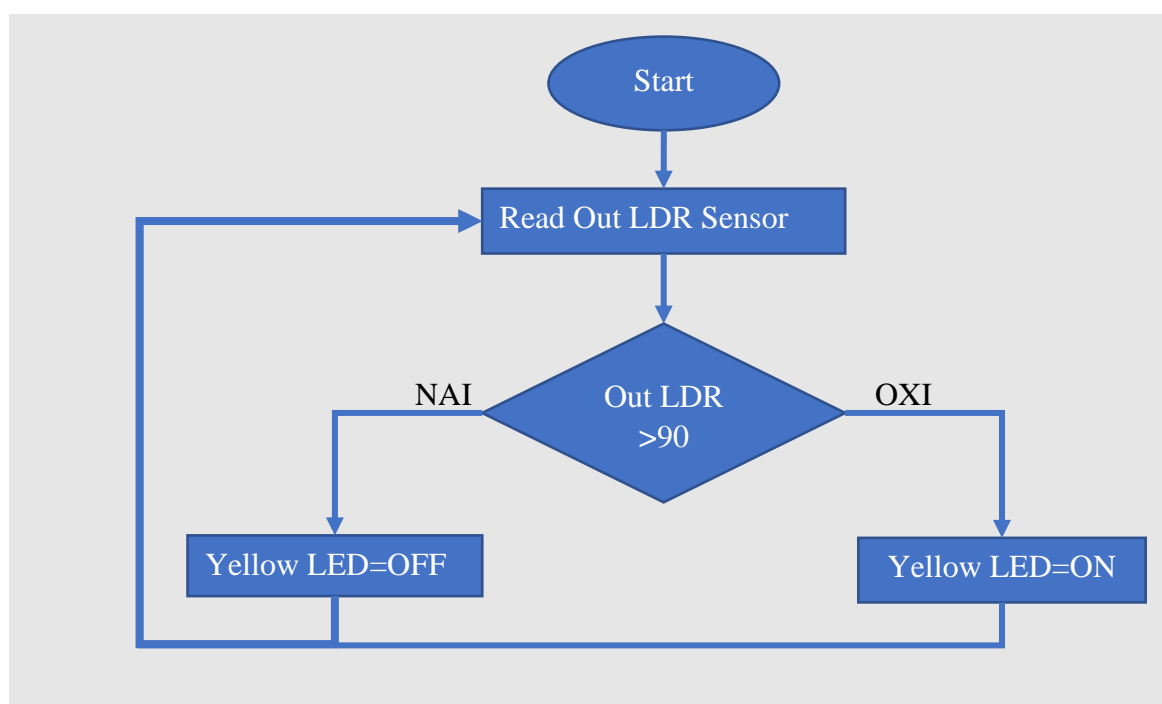
Κύκλωμα 5.3 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου της ηλιακής ακτινοβολίας.

Περιγραφή σεναρίου:

Το σύστημα σκίασης προστατεύει τα φυτά από τις ακτίνες του ηλίου, όταν υπάρχει μεγάλη ηλιοφάνεια και βοηθάει στην μείωση της θερμοκρασίας.

Αρχικά διαβάζεται η εξωτερική φωτεινότητα με τον αισθητήρα LDR και με βάση την τιμή της ενεργοποιείται το σύστημα σκίασης, όταν η εξωτερική φωτεινότητα είναι πάνω από 90%.

Παρακάτω ακολουθεί το λογικό διάγραμμα ροής του σεναρίου (Σχήμα 5.4).



Σχήμα 5.4 Το λογικό διάγραμμα από του σεναρίου αυτοματισμού ελέγχου της ηλιακής ακτινοβολίας.

Ακολουθεί ο κώδικας Arduino για το σενάριο αυτοματισμού ελέγχου της ηλιακής ακτινοβολίας.

```
unsigned int SEN_LdrOut = A9; //pin for Out LDR Sensor
int Val_OutLdr = 0; // variable for the out LDR data
// Xbee sensor variables for frame
unsigned int analogMSB1 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB1 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_SoilHygro = 0; // variable for the soil hygrometer data
unsigned int analogMSB2 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB2 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_InLdr = 0; // variable for the LDR in data
unsigned int analogMSB3 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB3 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_Mq07Gas = 0; // variable for the Mq-07 data
int My_Val_Max_OutLdr=90; // μέγιστη τιμή εξωτερικής φωτεινότητας
int My_Val_Min_OutLdr=50; // ελάχιστη τιμή εξωτερικής φωτεινότητας
void setup() {
  Serial.begin(9600); //start serial to 9600 baud
```

```

pinMode(SEN_LdrOut, INPUT); // set the pin mode to input
setRem_LED_RED_Org_ALERT_DIO4(0x4); //set the Xbee pin D4 to LOW
setRem_LED_Yellow_MOV_Sunshade_DIO11(0x4); //set the Xbee pin to LOW
setRem_LED_Blue_Blue_ForggerSpraySytem_DIO12(0x4); //Set the to LOW
}
void loop() {
  Read_Sen_OutLDR_Loop();
  Print_Serial_loop();
  System_SunsHade_Loop();
}
void Read_Sen_OutLDR_Loop() {
  //Read the value of ldr out of greenhouse
  Val_OutLdr = analogRead(SEN_LdrOut);
  //Αντιστοίχιση των Bytes σε επί τοις εκατό
  Val_OutLdr = map(Val_OutLdr, 400, 1030, 0, 100);
}
//----Read the value of Router Xbee-----
void Read_RemoteXbee_Loop() {
  //checks for Xbee
  //έλεγχος εάν το frame έχει τουλάχιστον 29 bits
  if (Serial.available() >= 29) {
    if (Serial.read() == 0x7E) { // έλεγχος εάν έχει το start byte 7E
      // Απόρριψη των πρώτων 21 byte
      for (int i = 1; i < 21; i++) {
        byte discardbyte = Serial.read();
      }
      //τα Πρώτα bytes είναι η τιμή του Soil hygrometer
      // Read the most significant bit of the analog data
      analogMSB1 = Serial.read();
      analogLSB1 = Serial.read(); // Read least significant bit
      //μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα //byte
      Val_SoilHygro = analogLSB1 + (analogMSB1 * 256);
      //Αντιστοιχίσει των Bytes σε επί τοις εκατό
      Val_SoilHygro = map(Val_SoilHygro, 0, 1030, 100, 0);
      //τα Επόμενα Bytes είναι η τιμή του LDR
      // Read the most significant bit of the analog data
      analogMSB2 = Serial.read();
      analogLSB2 = Serial.read(); // Read least significant bit
      //μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα //byte
      Val_InLdr = analogLSB2 + (analogMSB2 * 256);
      //Αντιστοιχίσει των Bytes σε επί τοις εκατό
      Val_InLdr = map(Val_InLdr, 0, 1030, 0, 100);
      //τα τελευταία byte είναι η τιμή το Mq07
      //Read the most significant bit of the analog data
      analogMSB3 = Serial.read();
      analogLSB3 = Serial.read(); // Read least significant bit
      //μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα //byte
      Val_Mq07Gas = analogLSB3 + (analogMSB3 * 256);
      //Αντιστοιχίσει των Bytes σε επί τοις εκατό
      Val_Mq07Gas = map(Val_Mq07Gas, 0, 1030, 0, 100);
    }
  }
}
//CONTROL THE SUNLIGHT
void System_SunsHade_Loop() {
  //εάν η τιμή του LDR είναι μεγαλύτερη από 90% τότε
  if (Val_OutLdr > My_Val_Max_OutLdr) {
    //Κάνε το value=0x5 ώστε το yellow led να είναι ON
    setRem_LED_Yellow_MOV_Sunshade_DIO11(0x5);
    //Εμφάνιση της κατάστασης στην σειριακή οθόνη
    Serial.println(F("Yellow Led=ON"));
  } else { // αλλιώς θα είναι κάτω από 90% και τότε
    //Κάνε το value=0x4 ώστε το yellow led να είναι Off
    setRem_LED_Yellow_MOV_Sunshade_DIO11(0x4);
  }
}

```



```

//Εμφάνιση της κατάστασης στην σειριακή οθόνη
    Serial.println(F("Yellow Led=OFF"));
}
}
// Displays the value on the Serial Monitor
void Print_Serial_loop() {
Serial.print(F("LDR OUT Value: "));
Serial.print(Val_OutLdr);
Serial.println(F("%"));
}
// create the frame to send to router
// πλαίσιο για τον ελέγχου του κόκκινου LED και του buzzer που
//εξομοιώνει το σύστημα ειδοποίησης
void setRem_LED_RED_Org_ALERT_DIO4(char value) {
    Serial.write(0x7E); //0
    Serial.write(byte(0x0)); //1
    Serial.write(0x10); //2
    Serial.write(0x17); // 3
    Serial.write(byte(0x0)); //4
    Serial.write(byte(0x0)); //5
    Serial.write(byte(0x0)); //6
    Serial.write(byte(0x0)); //7
    Serial.write(byte(0x0)); //8
    Serial.write(byte(0x0)); //9
    Serial.write(byte(0x0)); //10
    Serial.write(0xFF); //12
    Serial.write(0xFF); // 13
    Serial.write(0xFF); //14
    Serial.write(0xFE); //15
    Serial.write(0x02); //16
    Serial.write('D'); //17
    Serial.write('4'); //18
    Serial.write(value); // 19
    long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'D' + '4' + value;
    Serial.write(0xFF - (sum & 0xFF));
}
//πλαίσιο για τον ελέγχου του κίτρινου LED που εξομοιώνει το σύστημα
//σκίασης
void setRem_LED_Yellow_MOV_Sunshade_DIO11(char value) {
    Serial.write(0x7E); //start byte
    Serial.write(byte(0x0)); // high part of length (always zero)
//low part of length (the number of bytes that follow, not including
//check
    Serial.write(0x10);
    Serial.write(0x17); //0x17 is a remote AT command
    Serial.write(byte(0x0)); //frame id set to zero for no reply
//Id of recipient, or use 0xFFFF for broadcast
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(0xFF); // 0xFF for broadcast
    Serial.write(0xFF); // 0xFF for broadcast
//16 bit of recipient or 0xFFFFE
    Serial.write(0xFF);
    Serial.write(0xFE);
    Serial.write(0x02); // 0x02 to apply changes immediately on remote
//AT command name in ASCII character
    Serial.write('P');
    Serial.write('1');
}

```

```

// command data in as many bytes as needed
Serial.write(value);
//checksum is all bytes after length bytes
long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '1' + value;
Serial.write(0xFF - (sum & 0xFF)); // calculate the proper checksum
}
// πλαίσιο για τον έλεγχο του μπλε LED που εξομοιώνει το σύστημα
//ψεκασμού ομίχλης νερού
void setRem_LED_Blue_Blue_ForggerSpraySytem_DIO12(char value) {
Serial.write(0x7E);
Serial.write(byte(0x0));
Serial.write(0x10);
Serial.write(0x17);
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(0xFF);
Serial.write(0xFF);
Serial.write(0xFF);
Serial.write(0xFE);
Serial.write(0x02);
Serial.write('P');
Serial.write('2');
Serial.write(value);
long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '2' + value;
Serial.write(0xFF - (sum & 0xFF));
}

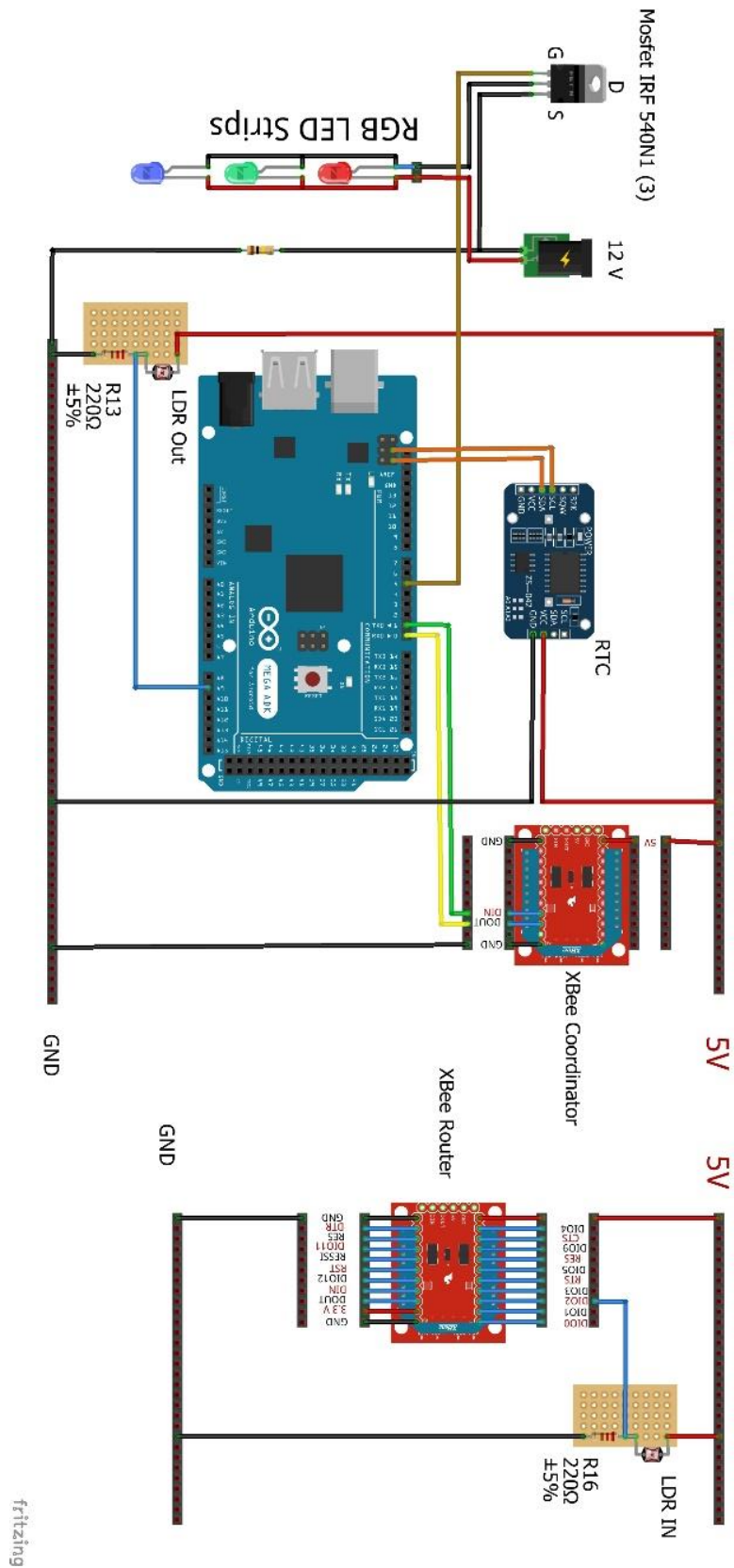
```

## 5.5 Περιγραφή σεναρίου αυτοματισμού επέκτασης φωτισμού

Το σενάριο επέκτασης φωτισμού αποτελείται από:

- 1 Xbee ως Coordinator του οποίου οι ακίδες του συνδέονται με το Arduino ως εξής: ακίδα D1-TX του Arduino με το DIN του Xbee και η ακίδα D0-RX του Arduino με το DOUT του Xbee.
- 1 Xbee ως Router στον οποίο συνδέεται ο εσωτερικός αισθητήρας φωτοαντίστασης LDR.
- 1 Αισθητήρα Φωτοαντίσταση LDR στο Εσωτερικό του θερμοκηπίου συνδέεται στην ακίδα DI02 του Xbee Router
- 1 Αισθητήρα Φωτοαντίστασης LDR στο Εξωτερικό του θερμοκηπίου που συνδέεται στην ακίδα A9 του Arduino mega.
- 2 Rgb Ταινίες LED (RGB Strips) σε σειρά που συνδέεται με το Mosfet-3.
- 1 Mosfet N- Channel 33A IRF540N-(3) που συνδέεται στην ακίδα D5 του Arduino.
- 1 ρολόι πραγματικού χρόνου Real Time Clock (RTC) που συνδέεται στις ακίδες SCL και SDA του Arduino mega όπως φαίνεται στο κύκλωμα παρακάτω.

Στο [Κύκλωμα 5.4](#) παρουσιάζεται ο τρόπος διασύνδεσης των εξαρτημάτων για την υλοποίηση του σεναρίου.

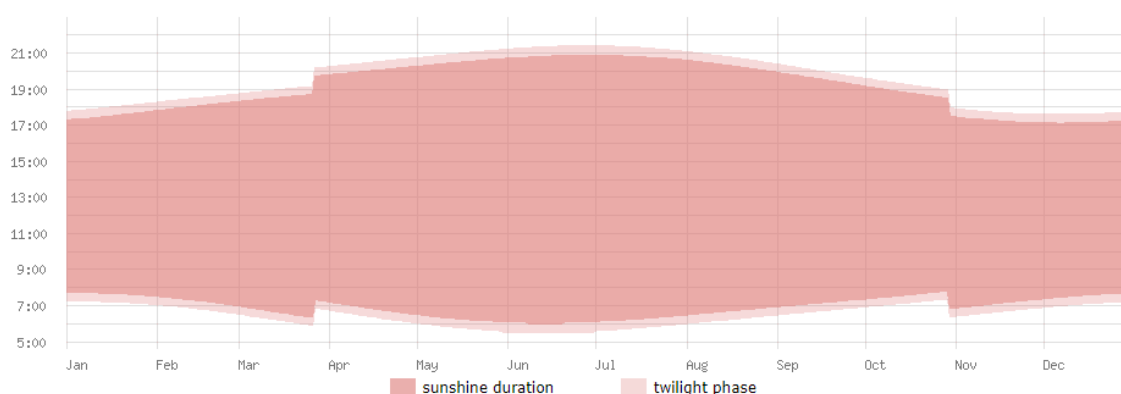


Κύκλωμα 5.4 Συνολική σύνδεση σεναρίου αυτοματισμού επέκτασης φωτισμού.

Περιγραφή σεναρίου:

Στα θερμοκήπια υπάρχει η ανάγκη επέκτασης της διάρκειας και της ποσότητας του φωτός στο εσωτερικό του, η οποία είναι απαραίτητη για την σωστή ανάπτυξη, ανθοφορία και καρποφορία των φυτών. Οπότε τις ημέρες που υπάρχει συννεφιά ή ο ήλιος δύει νωρίτερα ή ανατέλλει αργότερα τα φυτά θα φωτοσυνθέτουν, έτσι υπάρχει η δυνατότητα καλλιέργειας των φυτών όλο το χρόνο. Ο χρήστης μπορεί να ορίσει την περίοδο που χρειάζεται φως μέσω του RTC και ανάλογα με το ποσοστό φωτός που θα διαβάσει ο εσωτερικός αισθητήρας φωτός LDR στο εσωτερικό του θερμοκηπίου, ενεργοποιείται η LED strip.

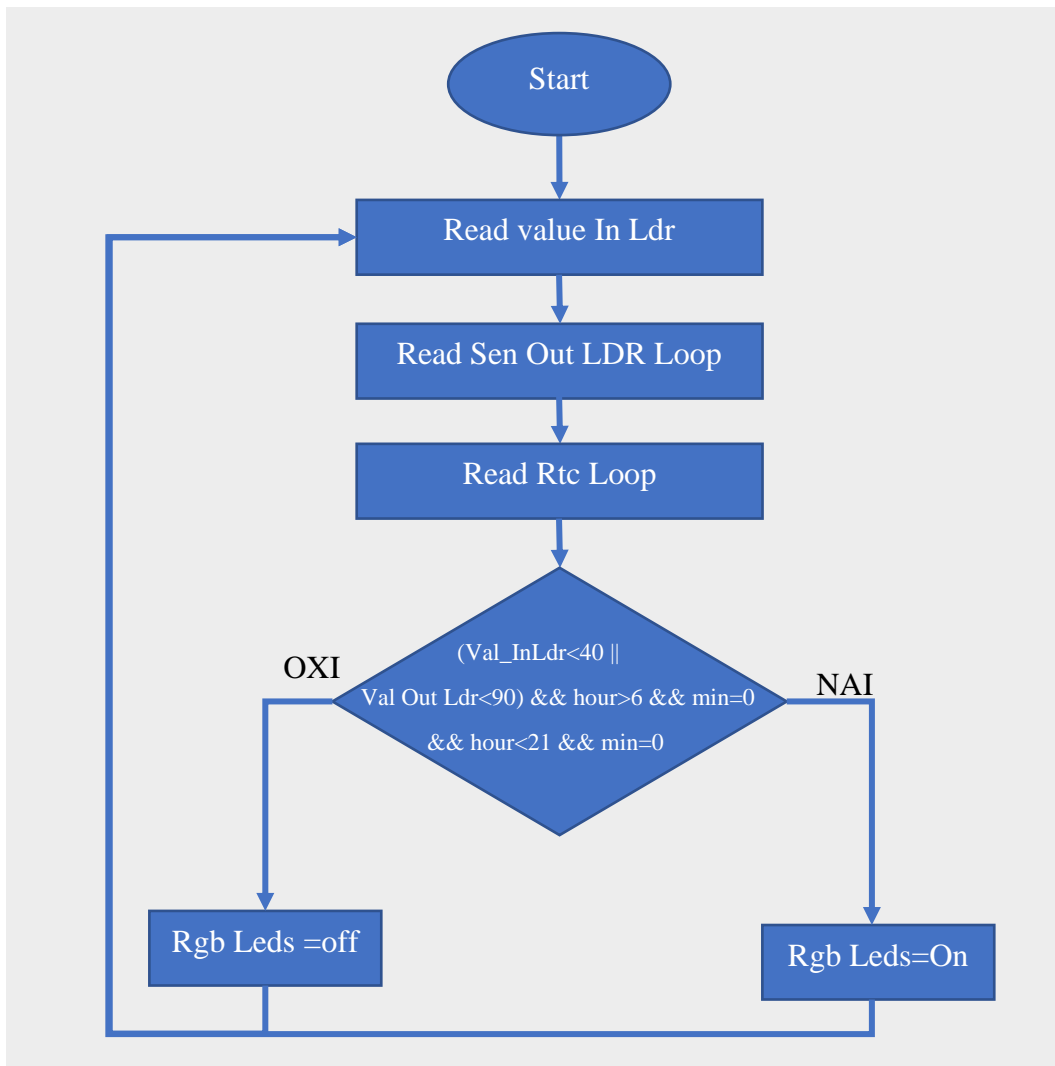
Average length of day in Athens



Εικόνα 5.1 Μέση διάρκεια ημέρας στην Αθήνα. [Πηγή](#)

Αρχικά, διαβάζεται το ποσοστό φωτεινότητας στο εσωτερικό και στο εξωτερικό του θερμοκηπίου και με βάση την χρονική περίοδο που έχει ορίσει ο χρήστης και τις τιμές που διαβάστηκαν από τους αισθητήρες LDR, ενεργοποιείται το σύστημα φωτισμού. Με βάση το παραπάνω σχεδιάγραμμα ([Εικόνα 5.1](#)) το νωρίτερο που ανατείλει ο ήλιος όλον τον χρόνο είναι 07:00 ενώ το αργότερο που θα δύση είναι 21:00, που είναι η χρονική περίοδος που έχει οριστεί από τον χρήστη.

Παρακάτω ακολουθεί το λογικό διάγραμμα ροής του σεναρίου ([Σχήμα 5.5](#)).



Σχήμα 5.5 Σενάριο αυτοματισμού επέκτασης φωτισμού.

Ακολουθεί ο κώδικας Arduino για το σενάριο αυτοματισμού επέκτασης φωτισμού.

```

/*-----( Import needed libraries )-----*/
//Βιβλιοθηκη που επιτρέπει να επικοινωνείτε με συσκευές I2C Mega2560 20
(SDA), 21 (SCL)
#include <Wire.h>
#include <DS3231.h> //library for real time clock Rtc ds3231
/*-----( Declare Constants and Pin Numbers )-----*/
int MOS_RgbLeds = 5; // Gste Mosfet attach to Rgb Led strip
unsigned int SEN_LdrOut = A9; //pin for Out Ldr Sensor
/*-----( Declare objects )-----*/
DS3231 rtc(SDA, SCL);
Time t;
/*-----( Declare Variables )-----*/
// Xbee sensor variables for frame
unsigned int analogMSB1 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB1 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_SoilHygro = 0; //variable for the soil hygrometer data

unsigned int analogMSB2 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB2 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_InLdr = 0; // variable for the LDR in data
  
```

```

unsigned int analogMSB3 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB3 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_Mq07Gas = 0; // variable for the Mq-07 data

int Val_OutLdr = 0; // variable for the out LDR data

int My_Val_Max_OutLdr=90; // μέγιστη τιμή εξωτερικής φωτεινότητας
int My_Val_Min_OutLdr=50; // ελάχιστη τιμή εξωτερικής φωτεινότητας
int My_Val_InLdr=60; //Give the desired light value
// Rtc ds3231
const int My_Val_OnHour = 6; //SET Hour TO ON the rgb led
const int My_Val_OnMin = 00; //SET min TO ON the rgb led
const int My_Val_OffHour = 14; //SET TIME TO OFF the rgb led
const int My_Val_OffMin = 15; //SET min TO OFF the rgb led
void setup() {
  Serial.begin(9600); //start serial to 9600 baud
  rtc.begin(); // Start Rtc dc331 on Arduino
  //Set the INPUT pin mode
  pinMode(SEN_LdrOut, INPUT);

  //Set the Outout pin mode
  pinMode(MOS_RgbLeds, OUTPUT);
}
void loop() {
  System_Light_Loop();
  Read_RemoteXbee_Loop();
  Read_Rtc_Loop();
  Read_Sen_OutLDR_Loop();
  Print_Serial_loop();
}
//---Read the value of ldr out of greenhouse
void Read_Sen_OutLDR_Loop() {
  //Read the value of ldr out of greenhouse
  Val_OutLdr = analogRead(SEN_LdrOut);
  //Αντιστοιχίσει των BYTE σε επι της εκατό
  Val_OutLdr = map(Val_OutLdr, 100, 1023, 0, 100);
}
//---Read the value of Router Xbee-----
void Read_RemoteXbee_Loop() {
  // έλεγχος εάν το frame έχει τουλάχιστον 29 bits
  if (Serial.available() >= 29) {
    // έλεγχος εάν έχει start byte 7E
    if (Serial.read() == 0x7E) {
      // Απόρριψη των πρώτων 21 byte
      for (int i = 1; i < 21; i++) {
        byte discardbyte = Serial.read();
      }
      //τα Πρώτα byte είναι η τιμή του Soil hygrometer
      // Read the most significant bit of the analog data
      analogMSB1 = Serial.read(); // Read the first analog byte data
      analogLSB1 = Serial.read(); // Read the second byte
      //μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
      //byte
      Val_SoilHygro = analogLSB1 + (analogMSB1 * 256); //900- νερό 800
      όχι νερό
      //Αντιστοιχίσει των Byte σε επι τοις εκατό
      Val_SoilHygro = map(Val_SoilHygro, 700, 1000, 100, 0);
      //τα Επόμενα byte είναι η τιμή του LDR
      analogMSB2 = Serial.read(); // Read the first analog byte data
      analogLSB2 = Serial.read(); // Read the second byte
      //μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
      //byte

```



```

Val_InLdr = analogLSB2 + (analogMSB2 * 256);
//Αντιστοιχίσει των Byte σε επι τοις εκατό
Val_InLdr = map(Val_InLdr, 0, 1030, 0, 100);
//τα τελευταία byte είναι η τιμή το Gas Sensor Mq07
analogMSB3 = Serial.read();// Read the first analog byte data
analogLSB3 = Serial.read();// Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
Val_Mq07Gas = analogLSB3 + (analogMSB3 * 256);
//Αντιστοιχίσει των Byte σε επι τοις εκατό
Val_Mq07Gas = map(Val_Mq07Gas, 0, 1030, 0, 100);
} }}
//----Read the value of Real Time Clock-----
void Read_Rtc_Loop() {
    t = rtc.getTime();
}
//control the inside Light System
void System_Light_Loop(){
//εάν η τιμή του εσωτερικού LDR είναι μικρότερη από 40 ή
//η τιμή εξωτερικού είναι μικρότερη από 50 και
// η ώρα είναι πάνω από 6 και
// τα λεπτά είναι πάνω από 0 και
// η ώρα είναι κάτω από 22 και
// τα λεπτά είναι κάτω από 0 τότε
    if ((Val_InLdr<=My_Val_InLdr || Val_OutLdr < My_Val_Min_OutLdr) &&
t.hour >= My_Val_OnHour && t.min >= My_Val_OnMin && t.hour <=
My_Val_OffHour && t.min <= My_Val_OffMin) {
        // οι λεντ ταινίες είναι ON
        digitalWrite(MOS_RgbLeds, HIGH);
        //εμφάνιση κατάστασης στην σειριακή οθόνη
        Serial.println(("RGB=ON"));
    }else { //εάν δεν ισχύει κάτι από αυτά τότε
        // οι λεντ ταινίες είναι OFF
        digitalWrite(MOS_RgbLeds, LOW);
        //εμφάνιση κατάστασης στην σειριακή οθόνη
        Serial.println(F("RGB=OFF"));
    }
}
// Displays the value on the Serial Monitor
void Print_Serial_loop() {
//εμφάνιση στην σειριακή οθόνη την τιμή του LDR
    Serial.print(F("LDR OUT Value: ")); Serial.print(Val_OutLdr);
    Serial.println(F("%"));
    Serial.print(F("Ldr IN Value: ")); Serial.print(Val_InLdr);
    Serial.println(F("%"));
//εμφάνιση στην σειριακή οθόνη την ημερομηνία
    Serial.print("Date : ");
    Serial.print(t.date); Serial.print("/"); Serial.print(t.mon);
Serial.print("/"); Serial.print(t.year); Serial.print(" ");
//εμφάνιση στην σειριακή οθόνη την ώρα
    Serial.print("Hour : ");
    Serial.print(t.hour); Serial.print(":"); Serial.print(t.min);
Serial.print(":"); Serial.println(t.sec);
    Serial.println(F("*****"));
}

//=====
// create the frame to send to router
// πλαίσιο για των ελέγχου του κόκκινου LED και του buzzer που
//εξομοιώνει το σύστημα ειδοποίησης
void setRem_LED_RED_Org_ALERT_DIO4(char value) {
    Serial.write(0x7E); //0

```

```

Serial.write(byte(0x0)); //1
Serial.write(0x10); //2
Serial.write(0x17); // 3
Serial.write(byte(0x0)); //4
Serial.write(byte(0x0)); //5
Serial.write(byte(0x0)); //6
Serial.write(byte(0x0)); //7
Serial.write(byte(0x0)); //8
Serial.write(byte(0x0)); //9
Serial.write(byte(0x0)); //10
Serial.write(0xFF); //12
Serial.write(0xFF); // 13
Serial.write(0xFF); //14
Serial.write(0xFE); //15
Serial.write(0x02); //16
Serial.write('D'); //17
Serial.write('4'); //18
Serial.write(value); // 19
long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'D' + '4' + value;
Serial.write(0xFF - (sum & 0xFF));
}
//πλαίσιο για των ελέγχου του κίτρινου LED που εξομοιώνει το σύστημα
//σκίασης
void setRem_LED_Yellow_MOV_Sunshade_DIO11(char value) {
    Serial.write(0x7E); //start byte
    Serial.write(byte(0x0)); // high part of length (always zero)
    //low part of length (the number of bytes that follow, not including
    //check
    Serial.write(0x10);
    Serial.write(0x17); //0x17 is a remote AT command
    Serial.write(byte(0x0)); //frame id set to zero for no reply
    //Id of recipient, or use 0xFFFF for broadcast
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(0xFF); // 0xFF for broadcast
    Serial.write(0xFF); // 0xFF for broadcast

    //16 bit of recipient or 0xFFFFE
    Serial.write(0xFF);
    Serial.write(0xFE);
    Serial.write(0x02); // 0x02 to apply changes immediately on remote

    //AT command name in ASCII character
    Serial.write('P');
    Serial.write('1');

    // command data in as many bytes as needed
    Serial.write(value);

    //checksum is all bytes after length bytes
    long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '1' + value;
    Serial.write(0xFF - (sum & 0xFF)); // calculate the proper checksum
}
// πλαίσιο για των έλεγχο του μπλε LED που εξομοιώνει το σύστημα
//ψεκασμού ομίχλης νερού
void setRem_LED_Blue_Blue_ForggerSpraySystem_DIO12(char value) {
    Serial.write(0x7E);
    Serial.write(byte(0x0));

```

```

Serial.write(0x10);
Serial.write(0x17);
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(0xFF);
Serial.write(0xFF);
Serial.write(0xFF);
Serial.write(0xFE);
Serial.write(0x02);
Serial.write('P');
Serial.write('2');
Serial.write(value);
long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '2' + value;
Serial.write(0xFF - (sum & 0xFF));
}

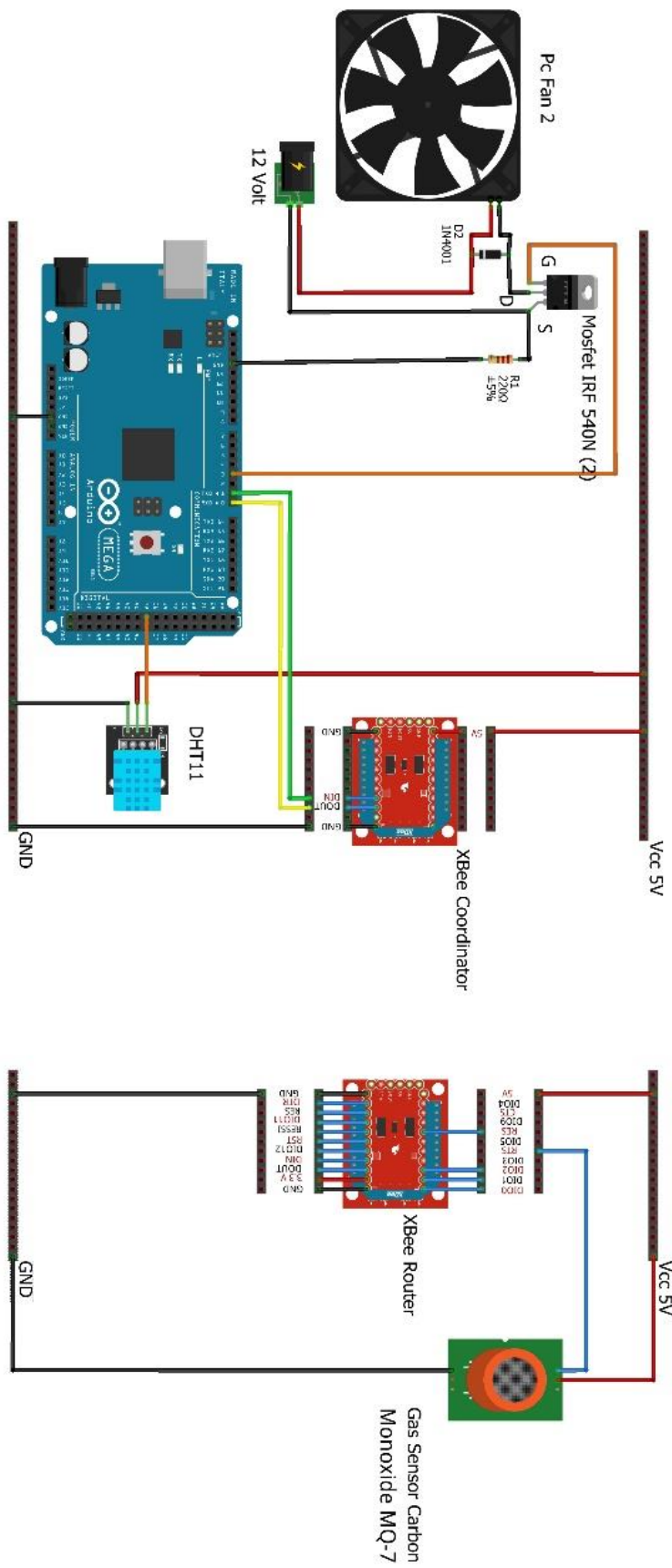
```

## 5.6 Περιγραφή σεναρίου αυτοματισμού ελέγχου εξαερισμού

Το σενάριο ελέγχου εξαερισμού αποτελείται από:

- 1 Xbee ως Coordinator του οποίου οι ακίδες του συνδέονται με το Arduino ως εξής: ακίδα D1-TX του Arduino με το DIN του Xbee και η ακίδα D0-RX του Arduino με το DOUT του Xbee.
- 1 Xbee ως Router στον οποίου συνδέεται ο αισθητήρας αερίου.
- 1 αισθητήρα διοξειδίου του άνθρακα MQ-07 (Gas Sensor Carbon Monoxide MQ-7) που συνδέεται στην ακίδα DI03 του Xbee Router
- 1 αισθητήρα θερμοκρασίας και υγρασίας DHT11 που συνδέεται στην ακίδα D38 του Arduino mega.
- 1 ανεμιστήρα 12 volt (Pc Fan 2) για εξομοίωση του συστήματος αερισμού που συνδέεται με το Mosfet 2.
- 1 Mosfet N- Channel 33A IRF540N-(2) που συνδέεται στην ακίδα D3 του Arduino mega.
- 1 δίοδο (D2) 1N4001 που συνδέεται όπως φαίνεται στο παρακάτω Κύκλωμα 5.5.
- 1 αντίσταση 220 Ω

Στο [Κύκλωμα 5.5](#) παρουσιάζεται ο τρόπος διασύνδεσης των εξαρτημάτων για την υλοποίηση του σεναρίου.



fritzing

Κύκλωμα 5.5 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου εξαερισμού.

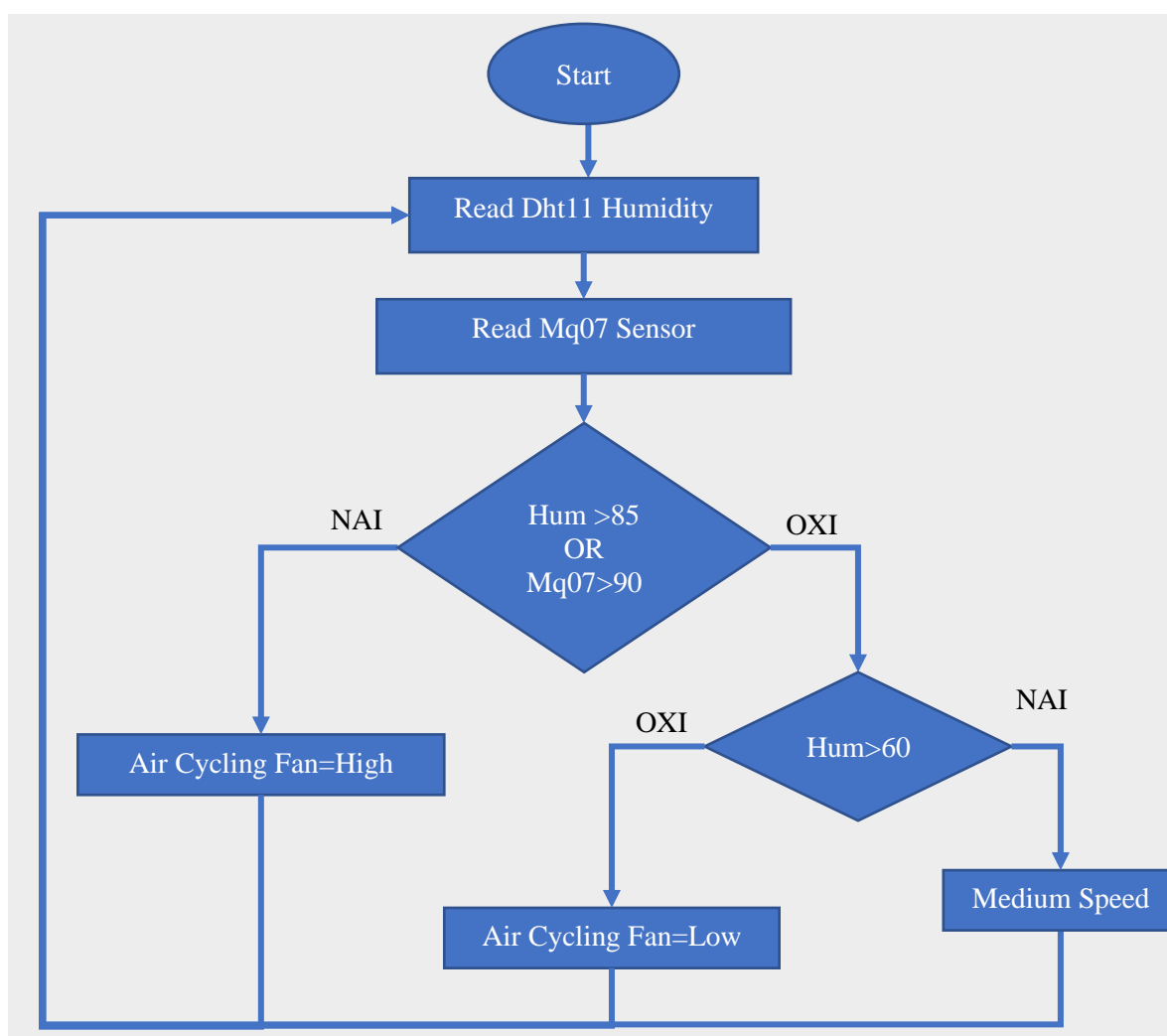
Περιγραφή σεναρίου:

Το σενάριο ελέγχου εξαερισμού μειώνει την υγρασία του χώρου και αερίζει τον χώρο από τον καπνό σε περίπτωση που έχει προκληθεί φωτιά. Αυτό επιτυγχάνεται ενεργοποιώντας τον ανεμιστήρα μέσω του Mosfet.

Αρχικά διαβάζεται η υγρασία από τον αισθητήρα DHT11 και από τον αισθητήρα MQ-07 το διοξείδιό του άνθρακα

- εάν η υγρασία είναι πάνω από 80% ή η τιμή του αισθητήρα διοξειδίου του άνθρακα MQ-07 είναι πάνω από 90% ενεργοποιείται ο ανεμιστήρας στις full στροφές.
- εάν η υγρασία είναι πάνω από 60%, ο ανεμιστήρας δουλεύει σε ελάχιστες στροφές.
- Εάν η υγρασία είναι κάτω από 65% ή η τιμή του αισθητήρα διοξειδίου του άνθρακα MQ-07 είναι κάτω από 90%, ο ανεμιστήρας είναι απενεργοποιημένος.

Παρακάτω ακολουθεί το λογικό διάγραμμα ροής του σεναρίου ([Σχήμα 5.6](#)).



Σχήμα 5.6 Σεναρίου αυτοματισμού ελέγχου εξαερισμού.

Ακολουθεί ο κώδικας Arduino για το σενάριο αυτοματισμού ελέγχου εξαερισμού.

```
/*-----( Import needed libraries )-----*/
#include "DHT.h" //Βιβλιοθήκη για αισθητήρα Dht11

/*-----( Declare Constants and Pin Numbers )-----*/
const int MOS_AirRecyclingFan = 2; //Gate mosfet 1 attach to d5 pin
arduino for Air Recycling area greenhouse

#define SEN_Dht11 38 // Digital pin connected to the DHT sensor

/*-----( Declare Variables )-----*/
float Val_humi = 0; // Humidity variables
float Val_temp = 0; // Temperature variables

// Xbee sensor variables for frame
unsigned int analogMSB1 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB1 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_SoilHygro = 0; //variable for the soil hygrometer data

unsigned int analogMSB2 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB2 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_InLdr = 0; // variable for the LDR in data

unsigned int analogMSB3 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB3 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_Mq07Gas = 0; // variable for the Mq-07 data

int My_Val_Max_humi=90; // μέγιστη τιμή υγρασίας %
int My_Val_Min_humi=65; // ελάχιστη τιμή υγρασίας %

int My_Val_Max_Mq07Gas=70; // μέγιστη τιμή αισθητήρα αερίου %

DHT dht(SEN_Dht11, DHT11); // επιλογή τύπου αισθητήρα DHT

void setup() {
  Serial.begin(9600); //start serial to 9600 baud
  dht.begin(); // start the dht11

  Serial.println(F("DHT11 test!!")); // print some text in Serial Monitor

  pinMode(SEN_Dht11, INPUT); //Set the INPUT pin mode

  pinMode(MOS_AirRecyclingFan, OUTPUT); //Set the Outout pin mode

  digitalWrite(MOS_AirRecyclingFan, LOW); //set the pin to LOW
}

void loop() {
  Read_Sen_Dht11_Loop();
  Read_RemoteXbee_Loop();

  System_AirRecycling_Loop();

  Print_Serial_loop();
}

void Read_Sen_Dht11_Loop() {
  Val_humi = dht.readHumidity(); // read the humidity
}

//-----Read the sensor value of Router Xbee-----
```



```

void Read_RemoteXbee_Loop() {
  // έλεγχος εάν το frame έχει τουλάχιστον 29 bits
  if (Serial.available() >= 29) {
    // έλεγχος εάν έχει start byte 7E
    if (Serial.read() == 0x7E) {
      // Απόρριψη των πρώτων 21 byte
      for (int i = 1; i < 21; i++) {
        byte discardbyte = Serial.read();
      }
      //τα Πρώτα byte είναι η τιμή του Soil hygrometer
      // Read the most significant bit of the analog data
      analogMSB1 = Serial.read(); // Read the first analog byte data
      analogLSB1 = Serial.read(); // Read the second byte
      //μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
      //byte
      Val_SoilHygro = analogLSB1 + (analogMSB1 * 256); //900- νερό 800
όχι νερό
      //Αντιστοιχίσει των Byte σε επι τοις εκατό
      Val_SoilHygro = map(Val_SoilHygro, 700, 1000, 100, 0);

      //τα Επόμενα byte είναι η τιμή του LDR
      analogMSB2 = Serial.read(); // Read the first analog byte data
      analogLSB2 = Serial.read(); // Read the second byte
      //μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
      //byte
      Val_InLdr = analogLSB2 + (analogMSB2 * 256);
      //Αντιστοιχίσει των Byte σε επι τοις εκατό
      Val_InLdr = map(Val_InLdr, 0, 1030, 0, 100);
      //τα τελευταία byte είναι η τιμή το Gas Sensor Mq07

      analogMSB3 = Serial.read(); // Read the first analog byte data
      analogLSB3 = Serial.read(); // Read the second byte
      //μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
      //byte
      Val_Mq07Gas = analogLSB3 + (analogMSB3 * 256);
      //Αντιστοιχίσει των Byte σε επι τοις εκατό
      Val_Mq07Gas = map(Val_Mq07Gas, 0, 1030, 0, 100);
    }
  }
}

//control the Humidity Sytem
void System_AirRecycling_Loop() {
  //εάν η τιμή της υγρασίας είναι πάνω από 90% ή
  if (Val_humi >= My_Val_Max_humi ||
  //η ποιότητα του αέρα είναι πάνω από 70% τότε
  Val_Mq07Gas > My_Val_Max_Mq07Gas ) {
  //Ο δεύτερος ανεμιστήρας που είναι για των εξαερισμό δουλεύει στις
  μέγιστες στροφές
  digitalWrite(MOS_AirRecyclingFan, HIGH);
  //Εμφάνιση της κατάστασης στην σειριακή οθόνη
  Serial.println(F("AirRecycl=Full"));

  //εάν η υγρασία είναι πάνω απο 65% τότε
  }else if (Val_humi >= My_Val_Min_humi) {
  // ο ανεμιστήρας θα δουλεύει στις μισές στροφές
  analogWrite(MOS_AirRecyclingFan, 127); //PWM
  //Εμφάνιση της κατάστασης στην σειριακή οθόνη
  Serial.println(F("AirRecycl=Medium"));
  //αλλιώς εάν είναι χαμηλότερα από 65% τότε
  }else {

```

```

// Ο ανεμιστήρας του εξαερισμού είναι σβηστός
digitalWrite(MOS_AirRecyclingFan, LOW);
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("AirRecycl=Stop"));
}
}

// Displays the value on the Serial Monitor
void Print_Serial_loop() {
  Serial.println(F("*****"));
  Serial.print(F("Humidity: ")); Serial.print(Val_humi);
Serial.println(F("%"));
  Serial.print(F("Gas Value: ")); Serial.print(Val_Mq07Gas);
Serial.println(F("%"));
  Serial.println(Val_SoilHygro);
  Serial.println(Val_InLdr);
}

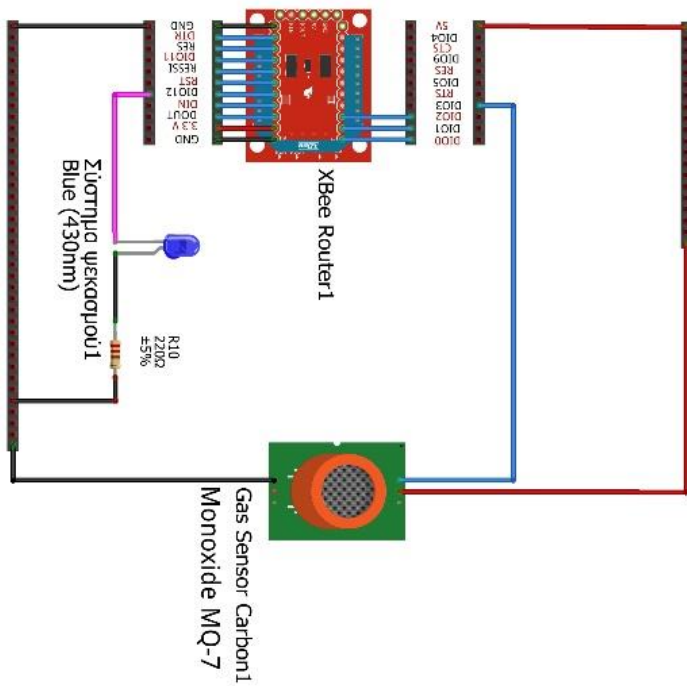
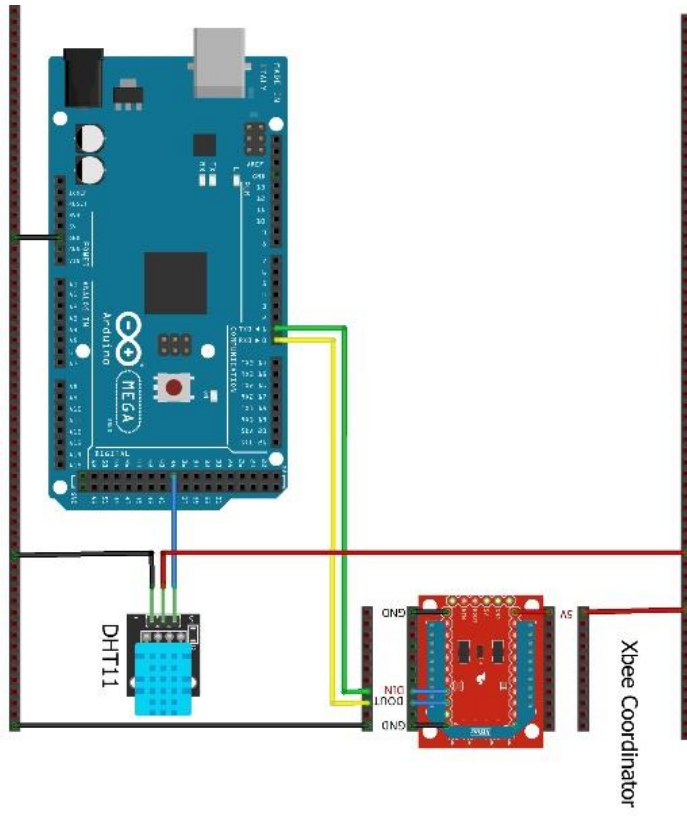
```

## 5.7 Περιγραφή σεναρίου αυτοματισμού ελέγχου ψεκασμού ομίχλης

Το σενάριο ελέγχου ψεκασμού ομίχλης από νερό αποτελείται από:

- 1 Xbee ως Coordinator του οποίου οι ακίδες του συνδέονται με το Arduino ως εξής ακίδα D1-TX του Arduino με το DIN του Xbee και η ακίδα D0-RX του Arduino με το DOUT του Xbee.
- 1 Xbee ως Router στον οποίου συνδέεται ο αισθητήρας αερίου και το μπλε LED ως εξομοίωση του συστήματος ψεκασμού ομίχλης από νερό.
- 1 μπλε LED ως εξομοίωση του συστήματος ψεκασμού ομίχλης από νερό που συνδέεται στην ακίδα DIO12 του Xbee Router.
- 1 αισθητήρα διοξειδίου του άνθρακα MQ-07 (Gas Sensor Carbon Monoxide MQ-7) που συνδέεται στην ακίδα DI03 του Xbee Router
- 1 αισθητήρα θερμοκρασίας και υγρασίας DHT11 που συνδέεται στην ακίδα D38 του Arduino mega.

Στο [Κύκλωμα 5.6](#) παρουσιάζεται ο τρόπος διασύνδεσης των εξαρτημάτων για την υλοποίηση του σεναρίου.



fritzing

Κύκλωμα 5.6 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου ψεκασμού ομίχλης από νερό.

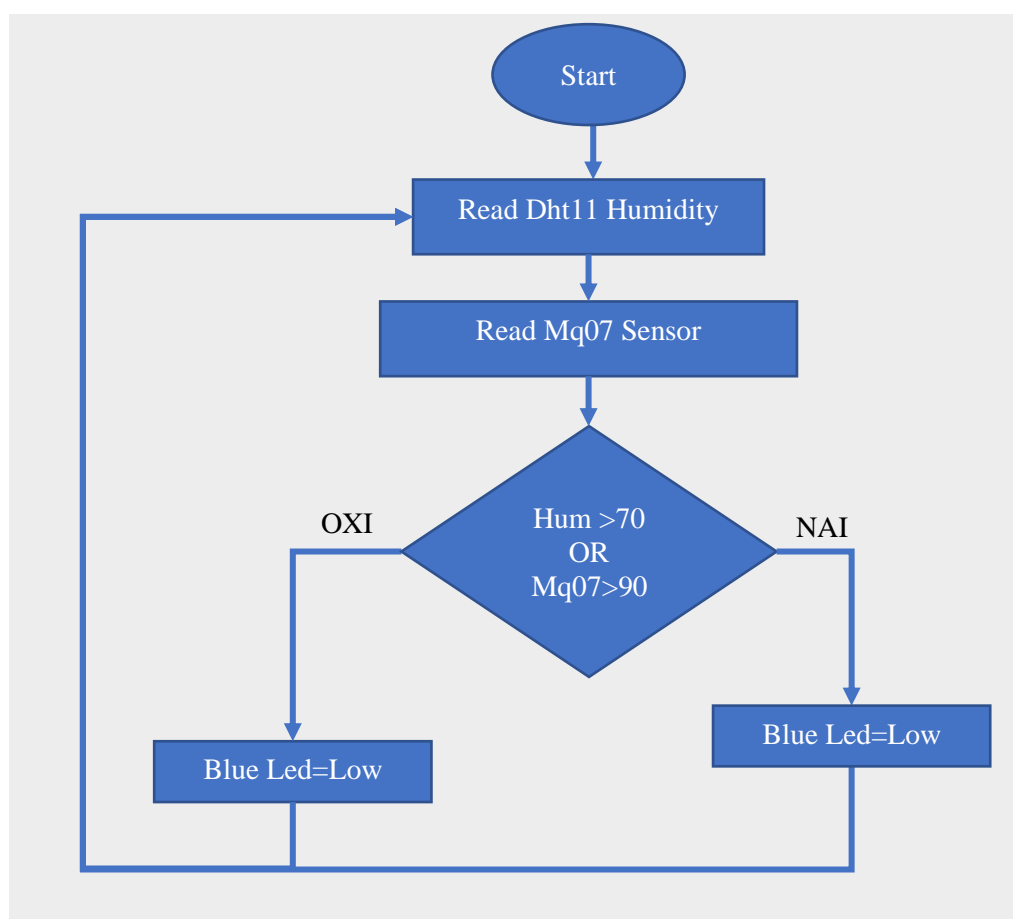
Περιγραφή σεναρίου:

Το σενάριο αυτοματισμού ελέγχου ψεκασμού ομίχλης από νερό είναι υπεύθυνο για την αύξηση της υγρασίας του χώρου και για την πυρόσβεση σε περίπτωση φωτιάς.

Αρχικά διαβάζεται η τιμή της υγρασίας από τον αισθητήρα θερμοκρασίας και υγρασίας DHT11 και η τιμή από τον αισθητήρα διοξειδίου του άνθρακα MQ-07

- Εάν η τιμή της υγρασίας είναι κάτω από 80% ή η τιμή του διοξειδίου του άνθρακα MQ-07 είναι πάνω από 90%, ενεργοποιείται το σύστημα ψεκασμού.
- Εάν δεν ισχύουν αυτά, τότε το σύστημα ψεκασμού είναι απενεργοποιημένο.

Παρακάτω ακολουθεί το λογικό διάγραμμα ροής του σεναρίου ([Σχήμα 5.7](#)).



Σχήμα 5.7 Σεναρίου αυτοματισμού ελέγχου ψεκασμού ομίχλης από νερό

Ακολουθεί ο κώδικας Arduino για το σενάριο αυτοματισμού ελέγχου ψεκασμού ομίχλης από νερό.

```
/*----- ( Import needed libraries )-----*/  
#include "DHT.h" // Βιβλιοθήκη για των αισθητήρα DHT11
```

```

/*-----( Declare Constants and Pin Numbers )-----*/
const int MOS_AirRecyclingFan = 2; //Gate mosfet 1 attach to d2 pin
Arduino for Air Recycling area greenhouse

#define SEN_Dht11 38 // Digital pin connected to the DHT sensor

/*-----( Declare objects )-----*/
DHT dht(SEN_Dht11, DHT11); // Select type Dht11

/*-----( Declare Variables )-----*/
float Val_humi = 0; // Humidity variables
float Val_temp = 0; // Temperature variables

// Xbee sensor variables for frame
unsigned int analogMSB1 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB1 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_SoilHygro = 0; //variable for the soil hygrometer data

unsigned int analogMSB2 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB2 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_InLdr = 0; // variable for the LDR in data

unsigned int analogMSB3 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB3 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_Mq07Gas = 0; // variable for the Mq-07 data

int My_Val_Max_humi=90; // Μέγιστη τιμή υγρασίας επι τοις εκατό
int My_Val_Min_humi=65; // Ελάχιστη τιμή υγρασίας επι τοις εκατό

int My_Val_Max_Mq07Gas=70; //Μέγιστη τιμή καπνού επι τοις εκατό

void setup(){
  Serial.begin(9600); //start serial to 9600 baud
  dht.begin(); // start the dht11
  Serial.println(F("DHT11 test!!")); // print some text in Serial Monitor
  pinMode(SEN_Dht11, INPUT); // set the pin mode to input
  pinMode(MOS_AirRecyclingFan, OUTPUT); // Sets the trigPin as an OUTPUT
  digitalWrite(MOS_AirRecyclingFan, LOW); //set the pin to LOW
}
void loop() {
  Read_Sen_Dht11_Loop();
  Read_RemoteXbee_Loop();
  System_ForggerSpray_Loop();
  Print_Serial_loop();
}
// Read Humidity and Temperature as Celsius (the default)
void Read_Sen_Dht11_Loop() {
  Val_humi = dht.readHumidity();
  Val_temp = dht.readTemperature();
}
//----Read the value of Router Xbee-----
void Read_RemoteXbee_Loop() {
  // έλεγχος εαν το frame έχει τουλάχιστον 29 bits
  if (Serial.available() >= 29) {
    // έλεγχος εαν έχει start byte 7E
    if (Serial.read() == 0x7E) {
      // Απόρριψη των πρώτων 21 byte
      for (int i = 1; i < 21; i++) {
        byte discardbyte = Serial.read();
      }
      //τα Πρώτα byte είναι η τιμή του Soil hygrometer
    }
  }
}

```

```

// Read the most significant bit of the analog data
analogMSB1 = Serial.read(); // Read the first analog byte data
analogLSB1 = Serial.read(); // Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
Val_SoilHygro = analogLSB1 + (analogMSB1 * 256);
//Αντιστοιχίσει των Byte σε επι τοις εκατό
Val_SoilHygro = map(Val_SoilHygro, 700, 1000, 100, 0);

//τα Επόμενα byte είναι η τιμή του LDR
analogMSB2 = Serial.read();// Read the first analog byte data
analogLSB2 = Serial.read();// Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
Val_InLdr = analogLSB2 + (analogMSB2 * 256);
//Αντιστοιχίσει των Byte σε επι τοις εκατό
Val_InLdr = map(Val_InLdr, 0, 1030, 0, 100);
//τα τελευταία byte είναι η τιμή το Gas Sensor Mq07

analogMSB3 = Serial.read();// Read the first analog byte data
analogLSB3 = Serial.read();// Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
Val_Mq07Gas = analogLSB3 + (analogMSB3 * 256);
//Αντιστοιχίσει των Byte σε επι τοις εκατό
Val_Mq07Gas = map(Val_Mq07Gas, 0, 1030, 0, 100);
}
}
}
//control ForggerSpraySystem at the fire extinguishing system and humidity
void System_ForggerSpray_Loop() {
//Εάν η τιμή του αισθητήρα αερίου ή υγρασίας είναι μικρότερο από 65% τότε
if (Val_Mq07Gas > My_Val_Max_Mq07Gas || Val_humi < My_Val_Min_humi)
{//High Humidity ->ForggerSpray system OFF
//το value=0x5 ώστε το Blue led να είναι ON
setRem_LED_Blue_Blue_ForggerSpraySystem_DIO12(0x5); // 0x5 -> on
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("blueLed=on"));
}else {
//το value=0x4 ώστε το Blue led να είναι OFF
setRem_LED_Blue_Blue_ForggerSpraySystem_DIO12(0x4); //0x4 -> off
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("blueLed=off"));
}
}
// Displays the value on the Serial Monitor
void Print_Serial_loop() {
Serial.println(F("*****"));
Serial.print(F("Humidity: ")); Serial.print(Val_humi);
Serial.println(F("%"));
Serial.print(F("Gas Value: ")); Serial.print(Val_Mq07Gas);
Serial.println(F("%"));
}
//=====
// create the frame to send to router
// πλαίσιο για των ελέγχου του κόκκινου LED και του buzzer που
//εξομοιώνει το σύστημα ειδοποίησης
void setRem_LED_RED_Org_ALERT_DIO4(char value) {
Serial.write(0x7E); //0
Serial.write(byte(0x0)); //1
Serial.write(0x10); //2
Serial.write(0x17); // 3
}

```



```

Serial.write(byte(0x0)); //4

Serial.write(byte(0x0)); //5
Serial.write(byte(0x0)); //6
Serial.write(byte(0x0)); //7
Serial.write(byte(0x0)); //8
Serial.write(byte(0x0)); //9
Serial.write(byte(0x0)); //10
Serial.write(0xFF); //12
Serial.write(0xFF); // 13

Serial.write(0xFF); //14
Serial.write(0xFE); //15

Serial.write(0x02); //16
Serial.write('D'); //17
Serial.write('4'); //18
Serial.write(value); // 19
long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'D' + '4' + value;
Serial.write(0xFF - (sum & 0xFF));
}
//πλαίσιο για των ελέγχου του κίτρινου LED που εξομοιώνει το σύστημα
//σκίασης
void setRem_LED_Yellow_MOV_Sunshade_DIO11(char value) {
    Serial.write(0x7E); //start byte
    Serial.write(byte(0x0)); // high part of length (always zero)
    //low part of length (the number of bytes that follow, not including
    //check
    Serial.write(0x10);
    Serial.write(0x17); //0x17 is a remote AT command
    Serial.write(byte(0x0)); //frame id set to zero for no reply
    //Id of recipient, or use 0xFFFF for broadcast
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(0xFF); // 0xFF for broadcast
    Serial.write(0xFF); // 0xFF for broadcast

    //16 bit of recipient or 0xFFFFE
    Serial.write(0xFF);
    Serial.write(0xFE);

    Serial.write(0x02); // 0x02 to apply changes immediately on remote

    //AT command name in ASCII character
    Serial.write('P');
    Serial.write('1');

    // command data in as many bytes as needed
    Serial.write(value);

    //checksum is all bytes after length bytes
    long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '1' + value;
    Serial.write(0xFF - (sum & 0xFF)); // calculate the proper checksum
}
// πλαίσιο για των έλεγχο του μπλε LED που εξομοιώνει το σύστημα
//ψεκασμού ομίχλης νερού
void setRem_LED_Blue_Blue_ForggerSpraySystem_DIO12(char value) {
    Serial.write(0x7E);

```

```

Serial.write(byte(0x0));
Serial.write(0x10);
Serial.write(0x17);
Serial.write(byte(0x0));

Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(0xFF);
Serial.write(0xFF);
Serial.write(0xFF);
Serial.write(0xFE);

Serial.write(0x02);
Serial.write('P');
Serial.write('2');
Serial.write(value);
long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '2' + value;
Serial.write(0xFF - (sum & 0xFF));
}
//=====

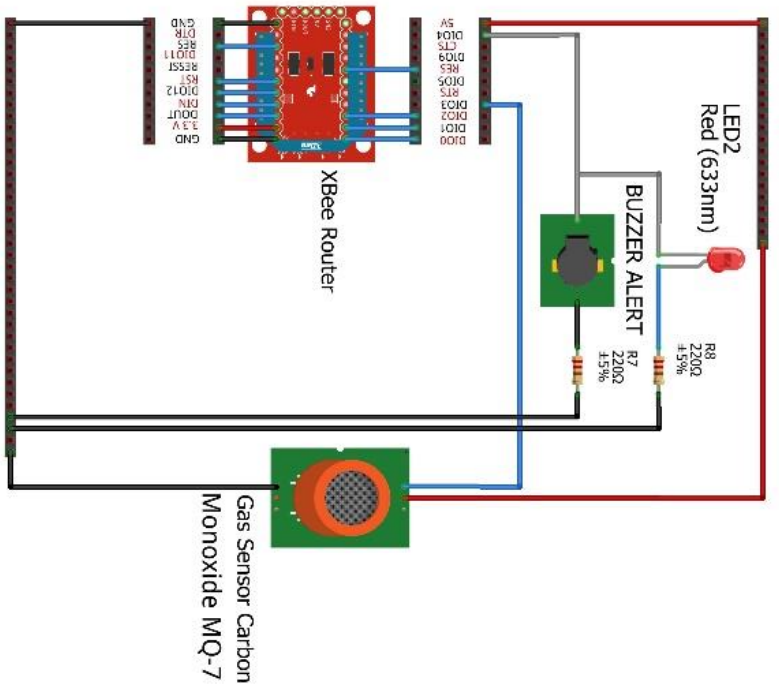
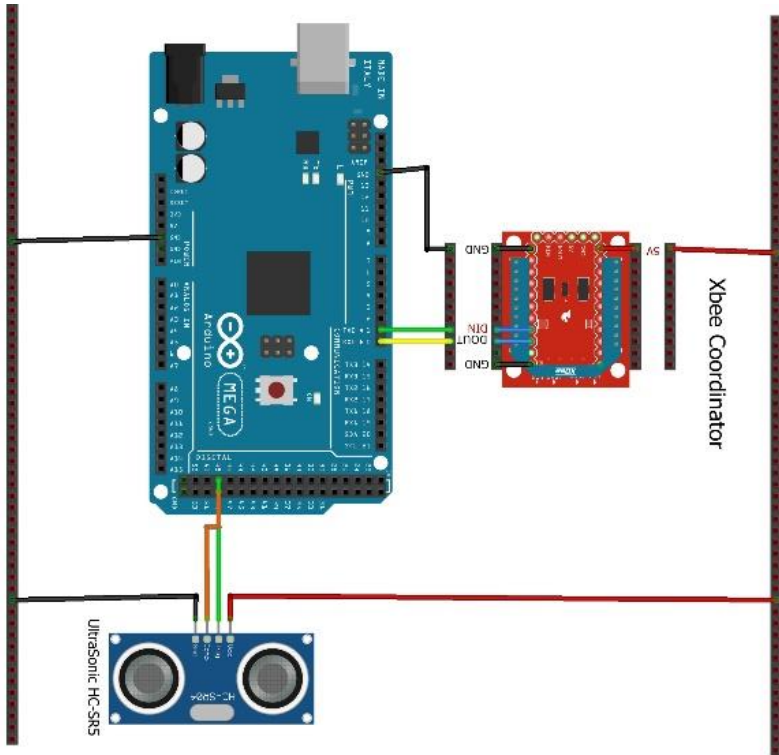
```

## 5.8 Περιγραφή σεναρίου αυτοματισμού ελέγχου συναγερμού

Το σενάριο αυτοματισμού ελέγχου συναγερμού αποτελείται από :

- 1 Xbee ως **Coordinator** του οποίου οι ακίδες του συνδέονται με το Arduino ως εξής: ακίδα D1 TX του Arduino με το DIN του Xbee και η ακίδα D0 RX του Arduino με το DOUT του Xbee.
- 1 Xbee ως Router στον οποίο συνδέεται ο αισθητήρας αερίου, το κόκκινο LED που και το Buzzer.
- 1 **buzzer** που συνδέεται σε **παράλληλα** με ένα κόκκινο LED.
- 1 κόκκινο LED που συνδέεται στην ακίδα **DIO4** του Xbee Router.
- 1 αισθητήρα διοξειδίου του άνθρακα MQ-07 (**Gas Sensor Carbon Monoxide MQ-7**) που συνδέεται στην ακίδα DI03 του Xbee Router.
- 1 αισθητήρα απόστασης **Distance HC-SR04 Ultrasonic** που συνδέεται στις ακίδες **D48** το Trig και το **D49** το Echo του Arduino mega.

Στο [Κύκλωμα 5.7](#) παρουσιάζεται ο τρόπος διασύνδεσης των εξαρτημάτων για την υλοποίηση του σεναρίου.



fritzing

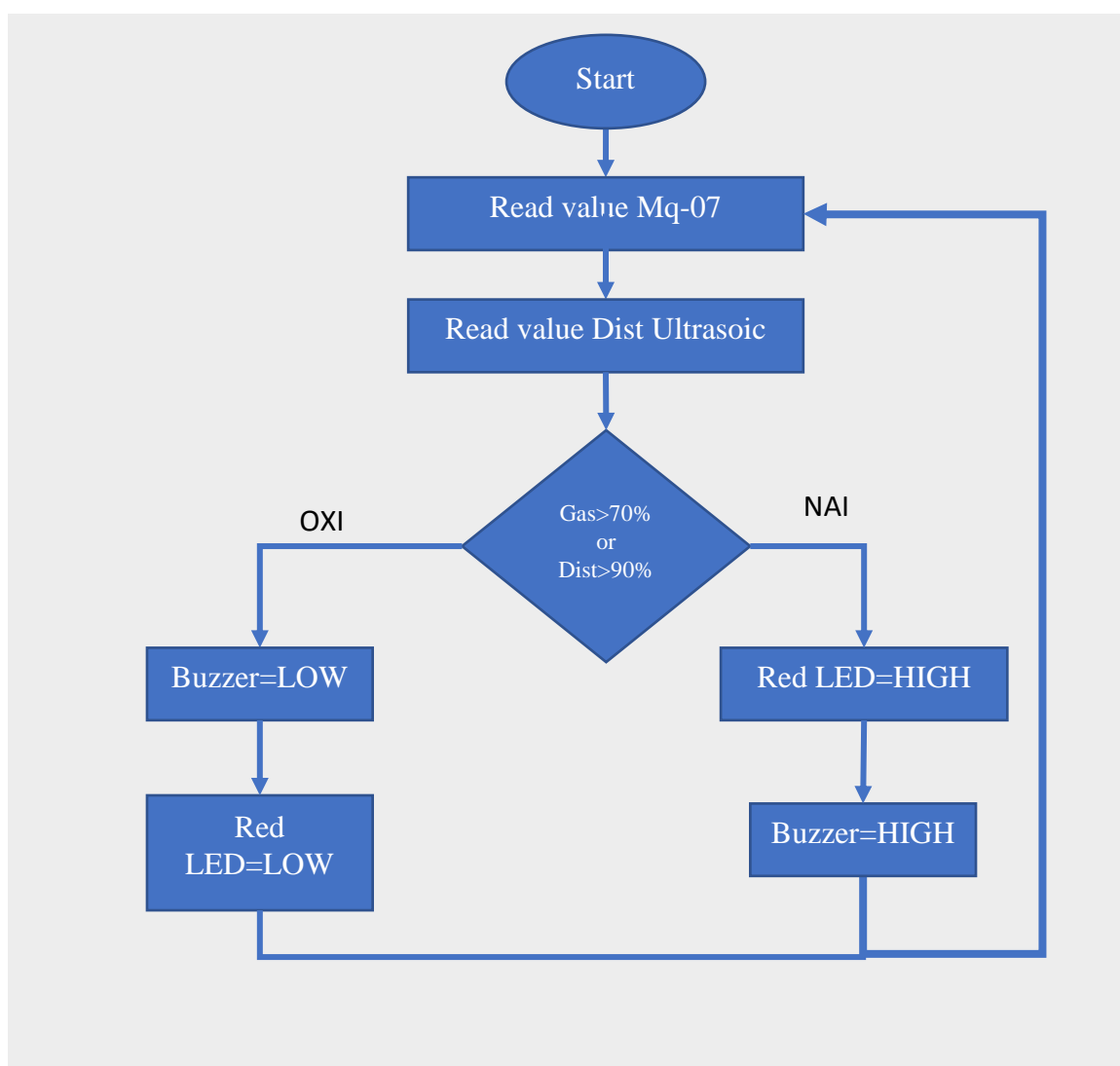
Κύκλωμα 5.7 Συνολική σύνδεση του σεναρίου αυτοματισμού ελέγχου συναγερμού.

Περιγραφή σεναρίου:

Το σενάριο αυτοματισμού ελέγχου συναγερμού είναι υπεύθυνο στο να ανιληφθεί την φωτιά μέσω του αισθητήρα Monoxide MQ-07 ή να ανιληφθεί μια υπερχειλίση στο δοχείο νερού και να ειδοποιήσει τον χρήστη μέσω ενός κόκκινου LED και ενός Buzzer.

Η λειτουργία του αυτοματισμού ελέγχου συναγερμού λειτουργεί ως εξής: όταν η τιμή του αισθητήρα Monoxide MQ-7 γίνει πάνω από **90%** ή η τιμή του αισθητήρα απόστασης **Distance HC-SR04 Ultrasonic** είναι πάνω από 95% τότε υπάρχει η περίπτωση να υπάρχει φωτιά στον χώρο ή υπερχειλίση, με αποτέλεσμα να ενεργοποιείται το κόκκινο LED και το buzzer. Εάν η τιμή του αισθητήρα Monoxide MQ-7, είναι κάτω από 90% ή η τιμή του αισθητήρα απόστασης **Distance HC-SR04 Ultrasonic** είναι κάτω από 95%, τότε είναι απενεργοποιημένο το κόκκινο LED και το buzzer

Παρακάτω ακολουθεί το λογικό διάγραμμα ροής του σεναρίου ([Σχήμα 5.8](#)).



Σχήμα 5.8 Σεναρίου αυτοματισμού ελέγχου συναγερμού.

Ακολουθεί ο κώδικας Arduino για το σενάριο αυτοματισμού ελέγχου συναγερμού.

```
/*-----( Import needed libraries )-----*/
#include <HCSR04.h> //librari Ultrasonic Sensor HC-SR04

/*-----( Declare Constants and Pin Numbers )-----*/
//attach pin D48 Arduino to pin Trig of HC-SR04
#define SEN_UltraSonicTrigPin 48
// attach pin D49 Arduino to pin Echo of HC-SR04
#define SEN_UltraSonicEchoPin 49

/*-----( Declare Variables )-----*/
int Val_distance; // variable for the distance measurement
int Val_distancePerCent;// variable for the distance per cent measurement

// Xbee sensor variables for frame
unsigned int analogMSB1 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB1 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_SoilHygro =0;//variable for the soil hygrometer data

unsigned int analogMSB2 = 0;// πιο σημαντικό ψηφίο bit
unsigned int analogLSB2 = 0;// λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_InLdr = 0;// variable for the LDR in data

unsigned int analogMSB3 = 0;// πιο σημαντικό ψηφίο bit
unsigned int analogLSB3 = 0;// λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_Mq07Gas = 0;// variable for the Mq-07 data

//-----Set the preference Constants
int My_Val_Max_distancePerCent=90; // μέγιστη τιμή νερού στο δοχείο
int My_Val_Min_distancePerCent=60; // ελάχιστη τιμή νερού στο δοχείο

int My_Val_Max_OutLdr=85; // μέγιστη τιμή εξωτερικής φωτεινότητας
int My_Val_Min_OutLdr=60; // ελάχιστη τιμή εξωτερικής φωτεινότητας

int My_Val_InLdr=40; //Give the desired light value
int My_Val_Max_Mq07Gas=70; //Μέγιστη τιμή καπνού επι της εκατό

HCSR04 hc(SEN_UltraSonicTrigPin, SEN_UltraSonicEchoPin); //initialization
class HCSR04 (trig pin , echo pin)

void setup() {
  Serial.begin(9600); //start serial to 9600 baud

  // print some text in Serial Monitor
  Serial.println(F("Ultrasonic Sensor HC-SR04 Test!!"));
  Serial.println(F("with Arduino UNO R3"));

  pinMode(SEN_UltraSonicEchoPin, INPUT); // Sets the echoPin as an INPUT
  pinMode(SEN_UltraSonicTrigPin, OUTPUT); // Sets the trigPin as an
OUTPUT
}

void loop() {
  Read_Sen_UltraSonic_Loop();
  Read_RemoteXbee_Loop();
  Print_Serial_loop();
  System_Alert_Loop();
}

//----Read the distance at tank-----
void Read_Sen_UltraSonic_Loop(){
```

```

Val_distance = hc.dist(); // Read Sensor ultrasonic data
//Μαping των αισθητήρα ultrasonic για την στάθμη νερού στο δοχείο.
//Το δοχείο είναι 17 εκατοστά αρά τότε θα έχει 0% νερό
//ενώ εάν είναι 0 εκ. τότε έχουμε 100% νερό μέσα στο δοχείο.
Val_distancePerCent = map(Val_distance, 17, 0, 0, 99);
}

//----Read the value of Router Xbee-----
void Read_RemoteXbee_Loop() {
  // έλεγχος εάν το frame έχει τουλάχιστον 29 bits
  if (Serial.available() >= 29) {
    // έλεγχος εάν έχει start byte 7E
    if (Serial.read() == 0x7E) {
      // Απόρριψη των πρώτων 21 byte
      for (int i = 1; i < 21; i++) {
        byte discardbyte = Serial.read();
      }
      //τα Πρώτα byte είναι η τιμή του Soil hygrometer
      // Read the most significant bit of the analog data
      analogMSB1 = Serial.read(); // Read the first analog byte data
      analogLSB1 = Serial.read(); // Read the second byte
      //μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
      //byte
      Val_SoilHygro = analogLSB1 + (analogMSB1 * 256);
      //Αντιστοιχίσει των Byte σε επι τοις εκατό
      Val_SoilHygro = map(Val_SoilHygro, 700, 1000, 100, 0);

      //τα Επόμενα byte είναι η τιμή του LDR
      analogMSB2 = Serial.read(); // Read the first analog byte data
      analogLSB2 = Serial.read(); // Read the second byte
      //μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
      //byte
      Val_InLdr = analogLSB2 + (analogMSB2 * 256);
      //Αντιστοιχίσει των Byte σε επι τοις εκατό
      Val_InLdr = map(Val_InLdr, 0, 1030, 0, 100);
      //τα τελευταία byte είναι η τιμή το Gas Sensor Mq07

      analogMSB3 = Serial.read(); // Read the first analog byte data
      analogLSB3 = Serial.read(); // Read the second byte
      //μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
      //byte
      Val_Mq07Gas = analogLSB3 + (analogMSB3 * 256);
      //Αντιστοιχίσει των Byte σε επι τοις εκατό
      Val_Mq07Gas = map(Val_Mq07Gas, 0, 1030, 0, 100);
    }
  }
}

//controll the Alert Sytem ->Buzzer Red led
void System_Alert_Loop() {
  //εάν η τιμή του αισθητήρα αερίου είναι πάνω από 70 ή
  //η το επίπεδο του νερό ξεπεράσει το 90% τότε
  if ((Val_Mq07Gas >= My_Val_Max_Mq07Gas) || (Val_distancePerCent >
  My_Val_Max_distancePerCent)){
    //κάνει το value=0x5 ώστε το Red Led και Buzzer να είναι ON
    setRem_LED_RED_Org_ALERT_DIO4(0x5);
    //Εμφάνιση της κατάστασης στην σειριακή οθόνη
    Serial.println(F("Alert=On"));
  }
  else{
    //το value=0x4 ώστε το Red Led και Buzzer να είναι Off

```



```

    setRem_LED_RED_Org_ALERT_DIO4(0x4);
    //Εμφάνιση της κατάστασης στην σειριακή οθόνη
    Serial.println(F("Alert=Off"));
}
}

// Displays the value on the Serial Monitor
void Print_Serial_loop() {
    Serial.println(F("*****"));
    Serial.print(F("Distance: ")); Serial.print(Val_distancePerCent);
    Serial.println(F("%"));
    Serial.print(F("Distance: ")); Serial.print(Val_distance);
    Serial.println(F("cm"));
    Serial.print(F("Soil moisture: ")); Serial.print(Val_SoilHygro);
    Serial.println(F("%"));
    Serial.print(F("Ldr IN Value: ")); Serial.print(Val_InLdr);
    Serial.println(F("%"));
    Serial.print(F("Gas Value: ")); Serial.print(Val_Mq07Gas);
    Serial.println(F("%"));
    Serial.println(F("*****"));
}

//=====
// create the frame to send to router
// πλαίσιο για των ελέγχου του κόκκινου LED και του buzzer που
//εξομοιώνει το σύστημα ειδοποίησης
void setRem_LED_RED_Org_ALERT_DIO4(char value) {
    Serial.write(0x7E); //0
    Serial.write(byte(0x0)); //1
    Serial.write(0x10); //2
    Serial.write(0x17); // 3
    Serial.write(byte(0x0)); //4

    Serial.write(byte(0x0)); //5
    Serial.write(byte(0x0)); //6
    Serial.write(byte(0x0)); //7
    Serial.write(byte(0x0)); //8
    Serial.write(byte(0x0)); //9
    Serial.write(byte(0x0)); //10
    Serial.write(0xFF); //12
    Serial.write(0xFF); // 13

    Serial.write(0xFF); //14
    Serial.write(0xFE); //15

    Serial.write(0x02); //16
    Serial.write('D'); //17
    Serial.write('4'); //18
    Serial.write(value); // 19
    long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'D' + '4' + value;
    Serial.write(0xFF - (sum & 0xFF));
}

//πλαίσιο για των ελέγχου του κίτρινου LED που εξομοιώνει το σύστημα
//σκίασης
void setRem_LED_Yellow_MOV_Sunshade_DIO11(char value) {
    Serial.write(0x7E); //start byte
    Serial.write(byte(0x0)); // high part of length (always zero)
    //low part of length (the number of bytes that follow, not including
    //check
    Serial.write(0x10);
    Serial.write(0x17); //0x17 is a remote AT command
}

```

```

Serial.write(byte(0x0)); //frame id set to zero for no reply
//Id of recipient, or use 0xFFFF for broadcast
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(0xFF); // 0xFF for broadcast
Serial.write(0xFF); // 0xFF for broadcast

//16 bit of recipient or 0xFFFFE
Serial.write(0xFF);
Serial.write(0xFE);

Serial.write(0x02); // 0x02 to apply changes immediately on remote

//AT command name in ASCII character
Serial.write('P');
Serial.write('1');

// command data in as many bytes as needed
Serial.write(value);

//checksum is all bytes after length bytes
long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '1' + value;
Serial.write(0xFF - (sum & 0xFF)); // calculate the proper checksum
}

// πλαίσιο για των έλεγχο του μπλε LED που εξομοιώνει το σύστημα
//ψεκασμού ομίχλης νερού
void setRem_LED_Blue_Blue_ForggerSpraySytem_DI012(char value) {
Serial.write(0x7E);
Serial.write(byte(0x0));
Serial.write(0x10);
Serial.write(0x17);
Serial.write(byte(0x0));

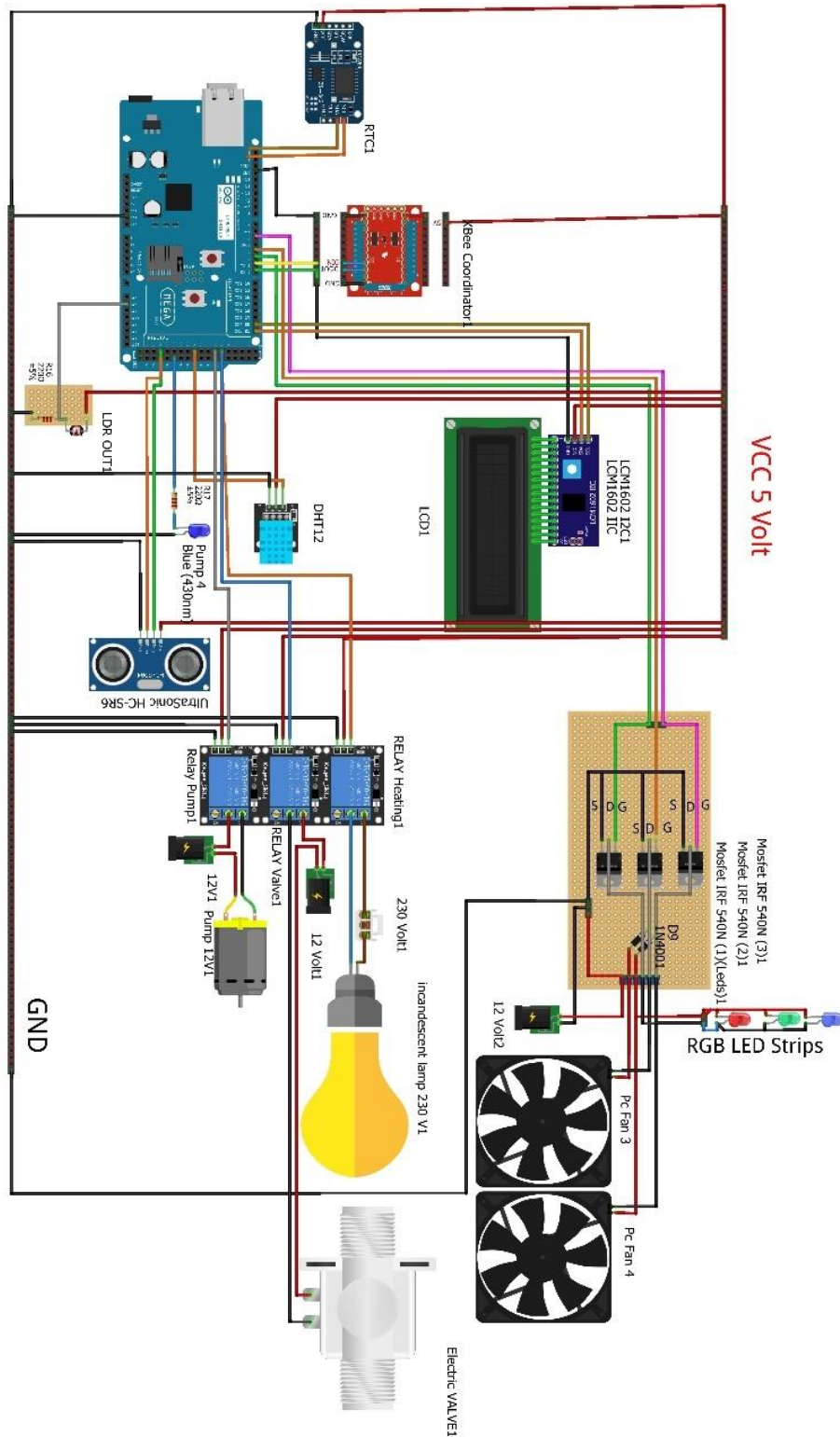
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(0xFF);
Serial.write(0xFF);
Serial.write(0xFF);
Serial.write(0xFE);

Serial.write(0x02);
Serial.write('P');
Serial.write('2');
Serial.write(value);
long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '2' + value;
Serial.write(0xFF - (sum & 0xFF));
}
//=====

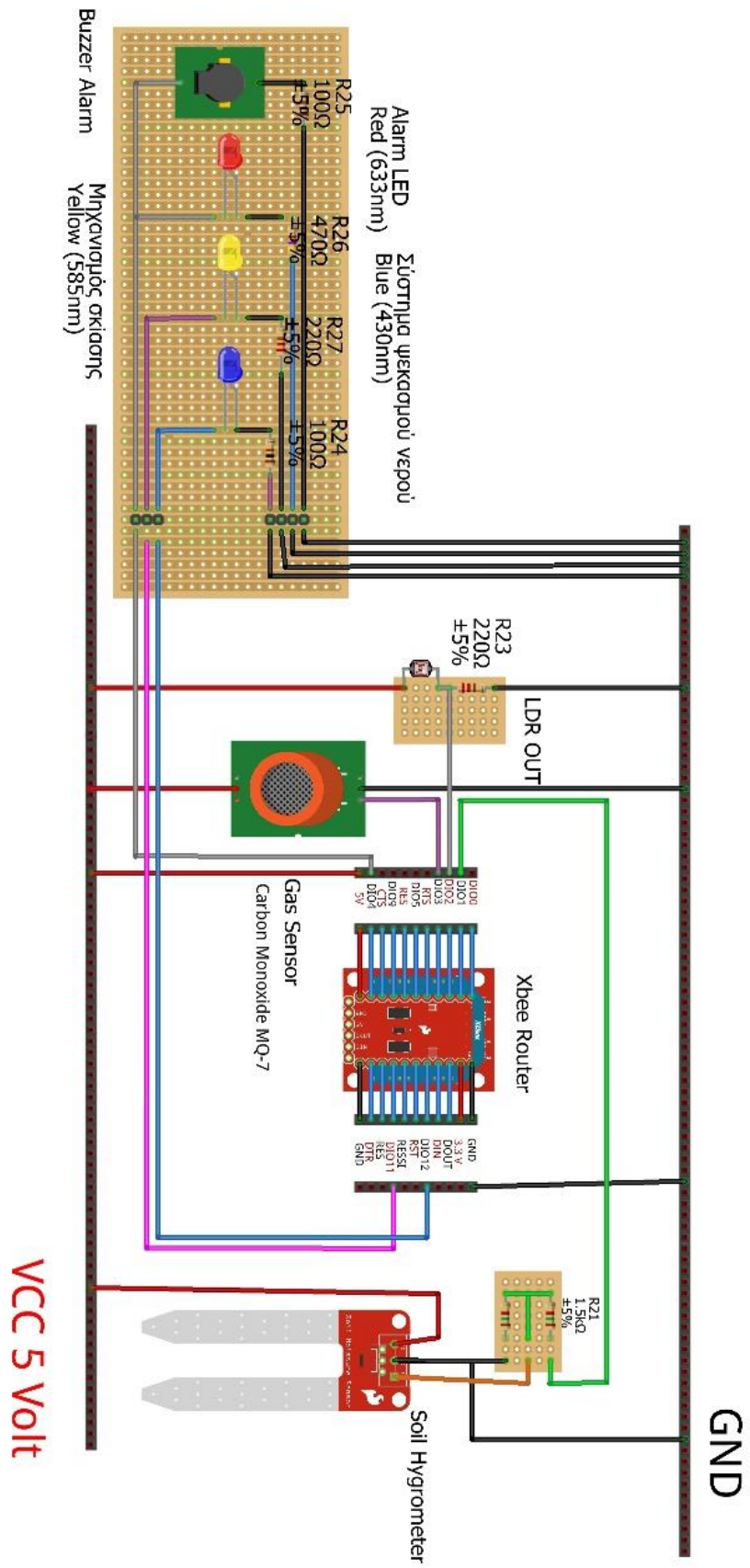
```

## 5.9 Το συνολικό κύκλωμα του IoT Θερμοκηπίου

Το ολοκληρωμένο κύκλωμα του θερμοκηπίου με όλα τα σενάρια αυτοματισμού ελέγχου παρουσιάζεται στα παρακάτω κυκλώματα ([Κύκλωμα 5.8](#) και [Κύκλωμα 5.9](#)).



Κύκλωμα 5.8 Το συνολικό κύκλωμα που συνδέεται στο Arduino mega.



fritzing

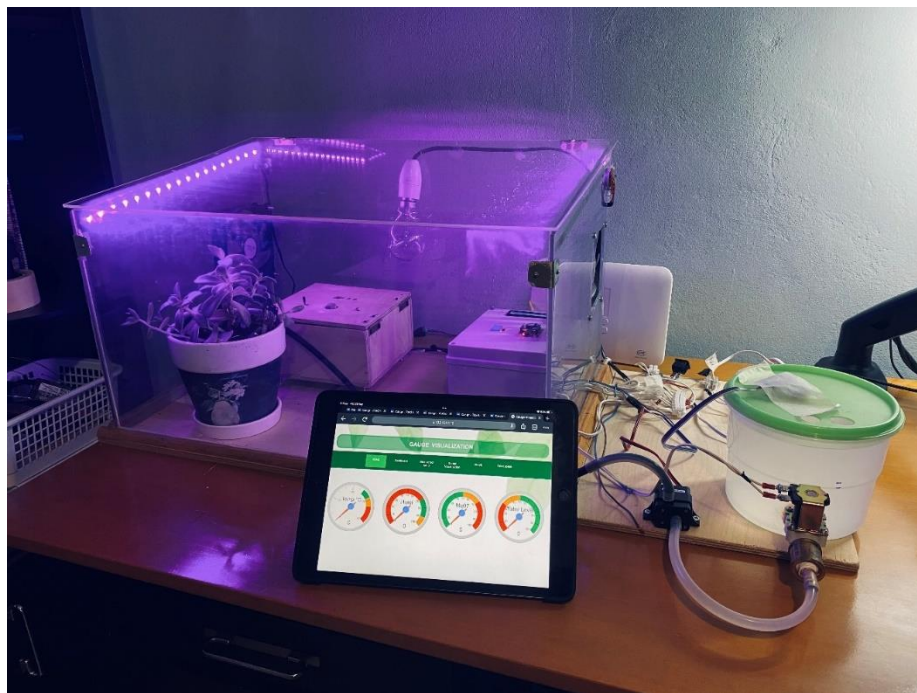
Κύκλωμα 5.9 Το συνολικό κύκλωμα που συνδέετε στο Xbee Router.

Όλα τα υλικά που χρησιμοποιήθηκαν για την υλοποίηση του θερμοκηπίου αναφέρονται παρακάτω.

- 1 Ethernet Shield
- 1 Xbee ως Coordinator του οποίου οι ακίδες του συνδέονται με το Arduino ως εξής: ακίδα D1 TX του Arduino με το DIN του Xbee και η ακίδα D0 RX του Arduino με το DOUT του Xbee.
- 1 Xbee ως Router στον οποίο συνδέεται ο αισθητήρας αερίου, LDR, Soil Hygrometer, το κίτρινο LED, το μπλε LED, το κόκκινο LED που και το Buzzer.
- 1 buzzer που συνδέεται σε παράλληλη συνδεσμολογία με ένα κόκκινο LED.
- 1 κόκκινο LED που συνδέεται στην ακίδα DIO4 του Xbee Router.
- 1 κίτρινο LED που εξομοιώνει ένα μηχανισμό σκίασης του θερμοκηπίου που συνδέεται στην ακίδα DIO11 του Xbee Router.
- 1 μπλε LED ως εξομοίωση του συστήματος ψεκασμού ομίχλης από νερό που συνδέεται στην ακίδα DIO12 του Xbee Router.
- 1 Αισθητήρα Φωτοαντίσταση LDR στο Εσωτερικό του θερμοκηπίου συνδέεται στην ακίδα DI02 του Xbee Router
- 1 αισθητήρα διοξειδίου του άνθρακα MQ-07 (Gas Sensor Carbon Monoxide MQ-7) που συνδέεται στην ακίδα DI03 του Xbee Router.
- 1 αισθητήρας υγρασίας χρώματος Soil Hygrometer που συνδέεται στην ακίδα DI01 του Xbee Router
- 1 αισθητήρα θερμοκρασίας και υγρασίας DHT11 που συνδέεται στην ακίδα D38 του Arduino mega.
- 1 αισθητήρα απόστασης Distance HC-SR04 Ultrasonic που συνδέεται στις ακίδες D48 το Trig και το D49 το Echo του Arduino mega.
- 1 Αισθητήρα Φωτοαντίστασης LDR στο Εξωτερικό του θερμοκηπίου που συνδέεται στην ακίδα A9 του Arduino mega.
- 1 ρολόι πραγματικού χρόνου Real Time Clock (RTC) που συνδέεται στις ακίδες SCL και SDA του Arduino mega.
- 1 οθόνη LCD που συνδέεται στις ακίδες SDA D20 και SCL D21 του Arduino mega.
- 1 μπλε LED που θα εξομοιώνει μια δεύτερη αντλία (pump2) που συνδέεται στην ακίδα D45 του Arduino mega
- 1 ηλεκτρονική βαλβίδα που συνδέεται σε ένα Valve Relay 5V.

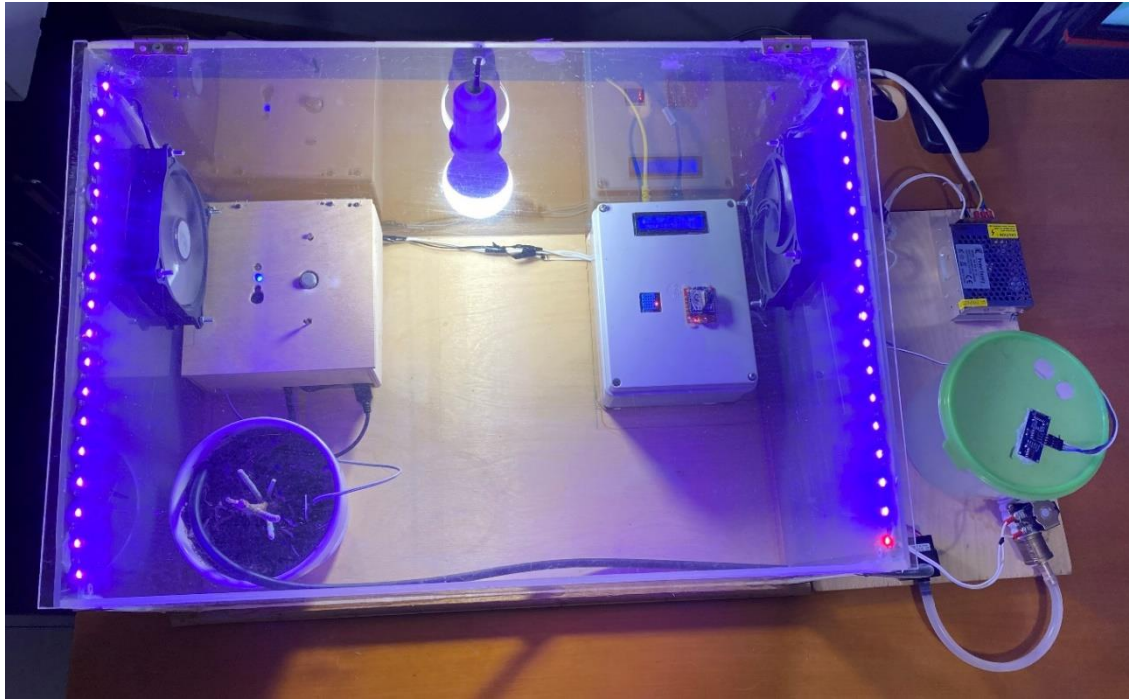
- 1 λάμπα πυρακτώσεως 230 volt ως εξομοίωση ενός θερμαντικού στοιχείου που συνδέεται στο Heating Relay 5V.
- 1 αντλία 12 volt (pump1) που συνδέεται στο Pump Relay 5V
- 1 Valve Relay 5V που συνδέεται στο ακίδα D30 του Arduino mega.
- 1 Heating Relay 5V που συνδέεται στο ακίδα D31 του Arduino mega.
- 1 Pump Relay 5V που συνδέεται στο ακίδα D32 του Arduino mega.
- 1 ανεμιστήρα 12 volt (Pc Fan 1) για εξομοίωση του συστήματος ψύξης που συνδέεται με το Mosfet (1).
- 1 ανεμιστήρα 12 volt (Pc Fan 2) για εξομοίωση του συστήματος αερισμού που συνδέεται με το Mosfet 2.
- 2 Ταινίες LED που θα συνδεθούν με το Mosfet-3
- 1 Mosfet N- Channel 33A IRF540N-(1) που συνδέεται στην ακίδα D2 του Arduino.
- 1 Mosfet N- Channel 33A IRF540N-(2) που συνδέεται στην ακίδα D3 του Arduino.
- 1 Mosfet N- Channel 33A IRF540N-(3) που συνδέεται στην ακίδα D5 του Arduino
- 1 δίοδο 1N4001

Ακολουθούν ενδεικτικές φωτογραφίες από την υλοποίηση του θερμοκηπίου.

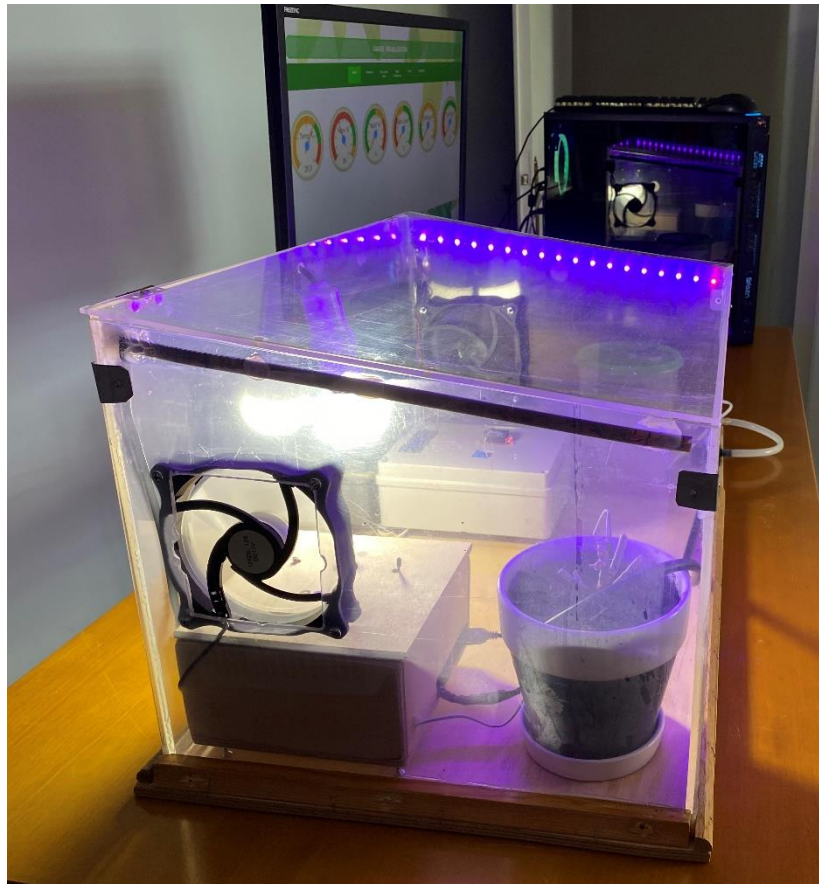


Εικόνα 5.2 Εικόνα από το θερμοκήπιο που υλοποιήσαμε.





*Εικόνα 5.3 Πάνο όψη θερμοκηπίου.*



*Εικόνα 5.4 Αριστερή όψη του θερμοκηπίου.*

## 6 Δημιουργία ιστοσελίδας απομακρυσμένου ελέγχου και παρακολούθησης

### 6.1 Πρόλογος

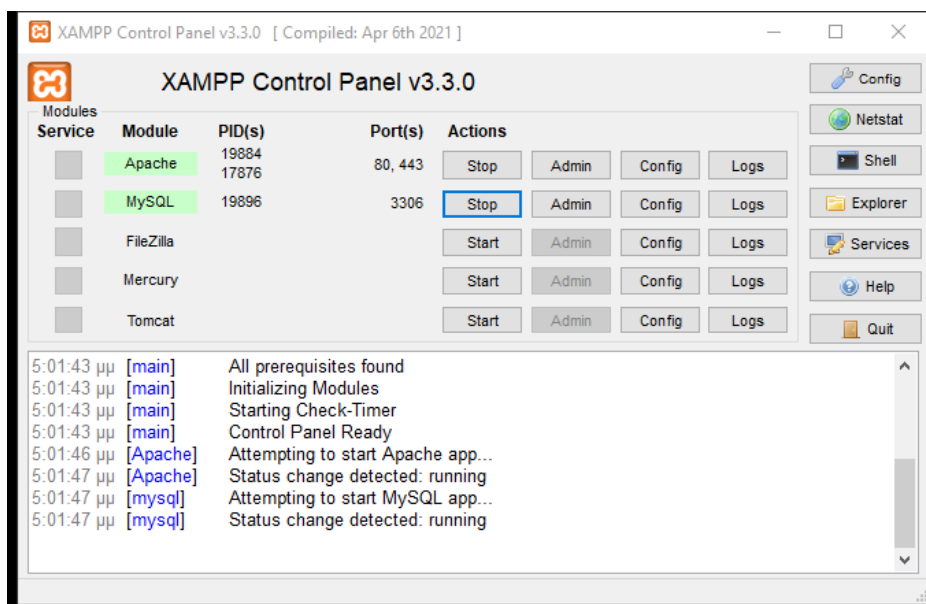
Σε αυτό το κεφάλαιο παρουσιάζεται η υλοποίηση της ιστοσελίδας για τον έλεγχο και την παρακολούθηση του θερμοκηπίου. Τα δεδομένα που παρουσιάζονται στην ιστοσελίδα είναι από την πλατφόρμα ThingSpeak και από την βάση δεδομένων MySQL χρησιμοποιώντας τον διακομιστή XAMPP αποστέλλονται μέσω του Arduino Ethernet shield και του δρομολογητή σε αυτήν.

Όταν εγκαθιστούμε το XAMPP δημιουργείται ένα φάκελος με την ονομασία htdocs οπότε εκεί θα δημιουργούμε ένα φάκελο με το όνομα db\_allSensor και μέσα σε αυτόν θα δημιουργήσουμε τα παρακάτω αρχεία για την λειτουργία της ιστοσελίδας.

- index.html
- Connection.php
- Data.php

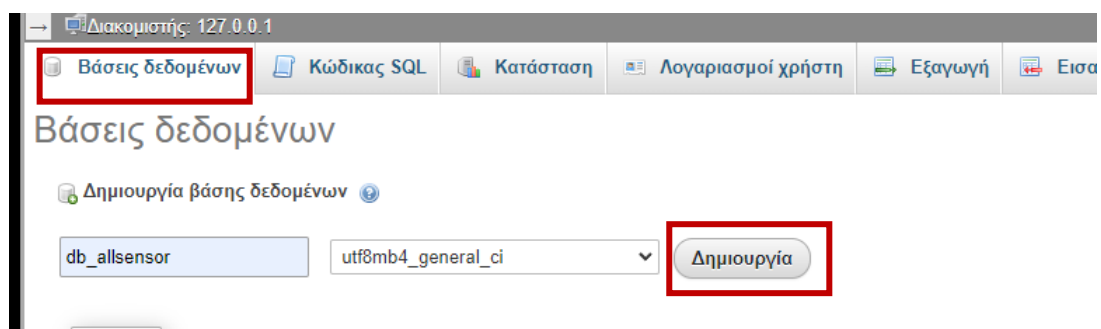
### 6.2 Δημιουργία βάσης δεδομένων

Για να μπορούμε να δημιουργήσουμε μια βάση δεδομένων για την εμφάνιση και την επεξεργασία των δεδομένων από τους αισθητήρες θα χρειαστεί να τρέξουμε το XAMPP και να γίνει START ο Apache και η MySQL.



Εικόνα 6.1 Εκκίνηση του Apache και της MySQL από τον πίνακα ελέγχου.

Για να δημιουργηθεί μια βάση δεδομένων με το όνομα db\_allsensor, χρειάστηκε η σύνδεση στην σελίδα <http://localhost/phpmyadmin/> και πατώντας πάνω αριστερά Βάσεις δεδομένων, εμφανίζεται η [Εικόνα 6.2](#) στην οποία επιλέγουμε το κουμπί δημιουργία.



Εικόνα 6.2 Απεικονίζει της βάσεις δεδομένων MySQL.

Τρέχοντας τον παρακάτω κώδικα SQL δημιουργείται ο πίνακας data με επτά στήλες η οποίες περιέχουν τα δεδομένα από των αισθητήρα θερμοκρασίας-υγρασίας DHT11, εξωτερικής φωτεινότητας LDR, εσωτερικής φωτεινότητας LDR, απόστασης ultrasonic, υγρασίας εδάφους soil hygrometer, και από των αισθητήρα αεριού MQ-07.

Ακόλουθη ο κώδικας mysql.sql.

```
--Δημιουργία πίνακα Data στην βάση δεδομένων db_allSensor
CREATE TABLE db_allSensor.data (
--Δημιουργία αριθμού ID που είναι ένας μοναδικός αριθμός που αυξάνετε
--αυτόματα κατά μία μονάδα όταν προστίθεται μια νέα εγγραφή στον πίνακα
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
--Δημιουργία στήλης που θα εμφανίζει την ώρα και την μέρα της εγγραφής
  event TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
--Εισαγωγή των δεδομένων από τους αισθητήρες στον πίνακα
--Τα δεδομένα θα είναι έως 10 BYTE και δεν θα δέχεται μηδενικά
  temperature VARCHAR(10) NOT NULL,
  humidity VARCHAR(10) NOT NULL,
  outLdr VARCHAR(10) NOT NULL,
  tank VARCHAR(10) NOT NULL,
  soilhygro VARCHAR(10) NOT NULL,
  inLdr VARCHAR(10) NOT NULL,
  Mq07Gas VARCHAR(10) NOT NULL
)
```

### 6.3 Σύνδεση Arduino με την βάση δεδομένων

Για την εισαγωγή των δεδομένων από τους αισθητήρες στην βάση δεδομένων που δημιουργήσαμε χρειάζεται ο κώδικας Arduino, connection.php και data.php.

Ακολουθεί ο κώδικας του Arduino για την σύνδεση με την βάση και την αποστολή των δεδομένων.

Επίσης απαραίτητο για την λειτουργία του παρακάτω κώδικα είναι εισαγωγή της βιβλιοθήκης SPI που χρησιμοποιείτε για την λειτουργία του Ethernet shield μέσω αυτού του

πρωτοκόλλου, η βιβλιοθήκη Ethernet για την λειτουργία των δυνατοτήτων του Ethernet Shield, η βιβλιοθήκη DHT\_sensor\_library-1.4.3, που χρησιμοποιείται για τους αισθητήρες Dht11, η βιβλιοθήκη HCSR04\_ultrasonic\_sensor-2.0.3, που χρησιμοποιείται για τον αισθητήρα Ultrasonic HC-SR04, η βιβλιοθήκη Wire που επιτρέπει την επικοινωνία με συσκευές I2C και της βιβλιοθήκης DS3231 και έκδοση 1.1.0 που είναι για το ρολόι πραγματικού DS3231.

```

/*-----( Import needed libraries )-----*/
#include <SPI.h> //βιβλιοθήκη για την σειριακή επικοινωνία
#include <Ethernet.h> //Ethernet Shield
#include <DHT.h> //Βιβλιοθήκη για αισθητήρα Dht11
#include <HCSR04.h> //library Ultrasonic Sensor HC-SR04
#include <DS3231.h> //library for real time clock Rtc ds3231
/*-----(Set the Digital pin connected to the sensor )-----*/
#define SEN_Dht11 38 // Digital pin connected to the DHT sensor
#define SEN_UltraSonicTrigPin 48 //attach pin D48 Arduino to pin Trig of
HC-SR04
#define SEN_UltraSonicEchoPin 49 // attach pin D49 Arduino to pin Echo of
HC-SR04
unsigned int SEN_LdrOut = A9; //pin for Out Ldr Sensor
/*-----( Declare objects )-----*/
#define DHTTYPE DHT11 // DHT 11
DHT dht(SEN_Dht11, DHT11); // Select type Dht11
HCSR04 hc(SEN_UltraSonicTrigPin, SEN_UltraSonicEchoPin); //initialization
class HCSR04 (trig pin , echo pin)
DS3231 rtc(SDA, SCL);
Time t;
// Initialize Arduino Ethernet Client
//Set MAC - IP - Server for ETHERNET SETTINGS
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };//Ορισμός της
διεύθυνσης MAC του Arduino Ethernet shield
byte ip[] = {192, 168, 1, 8 }; //Enter the IP of ethernet shield Must be
unique on local network
byte serv[] = {192, 168, 1, 12} ; //Εισαγωγή της διεύθυνσης IPv4 address
του υπολογιστή που είναι ο Server
EthernetClient cliente; //subnet mask
/*-----( Declare Variables )-----*/
// defines variables
float Val_humi = 0; // Humidity variables
float Val_temp = 0;// Temperature variables
int Val_OutLdr = 0;
int Val_distance; // variable for the distance measurement
int Val_distancePerCent;// variable for the distance per cent measurement

// define the Xbee sensor variables for frame
unsigned int analogMSB1 = 0;
unsigned int analogLSB1 = 0;
unsigned int Val_SoilHygro = 0;

unsigned int analogMSB2 = 0;
unsigned int analogLSB2 = 0;
unsigned int Val_InLdr = 0;

unsigned int analogMSB3 = 0;
unsigned int analogLSB3 = 0;
unsigned int Val_Mq07Gas = 0;

```

```

void setup() {
  Serial.begin(9600); //setting the baud rate at 9600

  rtc.begin();// Start Rtc dc331 on Arduino

  dht.begin();// Start Dht11 sensor on Arduino
  Serial.println(F("DHT11 test!"));

  Serial.println(F("Ultrasonic Sensor HC-SR04 Test")); // print some text
  in Serial Monitor
  Serial.println(F("with Arduino UNO R3"));

  Ethernet.begin(mac, ip);// initialize Ethernet device

  pinMode(SEN_UltraSonicTrigPin, OUTPUT); // Sets the trigPin as an
  OUTPUT
  pinMode(SEN_UltraSonicEchoPin, INPUT); // Sets the echoPin as an INPUT
}

void loop() {
  Read_Sen_Dht11_Loop();
  Read_Sen_UltraSonic_Loop();
  Read_Sen_OutLDR_Loop();

  Read_RemoteXbee_Loop();
  Read_Rtc_Loop();

  Print_Serial_loop();
  Print_DataBase_loop();
}

//----Read the value of Real Time Clock-----
void Read_Rtc_Loop() {
  t = rtc.getTime();
}
// Read data Dht11 Data sensor
void Read_Sen_Dht11_Loop() {
  // read the humidity
  Val_humi = dht.readHumidity();
  // Read temperature as Celsius (the default)
  Val_temp = dht.readTemperature();
}
//----Read the distance at tank-----
void Read_Sen_UltraSonic_Loop(){
  Val_distance = hc.dist(); // Read Sensor ultrasonic data
  //Κάνουμε αντιστοίχιση των τιμών του αισθητήρα UltraSonic σε
  //εκατοστά σε εκατό τις εκατό για την καλύτεροί απεικόνιση της
  //στάθμης νερού στο δοχείο. Το δοχείο είναι 17 εκατοστά αρά
  //τότε θα έχει 0% νερό ενώ εάν είναι 0 εκ. τότε έχουμε 100%
  //νερό μέσα στο δοχείο.
  Val_distancePerCent = map(Val_distance, 17, 0, 0, 99);
}
//----Read the value of ldr out of greenhouse-----
----
void Read_Sen_OutLDR_Loop() {
  Val_OutLdr = analogRead(SEN_LdrOut);// read the analog data 0-1023 byte
  //Αντιστοιχίσει των Bytes σε επί τοις εκατό
  Val_OutLdr = map(Val_OutLdr, 400, 1030, 0, 100);
}
//----Read the sensor value of Router Xbee-----
-

```

```

void Read_RemoteXbee_Loop() {
    // έλεγχος εάν το frame έχει τουλάχιστον 29 bits
    if (Serial.available() >= 29) {
        // έλεγχος εάν έχει start byte 7E
        if (Serial.read() == 0x7E) {
            // Απόρριψη των πρώτων 21 byte
            for (int i = 1; i < 21; i++) {
                byte discardbyte = Serial.read();
            }
            //τα Πρώτα byte είναι η τιμή του Soilhygrometer
            // Read the most significant bit of the analog data
            analogMSB1 = Serial.read(); // Read the first analog byte
data
            analogLSB1 = Serial.read(); // Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
Val_SoilHygro = analogLSB1 + (analogMSB1 * 256);
//Αντιστοιχίσει των Byte σε επι τοις εκατό
Val_SoilHygro = map(Val_SoilHygro, 700, 1000, 100, 0);

            //τα Επόμενα byte είναι η τιμή του LDR
            analogMSB2 = Serial.read();// Read the first analog byte data
            analogLSB2 = Serial.read();// Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
Val_InLdr = analogLSB2 + (analogMSB2 * 256);
//Αντιστοιχίσει των Byte σε επι τοις εκατό
Val_InLdr = map(Val_InLdr, 0, 1030, 0, 100);
//τα τελευταία byte είναι η τιμή το Gas Sensor Mq07

            analogMSB3 = Serial.read();// Read the first analog byte data
            analogLSB3 = Serial.read();// Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
Val_Mq07Gas = analogLSB3 + (analogMSB3 * 256);
//Αντιστοιχίσει των Byte σε επι τοις εκατό
Val_Mq07Gas = map(Val_Mq07Gas, 0, 1030, 0, 100);
        }
    }
}

void Print_DataBase_loop(){
    if (cliente.connect(serv, 80)) { //Connecting at the IP address and
port we saved before
        Serial.println("connected");
        cliente.print("GET /db_allSensor/data.php?"); //Connecting and
Sending values to database

        //send the value to data.php
        cliente.print("temperature="); cliente.print(Val_temp);
        cliente.print("&humidity="); cliente.print(Val_humi);
        cliente.print("&outLdr="); cliente.print(Val_OutLdr);
        cliente.print("&tank="); cliente.print(Val_distance);
        cliente.print("&soilhygro="); cliente.print(Val_SoilHygro);
        cliente.print("&inLdr="); cliente.print(Val_InLdr);
        cliente.print("&Mq07Gas="); cliente.print(Val_Mq07Gas);

        cliente.println(" HTTP/1.1"); // Part of the GET request
        cliente.println("Host: 192.168.1.12");
        cliente.println("Connection: close"); // Part of the GET request
telling the server that we are over transmitting the message
    }
}

```



```

    cliente.println(); // Empty line
    cliente.println(); // Empty line
    cliente.stop();    // Closing connection to server
}else {
    // if you didn't get a connection to the server:
    Serial.println("connection failed");
}
delay(5000);
}
// Displays the value on the Serial Monitor
void Print_Serial_loop() {
    Serial.println(F("*****"));
    Serial.print(F("Temperature: ")); Serial.print(Val_temp);
Serial.println(F("°C"));
    Serial.print(F("Humidity: ")); Serial.print(Val_humi);
Serial.println(F("%"));
    Serial.print(F("LDR OUT Value: ")); Serial.print(Val_OutLdr);
Serial.println(F("%"));
    Serial.print(F("Distance: ")); Serial.print(Val_distancePerCent);
Serial.println(F("%"));
    Serial.print(F("Distance: ")); Serial.print(Val_distance);
Serial.println(F("cm"));
    Serial.print(F("Soil moisture: ")); Serial.print(Val_SoilHygro);
Serial.println(F("%"));
    Serial.print(F("Ldr IN Value: ")); Serial.print(Val_InLdr);
Serial.println(F("%"));
    Serial.print(F("Gas Value: ")); Serial.print(Val_Mq07Gas);
Serial.println(F("%"));
    Serial.print("Date : ");
    Serial.print(t.date); Serial.print("/"); Serial.print(t.mon);
Serial.print("/"); Serial.print(t.year); Serial.print("  ");
    Serial.print("Hour : ");
    Serial.print(t.hour); Serial.print(":"); Serial.print(t.min);
Serial.print(":"); Serial.println(t.sec);
    Serial.println(F("*****"));
}

// Loops For Xbee Part
// create the frame to send to router
void setRem_LED_RED_Org_ALERT_DIO4(char value) {
    Serial.write(0x7E); //0
    Serial.write(byte(0x0)); //1
    Serial.write(0x10); //2
    Serial.write(0x17); // 3
    Serial.write(byte(0x0)); //4

    Serial.write(byte(0x0)); //5
    Serial.write(byte(0x0)); //6
    Serial.write(byte(0x0)); //7
    Serial.write(byte(0x0)); //8
    Serial.write(byte(0x0)); //9
    Serial.write(byte(0x0)); //10
    Serial.write(0xFF); //12
    Serial.write(0xFF); // 13

    Serial.write(0xFF); //14
    Serial.write(0xFE); //15

    Serial.write(0x02); //16
    Serial.write('D'); //17
    Serial.write('4'); //18
    Serial.write(value); // 19
}

```

```

    long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'D' + '4' + value;
    Serial.write(0xFF - (sum & 0xFF));
}
void setRem_LED_Yellow_MOV_Sunshade_DIO11(char value) {
    Serial.write(0x7E); //start byte
    Serial.write(byte(0x0)); // high part of length (always zero)
    Serial.write(0x10); // low part of length (the number of bytes that
follow, not including check)
    Serial.write(0x17); //0x17 is a remote AT command
    Serial.write(byte(0x0)); //frame id set to zero for no reply
    //Id of recipient, or use 0xFFF for broadcast
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(0xFF); // 0xFF for broadcast
    Serial.write(0xFF); // 0xFF for broadcast

    //16 bit of recipient or 0xFFFE
    Serial.write(0xFF);
    Serial.write(0xFE);

    Serial.write(0x02); // 0x02 to apply changes immediately on remote

    //AT command name in ASCII character
    Serial.write('P');
    Serial.write('1');

    // command data in as many bytes as needed
    Serial.write(value);

    //checksum is all bytes after length bytes
    long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '1' + value;
    Serial.write(0xFF - (sum & 0xFF)); // calculate the proper checksum
}
void setRem_LED_Blue_Blue_ForggerSpraySytem_DIO12(char value) {
    Serial.write(0x7E);
    Serial.write(byte(0x0));
    Serial.write(0x10);
    Serial.write(0x17);
    Serial.write(byte(0x0));

    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(0xFF);
    Serial.write(0xFF);
    Serial.write(0xFF);
    Serial.write(0xFE);

    Serial.write(0x02);
    Serial.write('P');
    Serial.write('2');
    Serial.write(value);
    long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '2' + value;
    Serial.write(0xFF - (sum & 0xFF));
}

```

Ακολουθεί ο κώδικας connection.php για την σύνδεση στη βάση δεδομένων.

```
<?php
    $username = "root"; //εισαγωγή username της MySQL
    $pass = ""; //εισαγωγή password της mysql
    $host = "localhost";

    $db_name = "db_allSensor"; //εισαγωγή ονόματος βάσεις δεδομένων MySQL

//εισαγωγή μεταβλητής με τα στοιχεία συνδέσεις Host, username και
password
    $con = mysqli_connect ($host, $username, $pass);
//σύνδεση στην βάση δεδομένων db_allSensor με τα στοιχεία συνδέσεις της
μεταβλητής con
    $db = mysqli_select_db ( $con, $db_name );

    //έλεγχος εάν θα γίνει σύνδεση
    if(!$con){
        die("Connection failed: " . mysqli_connect_error());
    }
?>
```

Ακολουθεί ο κώδικας data.php για την εισαγωγή των δεδομένων από το Arduino στην βάση δεδομένων.

```
<?php
    include ('connection.php'); // Εισαγωγή αρχείου connection

    //εισαγωγή των παρακάτω εγγράφων στο πίνακα Data
    $sql_insert = "INSERT INTO data (temperature, humidity, outLdr, tank,
soilhygro, inLdr, Mq07Gas)
VALUES
('".$_GET["temperature"]."', '".$_GET["humidity"]."', '".$_GET["outLdr"]."',
'".$_GET["tank"]."', '".$_GET["soilhygro"]."', '".$_GET["inLdr"]."', '".$_G
ET["Mq07Gas"]."')";

    //έλεγχος εάν γίνει εισαγωγή εγγραφών
    if(mysqli_query($con,$sql_insert)){
        echo "Done";
        mysqli_close($con);
    }
    else{
        echo "error is ".mysqli_error($con );
    }
?>
```

Έπειτα από αυτό έχουμε όλα τα δεδομένα μας από τους αισθητήρες στη βάση δεδομένων.

	id	event	temperature	humidity	outLdr	tank	soilhygro	inLdr	Mg07Gas
<input type="checkbox"/>	1	2022-09-19 20:33:01	26.80	35.00	99	69	0	0	0
<input type="checkbox"/>	2	2022-09-19 20:33:09	26.70	35.00	99	69	0	0	0
<input type="checkbox"/>	3	2022-09-19 20:33:19	26.80	35.00	99	69	0	0	0
<input type="checkbox"/>	4	2022-09-19 20:33:24	26.80	35.00	99	69	0	0	0
<input type="checkbox"/>	5	2022-09-19 20:33:29	26.80	35.00	99	69	0	0	0
<input type="checkbox"/>	6	2022-09-19 20:33:35	26.70	35.00	99	69	0	0	0
<input type="checkbox"/>	7	2022-09-19 20:33:40	26.80	35.00	99	69	5	11	41
<input type="checkbox"/>	8	2022-09-19 20:33:45	26.70	35.00	99	69	5	11	41
<input type="checkbox"/>	9	2022-09-19 20:33:50	26.80	35.00	99	69	5	11	41
<input type="checkbox"/>	10	2022-09-19 20:33:56	26.70	35.00	99	69	5	11	41
<input type="checkbox"/>	11	2022-09-19 20:34:01	26.80	35.00	99	69	5	11	41
<input type="checkbox"/>	12	2022-09-19 20:34:06	26.70	35.00	99	69	5	11	41
<input type="checkbox"/>	13	2022-09-19 20:34:11	26.90	36.00	99	69	247	700	347
<input type="checkbox"/>	14	2022-09-19 20:34:17	26.80	36.00	99	69	247	700	347

Εικόνα 6.3 Πίνακας βάσεις δεδομένων

## 6.4 Δημιουργία αρχείου Index για την πλοήγηση στην ιστοσελίδα

Για την πλοήγηση στην ιστοσελίδα γίνεται χρήση του κώδικα index.html.

```

<!DOCTYPE html>
<html lang="el">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Smart Green House</title>
<link rel="stylesheet" href="../db_allSensor/stylepage/style3.css">
</head>
<body>
<h1>Smart Green House</h1>
<!-- Δημιουργία menu -->
<div class="menu-bar">
<ul>
<li class="active"><a href="index.html"><i class="fa-solid fa-house-blank"></i>HOME</a></li>
<li> <a href="http://192.168.1.7/"><i class="fa-solid fa-house-blank"></i>Control Area </a></li>
<li> <a href="menu/dbtable/display.php"><i class="fa-solid fa-house-blank"></i>Data sensor table</a></li>
<li> <a href="https://localhost/db_allsensor/menu/dbgauge/gauge.html"><i class="fa-solid fa-house-blank"></i>Gauge Visualization</a></li>
<li>
<a href="#">Graph</a>
<Div class="sub_menu-1">
<ul>
<li><a href="menu/dbgraph/graph.php">Temprature</a></li>
<li><a href="menu/dbgraph/humgraph.php">Humidity</a></li>
<li><a href="menu/dbgraph/outldr.php">Ldr out</a></li>
<li><a href="menu/dbgraph/tankgraph.php">Level Tank</a></li>
<li><a href="menu/dbgraph/soilhygrograph.php">Soil Hugro</a></li>
<li><a href="menu/dbgraph/inldrgraph.php">Ldr IN</a></li>
<li><a href="menu/dbgraph/gas.php">Gas</a></li>
</ul>
</Div>
</li>

```

```

</li>
<a href="#">ThingSpeak</a>
<div class="sub_menu-1">
<ul>
<li><a href="menu/thingspeak/graph.html">Graph</a></li>
<li><a href="menu/thingspeak/NumericDisplay.html">NumericDisplay</a></li>
</ul>
</div>
</li>
</ul>
</div>
<!-- Προσθήκη εικόνας -->
<div>
<ul>
<p class="image">
<br>

</p>
</ul>
</div>
<!-- Κείμενο -->
<div>
<p class="BorderText">
Στην διπλωματική αυτή θα υλοποιηθεί ένα IoT θερμοκήπιο που θα έχει ως στόχο των έλεγχο και την παρακολούθηση των κλιματικών συνθηκών για την καλύτερη δυνατή ανάπτυξη των φυτών του θερμοκηπίου αλλά και την εξοικονόμηση των πόρων.
<br>
Αρχικά θα πρέπει να αναφερθεί ότι το λειτουργικό κομμάτι του θερμοκηπίου βασίζεται στην πλατφόρμα Arduino και στις διάφορες συσκευές που θα συνδεθούν σε αυτόν όπως είναι ο αισθητήρας θερμοκρασίας και υγρασίας, ο αισθητήρας φωτοαντίστασης, ο αισθητήρας απόστασης, οι συσκευές εξόδου, οι συσκευές Xbee, το Ethernet shield και άλλα.
<br>
Επιπλέον, χρησιμοποιώντας τα δύο Xbee γίνεται χρήση ασύρματης επικοινωνίας ZigBee. Επομένως η μια συσκευή Xbee λειτουργεί ως Coordinator και η άλλη συσκευή λειτουργεί ως Router. Συνεπώς το Arduino θα επικοινωνεί απευθείας μέσω του πρωτοκόλλου UART με το Xbee coordinator ενώ η δεύτερη συσκευή Xbee (Router) θα στέλνει και θα λαμβάνει δεδομένα από Xbee Coordinator μέσω του πρωτοκόλλου Zigbee. Στο Xbee Router θα συνδεθούν ο αισθητήρας υγρασίας χρώματος, ο αισθητήρας αερίου, ο δεύτερος αισθητήρας φωτοαντίστασης και τρία LED.
<br>
Επιπρόσθετα, θα υπάρχει η δυνατότητα απομακρυσμένου ελέγχου αλλά και παρακολούθησης μέσω του τοπικού μας δικτύου από την ιστοσελίδα που θα υλοποιηθεί. Για να είναι εφικτό αυτό χρειάζεται η σύνδεση ενός Ethernet Shield με το Arduino ώστε να λειτουργεί ως Web Server και η δημιουργία μιας βάσης δεδομένων. Επομένως για να γίνει η οπτικοποίηση των δεδομένων από τους αισθητήρες στην ιστοσελίδα μας χρειάζεται η δημιουργία της βάσης δεδομένων και από εκεί να γίνει η οπτικοποίηση τους μέσω ιστοσελίδων με την βοήθεια της php και των google charts.
<br>
Επίσης, μέσω της ιστοσελίδας θα υπάρχει η επιλογή της λειτουργίας του συστήματος θερμοκηπίου σε δύο καταστάσεις, είτε στην χειροκίνητη λειτουργία, είτε στην αυτόματη λειτουργία.
<br>
Στην χειροκίνητη λειτουργία ο χρήστης έχει την δυνατότητα να ενεργοποιεί και να απενεργοποιεί κάθε συσκευή εξόδου ξεχωριστά. Για να γίνει ο έλεγχος χειροκίνητα μέσω της ιστοσελίδας χρειάζεται να χρησιμοποιήσουμε της μεθόδους HTTP POST και GET έτσι θα μπορούμε να ελέγχουμε το πότισμα, τον εξαερισμό, την θέρμανση, την ψύξη, τον φωτισμό αλλά θα μπορούμε και να

```

ελέγχουμε τα τρία LED που είναι συνδεδεμένα στην συσκευή Xbee 2 που εξομοιώνουν το σύστημα ψεκασμού ομίχλης από νερό, το σύστημα σκίασης και το σύστημα ειδοποίησης.

<br>

Στην αυτόματη λειτουργία θα εκτελούνται τα σενάρια αυτοματισμού για την διευκόλυνση των εργασιών που απαιτούνται σε ένα θερμοκήπιο όπως είναι ο έλεγχος θερμοκρασίας, υγρασίας, ποτίσματος, φωτισμού, ηλιακής ακτινοβολίας και ο έλεγχος της ασφάλειας του θερμοκηπίου.

<br>

Όλα αυτά θα ενσωματωθούν σε μια κατασκευή: που θα απεικονίζει ένα θερμοκήπιο με το σύστημα ποτίσματος και 2 μικρότερες κατασκευές η πρώτη κατασκευή θα περιέχει το Arduino mega, το Xbee coordinator και τις συσκευές εισόδου εξόδου που συνδέονται σε αυτό και στη δεύτερη κατασκευή θα περιέχει το Xbee router και τις συσκευές εισόδου εξόδου που συνδέονται σε αυτό.

<br>

</p>

</div>

</body>

</html>

Το αποτέλεσμα του κώδικα index.html μαζί με την χρήση του αρχείου style3.css.



Εικόνα 6.4 Αρχική σελίδα της ιστοσελίδας.

## 6.5 Οπτικοποίηση δεδομένων σε μορφή πίνακα

Έχοντας τα δεδομένα στην βάση δεδομένων θα δημιουργήσουμε έναν πίνακα για οπτικοποίηση της βάσεις δεδομένων στην ιστοσελίδα χρησιμοποιώντας των παρακάτω κώδικα display.php.

```
<?php
    $url=$_SERVER['REQUEST_URI'];
    header("Refresh: 5; URL=$url"); // Refresh the webpage every 5
seconds
?>
<!DOCTYPE html>
<html lang="el">
<head>
```



```

<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Arduino DataBase sensors.</title>
<link rel="stylesheet" href="../../stylepage/style3.css">
</head>
<body>
<h1>Arduino Data sensor Logging to Database</h1>
<!-- Δημιουργία menu -->
<div class="menu-bar">
<ul>
<li class="active"><a href="http://localhost/db_allSensor/"><i class="fa-solid fa-house-blank"></i>HOME</a></li>
<li> <a href="http://192.168.1.7/"><i class="fa-solid fa-house-blank"></i>Dashboard </a></li>
<li><a href="display.php"><i class="fa-solid fa-house-blank"></i>Data sensor table</a></li>
<li> <a href="https://localhost/db_allsensor/menu/dbgauge/gauge.html"><i class="fa-solid fa-house-blank"></i>Gauge Visualization</a></li>
<li>
<a href="#">Graph</a>
<Div class="sub_menu-1">
<ul>
<li><a href="../../menu/dbgraph/graph.php">Temprature</a></li>
<li><a href="../../menu/dbgraph/humgraph.php">Humidity</a></li>
<li><a href="../../menu/dbgraph/outldr.php">Ldr out</a></li>
<li><a href="../../menu/dbgraph/tankgraph.php">Level Tank</a></li>
<li><a href="../../menu/dbgraph/soilhygrograph.php">Soil Hugro</a></li>
<li><a href="../../menu/dbgraph/inldrgraph.php">Ldr IN</a></li>
<li><a href="../../menu/dbgraph/gas.php">Gas</a></li>
</ul>
</Div>
</li>
<li>
<a href="#">ThingSpeak</a>
<div class="sub_menu-1">
<ul>
<li><a href="../../menu/thingspeak/graph.html">Graph</a></li>
<li><a href="../../menu/thingspeak/NumericDisplay.html">NumericDisplay</a></li>
</ul>
</div>
</li>
</ul>
</div>
<!-- Δημιουργία Πίνακα-->
<table>
<!--Τίτλος πίνακα-->
<tr>
<td class="table_titles">ID</td>
<td class="table_titles">Date and Time</td>
<td class="table_titles">Temperature</td>
<td class="table_titles">Humidity</td>
<td class="table_titles">Out Ldr</td>
<td class="table_titles">Level at Water tank</td>
<td class="table_titles">Soil Hygro</td>
<td class="table_titles">In Ldr</td>
<td class="table_titles">Mq-07 Gas</td>
</tr>
<?php
//Εισαγωγή του αρχείου connection.php για να γίνει σύνδεση με την βάση δεδομένων

```

```

include('C:\xampp\htdocs\db_allSensor\connection.php');
$result = mysqli_query($con, 'SELECT * FROM data ORDER BY id DESC');
// Process every record
$row = true;
while($row = mysqli_fetch_array($result)){
if ($row){
$css_class=' class="table_cells_odd"';
}
else{
$css_class=' class="table_cells_even"';
}
$row = !$row;
//καταγραφή των δεδομένων των αισθητήρων στον πίνακα
echo "<tr>";
echo "<td '." . $row['id'] . "</td>";
echo "<td '." . $row['event'] . "</td>";
echo "<td '." . $row['temperature'] . "</td>";
echo "<td '." . $row['humidity'] . "</td>";
echo "<td '." . $row['outLdr'] . "</td>";
echo "<td '." . $row['tank'] . "</td>";
echo "<td '." . $row['soilhygro'] . "</td>";
echo "<td '." . $row['inLdr'] . "</td>";
echo "<td '." . $row['Mq07Gas'] . "</td>";
echo "</tr>";
}
// Close the connection
mysqli_close($con);
?>
</table>
</body>
</html>

```

Το αποτέλεσμα από τον παραπάνω κώδικα μαζί με την χρήση του αρχείου style3.css.

ID	Date and Time	Temperature	Humidity	Out Ldr	Level at Water tank	Soil Hygro	In Ldr	Mq-07 Gas
2567	2022-08-22 21:08:37	29.20	63.00	98	2	19	14	44
2566	2022-08-22 21:08:37	29.20	63.00	98	2	19	14	44
2565	2022-08-22 21:08:36	29.20	63.00	98	2	19	14	44
2564	2022-08-22 21:08:36	29.20	63.00	98	2	19	14	44
2563	2022-08-22 21:08:36	29.20	63.00	98	2	19	14	44
2562	2022-08-22 21:08:35	29.20	63.00	98	2	19	14	44
2561	2022-08-22 21:08:35	29.20	63.00	98	2	19	14	44
2560	2022-08-22 21:08:35	29.10	63.00	98	2	19	14	44
2559	2022-08-22 21:08:34	29.10	63.00	98	2	19	14	44
2558	2022-08-22 21:08:34	29.10	63.00	98	2	19	14	44
2557	2022-08-22 21:08:34	29.10	63.00	98	2	19	14	44
2556	2022-08-22 21:08:34	29.10	63.00	98	2	19	14	44
2555	2022-08-22 21:08:33	29.10	63.00	98	2	19	14	44
2554	2022-08-22 21:08:33	29.10	63.00	98	2	19	14	44
2553	2022-08-22 21:08:33	29.20	63.00	98	2	19	14	44
2552	2022-08-22 21:08:32	29.20	63.00	98	2	19	14	44
2551	2022-08-22 21:08:32	29.20	63.00	98	2	19	14	44
2550	2022-08-22 21:08:32	29.20	63.00	98	2	19	14	44
2549	2022-08-22 21:08:32	29.20	63.00	98	2	19	14	44
2548	2022-08-22 21:08:31	29.20	63.00	98	2	19	14	44
2547	2022-08-22 21:08:31	29.20	63.00	98	2	19	14	44
2546	2022-08-22 21:08:31	29.60	63.00	98	2	19	14	44
2545	2022-08-22 21:08:30	29.60	63.00	98	2	19	14	44
2544	2022-08-22 21:08:30	29.60	63.00	98	2	19	14	44
2543	2022-08-22 21:08:30	29.60	63.00	98	2	19	14	44
2542	2022-08-22 21:08:30	29.60	63.00	98	2	19	14	44
2541	2022-08-22 21:08:29	29.60	63.00	98	2	19	14	44
2540	2022-08-22 21:08:29	29.60	63.00	98	2	19	14	44
2539	2022-08-22 21:08:29	29.50	63.00	98	2	19	14	44

Εικόνα 6.5 Στιγμιότυπο θόνης ιστοσελίδας όπου απεικονίζεται ο πίνακας δεδομένων.

## 6.6 Οπτικοποίηση δεδομένων σε μορφή γραφικών πραγματικού χρόνου

Στην ιστοσελίδα έχει δημιουργηθεί οπτικοποίηση των δεδομένων από τους αισθητήρες σε μορφή γραφικών πραγματικού χρόνου. Αυτά τα δεδομένα μέσω του ethernet shield και

έπειτα μέσω του δρομολογητή στέλνονται στην βάση δεδομένων MySQL και από εκεί με την βοήθεια των google charts έχουμε την οπτικοποίηση των γραφικών στην ιστοσελίδα [Εικόνα 6.6](#).

Ακολουθεί ο κώδικας graph.php για την εμφάνιση της γραφικής με τα δεδομένα του αισθητήρα θερμοκρασίας DHT11.

```
<!--Σύνδεση με την βάση δεδομένων-->
<?php
//Στοιχεία σύνδεσης
$conn = mysqli_connect("localhost", "root", "", "db_allSensor");
$query = "SELECT temperature, event FROM data";
$result = mysqli_query($conn, $query);
$url=$_SERVER['REQUEST_URI'];
header("Refresh: 5; URL=$url"); // Refresh the webpage every 5 seconds
?>

<!DOCTYPE html>
<html lang="el">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Graph</title>
    <link rel="stylesheet" href="../../stylepage/style3.css">
    <link rel="stylesheet" href="../../stylepage/btnstyle.css">
</head>
<html>
<head>
<!--Προσθήκη βιβλιοθήκης-->
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
//επιλογή εμφάνισης γραφικής
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawChart);
function drawChart() {
//Επιλογή δεδομένων από τον πίνακα για την εμφάνιση στην γραφική
var data = google.visualization.arrayToDataTable([
['event', 'temperature'],
<?php
while($row = mysqli_fetch_array($result)) {
echo "[".$row["event"].", ".$row["temperature"].",";
}
?>
]);
//Επιλογές εμφάνισης γραφικής
var options = {
areaOpacity:1.0,
vAxis: { ticks: [5,10,15,20] },
chartArea:{left:80,top:80,width:'90%',height:'50%'},
lineWidth: 4,
hAxis: {
title: 'Time',
```

```

textStyle: {
color: '#097138',
fontSize: 24,
},
titleTextStyle: {
color: '#097138',
fontSize: 20,
bold: true
}
},
vAxis: {
title: 'Temperature °C',
textStyle: {
color: '#097138',
fontSize: 24,
bold: true
},
titleTextStyle: {
color: '#097138',
fontSize: 24,
bold: true
},
viewWindowMode: 'explicit',
viewWindow: {
max: 50,
min: 0
}
},
colors: ['#097138', '#097138'],
background-color: '#f1f8e9'
};

var chart = new
google.visualization.LineChart(document.getElementById('curve_chart'));
chart.draw(data, options);
}
</script>
</head>
<body>
<h1>TEMPERATURE GRAPH FOR TEMPERATURE OF DHT11 SENSOR</h1>
<!-- Δημιουργία menu -->
<div class="menu-bar">
<ul>
<li class="active"><a
href="http://192.168.1.12/db_allSensor/index.html"><i class="fa-solid fa-
house-blank"></i>HOME</a></li>
<li> <a href="http://192.168.1.7/"><i class="fa-solid fa-house-
blank"></i>Control Area </a></li>
<li> <a href="../../menu/dbtable/display.php"><i class="fa-solid fa-
house-blank"></i>Data sensor table</a></li>
<li> <a href="https://localhost/db_allsensor/menu/dbgauge/gauge.html"><i
class="fa-solid fa-house-blank"></i>Gauge Visualization</a></li>
<li>
<a href="#">Graph</a>
<Div class="sub_menu-1">
<ul>
<li><a href="../../menu/dbgraph/graph.php">Temprature</a></li>
<li><a href="../../menu/dbgraph/humgraph.php">Humidity</a></li>
<li><a href="../../menu/dbgraph/outldr.php">Ldr out</a></li>
<li><a href="../../menu/dbgraph/tankgraph.php">Level Tank</a></li>
<li><a href="../../menu/dbgraph/soilhygrograph.php">Soil Hugro</a></li>
<li><a href="../../menu/dbgraph/inldrgraph.php">Ldr IN</a></li>

```

```

<li><a href="../../../menu/dbgraph/gas.php">Gas</a></li>
</ul>
</Div>
</li>

<li>
<a href="#">ThingSpeak</a>
<div class="sub_menu-1">
<ul>
<li><a href="../../../menu/thingspeak/graph.html">Graph</a></li>
<li><a href="../../../menu/thingspeak/NumericDisplay.html">Numeric
Display</a></li>
</ul>
</div>
</li>
</ul>
</div>

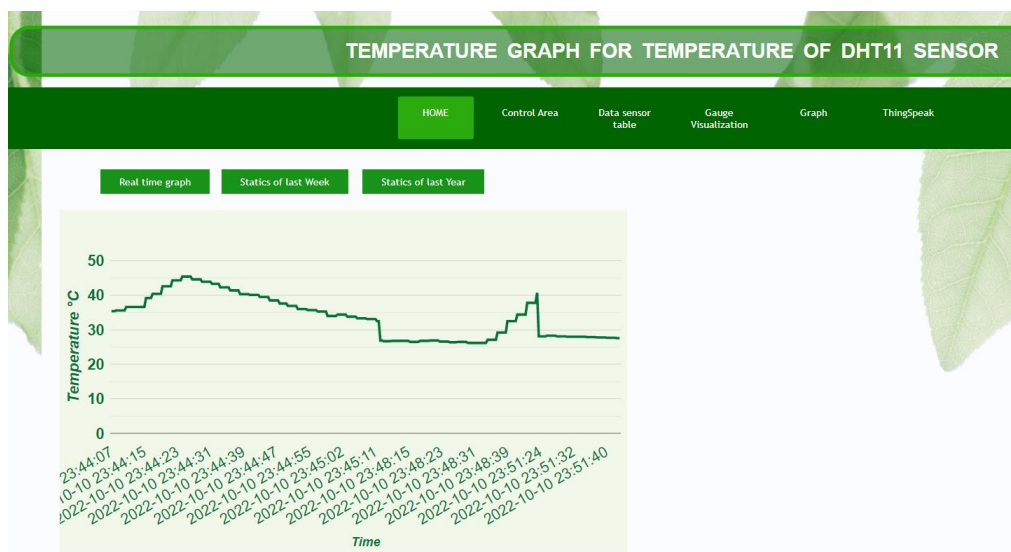
<!--Link για την εμφάνιση της γραφικής παράστασης της θερμοκρασία σε
πραγματικό χρόνο, για την τελευταία εβδομάδα και για τον τελευταίο χρόνο-->
<ul>
<li><a class="livebtn" href="../dbgraph/graph.php">Real time
graph</a></li>
<li><a class="daybtn" href="../Statics/WeeklyStats.php">Statics of last
Week</a></li>
<li><a class="monthbtn" href="../Statics/YearlyStats.php">Statics of last
Year</a></li>
</ul>

<div id="curve_chart" class="graph" style="width: 1400px; height:
550px"></div>

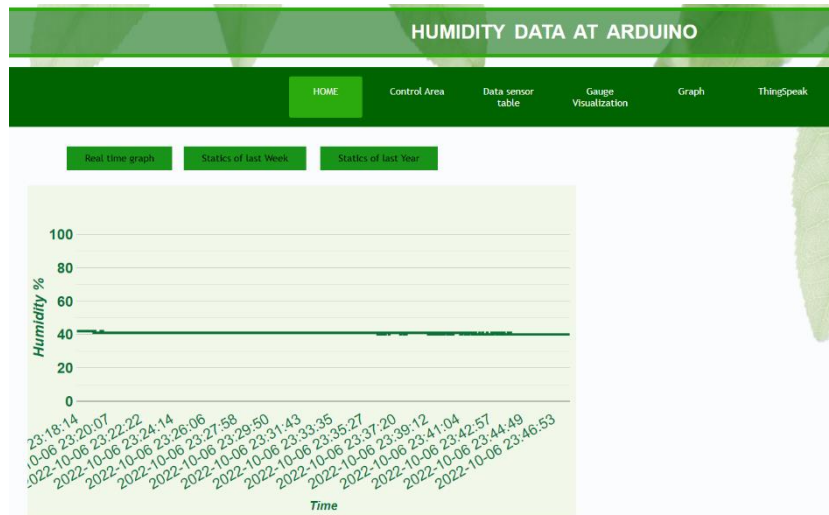
</body>
</html>

```

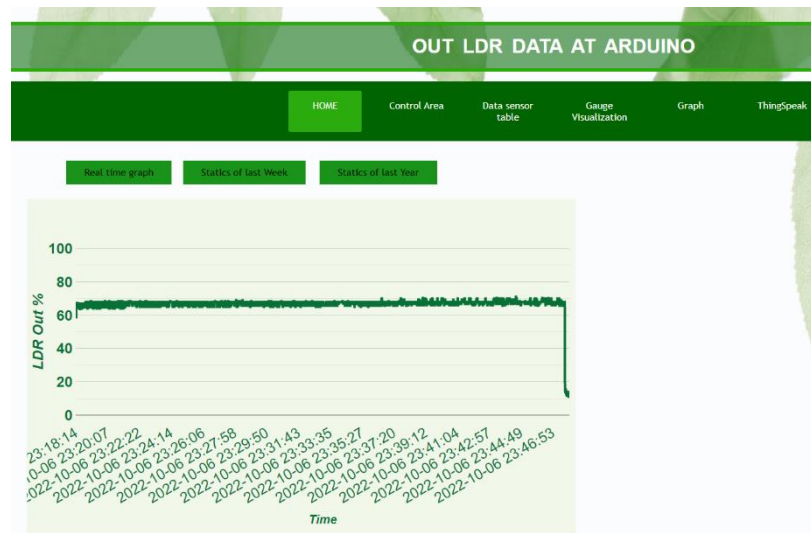
Το αποτέλεσμα του κώδικα μαζί με την χρήση του αρχείου style3.css.



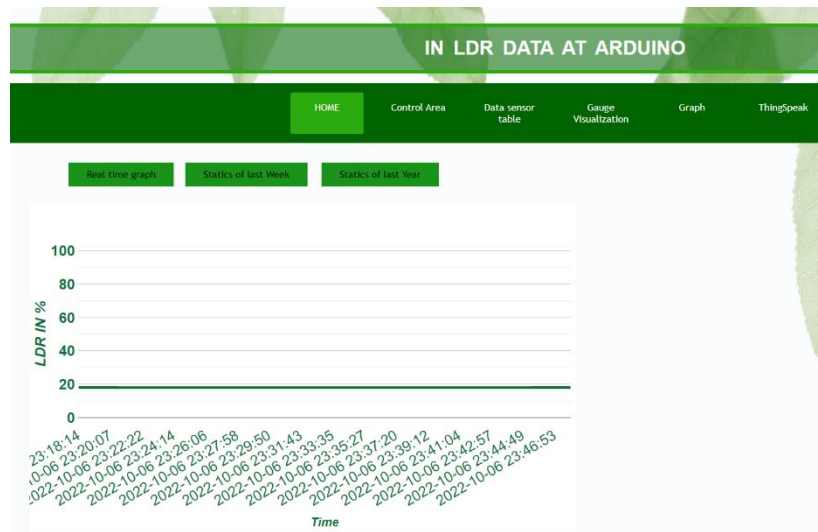
Εικόνα 6.6 Γραφική παράσταση της θερμοκρασίας.



Εικόνα 6.7 Γραφική παράσταση υγρασίας.

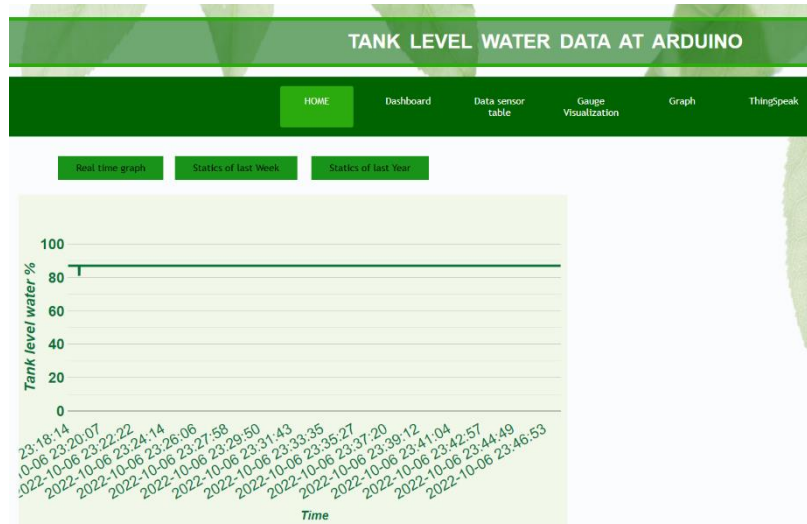


Εικόνα 6.8 Γραφική παράσταση εξωτερικής φωτοαντίστασης.

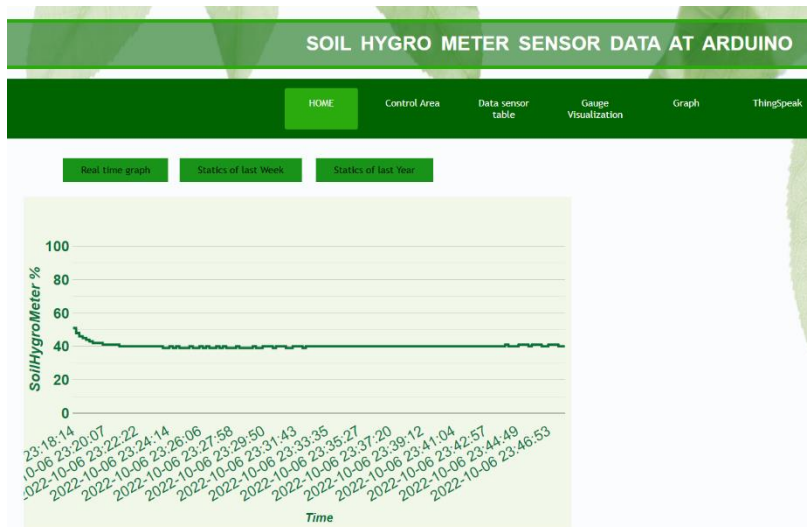


Εικόνα 6.9 Γραφική παράσταση εσωτερικής φωτοαντίστασης.

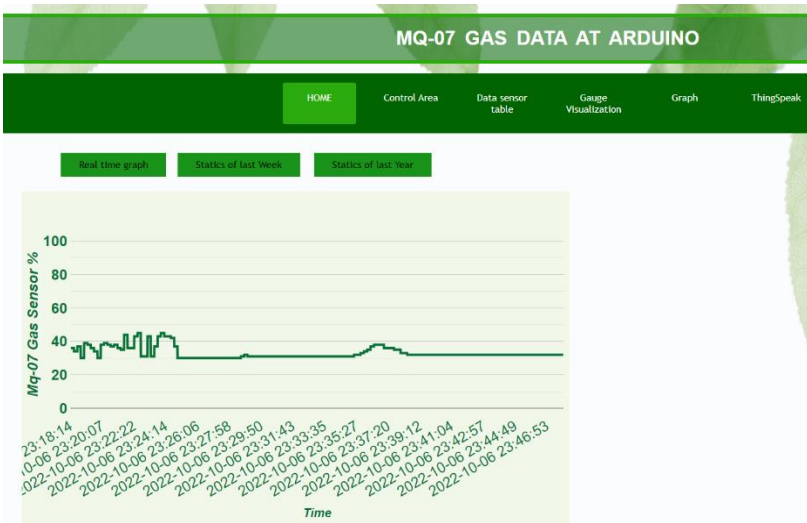




Εικόνα 6.10 Γραφική παράσταση επιπέδου νερού.



Εικόνα 6.11 Γραφική παράσταση υγρασίας εδάφους.



Εικόνα 6.12 Γραφική παράσταση διοξειδίου του άνθρακα.

## 6.7 Οπτικοποίηση στατιστικών

Εκτός από την οπτικοποίηση των διαγραμμάτων σε πραγματικό χρόνο υπάρχει η δυνατότητα οπτικοποίησης των δεδομένων της βάσης δεδομένων σε διάγραμμα εβδομαδιαίων στατιστικών και μηνιαίων στατιστικών. Επίσης τα διαγράμματα εβδομαδιαίων και μηνιαίων στατιστικών θα απεικονίζουν τα δεδομένα του αισθητήρα DHT11 θερμοκρασίας και υγρασίας, τα δεδομένα από το εσωτερικό και εξωτερικό LDR, τα δεδομένα από των αισθητήρα υγρασίας εδάφους, τα δεδομένα από των αισθητήρα απόστασης για της διακύμανσης του επιπέδου νερού στο δοχείο και τα δεδομένα από τον αισθητήρα διοξειδίου του άνθρακα.

Τα διαγράμματα εβδομαδιαίων στατιστικών θα απεικονίσουν την υψηλότερη τιμή, την χαμηλότερη τιμή και την μεσαία τιμή για την τελευταία εβδομάδα.

Ακολουθεί ο κώδικας WeeklyStats.php σε συνδυασμό με των κώδικα css btnstyle.css για το γράφημα εβδομαδιαίων στατιστικών της θερμοκρασίας.

```
<!--Κώδικας php για την σύνδεση στην βάση δεδομένων και την άντληση
δεδομένων -->
<?php

// Σύνδεση με την βάση δεδομένων
$conn = mysqli_connect("localhost", "root", "", "db_allSensor");

//Δημιουργία του query sql για τους τελευταίους μήνες του τρέχουν έτους,
του μεγίστου, του ελάχιστου και της μέσης τιμή
$query = "SELECT
MAX(temperature),
MIN(temperature),
AVG(temperature),
dayname(event) as dayev
FROM data
WHERE
YEARWEEK(`event`, 1) = YEARWEEK( CURDATE() - INTERVAL 1 WEEK, 1)
group by day(event)
ORDER BY event ASC";

//δημιουργία μεταβλητής result για σύνδεση της σύνδεσης με το query
$result = mysqli_query($conn, $query);

// Refresh the webpage every 5 seconds
$url=$_SERVER['REQUEST_URI'];
header("Refresh: 5; URL=$url");
?>

<!--Κώδικας Html-->
<!DOCTYPE html>
<html lang="el">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Graph</title>
<link rel="stylesheet" href="../../stylepage/style3.css">
<link rel="stylesheet" href="../../stylepage/btnstyle.css">
</head>
<html>
<head>

<!--Προσθήκη βιβλιοθήκης-->
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
google.charts.load('current', {'packages':['bar']});
google.charts.setOnLoadCallback(drawChart);

function drawChart() {
var data = google.visualization.arrayToDataTable([
['Event', 'MIN', 'Max', 'AVG'],
<?php
while($row = mysqli_fetch_array($result)){

//Δημιουργία γραφικής
echo "
[
'".$row["dayev"]."',
'".$row["MIN(temperature)"]."',
'".$row["MAX(temperature)"]."',
'".$row["AVG(temperature)"]."',
],";
}
?>
]);

//Ρύθμισης εμφάνισης γραφήματος
var options = {
chart: {
title: 'Statics Temperature (°C ) of days for the last week',
subtitle: 'Max, Min, and Avg',
},
vAxis: {
title: 'Temperature °C',
textStyle: {
color: '#097138',
fontSize: 24,
}
},
hAxis: {
title: 'Month',
textStyle: {
color: '#097138',
fontSize: 24,
}
},
colors: ['#006400', '#6B8E23', '#228B22'],
backgroundColor: '#f1f8e9'
};

var chart = new
google.charts.Bar(document.getElementById('columnchart_material'));

chart.draw(data, google.charts.Bar.convertOptions(options));

```

```

}

</script>
</head>
<body>
<h1>TEMPERATURE GRAPH FOR TEMPERATURE OF DHT11 SENSOR</h1>

<!-- Δημιουργία menu -->
<div class="menu-bar">
<ul>

<li class="active"><a
href="http://192.168.1.12/db_allSensor/index.html"><i class="fa-solid fa-
house-blank"></i>HOME</a></li>

<li> <a href="http://192.168.1.7/"><i class="fa-solid fa-house-
blank"></i>Control Area </a></li>
<li> <a href="../../../menu/dbtable/display.php"><i class="fa-solid fa-
house-blank"></i>Data sensor table</a></li>

<li> <a href="https://localhost/db_allsensor/menu/dbgauge/gauge.html"><i
class="fa-solid fa-house-blank"></i>Gauge Visualization</a></li>
<li>

<a href="#">Graph</a>
<Div class="sub_menu-1">
<ul>
<li><a href="../../../menu/dbgraph/graph.php">Temprature</a></li>

<li><a href="../../../menu/dbgraph/humgraph.php">Humidity</a></li>

<li><a href="../../../menu/dbgraph/outldr.php">Ldr out</a></li>

<li><a href="../../../menu/dbgraph/tankgraph.php">Level Tank</a></li>

<li><a href="../../../menu/dbgraph/soilhygrograph.php">Soil Hugro</a></li>

<li><a href="../../../menu/dbgraph/inldrgraph.php">Ldr IN</a></li>

<li><a href="../../../menu/dbgraph/gas.php">Gas</a></li>

</ul>

</Div>
</li>
<li>

<a href="#">ThingSpeak</a>
<div class="sub_menu-1">
<ul>

<li><a href="../../../menu/thingspeak/graph.html">Graph</a></li>

<li><a href="../../../menu/thingspeak/NumericDisplay.html">Numeric
Display</a></li>

</ul>
</div>
</li>
</ul>
</div>

```

```

<ul>

<!--Link για την εμφάνιση της γραφικής παράστασης της θερμοκρασία σε
πραγματικό χρόνο, για την τελευταία εβδομάδα και για τον τελευταίο χρόνο-->
<ul>

<li><a class="livebtn" href="../dbgraph/graph.php">Real time
graph</a></li>

<li><a class="daybtn" href="../Statics/WeeklyStats.php">Statics of last
Week</a></li>

<li><a class="monthbtn" href="../Statics/YearlyStats.php">Statics of last
Year</a></li>

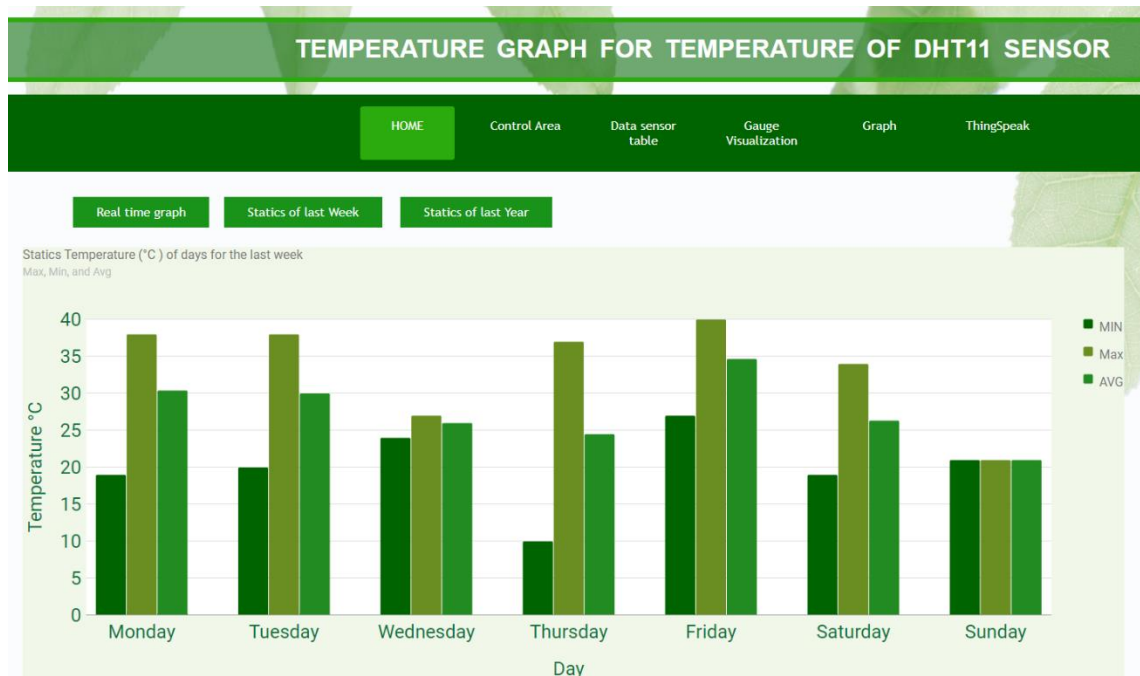
</ul>

<!-- Εμφάνιση γραφήματος -->
<div id="columnchart_material" class="graph" style="width: 1400px;
height: 550px;"></div>

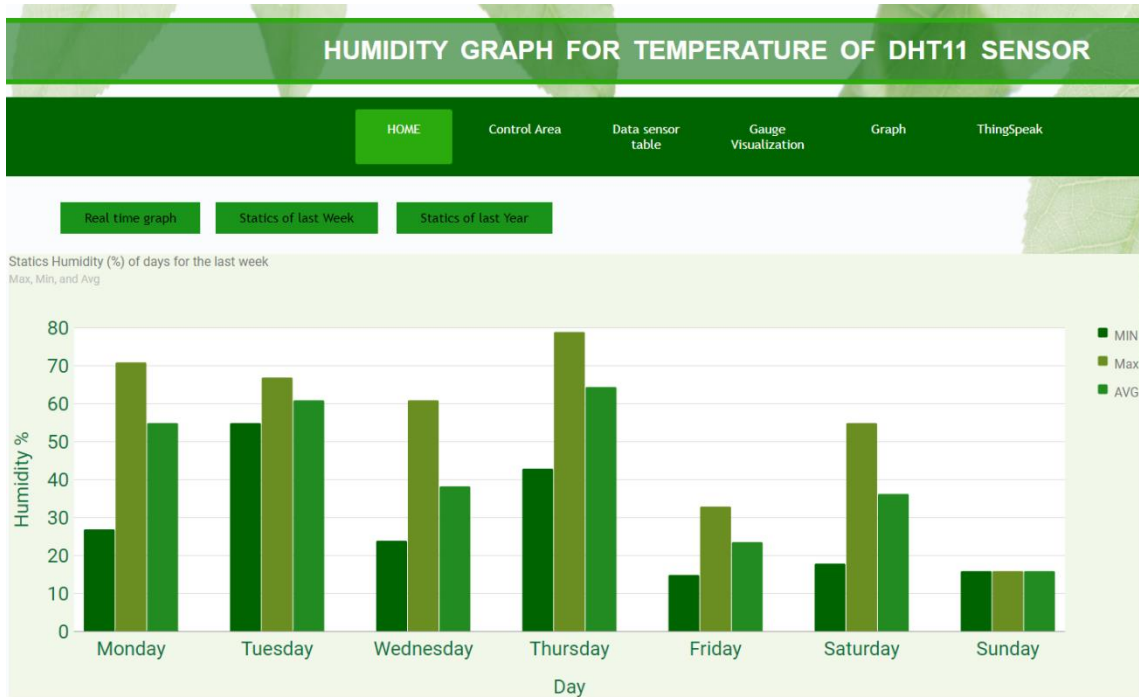
</body>
</html>

```

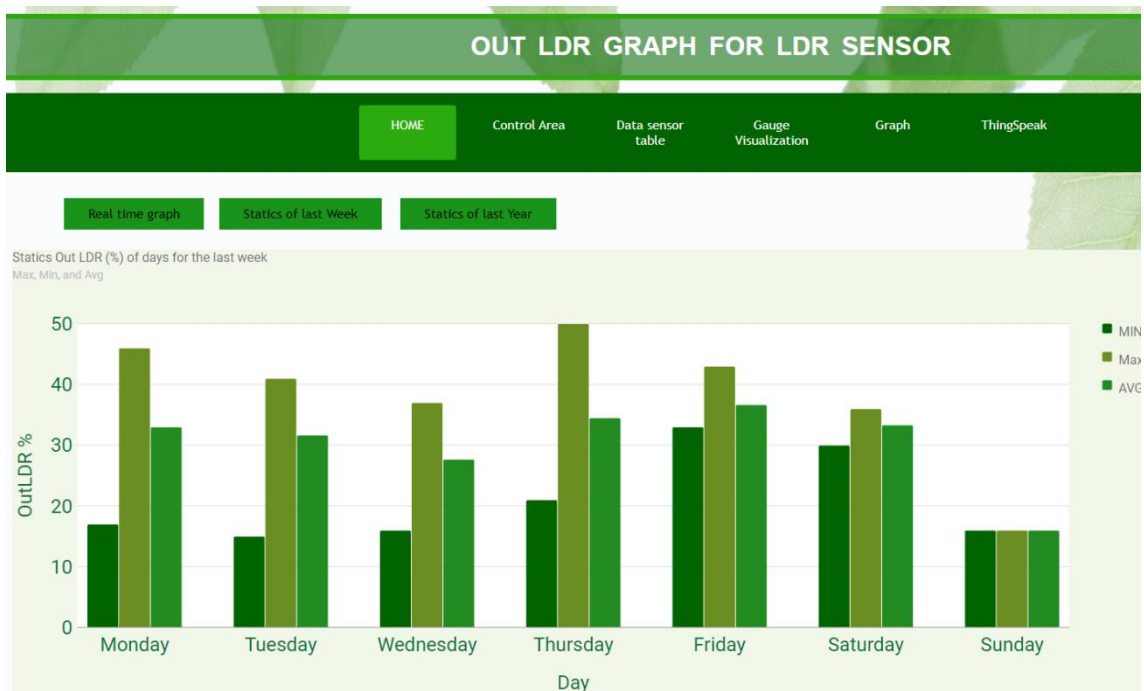
Το αποτέλεσμα του κώδικα είναι η [Εικόνα 6.13](#).



Εικόνα 6.13 Γράφημα εβδομαδιαίων στατιστικών θερμοκρασίας.

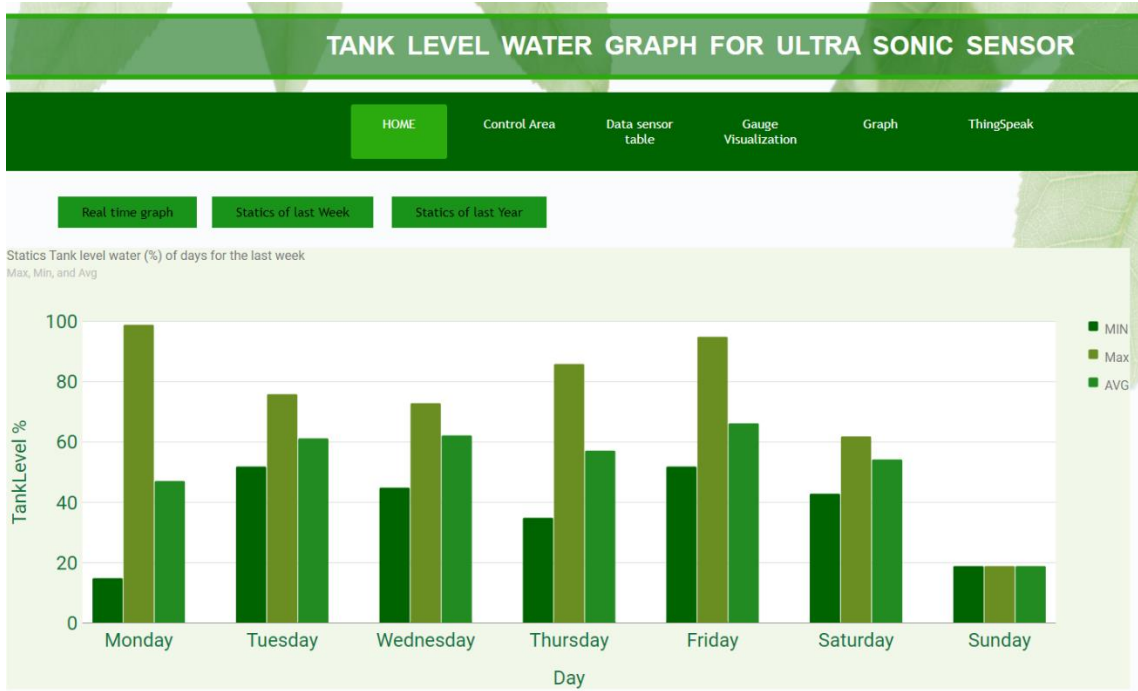


Εικόνα 6.14 Γράφημα εβδομαδιαίων στατιστικών υγρασίας.

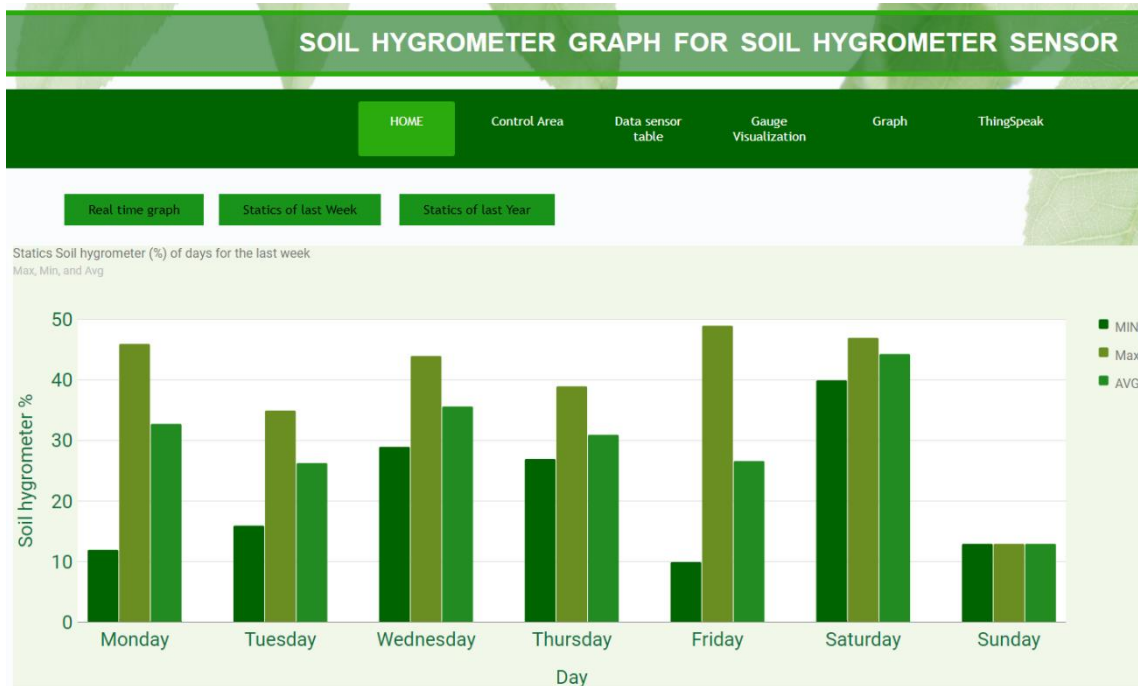


Εικόνα 6.15 Γράφημα εβδομαδιαίων στατιστικών εξωτερικής φωτοαντίστασης.

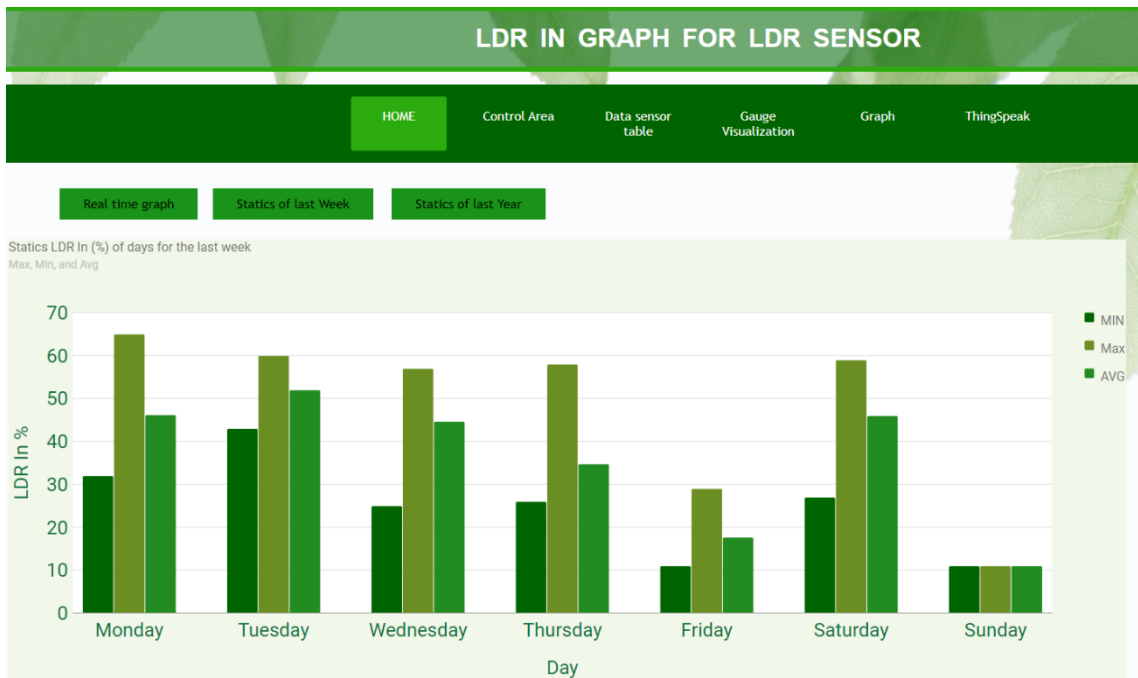




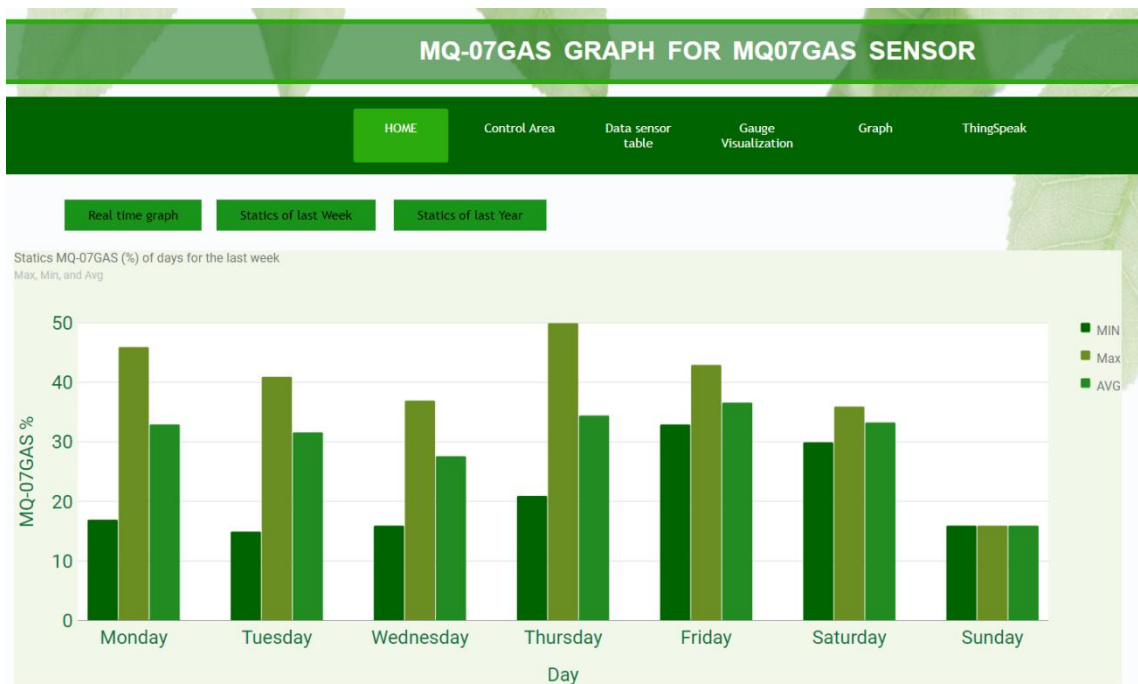
*Εικόνα 6.16 Γράφημα εβδομαδιαίων στατιστικών επιπέδου νερού.*



*Εικόνα 6.17 Γράφημα εβδομαδιαίων στατιστικών υγρασίας εδάφους.*



Εικόνα 6.18 Γράφημα εβδομαδιαίων στατιστικών εσωτερικής φωτοαντίστασης.



Εικόνα 6.19 Γράφημα εβδομαδιαίων στατιστικών διοξειδίου του άνθρακα.

Τα διαγράμματα μηνιαίων στατιστικών θα απεικονίσουν την υψηλότερη τιμή, την χαμηλότερη τιμή και την μεσαία τιμή για τους τελευταίους μήνες του τρέχουν έτους.

Ακολουθεί ο κώδικας YearlyStats.php σε συνδυασμό με τον κώδικα css btnstyle.css για το γράφημα μηνιαίων στατιστικών της θερμοκρασίας.

```

<!--Κώδικας php για την σύνδεση στην βάση δεδομένων και την άντληση
δεδομένων -->
<?php

// Σύνδεση με την βάση δεδομένων
$conn = mysqli_connect("localhost", "root", "", "db_allSensor");
//Δημιουργία του query sql για την εύρεση των ημερών της τελευταίας
εβδομάδας, του μεγίστου, του ελάχιστου και της μέσης τιμή
$query = "SELECT
Date(event) as dateev,
MAX(temperature),
MIN(temperature),
AVG(temperature)
from data
group by MONTHNAME(event)
ORDER BY event DESC
limit 12";

$result = mysqli_query($conn, $query); //δημιουργία μεταβλητής result
για σύνδεση της σύνδεσης με το query

// Refresh the webpage every 5 seconds
$url=$_SERVER['REQUEST_URI'];
header("Refresh: 5; URL=$url");
?>

<!--Κώδικας Html-->
<!DOCTYPE html>
<html lang="el">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Graph</title>
<link rel="stylesheet" href="../../stylepage/style3.css">
<link rel="stylesheet" href="../../stylepage/btnstyle.css">
</head>

<html>
<head>

<!--Προσθήκη βιβλιοθήκης-->
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>

<script type="text/javascript">
google.charts.load('current', {'packages':['bar']});
google.charts.setOnLoadCallback(drawChart);

function drawChart() {

var data = google.visualization.arrayToDataTable([
['DATE', 'Max', 'MIN', 'AVG'],

<?php
while($row = mysqli_fetch_array($result)){
//Δημιουργία γραφικής
echo "
[
'".$row["dateev"]."',
'".$row["MIN(temperature)"]."',
'".$row["MAX(temperature)"]."'

```

```

, ".$row["AVG (temperature)"]."
],";
}
?>
});

//Ρύθμισης εμφάνισης γραφήματος
var options = {
chart: {
title: 'Statics Temperature (°C ) of months for the last year',
subtitle: 'Max, Min, and Avg',
},
vAxis: {
title: 'Temperature °C',
textStyle: {
color: '#097138',
fontSize: 24
}
},
hAxis: {
title: 'DATE',
textStyle: {
color: '#097138',
fontSize: 17,
bold: true
}
},
colors: ['#006400', '#6B8E23', '#228B22' ],
backgroundColor: '#f1f8e9'
};

var chart = new
google.charts.Bar (document.getElementById('columnchart_material'));

chart.draw (data, google.charts.Bar.convertOptions (options));
}

</script>

</head>

<body>
<h1>TEMPERATURE GRAPH FOR TEMPERATURE OF DHT11 SENSOR</h1>
<!-- Δημιουργία menu -->
<div class="menu-bar">
<ul>

<li class="active"><a
href="http://192.168.1.12/db_allSensor/index.html"><i class="fa-solid fa-
house-blank"></i>HOME</a></li>

<li> <a href="http://192.168.1.7/"><i class="fa-solid fa-house-
blank"></i>Control Area </a></li>

<li> <a href="../../menu/dbtable/display.php"><i class="fa-solid fa-
house-blank"></i>Data sensor table</a></li>
<li> <a href="https://localhost/db_allsensor/menu/dbgauge/gauge.html"><i
class="fa-solid fa-house-blank"></i>Gauge Visualization</a></li>

<li>
<a href="#">Graph</a>

```

```

<Div class="sub_menu-1">
<ul>
<li><a href="../../../menu/dbgraph/graph.php">Temprature</a></li>
<li><a href="../../../menu/dbgraph/humgraph.php">Humidity</a></li>
<li><a href="../../../menu/dbgraph/outldr.php">Ldr out</a></li>
<li><a href="../../../menu/dbgraph/tankgraph.php">Level Tank</a></li>
<li><a href="../../../menu/dbgraph/soilhygrograph.php">Soil Hugro</a></li>
<li><a href="../../../menu/dbgraph/inldrgraph.php">Ldr IN</a></li>
<li><a href="../../../menu/dbgraph/gas.php">Gas</a></li>
</ul>

</Div>
</li>
<li>

<a href="#">ThingSpeak</a>
<div class="sub_menu-1">
<ul>

<li><a href="../../../menu/thingspeak/graph.html">Graph</a></li>

<li><a href="../../../menu/thingspeak/NumericDisplay.html">Numeric
Display</a></li>
</ul>
</div>
</li>
</ul>
</div>

<!--Link για την εμφάνιση της γραφικής παράστασης της θερμοκρασία σε
πραγματικό χρόνο, για την τελευταία εβδομάδα και για τον τελευταίο χρόνο-->

<ul>

<li><a class="livebtn" href="../../../dbgraph/graph.php">Real time
graph</a></li>

<li><a class="daybtn" href="../../../Statics/WeeklyStats.php">Statics of last
Week</a></li>

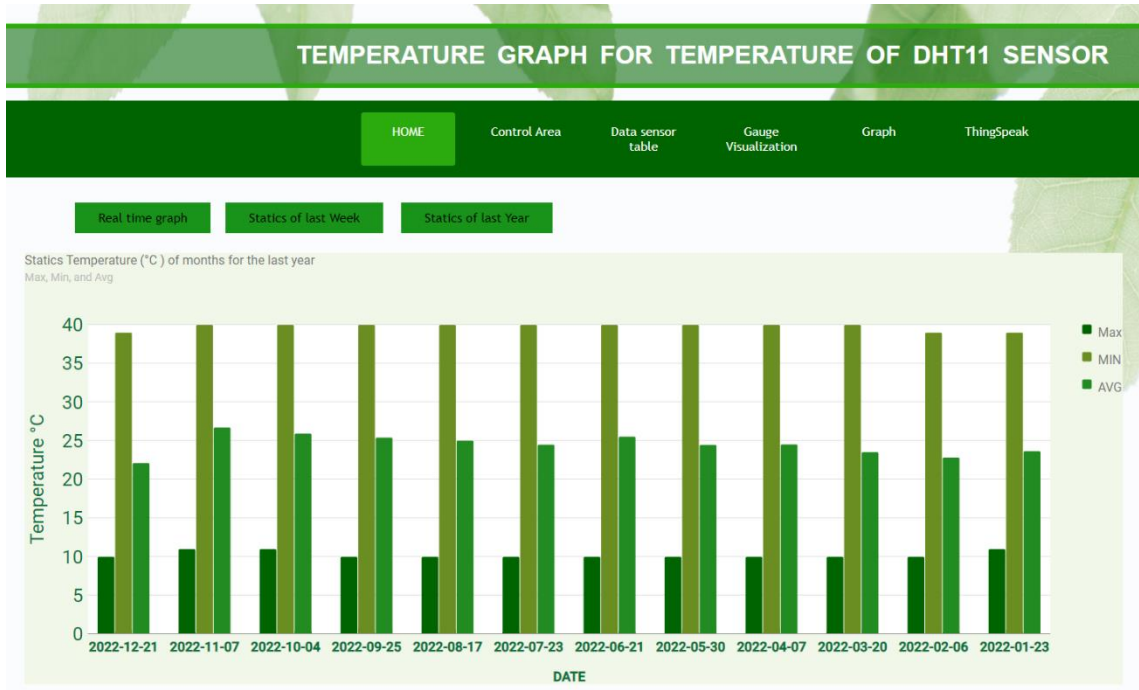
<li><a class="monthbtn" href="../../../Statics/YearlyStats.php">Statics of last
Year</a></li>

</ul>

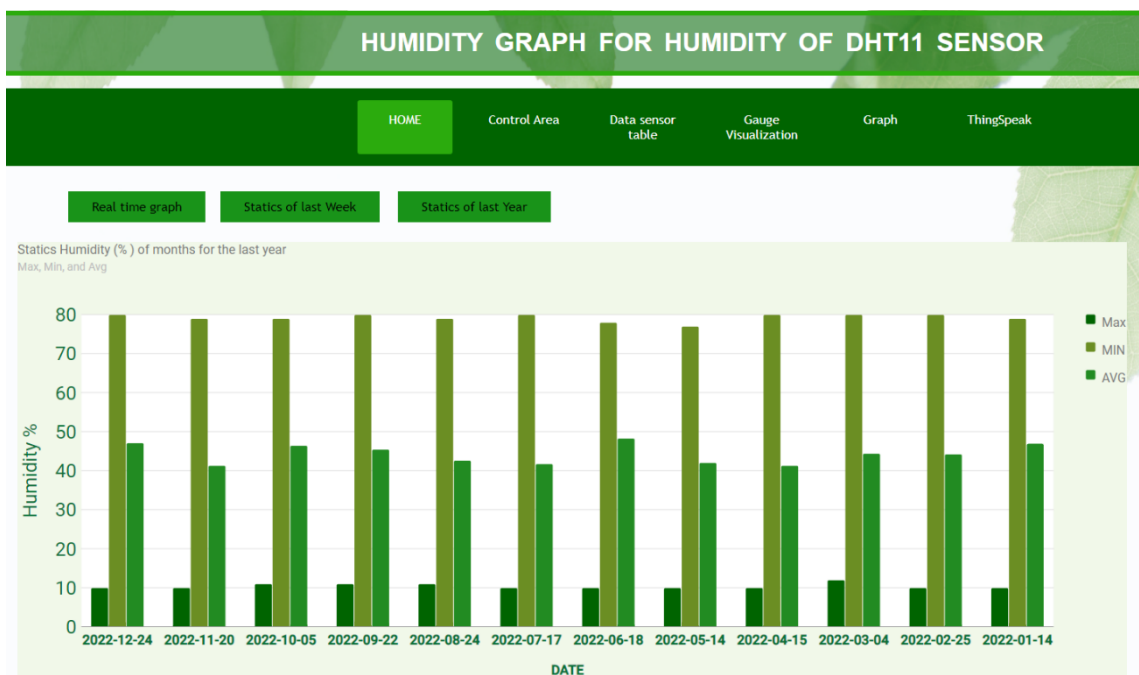
<!-- Εμφάνιση γραφήματος -->
<div id="columnchart_material" class="graph" style="width: 1400px;
height: 550px;"></div>
</body>
</html>

```

Το αποτέλεσμα του κώδικα είναι η [Εικόνα 6.20](#).

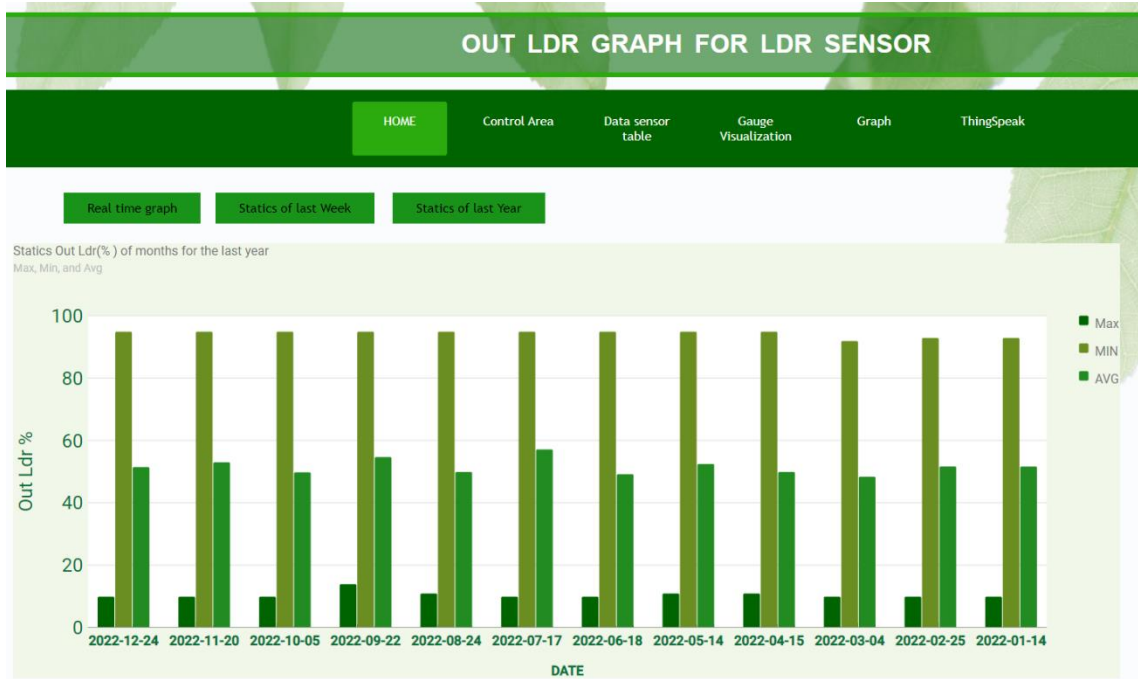


Εικόνα 6.20 Γράφημα μηνιαίων στατιστικών θερμοκρασίας.

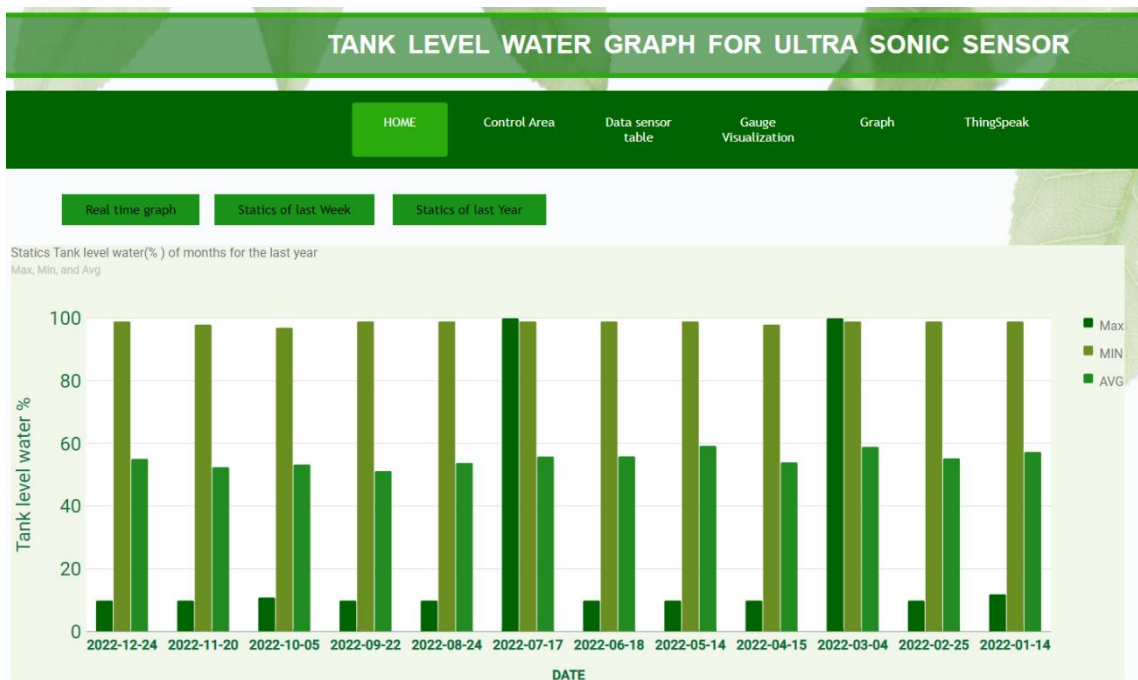


Εικόνα 6.21 Γράφημα μηνιαίων στατιστικών υγρασίας.

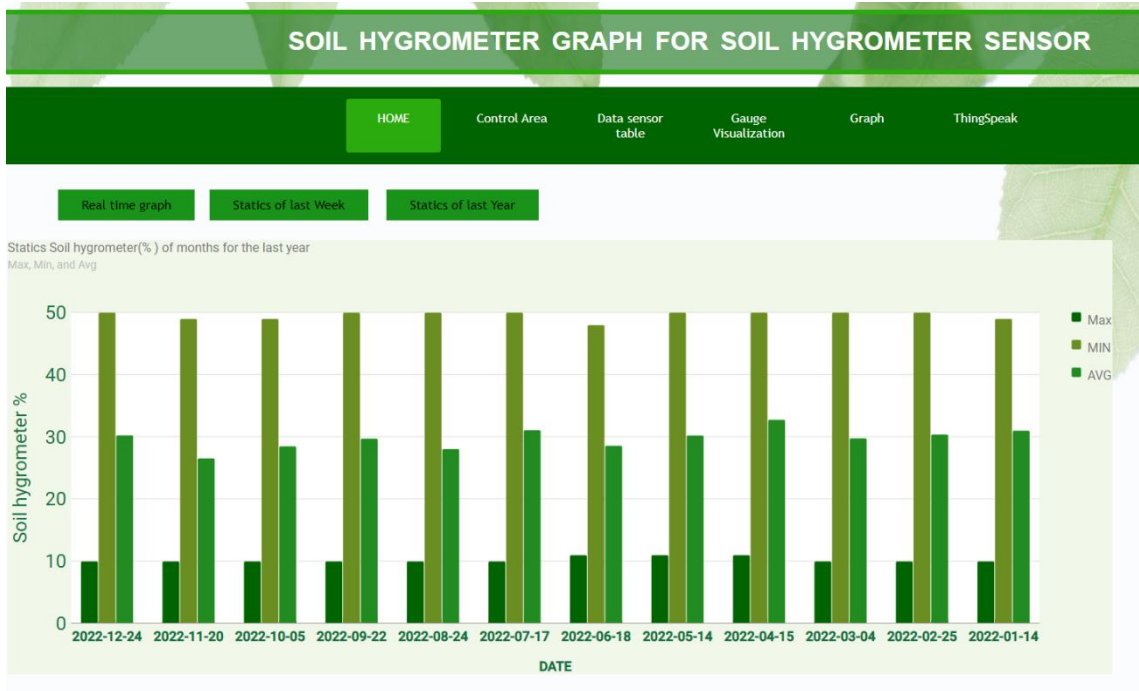




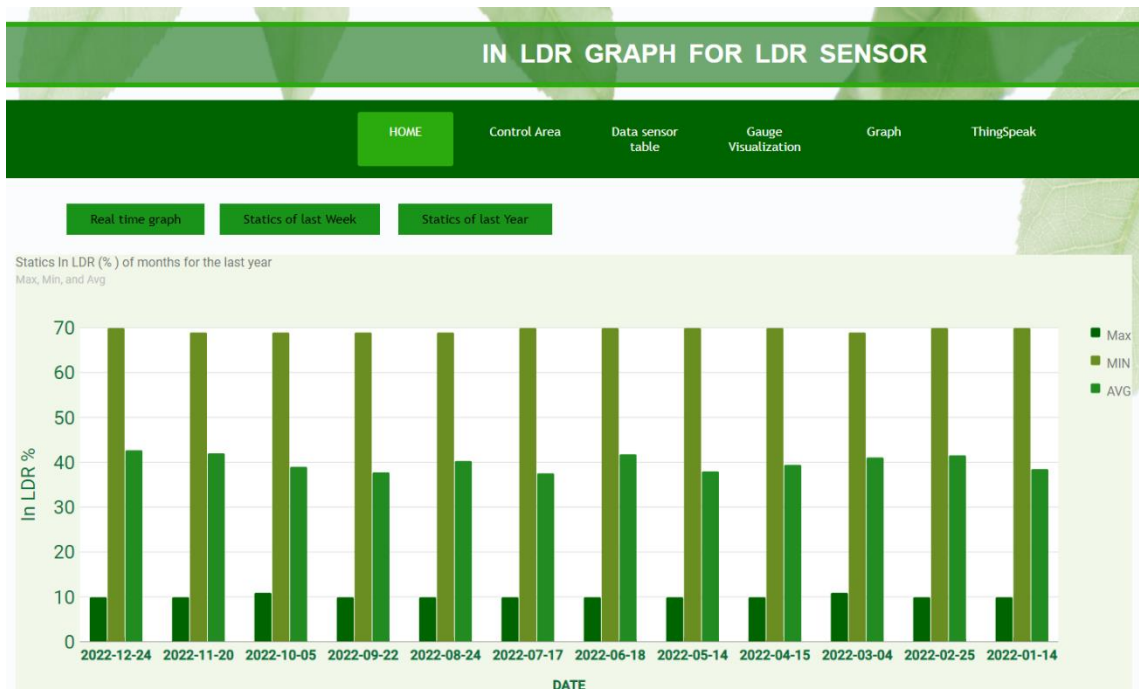
Εικόνα 6.22 Γράφημα μηνιαίων στατιστικών εξωτερικής φωτοαντίστασης.



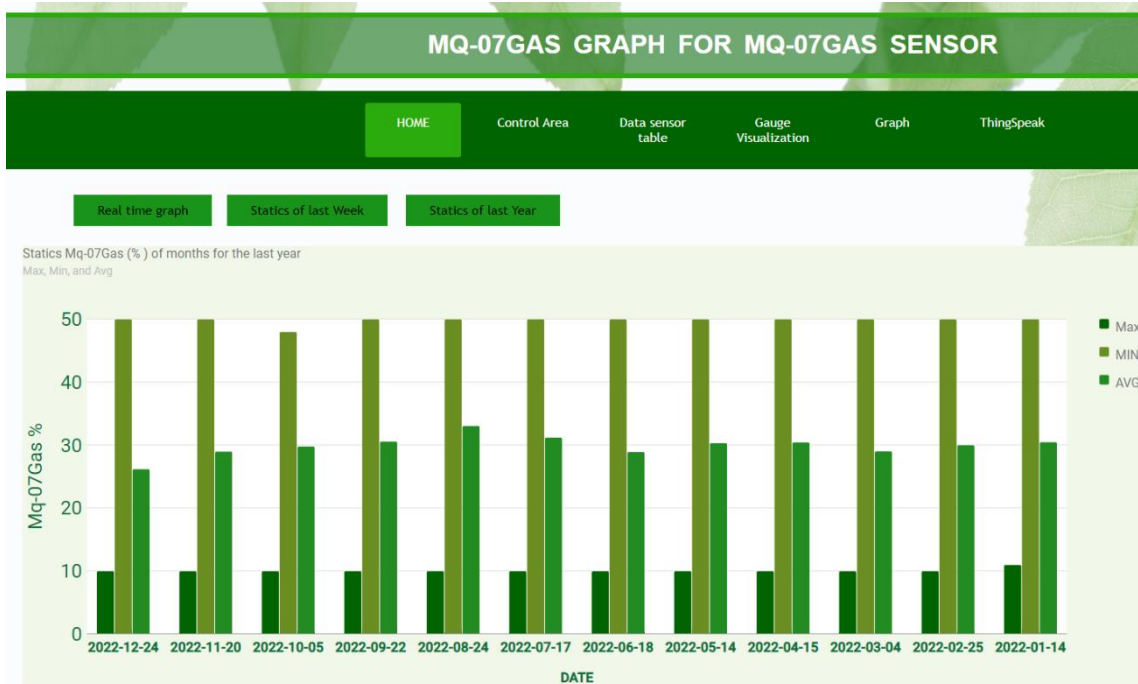
Εικόνα 6.23 Γράφημα μηνιαίων στατιστικών επιπέδου νερού.



*Εικόνα 6.24 Γράφημα μηνιαίων στατιστικών υγρασίας εδάφους.*



*Εικόνα 6.25 Γράφημα μηνιαίων στατιστικών εσωτερικής φωτοαντίστασης.*



Εικόνα 6.26 Γράφημα μηνιαίων στατιστικών διοξειδίου του άνθρακα.

## 6.8 Οπτικοποίηση δεδομένων σε μορφή μετρητών Gauge

Στην ιστοσελίδα έχει δημιουργηθεί οπτικοποίηση των δεδομένων μέσω μετρητών Gauge με την βοήθεια των google charts [Εικόνα 6.27](#).

Ακολουθεί ο κώδικας DataGauge.php που αντλεί τα δεδομένα από την βάση δεδομένων.

```
<?php
header('Content-Type: application/json');
$pdo=new PDO("mysql:dbname=db_allSensor;host=127.0.0.1","root","");
switch($_GET['q']){
// Ανάκτηση τελευταίων δεδομένων
case 1:
$stmtment=$pdo->prepare("SELECT temperature, humidity, outLdr, tank,
soilhygro, inLdr, Mq07Gas FROM data ORDER BY id DESC LIMIT 0,1");
$stmtment->execute();
$results=$stmtment->fetchAll(PDO::FETCH_ASSOC);
$json=json_encode($results);
echo $json;
break;
// Εύρεση όλων των δεδομένων
default:
$stmtment=$pdo->prepare("SELECT temperature, humidity, outLdr, tank,
soilhygro, inLdr, Mq07Gas FROM data ORDER BY id ASC");
$stmtment->execute();
$results=$stmtment->fetchAll(PDO::FETCH_ASSOC);
$json=json_encode($results);
echo $json;
break;
}
?>
```

Ο κώδικας **gauge.html** που εμφανίζει τα δεδομένα σε μορφή google gauge.

```
<!DOCTYPE html>
```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Gauge Visualization</title>
<link rel="stylesheet" href="../../stylepage/style3.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></
script>
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
google.charts.load('current', {'packages':['gauge']});
google.charts.setOnLoadCallback(drawChart1);
google.charts.setOnLoadCallback(drawChart2);
google.charts.setOnLoadCallback(drawChart3);
google.charts.setOnLoadCallback(drawChart4);
google.charts.setOnLoadCallback(drawChart5);
google.charts.setOnLoadCallback(drawChart6);

// Μετρητής 1 Θερμοκρασίας
function drawChart1() {
var data1 = google.visualization.arrayToDataTable([
  ['Label', 'Value'],
  ['Temp °C', 0]
]);
var options1 = {
width: 300, height: 300,
greenFrom: 26, greenTo: 33,
yellowFrom: 0, yellowTo: 26,
redFrom: 33, redTo: 40,
minorTicks: 5,
majorTicks: ['0', '10', '20', '30', '40'],
minorTicks: 3,
max: 40
};
var chart = new
google.visualization.Gauge(document.getElementById('chart_div1'));
chart.draw(data1, options1);
setInterval(function() {
var JSON=$.ajax({
url:"https://localhost/db_allsensor/menu/dbgauge/DataGauge.php?q=1",
dataType: 'json',
async: false}).responseText;
var Respuesta=jQuery.parseJSON(JSON);
data1.setValue(0, 1, Respuesta[0].temperature);
chart.draw(data1, options1);
}, 1300);
}
// Μετρητής 2 Υγρασίας
function drawChart2() {
var data2 = google.visualization.arrayToDataTable([
  ['Label', 'Value'],
  ['Humi %', 0]
]);
var options2 = {
width: 300, height: 300,
greenFrom: 70, greenTo: 90,
yellowFrom: 90, yellowTo: 100,
redFrom: 0, redTo: 70,
minorTicks: 11,
majorTicks: ['0', '10', '20', '30', '40', '50', '60', '70', '80', '90',
'100'],

```

```

minorTicks: 5
};
var chart = new
google.visualization.Gauge(document.getElementById('chart_div2'));
chart.draw(data2, options2);
setInterval(function() {
var JSON=$.ajax({
url:"https://localhost/db_allsensor/menu/dbgauge/DataGauge.php?q=1",
dataType: 'json',
async: false}).responseText;
var Respuesta=jQuery.parseJSON(JSON);
data2.setValue(0, 1,Respuesta[0].humidity);
chart.draw(data2, options2);
}, 1300);
}
// Μετρητής 3 διοξειδίου του άνθρακα
function drawChart3() {
var data3 = google.visualization.arrayToDataTable([
['Label', 'Value'],
['Mq07 %', 0]
]);
var options3 = {
width: 300, height: 300,
greenFrom: 0, greenTo: 50,
yellowFrom: 50, yellowTo: 70,
redFrom:70, redTo: 100,
minorTicks: 11,
majorTicks: ['0', '10', '20', '30', '40', '50', '60', '70', '80', '90',
'100'],
minorTicks: 5
};
var chart = new
google.visualization.Gauge(document.getElementById('chart_div3'));
chart.draw(data3, options3);
setInterval(function() {
var JSON=$.ajax({
url:"https://localhost/db_allsensor/menu/dbgauge/DataGauge.php?q=1",
dataType: 'json',
async: false}).responseText;
var Respuesta=jQuery.parseJSON(JSON);
data3.setValue(0, 1,Respuesta[0].Mq07Gas);
chart.draw(data3, options3);
}, 1300);
}
// Μετρητής 4 επιπέδου νερού
function drawChart4() {
var data4 = google.visualization.arrayToDataTable([
['Label', 'Value'],
['Water Level%', 0]
]);
var options4 = {
width: 300, height: 300,
greenFrom: 50, greenTo: 100,
yellowFrom: 30, yellowTo: 50,
redFrom:0, redTo: 30,
minorTicks: 11,
majorTicks: ['0', '10', '20', '30', '40', '50', '60', '70', '80', '90',
'100'],
minorTicks: 5
};
var chart = new
google.visualization.Gauge(document.getElementById('chart_div4'));

```

```

chart.draw(data4, options4);
setInterval(function() {
var JSON=$.ajax({
url:"https://localhost/db_allsensor/menu/dbgauge/DataGauge.php?q=1",
dataType: 'json',
async: false}).responseText;
var Respuesta=jQuery.parseJSON(JSON);
data4.setValue(0, 1,Respuesta[0].tank);
chart.draw(data4, options4);
}, 1300);
}
// Μετρητής 5 υγρασίας εδάφους
function drawChart5() {
var data5 = google.visualization.arrayToDataTable([
['Label', 'Value'],
['SoilHygro%', 0]
]);
var options5 = {
width: 300, height: 300,
greenFrom: 5, greenTo: 100,
yellowFrom: 20, yellowTo: 100,
redFrom:0, redTo: 5,
minorTicks: 11,
majorTicks: ['0', '10', '20', '30', '40', '50', '60', '70', '80', '90',
'100'],
minorTicks: 5
};
var chart = new
google.visualization.Gauge(document.getElementById('chart_div5'));
chart.draw(data5, options5);
setInterval(function() {
var JSON=$.ajax({
url:"https://localhost/db_allsensor/menu/dbgauge/DataGauge.php?q=1",
dataType: 'json',
async: false}).responseText;
var Respuesta=jQuery.parseJSON(JSON);
data5.setValue(0, 1,Respuesta[0].soilhygro);
chart.draw(data5, options5);
}, 1300);
}
// Μετρητής 6 εσωτερικής φωτοαντίστασης
function drawChart6() {
var data6 = google.visualization.arrayToDataTable([
['Label', 'Value'],
['Ldr IN%', 0]
]);
var options6 = {
width: 300, height: 300,
greenFrom: 30, greenTo: 70,
yellowFrom: 70, yellowTo: 100,
redFrom:0, redTo: 30,
minorTicks: 11,
majorTicks: ['0', '10', '20', '30', '40', '50', '60', '70', '80', '90',
'100'],
minorTicks: 5
};
var chart = new
google.visualization.Gauge(document.getElementById('chart_div6'));
chart.draw(data6, options6);
setInterval(function() {
var JSON=$.ajax({
url:"https://localhost/db_allsensor/menu/dbgauge/DataGauge.php?q=1",

```



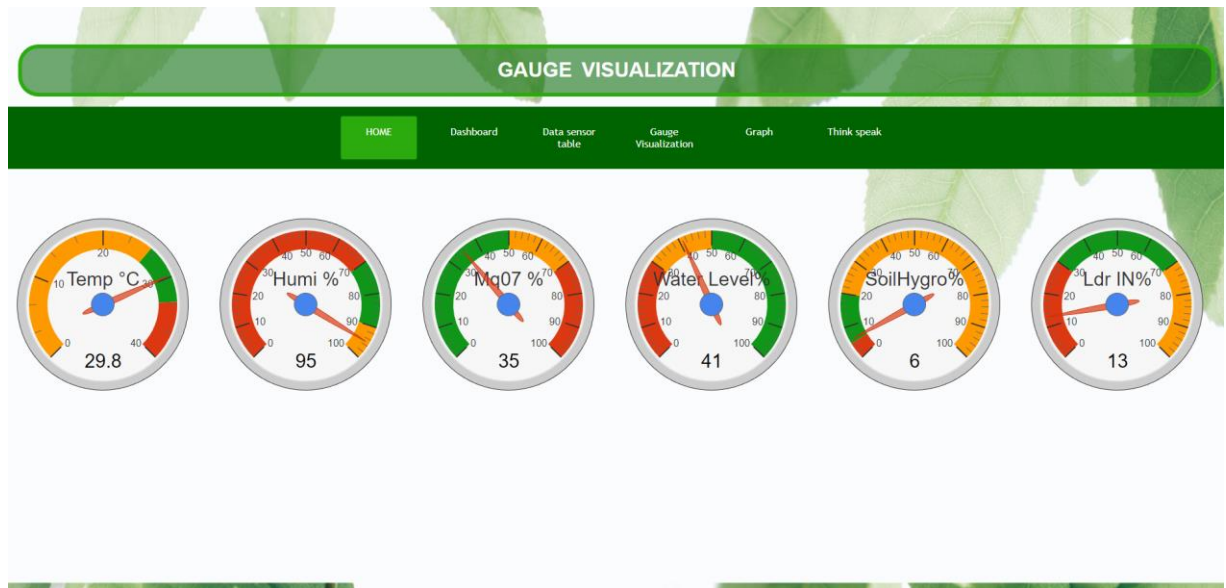
```

dataType: 'json',
async: false}).responseText;
var Respuesta=jQuery.parseJSON(JSON);
data6.setValue(0, 1,Respuesta[0].inLdr);
chart.draw(data6, options6);
}, 1300);
}

</script>
</head>
<body>
<h1>Gauge Visualization</h1>
<!-- Δημιουργία menu -->
<div class="menu-bar">
<ul>
<li class="active"><a href="../../index.html"><i class="fa-solid fa-house-blank"></i>HOME</a></li>
<li> <a href="http://192.168.1.7/"><i class="fa-solid fa-house-blank"></i>Dashboard </a></li>
<li> <a href="../../menu/dbtable/display.php"><i class="fa-solid fa-house-blank"></i>Data sensor table</a></li>
<li> <a href="https://localhost/db_allsensor/menu/dbgauge/gauge.html"><i class="fa-solid fa-house-blank"></i>Gauge Visualization</a></li>
<li>
<a href="#">Graph</a>
<Div class="sub_menu-1">
<ul>
<li><a href="../../menu/dbgraph/graph.php">Temprature</a></li>
<li><a href="../../menu/dbgraph/humgraph.php">Humidity</a></li>
<li><a href="../../menu/dbgraph/outldr.php">Ldr out</a></li>
<li><a href="../../menu/dbgraph/tankgraph.php">Level Tank</a></li>
<li><a href="../../menu/dbgraph/soilhygrograph.php">Soil Hugro</a></li>
<li><a href="../../menu/dbgraph/inldrgraph.php">Ldr IN</a></li>
<li><a href="../../menu/dbgraph/gas.php">Gas</a></li>
</ul>
</Div>
</li>
<li>
<a href="#">ThingSpeak</a>
<div class="sub_menu-1">
<ul>
<li><a href="../../menu/thingspeak/graph.html">Graph</a></li>
<li><ahref="../../menu/thingspeak/NumericDisplay.html">NumericDisplay</a>
</li>
</ul>
</div>
</li>
</ul>
</div>
<div id="chart_div1" class="GauseTemp"></div>
<div id="chart_div2" class="GauseHumi" ></div>
<div id="chart_div3" class="GauseMq07" ></div>
<div id="chart_div4" class="GauseTank" ></div>
<div id="chart_div5" class="GauseSoil" ></div>
<div id="chart_div6" class="GauseLdrIN" ></div>
</body>
</html>

```

Ακολουθεί το αποτέλεσμα του κώδικα gauge.html μαζί με την χρήση του αρχείου style3.css..



Εικόνα 6.27 Οι έξι μετρητές που απεικονίζουν της τιμές των αισθητήρων.

## 6.9 ThingSpeak

### 6.9.1 Αποστολή δεδομένων στο ThingSpeak

Εκτός από την οπτικοποίηση δεδομένων σε γραφικές παραστάσεις, σε μορφή πίνακα και σε μορφή Gauge θα υπάρχει και η επιλογή οπτικοποίησης μέσω του ThingSpeak. Επίσης μέσω του ThingSpeak και της πλατφόρμας IFTTT θα στέλνονται αυτόματα email όταν θα ισχύουν κάποιες συγκεκριμένες συνθήκες.

Το ThingSpeak βρίσκετε στην ιστοσελίδα στο Bar Menu στην επιλογή ThingSpeak και υπάρχουν δύο επιλογές: η οπτικοποίηση με γραφικές και με σταθερές τιμές.

Για να αποσταλούν οι τιμές των αισθητήρων στην πλατφόρμα ThingSpeak χρειάζεται η δημιουργία ενός λογαριασμού στο ThingSpeak. Έπειτα χρειάζεται να δημιουργηθεί ένα κανάλι επιλέγοντας το New Chanel.

The image shows the ThingSpeak website interface. At the top is a blue navigation bar with the ThingSpeak logo and menu items: Channels, Apps, Devices, and Support. Below the navigation bar is the 'My Channels' section. There is a 'New Channel' button and a search bar labeled 'Search by tag'. A table lists the user's channels:

Name	Created	Updated
Arduino Sensor Data	2022-01-09	2022-08-27 13:54

Below the table are buttons for 'Private', 'Public', 'Settings', 'Sharing', 'API Keys', and 'Data Import / Export'.

Εικόνα 6.28 Το κανάλι Arduino Sensor Data.

Υστερα εφαρμόζονται οι παρακάτω ρυθμίσεις (Εικόνα 6.29).

The screenshot shows the 'Channel Settings' page for an Arduino Sensor Data channel. The page is divided into several sections: Channel Information, Channel Settings, Help, and Using the Channel. The Channel Information section shows the Channel ID (1628938), Author (mwa000022454647), and Access (Public). The Channel Settings section includes a progress bar for 'Percentage complete' (50%), a list of fields (Field 1 to Field 7) with their names and checkboxes, a 'Metadata' field, a 'Tags' field, a 'Link to External Site' field, a 'Link to GitHub' field, an 'Elevation' field, a 'Show Channel Location' section with 'Latitude' and 'Longitude' fields, a 'Show Video' section with 'YouTube' and 'Vimeo' radio buttons and a 'Video URL' field, and a 'Show Status' field. A 'Save Channel' button is at the bottom. The Help section provides detailed instructions for each field. The Using the Channel section explains how to get data into a channel from a device, website, or another ThingsSpeak channel.

ThingSpeak™ Channels ▾ Apps ▾ Devices ▾ Support ▾ Commercial Use How to Buy NS

## Arduino Sensor Data

Channel ID: 1628938  
Author: mwa000022454647  
Access: Public

Arduino Sensors (DHT11, Soil Hygrometer, 2 LDR, HC-SR04 Ultrasonic and Mq-07) send this data to thingspeak and later to my web site.

Private View Public View Channel Settings Sharing API Keys Data Import / Export

### Channel Settings

Percentage complete 50%

Channel ID 1628938

Name Arduino Sensor Data

Description Arduino Sensors (DHT11, Soil Hygrometer, 2 LDR, HC-SR04 Ultrasonic and Mq-07) send this data to

Field 1 TEMPERATURE

Field 2 Humidity

Field 3 Soil Hygrometer

Field 4 LDR IN

Field 5 LDR OUT

Field 6 HC-SR04 Ultrasonic

Field 7 MQ-7 Gas Sensor

Field 8

Metadata

Tags   
(Tags are comma separated)

Link to External Site

Link to GitHub

Elevation

Show Channel Location

Latitude

Longitude

Show Video

YouTube

Vimeo

Video URL

Show Status

Save Channel

### Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

### Channel Settings

- Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:**
  - Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
  - Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
  - Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
- Video URL:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.
- Link to GitHub:** If you store your ThingSpeak code on GitHub®, specify the GitHub repository URL.

### Using the Channel

You can get data into a channel from a device, website, or another ThingsSpeak channel. You can then visualize data and transform it using ThingSpeak Apps.

See [Get Started with ThingSpeak™](#) for an example of measuring dew point from a weather station that acquires data from an Arduino® device.

[Learn More](#)

Εικόνα 6.29 Οι ρυθμίσεις στο ThingSpeak για την δημιουργία καναλιού.

Επίσης μαζί με την δημιουργία του καναλιού δημιουργούνται και τρεις κωδικοί. Οι οποίοι είναι ο Chanel ID, ο Write API Key και Read API Key. Ο Write API Key είναι για την εισαγωγή δεδομένων στο κανάλι και ο Read API Key είναι για να υπάρχει η δυνατότητα να διαβάζονται τα δεδομένα από το κανάλι.

The screenshot shows the ThingSpeak interface for a channel named 'Arduino Sensor Data' with ID 1628938. The 'API Keys' tab is active, displaying a 'Write API Key' (NFNEYSXD0MUBW6MG) and a 'Read API Key' (0J7LVWUMQSOJ88JS). A 'Note' field is present for the Read API Key. The right sidebar contains a 'Help' section with 'API Keys Settings' and 'API Requests'.

**Write API Key**

Key: NFNEYSXD0MUBW6MG

Generate New Write API Key

**Read API Keys**

Key: 0J7LVWUMQSOJ88JS

Note: [Empty text area]

Save Note Delete API Key

Add New Read API Key

**Help**

API Keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

**API Keys Settings**

- Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

**API Requests**

**Write a Channel Feed**

```
GET https://api.thingspeak.com/update?api_key=NFNEYSXD0MUBW6MG&field=
```

**Read a Channel Feed**

```
GET https://api.thingspeak.com/channels/1628938/feeds.json?results=
```

**Read a Channel Field**

```
GET https://api.thingspeak.com/channels/1628938/fields/1.json?result=
```

**Read Channel Status Updates**

```
GET https://api.thingspeak.com/channels/1628938/status.json
```

Learn More

Εικόνα 6.30 Τα κλειδιά ThingSpeak.

Τέλος, πρέπει να γραφτεί ο παρακάτω κώδικας Arduino, χρησιμοποιώντας το Write Api Key ώστε να αποσταλούν στο ThingSpeak οι τιμές από τους αισθητήρες.

Ακολουθεί κώδικας arduino για την αποστολή των δεδομένων από τους αισθητήρες στο ThingSpeak.

```
/*-----( Import needed libraries )-----*/  
// Use the Serial Peripheral Interface  
#include <SPI.h>  
//Ethernet Shield Library
```

```

#include <Ethernet.h>
//Βιβλιοθήκη για αισθητήρα Dht11
#include <DHT.h>
//librari Ultrasonic Sensor HC-SR04
#include <HCSR04.h>

/*-----( Declare Constants and Pin Numbers )-----*/
// Digital pin connected to the DHT sensor
#define SEN_Dht11 38
//attach pin D49 Arduino to pin Trig of HC-SR04
#define SEN_UltraSonicTrigPin 49
// attach pin D48 Arduino to pin Echo of HC-SR04
#define SEN_UltraSonicEchoPin 48
// Analog pin A9 connected to the LDR Sensor
int SEN_LdrOut = A9;

/*-----( Declare objects )-----*/
// selected sensor type the DHT 11
#define DHTTYPE DHT11
DHT dht(SEN_Dht11, DHT11);

//initialization class HCSR04 (trig pin , echo pin)
HCSR04 hc(SEN_UltraSonicTrigPin, SEN_UltraSonicEchoPin);

//Ορισμός της διεύθυνσης MAC του Arduino Ethernet shield
byte mac[] = { 0xD4, 0x28, 0xB2, 0xFF, 0xA0, 0xA1 };

// ThingSpeak Settings
//διακομιστής ThingSpeak
char thingSpeakAddress[] = "api.thingspeak.com";
//Write the Write api key
String writeAPIKey = "NFNEYSXDOMUBW6MG";
// Time interval in milliseconds to update ThingSpeak
//(number of seconds * 1000 = interval)
const int updateThingSpeakInterval = 16 * 1000;

// Variable Setup
long lastConnectionTime = 0;
boolean lastConnected = false;
int failedCounter = 0;

/*-----( Declare Variables )-----*/
// Xbee sensor variables for frame
unsigned int analogMSB1 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB1 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_SoilHygro = 0; //variable for the soil hygrometer data

unsigned int analogMSB2 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB2 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_InLdr = 0; // variable for the LDR in data

unsigned int analogMSB3 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB3 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_Mq07Gas = 0; // variable for the Mq-07 data

// Variable Setup for DHT11
float Val_humi = 0; // Humidity variables
float Val_temp = 0; // Temperature variables
int Val_OutLdr = 0; // variable for the out LDR data

int Val_distance; // variable for the distance measurement
int Val_distancePerCent; // variable for the distance per cent measurement

```

```

// Initialize Arduino Ethernet Client
EthernetClient client;

void setup(){
  //Serial Communication is starting with 9600 of baud rate speed
  Serial.begin(9600);
  dht.begin();// Start the DHT sensor

  // print some text in Serial Monitor
  Serial.println(F("DHT11 test!"));

  // print the Ip to Serial monitor
  Serial.print("ETHERNET Server SHIELD Started...\nLocal IP: ");
  Ethernet.begin(mac);

  // Start Ethernet connection on Arduino for ThingSpeak
  startEthernet();

  //Set the INPUT pin mode
  pinMode(SEN_LdrOut, INPUT);
  // set the pin mode to input
  pinMode(SEN_Dht11, INPUT);
  // Sets the echoPin as an INPUT
  pinMode(SEN_UltraSonicEchoPin, INPUT);

  //Set the Outout pin mode
  // Sets the trigPin as an OUTPUT
  pinMode(SEN_UltraSonicTrigPin, OUTPUT);
}

void loop(){
  Read_Sen_Dht11_Loop(); // read the data sensor
  Read_Sen_UltraSonic_Loop();
  Read_Sen_OutLDR_Loop(); //Read the data sensor
  Read_RemoteXbee_Loop();

  Print_Serial_loop(); // print the data sensor to serial monitor
  Print_ThingSpeak_loop();
}

// Read data Dht11 Data sensor
void Read_Sen_Dht11_Loop() {
  // read the humidity
  Val_humi = dht.readHumidity();
  // Read temperature as Celsius (the default)
  Val_temp = dht.readTemperature();
}

//----Read the distance at tank-----
void Read_Sen_UltraSonic_Loop(){
  Val_distance = hc.dist(); // Read Sensor ultrasonic data
  //Κάνουμε αντιστοίχιση των τιμών του αισθητήρα UltraSonic σε
  //εκατοστά σε εκατό τις εκατό για την καλύτεροί απεικόνιση της
  //στάθμης νερού στο δοχείο. Το δοχείο είναι 17 εκατοστά αρά
  //τότε θα έχει 0% νερό ενώ εάν είναι 0 εκ. τότε έχουμε 100%
  //νερό μέσα στο δοχείο.
  Val_distancePerCent = map(Val_distance, 17, 0, 0, 99);
}

//----Read the value of ldr out of greenhouse
void Read_Sen_OutLDR_Loop() {
  Val_OutLdr = analogRead(SEN_LdrOut);// read the analog data 0-1023 byte
  //Αντιστοιχίσει των Bytes σε επί τοις εκατό
  Val_OutLdr = map(Val_OutLdr, 400, 1030, 0, 100);
}

```



```

}
//----Read the sensor value of Router Xbee
void Read_RemoteXbee_Loop() {
    // έλεγχος εάν το frame έχει τουλάχιστον 29 bits
    if (Serial.available() >= 29) {
        // έλεγχος εάν έχει start byte 7E
        if (Serial.read() == 0x7E) {
            // Απόρριψη των πρώτων 21 byte
            for (int i = 1; i < 21; i++) {
                byte discardbyte = Serial.read();
            }
            //τα Πρώτα byte είναι η τιμή του Soil hygrometer
            // Read the most significant bit of the analog data
            analogMSB1 = Serial.read(); // Read the first analog byte
data
            analogLSB1 = Serial.read(); // Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
Val_SoilHygro = analogLSB1 + (analogMSB1 * 256);
//Αντιστοιχίσει των Byte σε επι τοις εκατό
Val_SoilHygro = map(Val_SoilHygro, 700, 1000, 100, 0);

            //τα Επόμενα byte είναι η τιμή του LDR
            analogMSB2 = Serial.read();// Read the first analog byte data
            analogLSB2 = Serial.read();// Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
Val_InLdr = analogLSB2 + (analogMSB2 * 256);
//Αντιστοιχίσει των Byte σε επι τοις εκατό
Val_InLdr = map(Val_InLdr, 0, 1030, 0, 100);
//τα τελευταία byte είναι η τιμή το Gas Sensor Mq07

            analogMSB3 = Serial.read();// Read the first analog byte data
            analogLSB3 = Serial.read();// Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
Val_Mq07Gas = analogLSB3 + (analogMSB3 * 256);
//Αντιστοιχίσει των Byte σε επι τοις εκατό
Val_Mq07Gas = map(Val_Mq07Gas, 0, 1030, 0, 100);
        }
    }
}

void Print_ThingSpeak_loop(){
    // μετατροπή της μεταβλητής Float σε String
    String temp = String (Val_temp);
    String hum = String (Val_humi);
    String SoilHygro = String (Val_SoilHygro);
    String InLdr = String (Val_InLdr);
    String OutLdr = String (Val_OutLdr);
    String Mq07Gas = String (Val_Mq07Gas);
    String distance = String (Val_distancePerCent);

    // Print Update Response to Serial Monitor
    if (client.available()){
        char c = client.read();
        Serial.print(c);
    }

    // Disconnect from ThingSpeak
    if (!client.connected() && lastConnected){
        Serial.println("...disconnected");
    }
}

```

```

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

//Ενημέρωση τιμών
//chart sensor
if(!client.connected() && (millis() - lastConnectionTime>
updateThingSpeakInterval)){ updateThingSpeak("field1="+temp +
"&field2="+hum + "&field3="+SoilHygro + "&field4="+InLdr +
"&field5="+OutLdr+"&field6="+distance + "&field7="+ Mq07Gas);
}

// Check if Arduino Ethernet needs to be restarted
if (failedCounter > 3 ) {
    startEthernet();
}

lastConnected = client.connected();
}
void updateThingSpeak(String tsData){
    if (client.connect(thingSpeakAddress, 80)){
        //Χρήση Http post για την ενημέρωση των δεδομένων
        //To Start line αποτελείται από
        //(method post/the protocol/έκδοση) αντίστοιχα
        client.print("POST /update HTTP/1.1\n");

        //Τα header για αναγνώριση μηνύματος post Request από το ThingSpeak
        //thingspeak Host
        client.print("Host: api.thingspeak.com\n");
        //Header Connection: το οποίο θα κλίσει την σύνδεση όταν στείλουμε κα
λάβουμε το αίτημα μας
        client.print("Connection: close\n");
        //Header X-THINGSPEAKAPIKEY που είναι το ApiKey
        client.print("X-THINGSPEAKAPIKEY: "+writeAPIKey+"\n");
        //Connection type: το οποίο υποδεικνύει το είδος αποστολής
        client.print("Content-Type: application/x-www-form-urlencoded\n");
        //connection length
        client.print("Content-Length: ");

        client.print(tsData.length());
        client.print("\n\n");
        client.print(tsData);
        lastConnectionTime = millis();
        if (client.connected()){
            Serial.println("Connecting to ThingSpeak...");
            Serial.println();
            failedCounter = 0;
        }else{
            failedCounter++;
            Serial.println("Connection to ThingSpeak failed
("+String(failedCounter, DEC)+")");
            Serial.println();
        }
        }else{
            failedCounter++;
            Serial.println("Connection to ThingSpeak Failed
("+String(failedCounter, DEC)+")");
            Serial.println();
            lastConnectionTime = millis();
        }
    }
}
}

```

```

void startEthernet() {
  client.stop();
  Serial.println("Connecting Arduino to network...");
  Serial.println();
  delay(1000);
  // Connect to network and obtain an IP address using DHCP
  if (Ethernet.begin(mac) == 0) {
    Serial.println("DHCP Failed, reset Arduino to try again");
    Serial.println();
  } else {
    Serial.println("Arduino connected to network using DHCP");
    Serial.println();
  }
  delay(1000); // Delay at 1 Seconds
}
//εμφάνιση τιμών στην σειριακή οθόνη
void Print_Serial_loop() {
  Serial.println(F("*****"));
  //Εμφάνιση θερμοκρασίας στην σειριακή οθόνη
  Serial.print(F("Temperature: "));
  Serial.print(Val_temp); Serial.println(F("°C"));
  // Εμφάνιση υγρασίας στην σειριακή οθόνη
  Serial.print(F("Humidity: "));
  Serial.print(Val_humi); Serial.println(F("%"));
  Serial.print(F("LDR OUT Value: "));
  Serial.print(Val_OutLdr); Serial.println(F("%"));
  Serial.print(F("Distance: "));
  Serial.print(Val_distancePerCent); Serial.println(F("%"));
  Serial.print(F("Distance: "));
  Serial.print(Val_distance); Serial.println(F("cm"));
  Serial.print(F("Soil moisture: "));
  Serial.print(Val_SoilHygro); Serial.println(F("%"));
  Serial.print(F("Ldr IN Value: "));
  Serial.print(Val_InLdr); Serial.println(F("%"));
  Serial.print(F("Gas Value: "));
  Serial.print(Val_Mq07Gas); Serial.println(F("%"));
}

```

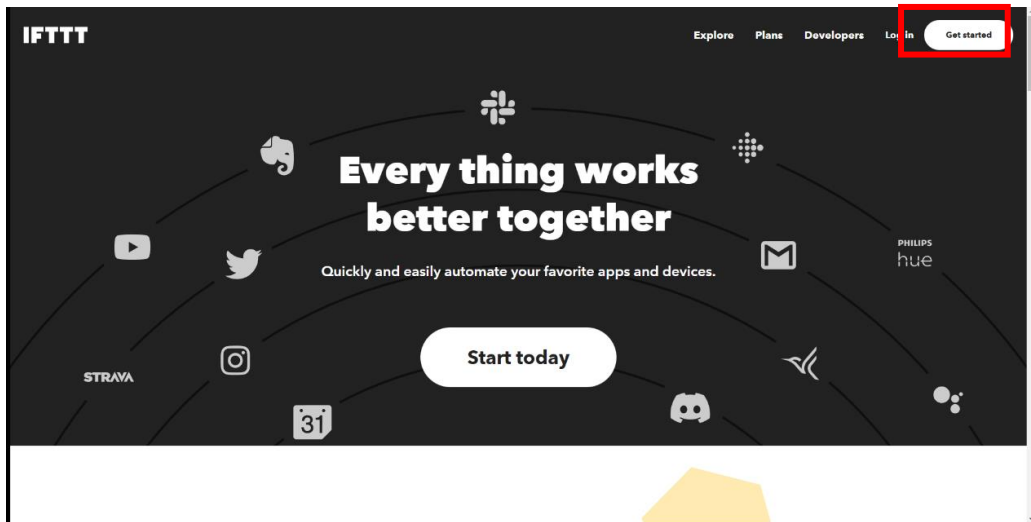
## 6.9.2 Σύνδεση ThingSpeak με IFTTT

Η σύνδεση μεταξύ ThingSpeak και IFTTT έχει ως σκοπό την αυτόματη αποστολή ενός email ειδοποιήσεις όταν υπάρχει:

- a) υψηλή θερμοκρασία στο θερμοκήπιο πάνω από 35 °C
- b) χαμηλή θερμοκρασία στο θερμοκήπιο κάτω από 25 °C
- c) υψηλή περιεκτικότητα διοξειδίου του άνθρακα στον χώρο του θερμοκήπιού πάνω από 50%

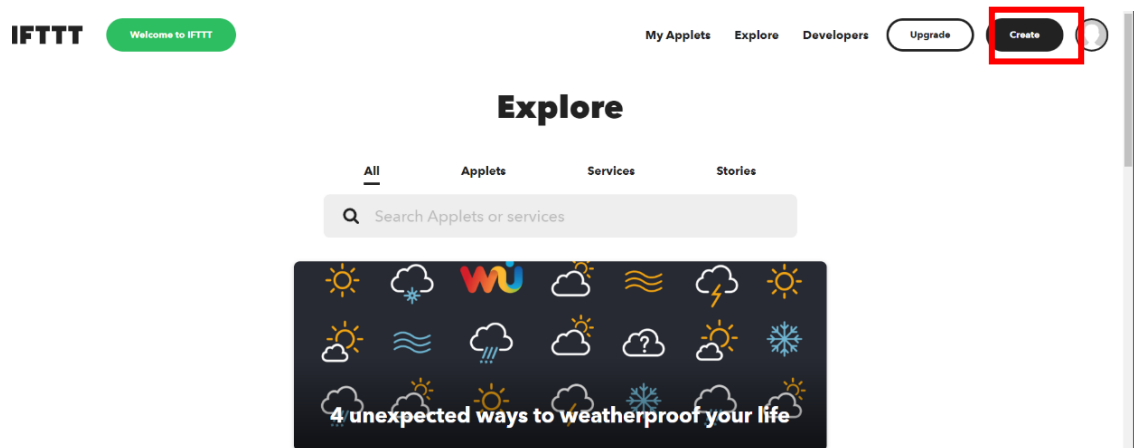
Το ThingSpeak καταγραφά τα δεδομένα από το Arduino όπως εξηγήθηκε στο παραπάνω κεφάλαιο [6.9.1](#) και ορίζει ποτέ θα σταλθεί το Email μέσω του IFTTT.

Για να γίνει χρήση του IFTTT απαιτείτε η δημιουργία ενός λογαριασμού. Πατώντας το Get Started αρχίζει η διαδικασία εγγραφής.



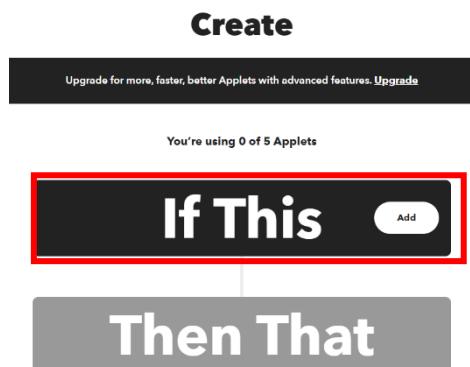
Εικόνα 6.31 Απεικόνιση της αρχικής σελίδας IFTTT.

Μόλις δημιουργηθεί ο λογαριασμός εμφανίζεται η [Εικόνα 6.32](#) και πατάμε Create για να δημιουργηθεί το Applet.



Εικόνα 6.32 Επιλογή Create για δημιουργία Applet.

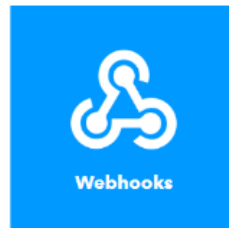
Για να δημιουργηθεί το Applet χρειάζεται να οριστεί το If this και το Then That. Αρχικά επιλέγουμε το If This.



Εικόνα 6.33 Επιλογή του If This.

Έπειτα στην γραμμή αναζήτησης γράφουμε Webhooks και το επιλέγουμε.


## Choose a service



Εικόνα 6.34 Επιλογή υπηρεσίας Webhooks.

Υστέρα επιλέγουμε το Receive a web request.

### Choose a trigger



## Webhooks

**Receive a web request with a JSON payload**

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap

**Receive a web request**

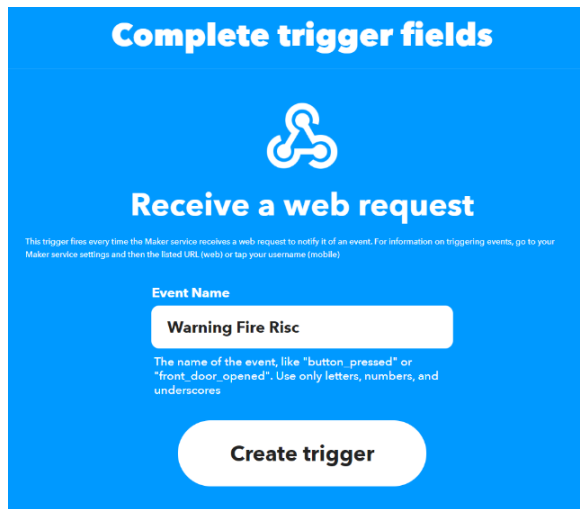
This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

**+**

**Suggest a new trigger**

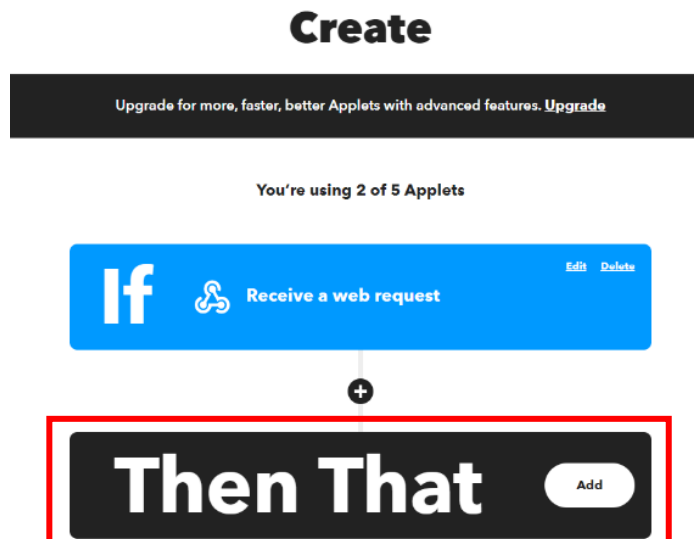
Εικόνα 6.35 Επιλογή του Receive a web request.

Στην συνέχεια δίνουμε ένα όνομα στο Event Name και πατάμε Create trigger.



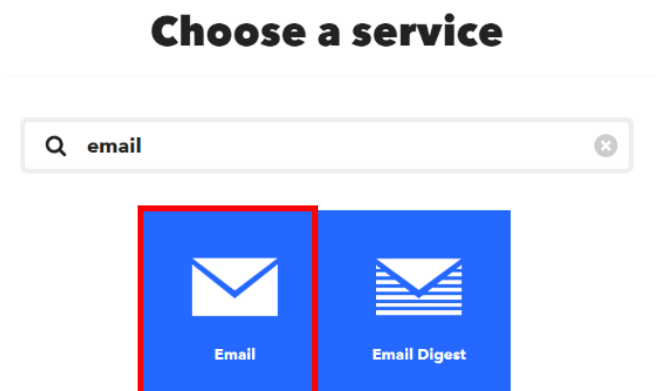
Εικόνα 6.36 Συμπλήρωση πεδίου Event Name.

Μόλις δημιουργηθεί το If then μπορούμε να επιλέξουμε το Then That.



Εικόνα 6.37 Επιλογή του Then That.

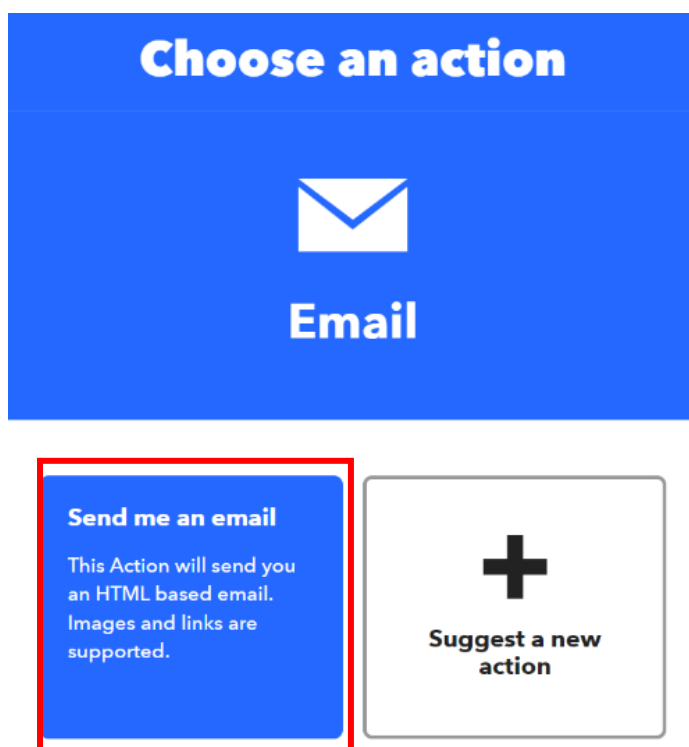
Κάνουμε αναζήτηση το service Email και το επιλέγουμε.



Εικόνα 6.38 Επιλογή υπηρεσίας Email.

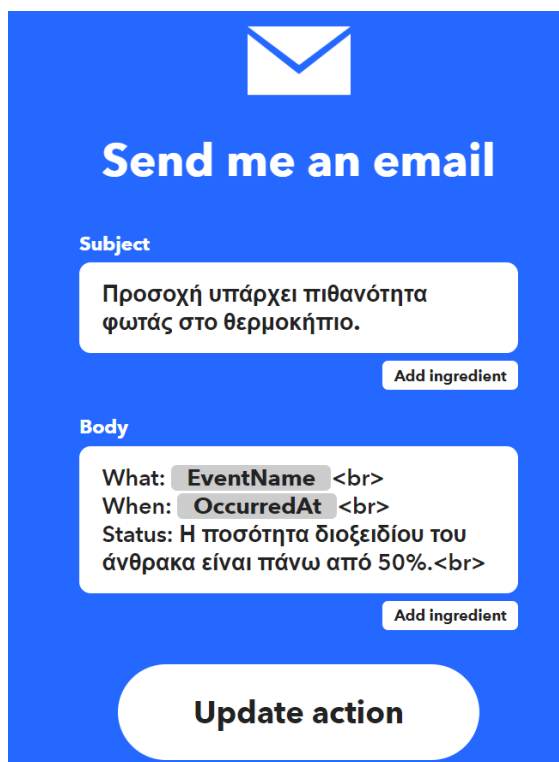


Στην συνέχεια επιλέγουμε το Send me an email



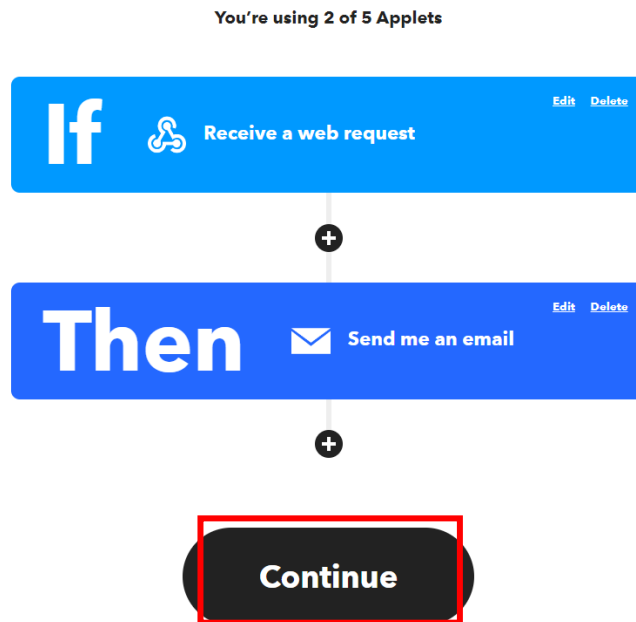
Εικόνα 6.39 Επιλογή της δράσης Send me an email.

Κατόπιν γράφουμε το θέμα (Subject) και το κείμενο (Body) που θα εμφανίζεται στο Email και πατάμε Crate action.



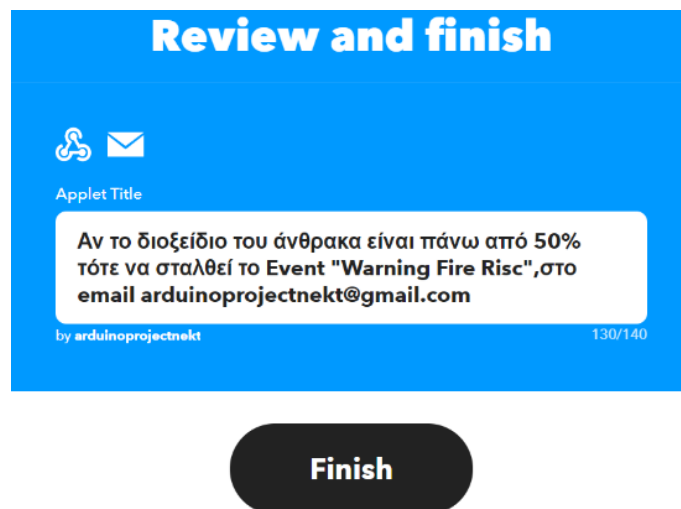
Εικόνα 6.40 Συντάζει του email που θα σταλθεί.

Έπειτα πατάμε Continue.



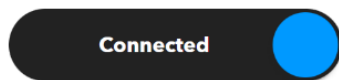
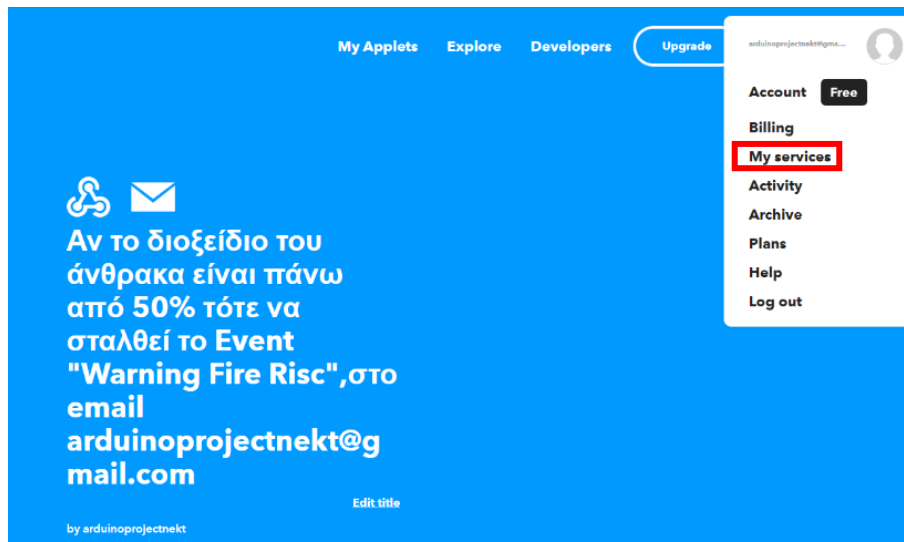
Εικόνα 6.41 Ολοκλήρωση του Applet.

Τέλος βάζουμε έναν τίτλο στο Applet που δημιουργήσαμε και πατάμε Finish.

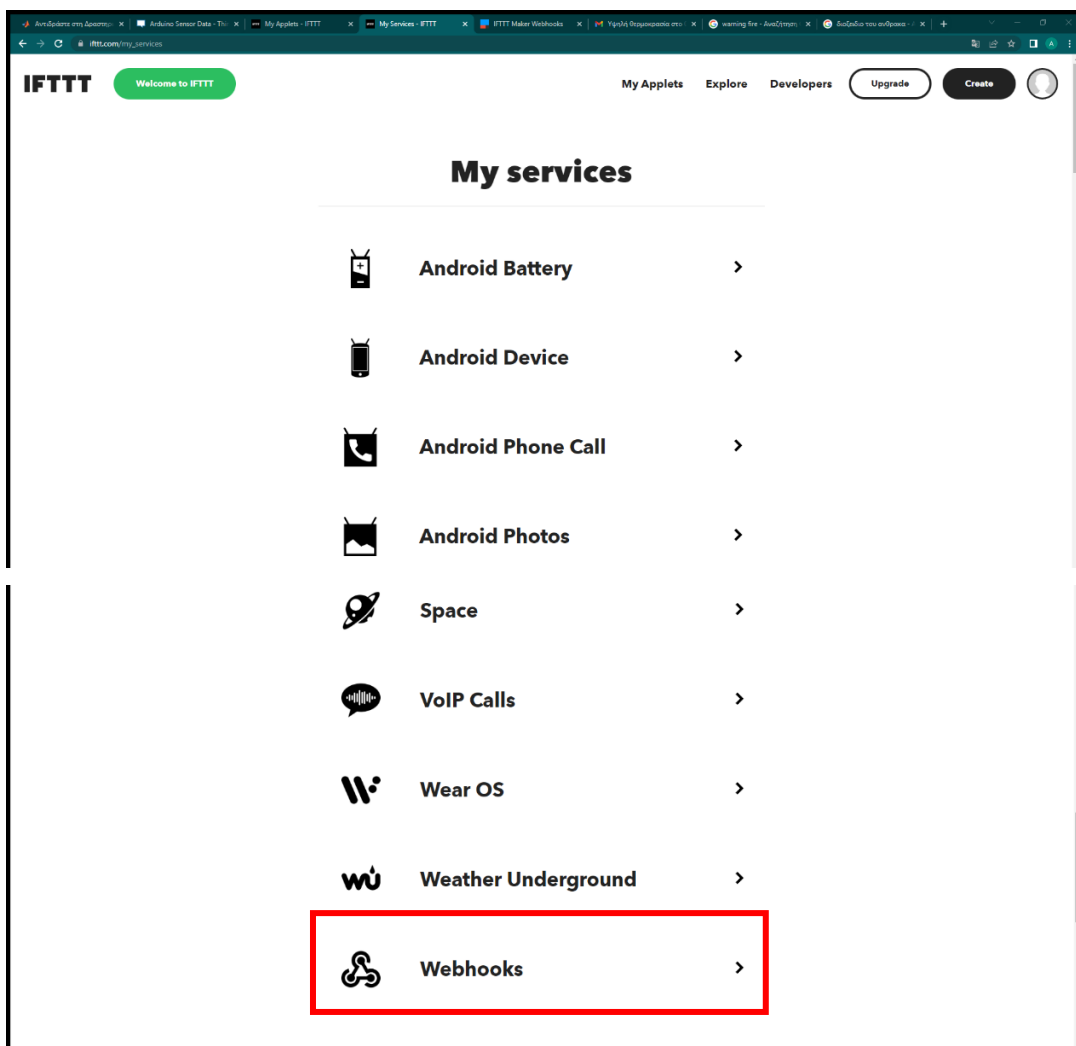


Εικόνα 6.42 Ονομασία του Applet.

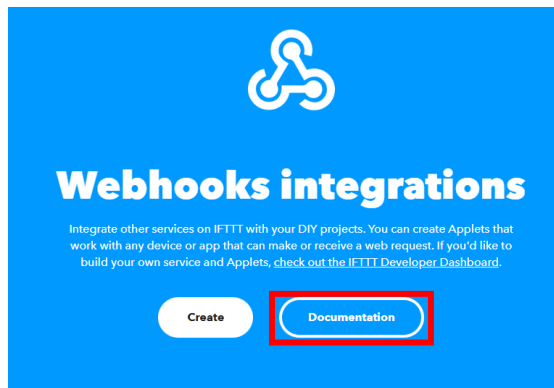
Ύστερα από την δημιουργία του Applet δημιουργείται ένα κλειδί που απαιτείται για να την σύνδεση με το ThingSpeak. Το κλειδί βρίσκεται στο My services([Εικόνα 6.43](#))>Webhooks([Εικόνα 6.44](#)) και ύστερα επιλέγουμε documentations ([Εικόνα 6.45](#)).



Εικόνα 6.43 Επιλογή του My services.



Εικόνα 6.44 Επιλογή του Webhooks.



[Applets](#) [My Applets](#) [Details](#) [Health](#)

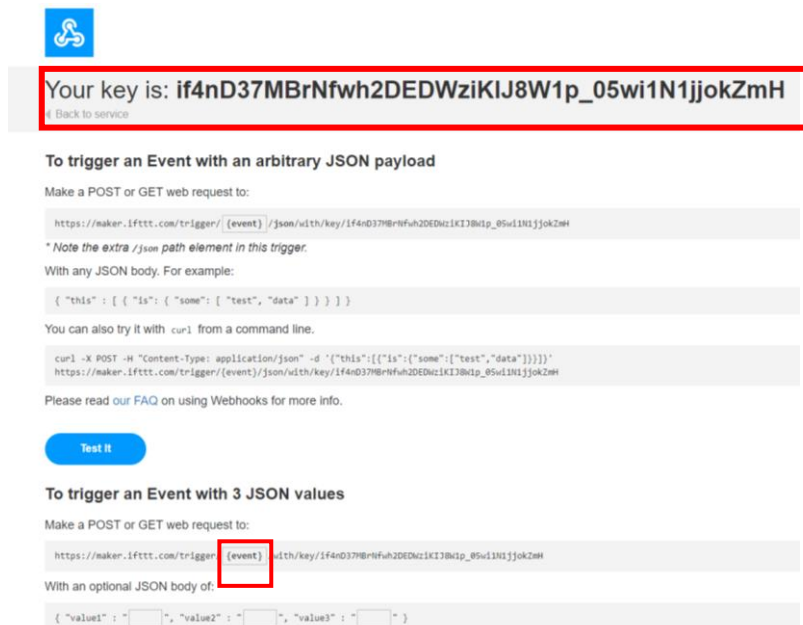
### Popular Webhooks workflows & automations

Εικόνα 6.45 Εισαγωγή στα Documentation του Webhooks.

Στο παράθυρο που εμφανίζεται βρίσκεται το Key. Ακόμα χρειάζεται η δημιουργία του URL που από εκεί θα γίνεται η αποστολή του email. Για την δημιουργία του URL χρειάζεται το key το οποίο συμπληρώνεται αυτόματα από το IFTTT και να συμπληρωθεί στο πεδίο event το event name που δημιουργήσαμε στην αρχή.

Ακολουθεί το URL που δημιουργήθηκε.

[https://maker.ifttt.com/trigger/Warning%20Fire%20Risc/with/key/if4nD37MBrNfwh2DEDWziKIJ8W1p\\_05wi1N1jjokZmH](https://maker.ifttt.com/trigger/Warning%20Fire%20Risc/with/key/if4nD37MBrNfwh2DEDWziKIJ8W1p_05wi1N1jjokZmH)



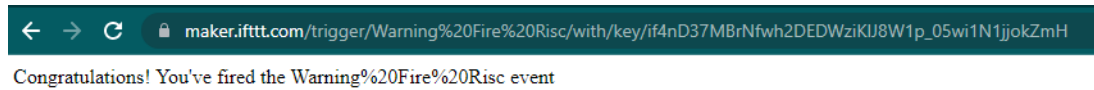
Εικόνα 6.46 Τοποθέτηση event name.

## To trigger an Event with 3 JSON values

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/Warning Fire Risc/with/key/if4nD37MBrNfwh2DEDWziKIJ8W1p_05wi1N1jjokZmH
```

Εικόνα 6.47 Το URL για την αποστολή του Email.



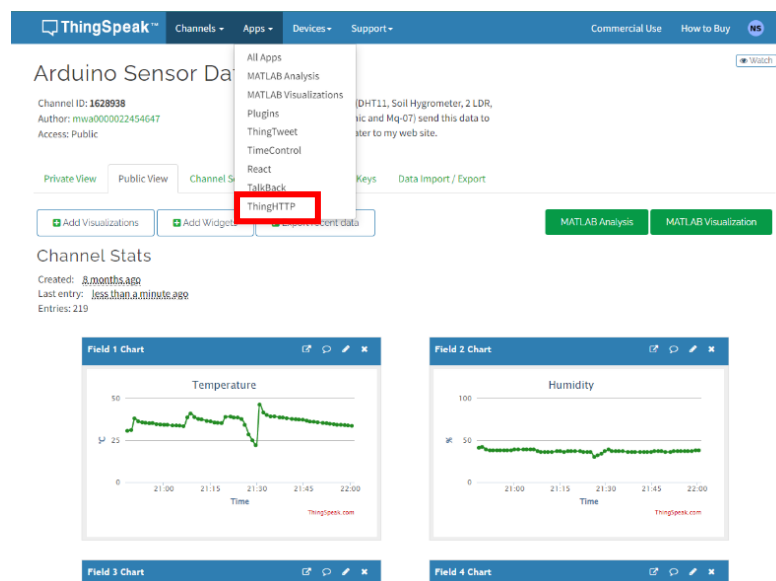
Εικόνα 6.48 Η δοκιμή του URL.

Αφού δημιουργήσαμε τα URL για να στείλουμε email σε περίπτωση αυξημένης θερμοκρασίας, μειωμένης θερμοκρασίας αλλά και σε περίπτωση πυρκαγιάς. Χρειάζεται να χρησιμοποιήσουμε από το ThingSpeak την εφαρμογή Thing Http και το React.

Το Thing Http επιτρέπει την επικοινωνία μεταξύ συσκευών, ιστοσελίδων και υπηρεσιών WEB χωρίς να χρειάζεται να εφαρμοστεί το πρωτόκολλο σε επίπεδο συσκευής. Επίσης μέσω του Thing Http καθορίζονται οι ενέργειες από της υπόλοιπες εφαρμογές του ThingSpeak (React, TweetControl, TimeControl ). [90]

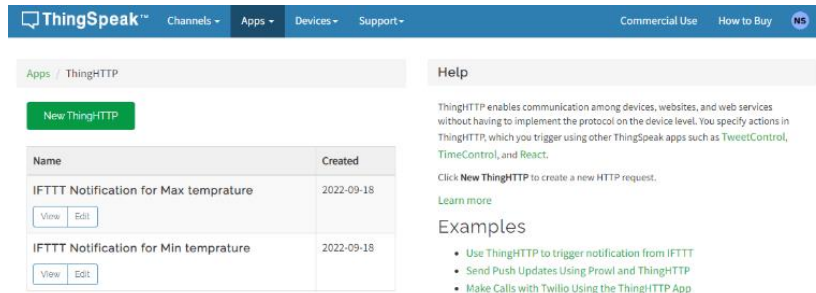
Το React app είναι μια εφαρμογή που συναρπάζεται με το Thing Http, το ThingTweet και MATLAB Analysis για την εκτέλεση ενεργειών όταν τα δεδομένα του καναλιού πληρούν μια συγκεκριμένη συνθήκη. [91]

Για να δημιουργηθεί μια εφαρμογή στο ThingHTTP χρειάζεται να επιλέξουμε στο menu το Apps και υστέρτα το Thing HTTP.

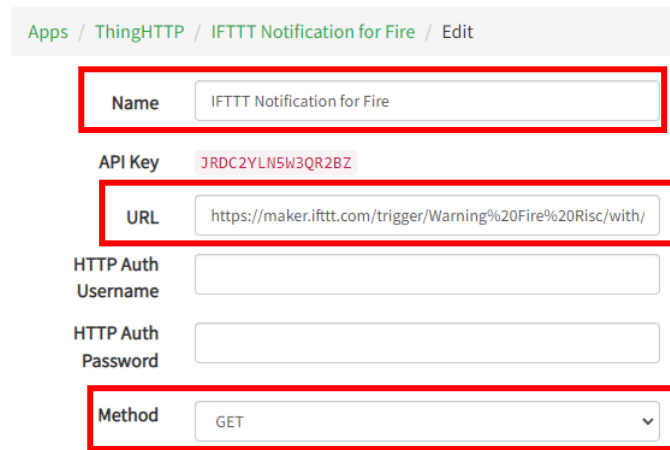


Εικόνα 6.49 Επιλογή του Thing HTTP.

Στην συνέχεια επιλέγεται το κουμπί New ThingHTTP και συμπληρώνεται το όνομα, το URL από το IFTT που δημιουργήθηκε, και η μέθοδος HTTP GET (Εικόνα 6.51).

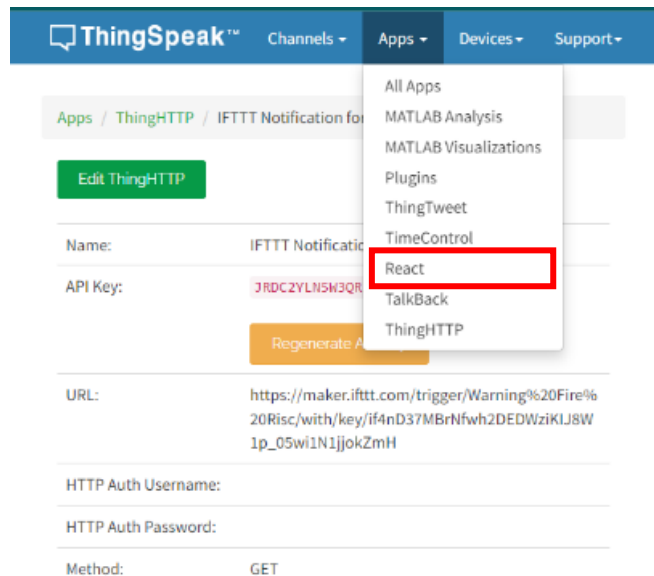


Εικόνα 6.50 Η σελίδα δημιουργίας Thing HTTP.



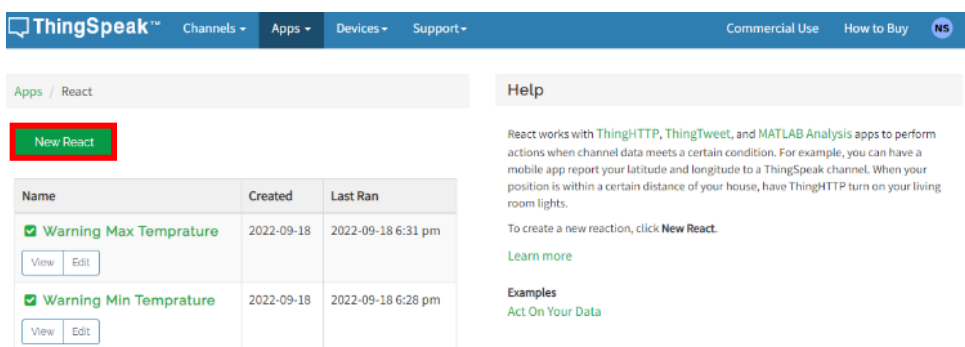
Εικόνα 6.51 Η απαραίτητες ρύθμισης του Thing HTTP.

Ύστερα από την δημιουργία του ThingHTTP πάμε στο menu επιλέγουμε το Apps και στην συνέχεια το React.



Εικόνα 6.52 Επιλογή του React από το menu ThingSpeak

Έπειτα επιλέγοντας το New React και συμπληρώνοντας το React Name με ένα όνομα, το Condition Type σε Numeric, στο Condition το if channel σε Arduino sensor data που είναι η επιλογή του καναλιού που θα αντλεί τα δεδομένα, το field σε 7 (MQ-07 Gas Sensor) και is greater than 50 που είναι εάν στο κανάλι τα δεδομένα από το field ξεπεράσει κάποια στιγμή το 50 τότε πήγαινε στο Action που είναι thingHTTP που είναι το IFTTT notification for fire και τότε στέλνεται το email.

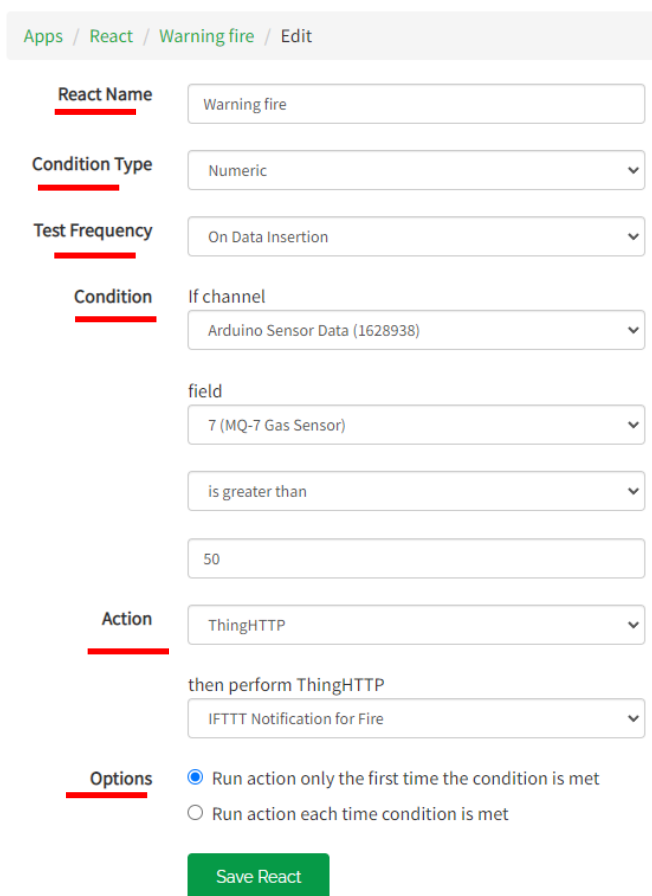


The screenshot shows the 'React' management page in ThingSpeak. At the top, there is a navigation bar with 'Channels', 'Apps', 'Devices', and 'Support'. Below this, the 'React' section is visible, featuring a 'New React' button highlighted in red. A table lists existing reacts:

Name	Created	Last Ran
Warning Max Temperature	2022-09-18	2022-09-18 6:31 pm
Warning Min Temperature	2022-09-18	2022-09-18 6:28 pm

Each row in the table has 'View' and 'Edit' buttons. To the right of the table, there is a 'Help' section with text explaining how Reacts work and a 'Learn more' link. Below that, there are 'Examples' with a link to 'Act On Your Data'.

Εικόνα 6.53 Επιλογή του New React.



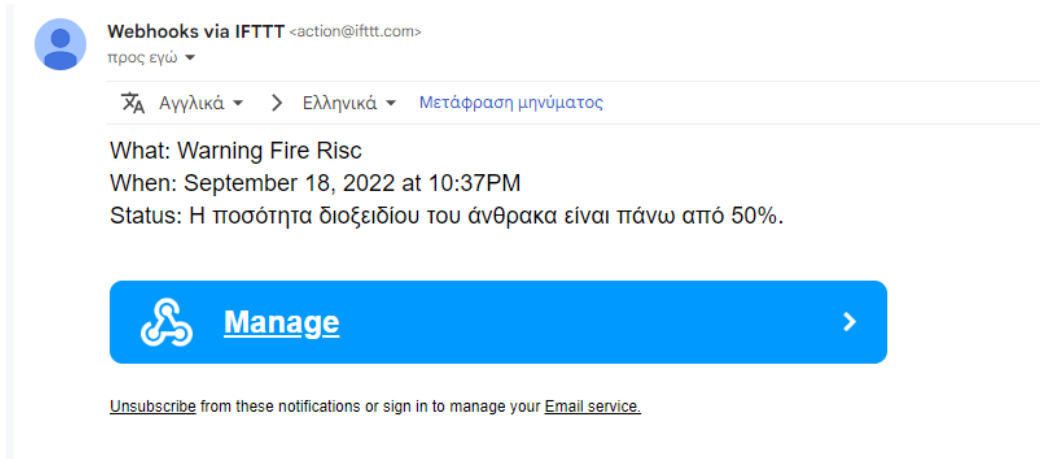
The screenshot shows the configuration page for a React named 'Warning fire'. The page has a breadcrumb trail: 'Apps / React / Warning fire / Edit'. The configuration is as follows:

- React Name:** Warning fire
- Condition Type:** Numeric
- Test Frequency:** On Data Insertion
- Condition:** If channel: Arduino Sensor Data (1628938)  
field: 7 (MQ-7 Gas Sensor)  
operator: is greater than  
value: 50
- Action:** ThingHTTP  
then perform ThingHTTP: IFTTT Notification for Fire
- Options:**  Run action only the first time the condition is met  
 Run action each time condition is met

A green 'Save React' button is located at the bottom of the form.

Εικόνα 6.54 Η απαραίτητες ρυθμίσεις στο React ώστε να σταλεί το email.

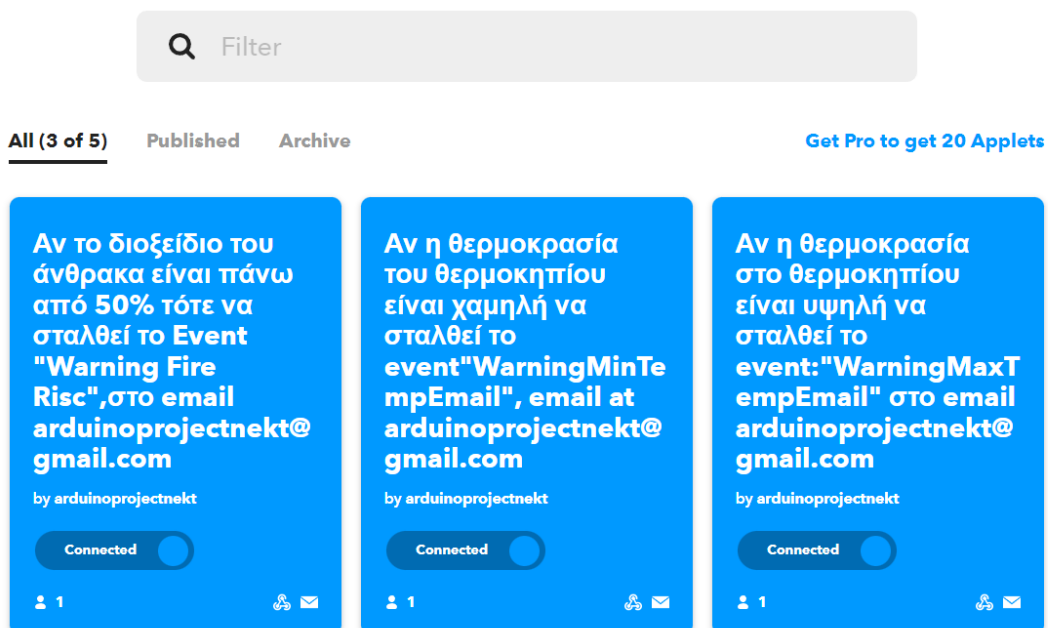




Εικόνα 6.55 Το email που στάλθηκε.

Με τον ίδιο τρόπο έγιναν και τα άλλα δυο Applets, ThingHTTP και τα React app.

## My Applets



Εικόνα 6.56 Τα τρία Applets που δημιουργήθηκαν στο IFTTT.

New ThingHTTP

Name	Created
IFTTT Notification for Max temprature <input type="button" value="View"/> <input type="button" value="Edit"/>	2022-09-18
IFTTT Notification for Min temprature <input type="button" value="View"/> <input type="button" value="Edit"/>	2022-09-18
IFTTT Notification for Fire <input type="button" value="View"/> <input type="button" value="Edit"/>	2022-09-18

Εικόνα 6.57 Τα τρία ThingHTTP που δημιουργήθηκαν στο ThingSpeak.

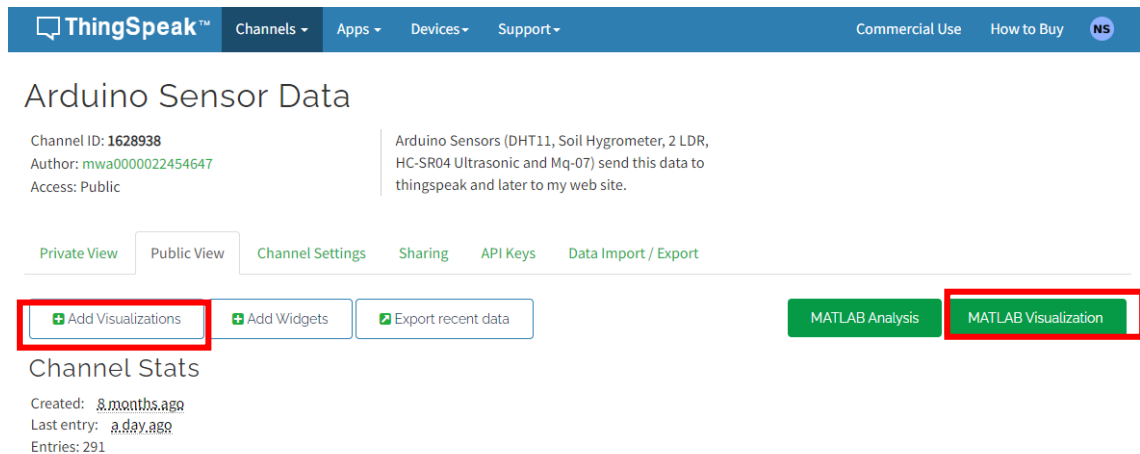
New React

Name	Created	Last Ran
<input checked="" type="checkbox"/> Warning Max Temperature <input type="button" value="View"/> <input type="button" value="Edit"/>	2022-09-18	2022-09-18 6:31 pm
<input checked="" type="checkbox"/> Warning Min Temperature <input type="button" value="View"/> <input type="button" value="Edit"/>	2022-09-18	2022-09-19 9:02 pm
<input checked="" type="checkbox"/> Warning fire <input type="button" value="View"/> <input type="button" value="Edit"/>	2022-09-18	2022-09-19 5:09 pm

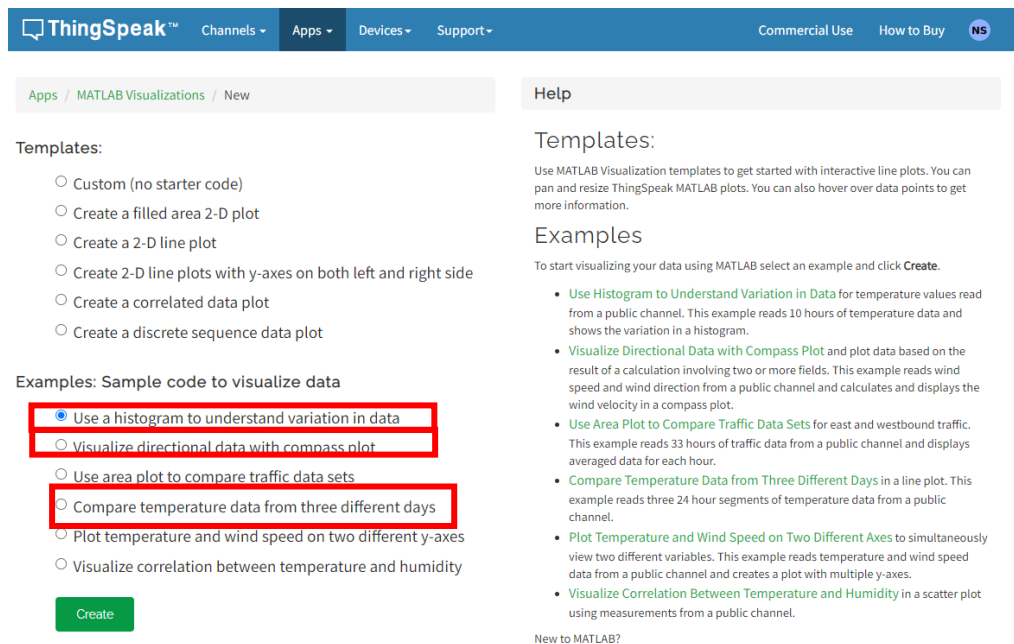
Εικόνα 6.58 Τα τρία React που δημιουργήθηκαν στο ThingSpeak.

### 6.9.3 Οπτικοποίηση δεδομένων μέσω του ThingSpeak

Για την οπτικοποίηση των γραφικών παραστάσεων στο ThingSpeak το μόνο που χρειάζεται είναι η επιλογή του Add Visualization στην καρτέλα Public View και εν συνέχεια να επιλέξουμε τα Field που χρειάζεται να προσθέσουμε.



Εικόνα 6.59 Προσθήκη των Public γραφικών στο ThingSpeak.



Εικόνα 6.60 Στιγμιότυπο γραφικής παράστασης μέσω ανάλυσης MATLAB.

Ενώ για την εμφάνιση των γραφικών

- Use a histogram to understand variation in data
- Compare temperature data from three different days
- Visualize correlation between temperature and humidity

Χρειάζεται να επιλέξουμε στο Public view το MATLAB Visualizations που εμφανίζεται στην [Εικόνα 6.59](#). Έπειτα επιλέγουμε οποιαδήποτε από της επιλογές που εμφανίζονται στην [Εικόνα 6.60](#) και ακολουθούμε την διαδικασία που ακολουθεί.

Για την εμφάνιση του Visualize correlation between temperature and humidity που οπτικοποιεί την σχέση μεταξύ της θερμοκρασίας και υγρασίας χρησιμοποιώντας το διάγραμμα SCATTER χρειάζεται να το επιλέξουμε από το MATLAB Visualizations ([Εικόνα 6.60](#)) και υστέρη στο MATLAB Code ([Εικόνα 6.61](#)) να εισάγουμε το κώδικα που ακολουθεί.

```
% Read temperature and humidity from a ThingSpeak channel and visualize
the
% relationship between them using the SCATTER plot

% Channel 1628938 contains data from Greenhouse in Nafpaktos Greece.
% The data is collected once every minute.
% Field 2 contains the humidity data and field 1 contains temperature
data.

% Channel ID to read data from
readChannelID = 1628938;
% Temperature Field ID
TemperatureFieldID = 1;
% Humidity Field ID
HumidityFieldID = 2;

% Channel Read API Key
% If your channel is private, then enter the read API
% Key between the '' below:
readAPIKey = '0J7LVWUMQSOJ88JS';

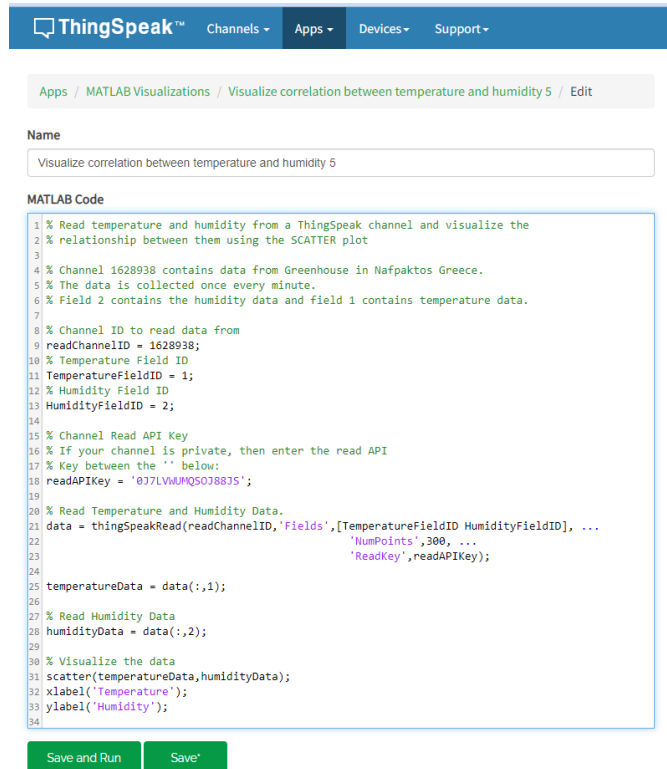
% Read Temperature and Humidity Data.
data = thingSpeakRead(readChannelID,'Fields',[TemperatureFieldID
HumidityFieldID], ...
'NumPoints',300, ...
'ReadKey',readAPIKey);

temperatureData = data(:,1);

% Read Humidity Data
humidityData = data(:,2);

% Visualize the data
scatter(temperatureData,humidityData);
xlabel('Temperature');
```

```
ylabel('Humidity');
```

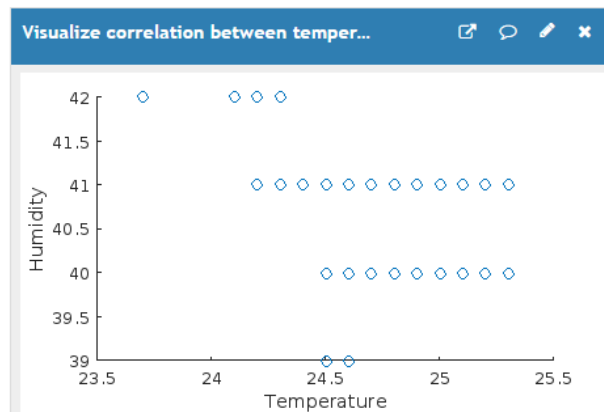


The screenshot shows the ThingSpeak interface for editing a MATLAB visualization. The title is "Visualize correlation between temperature and humidity 5". The MATLAB code is as follows:

```
1 % Read temperature and humidity from a ThingSpeak channel and visualize the
2 % relationship between them using the SCATTER plot
3
4 % Channel 1628938 contains data from Greenhouse in Nafpaktos Greece.
5 % The data is collected once every minute.
6 % Field 2 contains the humidity data and field 1 contains temperature data.
7
8 % Channel ID to read data from
9 readChannelID = 1628938;
10 % Temperature Field ID
11 TemperatureFieldID = 1;
12 % Humidity Field ID
13 HumidityFieldID = 2;
14
15 % Channel Read API Key
16 % If your channel is private, then enter the read API
17 % Key between the '' below:
18 readAPIKey = '0j7LVMMQ5038835';
19
20 % Read Temperature and Humidity Data.
21 data = thingSpeakRead(readChannelID,'Fields',[TemperatureFieldID HumidityFieldID], ...
22                       'NumPoints',300, ...
23                       'ReadKey',readAPIKey);
24
25 temperatureData = data(:,1);
26
27 % Read Humidity Data
28 humidityData = data(:,2);
29
30 % Visualize the data
31 scatter(temperatureData,humidityData);
32 xlabel('Temperature');
33 ylabel('Humidity');
```

Εικόνα 6.61 Εισαγωγή κώδικα Visualize correlation between temperature and humidity στο MATLAB Code.

Το αποτέλεσμα του κώδικα είναι η παρακάτω γραφική (Εικόνα 6.62).



Εικόνα 6.62 Η έξοδος του κώδικα Visualize correlation between temperature and humidity

Για την εμφάνιση του Use a histogram to understand variation in data που οπτικοποιεί τις διακυμάνσεις της θερμοκρασίας για τις τελευταίες 10 ώρες από ένα κανάλι ThingSpeak χρησιμοποιώντας τη λειτουργία MATLAB HISTOGRAM χρειάζεται να το επιλέξουμε από το MATLAB Visualizations (Εικόνα 6.60) και υστέρτα στο MATLAB Code (Εικόνα 6.61) να εισάγουμε το κώδικα που ακολουθεί.

```

% Read temperature for the last 10 hours from a ThingSpeak channel and
% visualize temperature variations using the MATLAB HISTOGRAM function.

% Channel 1628938 contains data from Greenhouse in Nafpaktos Greece.
% The data is collected once every minute.
% Field 1 contains temperature data.

% Channel ID to read data from
readChannelID = 1628938;

% Temperature Field ID
TemperatureFieldID = 1;

% Channel Read API Key
% Enter the read API key between the '' below:
readAPIKey = '0J7LVWUMQSOJ88JS';

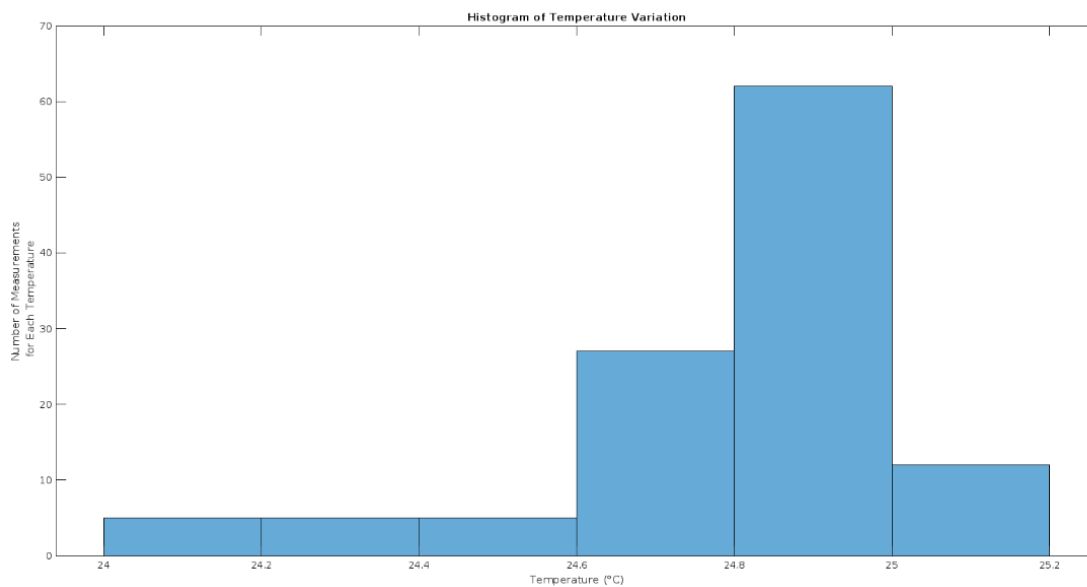
% Get temperature data from field 1 for the last 10 hours = 10 x 60 min

tempF = thingSpeakRead(readChannelID,'Fields',TemperatureFieldID,...
'NumMinutes',10*60, 'ReadKey',readAPIKey);

histogram(tempF);
xlabel('Temperature (°C)');
ylabel('Number of Measurements\nfor Each Temperature');
title('Histogram of Temperature Variation');

```

Το αποτέλεσμα του κώδικα είναι η παρακάτω γραφική ([Εικόνα 6.63](#)).



Εικόνα 6.63 Διακύμανσή της θερμοκρασίας για τις τελευταίες 10 ώρες.

Για την εμφάνιση του Compare temperature data from three different days που οπτικοποιή τα δεδομένα θερμοκρασίας από ένα κανάλι ThingSpeak για τρεις ξεχωριστές ημέρες σε ένα μόνο διάγραμμα χρησιμοποιώντας τη συνάρτηση PLOT χρειάζεται να το επιλέξουμε από το MATLAB Visualizations ([Εικόνα 6.60](#)) και υστέρτα στο MATLAB Code ([Εικόνα 6.61](#)) να εισάγουμε το κώδικα που ακολουθεί.

```
% Read temperature data from a ThingSpeak channel for three separate days
% and visualize the data in a single plot using the PLOT function.

% Channel 1628938 contains data from Greenhouse in Nafpaktos Greece.
% The data is collected once every minute.
% Field 1 contains temperature data.

% Channel ID to read data from
readChannelID = 1628938;
% Temperature Field ID
myFieldID = 1;
% One day date range
oneDay = [datetime('yesterday') datetime('today')];

% Channel Read API Key
% Enter the read API key between the '' below:
readAPIKey = '0J7LVWUMQSOJ88JS';

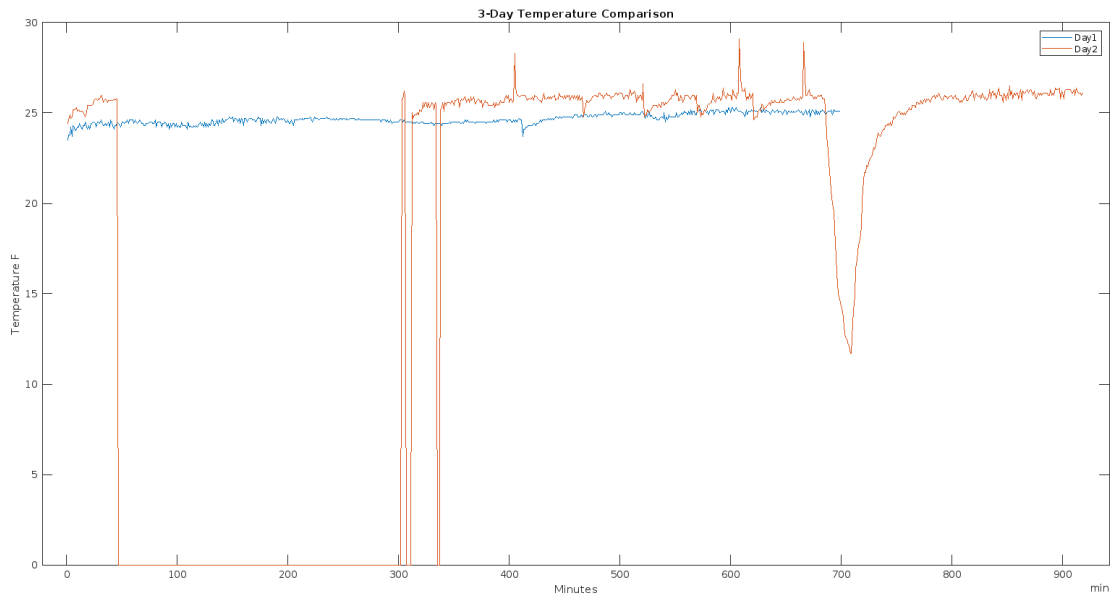
% Read Temperature Data.
temperatureDay1 = thingSpeakRead(readChannelID,'Fields',myFieldID, ...
    'dateRange', oneDay,
    'ReadKey',readAPIKey);
temperatureDay2 = thingSpeakRead(readChannelID,'Fields',myFieldID, ...
    'dateRange',oneDay-
days(1), 'ReadKey',readAPIKey);
temperatureDay3 = thingSpeakRead(readChannelID,'Fields',myFieldID, ...
    'dateRange', oneDay-
days(2), 'ReadKey',readAPIKey);

% Create array of durations
myTimes1 = minutes(1:length(temperatureDay1));
myTimes2 = minutes(1:length(temperatureDay2));
myTimes3 = minutes(1:length(temperatureDay3));

% Visualize the data
plot(myTimes1,temperatureDay1, myTimes2,temperatureDay2, myTimes3,
temperatureDay3);
legend({'Day1', 'Day2', 'Day3'});
xlabel('Minutes');
ylabel('Temperature F');
title('3-Day Temperature Comparison');
```

Το αποτέλεσμα του κώδικα είναι η παρακάτω γραφική ([Εικόνα 6.64](#)).





Εικόνα 6.64 Η θερμοκρασία για τρεις ξεχωριστές ημέρες σε ένα γράφημα.

Για την προσθήκη των γραφικών στην ιστοσελίδα χρειάζεται να πατήσουμε το κυκλωμένο κουμπί. Υστερα να γίνει αντιγράφι του κώδικα που εμφανίζεται στο παράθυρο και να το εισάγουμε στον κώδικα της ιστοσελίδας.

The screenshot shows the ThingSpeak dashboard interface. At the top, there are fields for ID (1628938) and API key (mwa0000022454647). Below these are buttons for "View", "Public View", and "Chart". A red box highlights a configuration window for "Field 1 Chart IFrame" containing an `<iframe>` tag. Below the configuration are buttons for "Add Widgets", "Export recent data", and a "MATLAB" button. The main dashboard area shows "Inel Stats" with a "Public View" button and a "Share" button circled in red. Two charts are displayed: "Field 1 Chart" showing "Temperature" and "Field 2 Chart" showing "Humidity". Both charts show data for the period from 19 Sep 08:00 to 20 Sep 08:00. The Temperature chart shows a value around 25, and the Humidity chart shows a value around 50.

Εικόνα 6.65 Προσθήκη γραφικών από το ThingSpeak στην ιστοσελίδα.

Παρακάτω ακολουθεί ο κώδικας graph.html για την εμφάνιση όλων των γραφικών από το ThingSpeak στην ιστοσελίδα.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>ThingSpeak Graph Plot</title>
<link rel="stylesheet" href="../../stylepage/style3.css">
</head>
<body>
<h1>Smart Green House</h1>

<!-- Δημιουργία menu -->
<div class="menu-bar">
<ul>

<li class="active"><a href="http://192.168.1.12/db_allSensor/"><i
class="fa-solid fa-house-blank"></i>HOME</a></li>

<li> <a href="http://192.168.1.8/"><i class="fa-solid fa-house-
blank"></i>Dashboard </a></li>
<li> <a href="../../menu/dbtable/display.php"><i class="fa-solid fa-
house-blank"></i>Data sensor table</a></li>
<li> <a href="https://localhost/db_allsensor/menu/dbgauge/gauge.html"><i
class="fa-solid fa-house-blank"></i>Gauge Visualization</a></li>

<li>
<a href="#">Graph</a>
<Div class="sub_menu-1">
<ul>
<li><a href="../../menu/dbgraph/graph.php">Temprature</a></li>
<li><a href="../../menu/dbgraph/humgraph.php">Humidity</a></li>
<li><a href="../../menu/dbgraph/outldr.php">Ldr out</a></li>
<li><a href="../../menu/dbgraph/tankgraph.php">Level Tank</a></li>
<li><a href="../../menu/dbgraph/soilhygrograph.php">Soil Hugro</a></li>
<li><a href="../../menu/dbgraph/inldrgraph.php">Ldr IN</a></li>
<li><a href="../../menu/dbgraph/gas.php">Gas</a></li>
</ul>
</Div>
</li>
<li>
<a href="#">ThingSpeak</a>
<div class="sub_menu-1">
<ul>
<li><a href="../../menu/thingspeak/graph.html">Graph</a></li>
<li><a href="../../menu/thingspeak/NumericDisplay.html">Numeric
Display</a></li>
</ul>
</div>
</li>
</ul>
</div>

<!-- Εμφάνιση των γραφικών μέσω του ThingSpeak στην ιστοσελίδα-->
<!-- temperature -->
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/charts/1?bgcolor=%23ffffff&c
olor=%23228B22&dynamic=true&max=50&min=0&results=60&title=Temperature&typ
e=spline&xaxis=Time&yaxis=%C2%B0C&yaxismax=50&yaxismin=0"></iframe>

```

```

<!-- Humidity -->
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/charts/2?bgcolor=%23FFFF&col
or=%23228B22&dynamic=true&max=100&min=0&results=60&title=Humidity&type=sp
line&xaxis=Time&yaxis=%25&yaxismax=100&yaxismin=0"></iframe>
<!-- Soil Hygrometer -->
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/charts/3?bgcolor=%23ffffff&c
olor=%23228B22&dynamic=true&max=100&min=0&results=60&title=SoilHygrometer
&type=spline&xaxis=Time&yaxis=%25&yaxismax=100&yaxismin=0"></iframe>

<!-- LDR Sensor IN -->
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/charts/4?bgcolor=%23ffffff&c
olor=%23008000&dynamic=true&max=100&min=0&results=60&title=LDR+Sensor+IN&
type=line&xaxis=Time&yaxis=%25&yaxismax=100&yaxismin=0"></iframe>

<!-- LDR Sensor Out -->
<iframe width="500" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/charts/5?bgcolor=%23ffffff&c
olor=%23008000&dynamic=true&max=100&min=0&results=60&title=LDR+Sensor+Out
&type=spline&xaxis=Time&yaxis=%25&yaxismax=100&yaxismin=0"></iframe>

<!-- Water Level -->
<iframe width="500" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/charts/6?bgcolor=%23ffffff&c
olor=%23228B22&dynamic=true&max=100&min=0&results=60&title=Water+Level&ty
pe=spline&xaxis=Time&yaxis=%25&yaxismax=100&yaxismin=0"></iframe>

<!-- Mq-07 Gas Sensor -->
<iframe width="500" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/charts/7?bgcolor=%23ffffff&c
olor=%23008000&dynamic=true&max=100&min=0&results=60&title=Mq-
07+Gas+Sensor&type=spline&xaxis=Time&yaxis=%25&yaxismax=100&yaxismin=0">
</iframe>

<!-- Visualize correlation between temperature and humidity-->
<iframe width="500" height="300" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/apps/matlab_visualizations/478119"></iframe>

<!-- Use a histogram to understand variation in data-->
<iframe width="500" height="300" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/apps/matlab_visualizations/478116"></iframe>

<!-- Compare temperature data from three different days-->
<iframe width="500" height="300" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/apps/matlab_visualizations/478123"></iframe>

<p></p>

<!-- URL για την εμφάνιση των γραφικών στην ιστοσελίδα ThingSpeak -->
<h1><a
href="https://thingspeak.com/channels/1628938">MyThinkSpeak</a></h1>
</body>
</html>

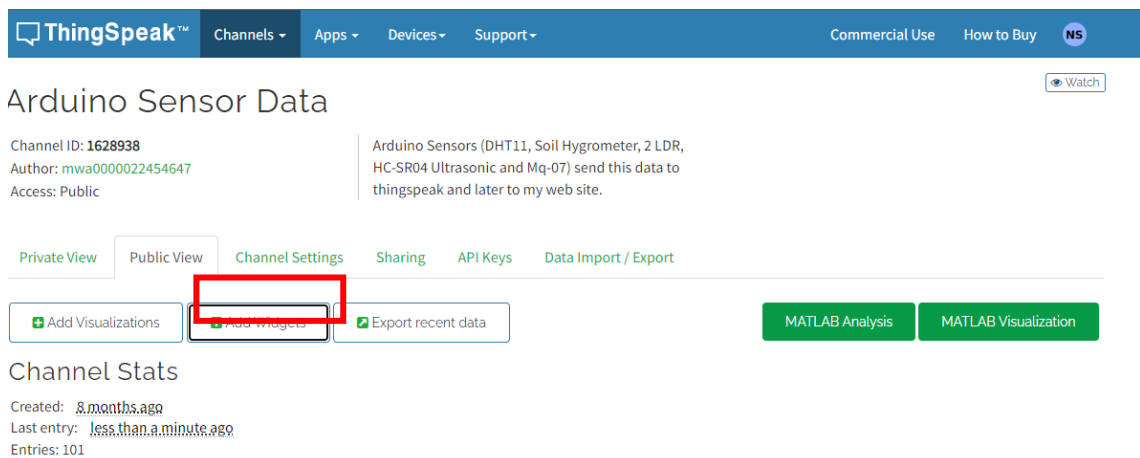
```

Το αποτέλεσμα του κώδικα μαζί με την χρήση του αρχείου style3.css (Εικόνα 6.66).



Εικόνα 6.66 Εμφάνιση γραφικών από το ThingSpeak.

Έπειτα για την οπτικοποίηση των δεδομένων μέσω των μετρητών αλλά και σταθερών τιμών χρειάστηκε να επιλεγθεί το Add Widgets (Εικόνα 6.67).

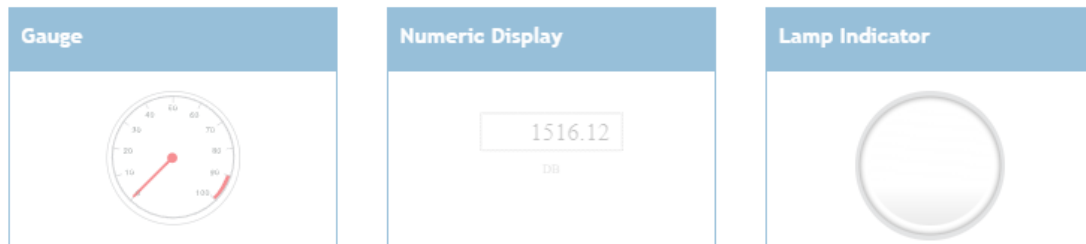


Εικόνα 6.67 Στιγμιότυπο οθόνης που φαίνεται η επιλογή Add Widgets.

Επιλέγονται τα Widgets που θα χρησιμοποιηθούν (Εικόνα 6.68).

Click on a widget to add it to the Channel

x



Next

Cancel

Εικόνα 6.68 Στιγμιότυπο οθόνης που φαίνονται οι επιλογές των Widgets που προσφέρει το ThingSpeak.

Για την εμφάνιση των διαγραμμάτων ThingSpeak στην ιστοσελίδα μας σε μορφή μετρητών, αλλά και σταθερών τιμών, χρειάζεται να χρησιμοποιηθεί ο κώδικας NumericDisplay.html που ακολουθεί.

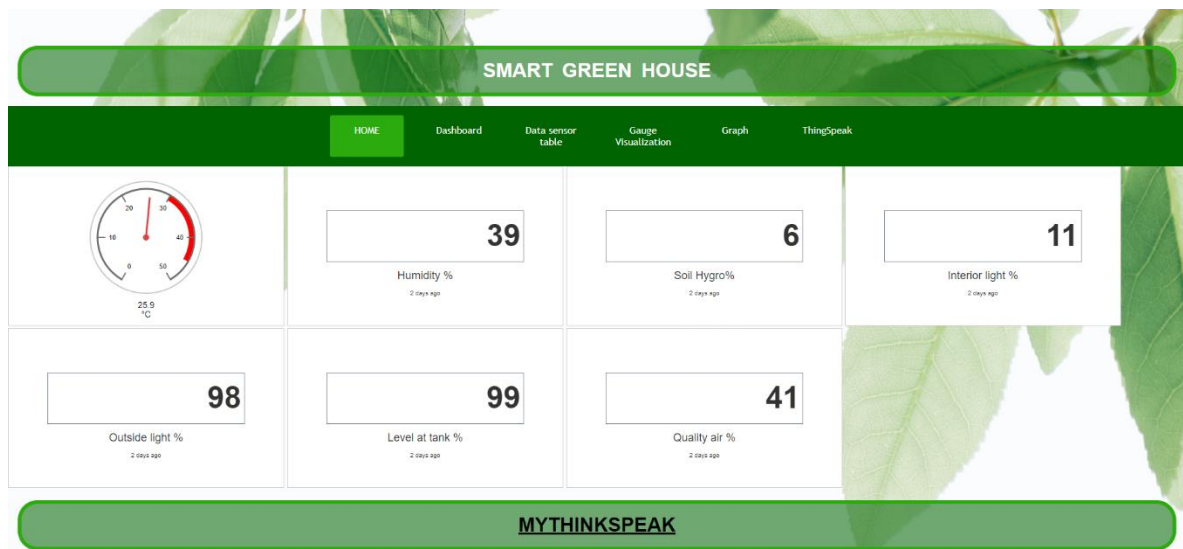
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>ThingSpeak Graph Plot</title>
<link rel="stylesheet" href="../../stylepage/style3.css">
</head>
<body>
<h1>Smart Green House</h1>
<!-- Δημιουργία menu -->
<div class="menu-bar">
<ul>
<li class="active"><a href="http://192.168.1.12/db_allSensor/"><i
class="fa-solid fa-house-blank"></i>HOME</a></li>
<li> <a href="http://192.168.1.7/"><i class="fa-solid fa-house-
blank"></i>Dashboard </a></li>
<li> <a href="../../menu/dbtable/display.php"><i class="fa-solid fa-
house-blank"></i>Data sensor table</a></li>
<li> <a href="https://localhost/db_allsensor/menu/dbgauge/gauge.html"><i
class="fa-solid fa-house-blank"></i>Gauge Visualization</a></li>
<li>
<a href="#">Graph</a>
<Div class="sub_menu-1">
<ul>
<li><a href="../../menu/dbgraph/graph.php">Temprature</a></li>
<li><a href="../../menu/dbgraph/humgraph.php">Humidity</a></li>
<li><a href="../../menu/dbgraph/outldr.php">Ldr out</a></li>
<li><a href="../../menu/dbgraph/tankgraph.php">Level Tank</a></li>
<li><a href="../../menu/dbgraph/soilhygrograph.php">Soil Hugro</a></li>
<li><a href="../../menu/dbgraph/inldrgraph.php">Ldr IN</a></li>
<li><a href="../../menu/dbgraph/gas.php">Gas</a></li>
</ul>
</Div>
</li>
</ul>
</div>
</li>
```

```

</li>
<a href="#">ThingSpeak</a>
<div class="sub_menu-1">
<ul>
<li><a href="../../menu/thingspeak/graph.html">Graph</a></li>
<li><a href="../../menu/thingspeak/NumericDisplay.html">Numeric
Display</a></li>
</ul>
</div>
</li>
</ul>
</div>
<!-- Εμφάνιση των Widgets ThingSpeak στην ιστοσελίδα -->
<!-- temperature -->
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/widgets/512557"></iframe>
<!-- Humidity -->
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/widgets/430274"></iframe>
<!-- SoilHygrometer -->
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/widgets/430275"></iframe>
<!-- LDR Sensor IN -->
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/widgets/430276"></iframe>
<!-- LDR Sensor Out -->
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/widgets/430277"></iframe>
<!-- Water Level -->
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/widgets/430278"></iframe>
<!-- Mq-07 Gas Sensor -->
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/1628938/widgets/430279"></iframe>
<p></p>
<!-- URL για την εμφάνιση των γραφικών στην ιστοσελίδα ThingSpeak -->
<h1><a
href="https://thingspeak.com/channels/1628938">MyThinkSpeak</a></h1>
</body>
</html>

```

Το αποτέλεσμα του κώδικα NumericDisplay.html μαζί με την χρήση του αρχείου style3.css (Εικόνα 6.69).



Εικόνα 6.69 Σταθερές τιμές στο ThingSpeak στην Ιστοσελίδα.

## 6.10 Έλεγχος συσκευών εξόδου

Στην ιστοσελίδα, εάν πάμε στο menu control device, έχουμε την δυνατότητα να ορίσουμε πως θα λειτουργεί το σύστημα. Να το ορίσουμε ως Auto, δηλαδή να λειτουργεί με τα σενάρια αυτοματισμού, ή να λειτουργεί ως manual που μπορούμε να ενεργοποιήσουμε τον ανεμιστήρα για μείωση της θερμοκρασίας, τον ανεμιστήρα εξαερισμού και τις υπόλοιπες συσκευές δράσης που είναι συνδεδεμένες και στο Xbee και στο Arduino.

Ακολουθεί ο κώδικας Arduino για τον έλεγχο των συσκευών

```
void control_NETarea_loop() {
  // Create a client connection
  EthernetClient client = server.available();
  if(client){
    Serial.println(F("new client"));
    boolean currentLineIsBlank = true;
    String postText=""; //for switch
    // an http request ends with a blank line
    while (client.connected()){
      if(client.available()){
        char c = client.read();
        //-----
        //read char by char HTTP request
        if (postText.length() < 10) {
          //store characters to string
          postText += c;
          Serial.print(c);
        }
        //-----
        HTTP_req += c;
        if(c == '\n' && currentLineIsBlank){
          if(HTTP_req.indexOf("read") > -1){ //AJAX request for DHT11
values
            AJAX_request(client);
          }else{
```



```

        HTML_webpage();
        client.println(webPage);
    }

    Serial.print(HTTP_req);
    HTTP_req = ""; //reset HTTP request string
    break;
}

if(c == '\n') currentLineIsBlank = true;
else if(c != '\r') currentLineIsBlank = false;
}
}

//controls the Arduino if you press the buttons

//-----sw1--auto/manual-----
if (postText.indexOf("?on1") >0){ //auto
    digitalWrite(Mode, LOW);
    Val_Set_Mode= true;
}
if (postText.indexOf("?off1") >0){ //manual
    digitalWrite(Mode, HIGH);
    Val_Set_Mode= false;
}

//-----sw2---Fans-----
if (postText.indexOf("?on2") >0){
    digitalWrite(MOS_AirRecyclingFan, LOW);
    digitalWrite(MOS_CoolingFan, LOW);
    Val_Fans= true;
}
if (postText.indexOf("?off2") >0){
    digitalWrite(MOS_AirRecyclingFan, HIGH);
    digitalWrite(MOS_CoolingFan, HIGH);
    Val_Fans= false;
}

//-----sw3-Watering plants valve and pump 1
if (postText.indexOf("?on3") >0){
    digitalWrite(RELAY_Pump1, LOW);
    digitalWrite(RELAY_ValveWater, LOW);
    digitalWrite(LED_Pump2, LOW);
    Val_Watering= true;
}
if (postText.indexOf("?off3") >0){
    digitalWrite(RELAY_Pump1, HIGH);
    digitalWrite(RELAY_ValveWater, HIGH);
    digitalWrite(LED_Pump2, HIGH);
    Val_Watering= false;
}

//-----sw4--Light-----
if (postText.indexOf("?on4") >0){
    digitalWrite(MOS_RgbLeds, LOW);
    digitalWrite(RELAY_HeatingLight, LOW);
    Val_Light= true;
}
if (postText.indexOf("?off4") >0){
    digitalWrite(MOS_RgbLeds, HIGH);
    digitalWrite(RELAY_HeatingLight, HIGH);
    Val_Light= false;
}
}
}

```

```

//-----sw5--Alarm-----
if (postText.indexOf("?on5") >0){
    setRem_LED_RED_Org_ALERT_DIO4(0x4);
    Val_Alert= true;
}
if (postText.indexOf("?off5") >0){
    setRem_LED_RED_Org_ALERT_DIO4(0x5);
    Val_Alert= false;
}

//*****
if (Val_Set_Mode== false){ //manual
    System_Light_Loop();
    System_SunsHade_Loop();
    System_Water_Loop();
    System_Temperature_Loop();
    System_ForggerSpray_Loop();
    System_Alert_Loop();
    System_AirRecycling_Loop();
}else{ //auto
    style_controll_loop(client);
}

style_controllMode(client);
delay(10);
client.stop(); //sever client connection with server
}
}
//*****Start***Switch
void style_controllMode(EthernetClient cl){
    Serial.print("Auto");
    if (Val_Set_Mode){
        cl.println("<a class=""status2"" href=\\\"./?off1\\\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/off1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }else{
        cl.println("<a class=""status2"" href=\\\"./?on1\\\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/on1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }
}

void style_controll_loop(EthernetClient cl){
    Serial.print("Manual");
    if (Val_Fans){
        cl.println("<a class=""status3"" href=\\\"./?off2\\\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/off1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }else{
        cl.println("<a class=""status3"" href=\\\"./?on2\\\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/on1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }
    if (Val_Watering){
        cl.println("<a class=""status4"" href=\\\"./?off3\\\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/off1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }else{
        cl.println("<a class=""status4"" href=\\\"./?on3\\\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/on1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }
}

```

```

    }
    if (Val_Light){
        cl.println("<a class=""status5"" href=""./?off4""><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/off1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }else{
        cl.println("<a class=""status5"" href=""./?on4""><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/on1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }
    if (Val_Alert){
        cl.println("<a class=""status6"" href=""./?off5""><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/off1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }else{
        cl.println("<a class=""status6"" href=""./?on5""><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/on1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }
}

void AJAX_request(EthernetClient client){
    client.print("Online");
}

void HTML_webpage() {
    //HTTP response header
    webPage = "HTTP/1.1 200 OK\r\nContent-Type:
text/html\r\n\r\n"; //versions protocol->HTTP/1.1 or HTTP/2

    //-----
    //webpage - HTML + JavaScript with AJAX calls
    webPage += R"*** (
<!DOCTYPE html>
<html lang="en">
<head>
<TITLE>Control area</TITLE>
<link rel='stylesheet' tye='text/css'
href='http://192.168.1.12/db_allSensor/stylepage/StyleArduino.css' />
</head>
<html>
<body>
<H1>Control area</H1>
<div class=""menu-bar"">

<ul>
    <li class="active"><a
href='https://192.168.1.12/db_allsensor/index.html'><i class="fa-solid
fa-house-blank"></i>HOME</a></li>

</UL>

</DIV>

<div class="table-box">
    <div class="table-head">

        <div class="table-cell first-cell">
            <p>Devices</p>
        </div>

        <div class="table-cell last-cell">

```

```

        <p>Switch</p>
    </div>

</div>
<div class="table-row">
    <span style="color:rgb(196, 3, 3)" id="IdVals">N/A</span>
</div>

<div class="table-row">
    <p>Auto </p>
</div>

<div class="table-row">
    <p>Fans</p>
</div>

<div class="table-row">
    <p>Watering plants</p>
</div>

<div class="table-row">
    <p>Lights</p>
</div>

<div class="table-row">
    <p>Xbee Device</p>
</div>
</div>

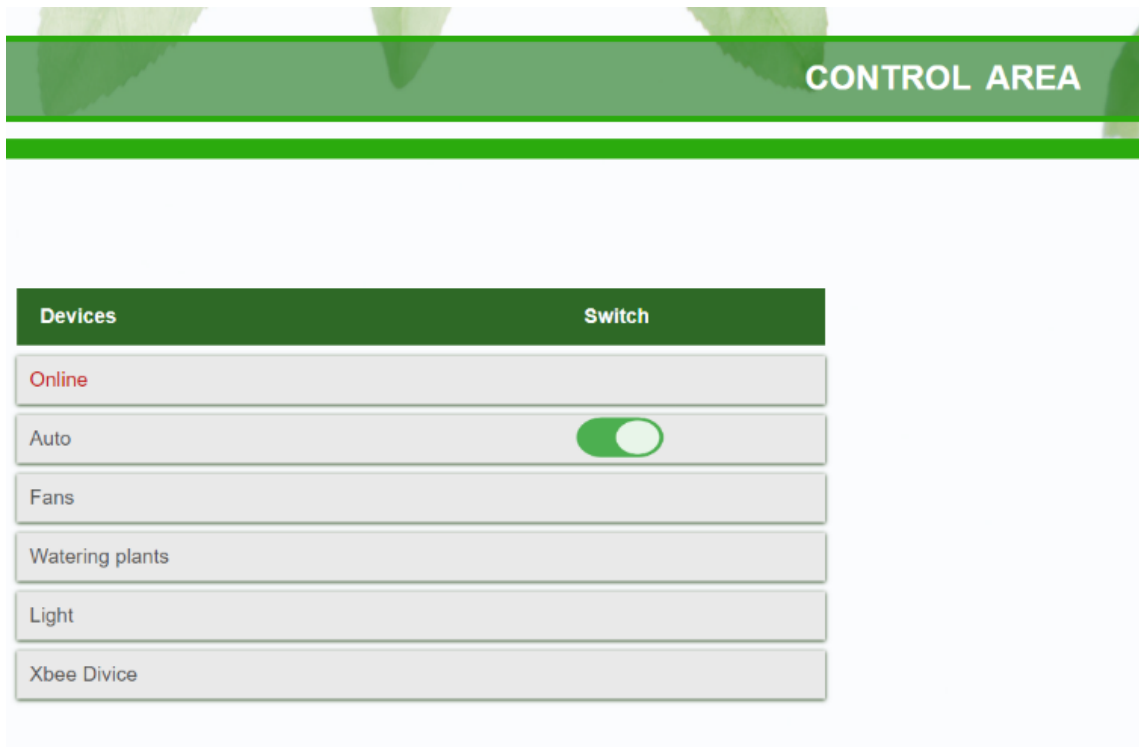
<!-------JavaScript----->
<script>
    setInterval(
        function(){
            getIdVals();
        }, 2000
    );

    function getIdVals(){
        var Request = new XMLHttpRequest();
        Request.onreadystatechange = function(){
            if(this.readyState == 4 && this.status == 200){
                document.getElementById("IdVals").innerHTML =
this.responseText;
            }
        };
        Request.open("GET", "read", true);
        Request.send();
    }
</script>

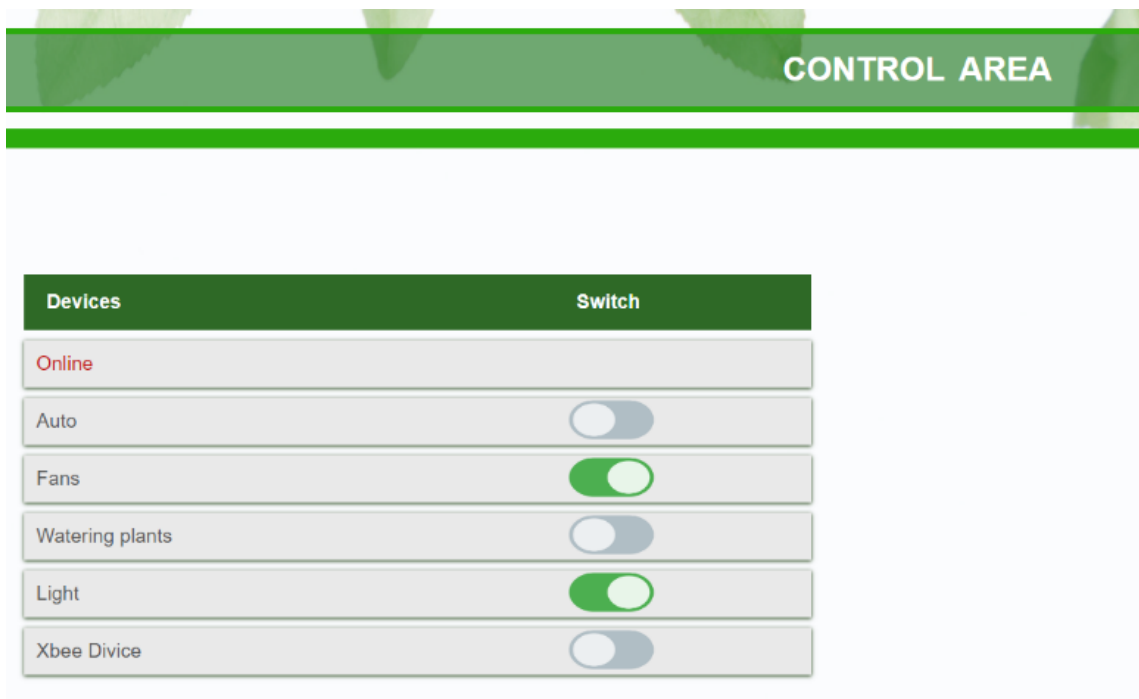
</body>
</html>
)***";
}
//////*****End***Switch

```

Το αποτέλεσμα του κώδικα.



Εικόνα 6.70 Απεικόνιση της ενεργοποίησης σε αυτόματη επιλογή.



Εικόνα 6.71 Απεικόνιση σε χειροκίνητη επιλογή όλα τα Switch κλειστά.

## 7 Συμπεράσματα και προτάσεις περαιτέρω ανάπτυξης

### 7.1 Συμπεράσματα και ανακεφαλαιώσει

Σε αυτήν την διπλωματική εργασία έγινε υλοποίηση ενός IoT θερμοκηπίου που έχει ως σκοπό τον έλεγχο και την παρακολούθηση των κλιματικών συνθηκών για την καλύτερη δυνατή ανάπτυξη των φυτών αλλά και την εξοικονόμηση των πόρων και ενέργειας. Το θερμοκήπιο το οποίο υλοποιήθηκε εκτελεί διάφορα σενάρια αυτοματισμού και έχει την δυνατότητα απομακρυσμένου ελέγχου και παρακολούθησης μέσω της ιστοσελίδας που δημιουργήθηκε.

Σε αυτή την διπλωματική εργασία χρησιμοποιήθηκε η πλατφόρμα Arduino, δημιουργήθηκαν σενάρια αυτοματισμού, έγινε εξομοίωση με τη λειτουργία του Internet και την ασύρματη επικοινωνία Zigbee. Επίσης, χρησιμοποιήθηκε η πλατφόρμα XCTU και για τον προγραμματισμό των Xbee. Έγινε χρήση της πλατφόρμας IoT ThingSpeak, της πλατφόρμας Ifttt και του προγράμματος Fritzng. Τέλος, δημιουργήθηκε βάση δεδομένων μέσω της MySQL και ιστοσελίδας με τις διαθέσιμες γλώσσες προγραμματισμού (Html, php,Css, JS,C++).

### 7.2 Προτάσεις περαιτέρω ανάπτυξης

Στη παραπάνω εργασία υπάρχουν πολλά περιθώρια αναβάθμισης τα οποία ανάλογα με την φαντασία του καθένα δεν έχουν όρια, παρακάτω αναφέρονται μερικές ιδέες. Μια αναβάθμιση που μπορεί να γίνει είναι η προσθήκη περισσότερων Xbee ώστε να γίνεται καλύτερη εφαρμογή του πρωτοκόλλου Zigbee. Επίσης, θα μπορούσε να συνδεθεί ένα Xbee με τον υπολογιστή για εμφάνιση της σειριακής οθόνης του Arduino ασύρματα αλλά και να είναι δυνατός ο έλεγχος και η παρακολούθηση από εκεί.

Μια καλή ιδέα είναι το Arduino να στέλνει τα δεδομένα σε ένα Raspberry Pi και αυτό να λειτουργεί ως Web Server. Ακόμα θα μπορούσαν να προστεθούν και άλλα σενάρια αυτοματισμού ελέγχου όπως ο έλεγχος του Ph του εδάφους, ο έλεγχος και ο εμπλουτισμός του οξυγόνου. Συμπληρωματικά θα μπορούσε να χρησιμοποιηθεί μια συσκευή όπως το ESP8266 για να γίνει σύνδεση στο τοπικό δίκτυο μέσω Wi-Fi είτε να χρησιμοποιεί ένα GSM Shield, ώστε να συνδεθεί στο διαδίκτυο μέσω κάρτας SIM .

Επιπρόσθετα, θα μπορούσε να προστεθεί ένας αναλυτής ενέργειας για την πλήρη εικόνα της ενέργειας που καταναλώνει το θερμοκήπιο. Επίσης μια καλή ιδέα είναι ο συνδυασμός του

αναλυτή ενέργειας με ένα φωτοβολταϊκό πάνελ και με μπαταρίες για την πλήρη αυτονομία και τον έλεγχο της ενέργειας του θερμοκηπίου.

Μια ακόμα αναβάθμιση που θα μπορούσε να γίνει είναι να προστεθούν και άλλες λειτουργίες στην ιστοσελίδα, όπως το να ορίζονται από τον χρήστη οι επιθυμητές τιμές του μικροκλίματος, όπως το να μπορεί ο χρήστης να επεξεργάζεται τις γραφικές.

Τέλος, θα μπορούσε να γίνεται ανάλυση των διάφορων τιμών που θα στέλνουν οι αισθητήρες ώστε να γίνεται πρόβλεψη των κλιματικών αλλαγών αλλά και των διάφορων ασθενειών που μπορεί να προκληθούν στην καλλιέργεια. Το ιδανικό θα ήταν ο συνδυασμός με την μηχανική μάθηση ώστε το σύστημα να λειτουργεί με βάση της ενέργειες που έχουν γίνει στο παρελθόν και έχουν λύση το πρόβλημα ώστε να επιλύονται τα προβλήματα της καλλιέργειας και του μικροκλίματος πιο ευκολά και γρήγορα.



## Παράρτημα

### Τελικός κώδικας Arduino

```
/*-----( Import needed libraries )-----*/
#include <SPI.h> // Use the Serial Peripheral Interface
#include <Wire.h> //Βιβλιοθήκη που επιτρέπει να επικοινωνείτε με συσκευές
I2C Mega2560 20 (SDA), 21 (SCL)
#include <LiquidCrystal_I2C.h>//Library for LCD
#include <Ethernet.h>//Ethernet Shield Library
#include <DHT.h> //Βιβλιοθήκη για αισθητήρα Dht11
#include <HC_SR04.h> //librari Ultrasonic Sensor HC-SR04
#include <DS3231.h> //library for real time clock Rtc ds3231
//


---


/*-----(Set the Digital pin connected to the sensor )-----*/
const int MOS_AirRecyclingFan = 2; //Gate mosfet 1 attach to d5 pin
arduino for Air Recycling area greenhouse
const int MOS_CoolingFan = 3; //Pc fan for cooling area Gate mosfet
Irf540N 2 attach to d5 pin arduino
int MOS_RgbLeds = 5; // Gate Mosfet attach to Rgb Led strip

int RELAY_Pump1 = 30;//Relay of real Pump 1 for watering plant
int RELAY_HeatingLight = 31;//Relay for heating the area
int RELAY_ValveWater = 32; //Relay Electric valve for watering

int LED_Pump2 = 45; //led blue pump for filling a tank

#define SEN_Dht11 38 // Digital pin connected to the DHT sensor

#define SEN_UltraSonicTrigPin 48 //attach pin D49 Arduino to pin Trig of
HC-SR04
#define SEN_UltraSonicEchoPin 49 // attach pin D48 Arduino to pin Echo of
HC-SR04

int SEN_LdrOut = A9; // Analog pin A9 connected to the LDR Sensor
//


---


/*-----( Declare Variables )-----*/
// LCD address and geometry and library initialization
const byte lcdAddr = 0x27; // Address of I2C backpack
const byte lcdCols = 16; // Number of character in a row
const byte lcdRows = 2; // Number of lines
LiquidCrystal_I2C lcd(lcdAddr, lcdCols, lcdRows); // set the LCD address
to 0x27 for a 16 chars and 2 line display

/*-----( Declare Variables )-----*/
// define the Xbee sensor variables for frame
unsigned int analogMSB1 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB1 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_SoilHygro = 0; //variable for the soil hygrometer data

unsigned int analogMSB2 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB2 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_InLdr = 0; // variable for the LDR in data

unsigned int analogMSB3 = 0; // πιο σημαντικό ψηφίο bit
unsigned int analogLSB3 = 0; // λιγότερο σημαντικό ψηφίο Bit
unsigned int Val_Mq07Gas = 0; // variable for the Mq-07 data

float Val_temp = 0; // Temperature variables
float Val_humi = 0; // Humidity variables
```

```

int Val_OutLdr = 0; // variable for the out LDR data

int Val_distance=0; // variable for the distance measurement
int Val_distancePerCent=0; // variable for the distance per cent
measurement

//Value for swithc buton in web site
int Mode =0;
boolean Val_Set_Mode = false;
boolean Val_Fans = false;
boolean Val_Light = false;
boolean Val_Watering= false;
boolean Val_Alert= false;
//-----Set the preference Constants

int My_Val_Max_temp=30; // Μέγιστη τιμή θερμοκρασίας σε βαθμούς Celsius
int My_Val_Min_temp=27; // Ελάχιστη τιμή θερμοκρασίας σε βαθμούς Celsius

int My_Val_Max_humi=90; // μέγιστη τιμή υγρασίας %
int My_Val_Min_humi=65; // ελάχιστη τιμή υγρασίας %

int My_Val_Max_distancePerCent=100; // μέγιστη τιμή νερού στο δοχείο
int My_Val_refill_distancePerCent=70;
int My_Val_Min_distancePerCent=60; // ελάχιστη τιμή νερού στο δοχείο

int My_Val_Max_OutLdr=90; // μέγιστη τιμή εξωτερικής φωτεινότητας
int My_Val_Min_OutLdr=50; // ελάχιστη τιμή εξωτερικής φωτεινότητας

int My_Val_InLdr=60; //Give the desired light value

int My_Val_Max_Mq07Gas=70; // μέγιστη τιμή αισθητήρα αερίου %

//he have WATER Give the desired soil moisture value
int My_Val_Max_SoilHygro=80;
// don't have WATER Give the desired soil moisture value
int My_Val_Min_SoilHygro=30;

// Rtc ds3231
const int My_Val_OnHour = 6; //SET Hour TO ON the rgb led
const int My_Val_OnMin = 01; //SET min TO ON the rgb led
const int My_Val_OffHour = 21; //SET TIME TO OFF the rgb led
const int My_Val_OffMin = 01; //SET min TO OFF the rgb led

byte thermometer[8] = { //icon for thermometer
    B00100,
    B01010,
    B01010,
    B01110,
    B01110,
    B11111,
    B11111,
    B01110
};
byte drop[8] = { //icon for water droplet
    B00100,
    B00100,
    B01010,
    B01010,
    B10001,
    B10001,
    B10001,
    B01110,

```

```

};

/*-----( Declare objects )-----*/
// selected sensor type the DHT 11
#define DHTTYPE DHT11
DHT dht(SEN_Dht11, DHT11);
//initialization class HCSR04 (trig pin , echo pin)
HCSR04 hc(SEN_UltraSonicTrigPin, SEN_UltraSonicEchoPin);

//RTC
DS3231 rtc(SDA, SCL);
Time t;

//Ορισμός της διεύθυνσης MAC του Arduino Ethernet shield
//Set MAC - IP - Server for ETHERNET SETTINGS
// Must be unique on local network
byte mac[] = { 0xD4, 0x28, 0xB2, 0xFF, 0xA0, 0xA1 };
//Set IP - Server for ETHERNET SETTINGS
//Enter the IP of ethernet shield Must be unique on local network
byte ip[] = {192, 168, 1, 100 };
//Εισαγωγή της διεύθυνσης IPv4 address του υπολογιστεί που είναι ο Server
//Enter the IPv4 pc address
byte serv[] = {192, 168, 1, 12} ;

//run cmd-> ipconfig Default Gateway = 192.168.1.1
byte gateway[] = { 192, 168, 1, 1};
byte subnet[] = { 255, 255, 255, 0};

// Initialize Arduino Ethernet Client
EthernetClient client;
EthernetClient cliente; //subnet mask
EthernetServer server(80); //server port
String HTTP_req, webPage;

/*-----( Declare Variables )-----*/
// ThingSpeak Settings
//διακομιστής ThingSpeak
char thingSpeakAddress[] = "api.thingspeak.com";
//Write the Write api key
String writeAPIKey = "NFNEYSXDOMUBW6MG";
// Time interval in milliseconds to update ThingSpeak
//(number of seconds * 1000 = interval)
const int updateThingSpeakInterval = 16 * 1000;

// Variable Setup
long lastConnectionTime = 0;
boolean lastConnected = false;
int failedCounter = 0;

void setup(){
  //Serial Communication is starting with 9600 of baud rate speed
  Serial.begin(9600); //start serial to 9600 baud
  dht.begin();// Start the DHT sensor
  rtc.begin();// Start Rtc dc331 on Arduino
  Wire.begin();//Start the I2C protocol

  //for lcd
  lcd.begin(16, 2);//Να ξεκινήσει η οθόνη για 16 στήλες και 2 γραμμές
  lcd.init();// initialize the lcd
  lcd.backlight();// Turns the backlight on
  lcd.createChar(1, thermometer); // Δημιουργία χαρακτήρα θερμομέτρου
  lcd.createChar(2, drop);// Δημιουργία χαρακτήρα σταγόνας

```

```

lcd.clear(); //Καθαρισμός οθόνης

// print some text in Serial Monitor
Serial.println(F("DHT11 test!"));
Serial.println(F("Ultrasonic Sensor HC-SR04 Test")); // print some text
in Serial Monitor
Serial.println(F("with Arduino UNO R3"));

// print the Ip to Serial monitor
// Serial.print(F("server is at "));
//start the Ethernet connection and the server:
Serial.print("ETHERNET Server SHIELD Started...\nLocal IP: ");

// Serial.println(Ethernet.localIP());

Ethernet.begin(mac, ip); // initialize Ethernet device

// Start Ethernet connection on Arduino for ThingSpeak
startEthernet();

//Set the INPUT pin mode
pinMode(SEN_LdrOut, INPUT);
pinMode(SEN_Dht11, INPUT);
// Sets the echoPin as an INPUT
pinMode(SEN_UltraSonicEchoPin, INPUT);

//Set the Outout pin mode
// Sets the trigPin as an OUTPUT
pinMode(SEN_UltraSonicTrigPin, OUTPUT);
pinMode(MOS_AirRecyclingFan, OUTPUT);
pinMode(MOS_CoolingFan, OUTPUT);
pinMode(MOS_RgbLeds, OUTPUT);
pinMode(RELAY_ValveWater, OUTPUT);
pinMode(RELAY_Pump1, OUTPUT);
pinMode(LED_Pump2, OUTPUT);
pinMode(RELAY_HeatingLight, OUTPUT);

digitalWrite(MOS_AirRecyclingFan, LOW); // set the pin to LOW
digitalWrite(MOS_CoolingFan, LOW); // set the pin to LOW
digitalWrite(MOS_RgbLeds, LOW); // set the pin to LOW

setRem_LED_RED_Org_ALERT_DIO4(0x4); //set the Xbee pin D4 to LOW
setRem_LED_Yellow_MOV_Sunshade_DIO11(0x4); //set the Xbee pin to LOW
setRem_LED_Blue_Blue_ForggerSpraySytem_DIO12(0x4); //Set the to LOW

// For switch buton in web site
digitalWrite(Mode, Val_Set_Mode);

digitalWrite(MOS_AirRecyclingFan, Val_Fans);
digitalWrite(MOS_CoolingFan, Val_Fans);

digitalWrite(RELAY_ValveWater, Val_Watering);
digitalWrite(RELAY_Pump1, Val_Watering);
digitalWrite(LED_Pump2, Val_Watering);

digitalWrite(RELAY_HeatingLight, Val_Light);
digitalWrite(MOS_RgbLeds, Val_Light);

digitalWrite(setRem_LED_RED_Org_ALERT_DIO4, Val_Alert);
digitalWrite(setRem_LED_Yellow_MOV_Sunshade_DIO11, Val_Alert);
digitalWrite(setRem_LED_Blue_Blue_ForggerSpraySytem_DIO12, Val_Alert);

```

```

}

void loop(){
  Read_RemoteXbee_Loop();
  Read_Rtc_Loop();
  Read_Sen_Dht11_Loop();
  Read_Sen_UltraSonic_Loop();
  Read_Sen_OutLDR_Loop();

  Print_Serial_loop(); // print the data sensor to serial monitor
  Print_Lcd_loop();
  Print_DataBase_loop();
  Print_ThingSpeak_loop();

  control_NETarea_loop();
}

//---Read the sensor value of Router Xbee-----
void Read_RemoteXbee_Loop() {
  // έλεγχος εάν το frame έχει τουλάχιστον 29 bits
  if (Serial.available() >= 29) {
    // έλεγχος εάν έχει start byte 7E
    if (Serial.read() == 0x7E) {
      // Απόρριψη των πρώτων 21 byte
      for (int i = 1; i < 21; i++) {
        byte discardbyte = Serial.read();
      }
      //τα Πρώτα byte είναι η τιμή του Soil hygrometer
      // Read the most significant bit of the analog data
      analogMSB1 = Serial.read(); // Read the first analog byte
data
      analogLSB1 = Serial.read(); // Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
      Val_SoilHygro = analogLSB1 + (analogMSB1 * 256);
      //Αντιστοιχίσει των Byte σε επι τοις εκατό
      Val_SoilHygro = map(Val_SoilHygro, 600, 1000, 100, 0);

      //τα Επόμενα byte είναι η τιμή του LDR
      analogMSB2 = Serial.read();// Read the first analog byte data
      analogLSB2 = Serial.read();// Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
      Val_InLdr = analogLSB2 + (analogMSB2 * 256);
      //Αντιστοιχίσει των Byte σε επι τοις εκατό
      Val_InLdr = map(Val_InLdr, 0, 1023, 0, 100);
      //τα τελευταία byte είναι η τιμή το Gas Sensor Mq07

      analogMSB3 = Serial.read();// Read the first analog byte data
      analogLSB3 = Serial.read();// Read the second byte
//μαθηματική πράξη για την δημιουργία από 2 bit LSB και MSB σε ένα
//byte
      Val_Mq07Gas = analogLSB3 + (analogMSB3 * 256);
      //Αντιστοιχίσει των Byte σε επι τοις εκατό
      Val_Mq07Gas = map(Val_Mq07Gas, 0, 1023, 0, 100);
    }
  }
}

void control_NETarea_loop(){
  // Create a client connection
  EthernetClient client = server.available();

```

```

if(client){
  Serial.println(F("new client"));
  boolean currentLineIsBlank = true;
  String postText=""; //for switch
  // an http request ends with a blank line
  while (client.connected()){
    if(client.available()){
      char c = client.read();
      //-----
-
      //read char by char HTTP request
      if (postText.length() < 10) {
        //store characters to string
        postText += c;
        Serial.print(c);
      }
      //-----
      HTTP_req += c;
      if(c == '\n' && currentLineIsBlank){
        if(HTTP_req.indexOf("read") > -1){ //AJAX request for DHT11
values
          AJAX_request(client);
        }else{
          HTML_webpage();
          client.println(webPage);
        }

        Serial.print(HTTP_req);
        HTTP_req = ""; //reset HTTP request string
        break;
      }

      if(c == '\n') currentLineIsBlank = true;
      else if(c != '\r') currentLineIsBlank = false;
    }
  }
  //controls the Arduino if you press the buttons
  //-----sw1--auto/manual-----
  if (postText.indexOf("?on1") >0){ //auto
    digitalWrite(Mode, LOW);
    Val_Set_Mode= true;
  }
  if (postText.indexOf("?off1") >0){ //manual
    digitalWrite(Mode, HIGH);
    Val_Set_Mode= false;
  }
  //-----sw2---Fans-----
  if (postText.indexOf("?on2") >0){
    digitalWrite(MOS_CoolingFan, LOW);
    digitalWrite(MOS_AirRecyclingFan, LOW);
    Val_Fans= true;
  }
  if (postText.indexOf("?off2") >0){
    digitalWrite(MOS_CoolingFan, HIGH);
    digitalWrite(MOS_AirRecyclingFan, HIGH);
    Val_Fans= false;
  }
  //-----sw3-Watering plants valve and pump 1

```

```

if (postText.indexOf("?on3") >0){
    digitalWrite(RELAY_Pump1, LOW);
    digitalWrite(RELAY_ValveWater, LOW);
    digitalWrite(LED_Pump2, LOW);
    Val_Watering= true;
}
if (postText.indexOf("?off3") >0){
    digitalWrite(RELAY_Pump1, HIGH);
    digitalWrite(RELAY_ValveWater, HIGH);
    digitalWrite(LED_Pump2, HIGH);
    Val_Watering= false;
}
//-----sw4--Light-----
if (postText.indexOf("?on4") >0){
    digitalWrite(MOS_RgbLeds, LOW);
    digitalWrite(RELAY_HeatingLight, LOW);
    Val_Light= true;
}
if (postText.indexOf("?off4") >0){
    digitalWrite(MOS_RgbLeds, HIGH);
    digitalWrite(RELAY_HeatingLight, HIGH);
    Val_Light= false;
}
//-----sw5--Alarm-----
if (postText.indexOf("?on5") >0){
    setRem_LED_RED_Org_ALERT_DIO4(0x4);
    setRem_LED_Yellow_MOV_Sunshade_DIO11(0x4);
    setRem_LED_Blue_Blue_ForggerSpraySystem_DIO12(0x4);
    Val_Alert= true;
}
if (postText.indexOf("?off5") >0){
    setRem_LED_RED_Org_ALERT_DIO4(0x5);
    setRem_LED_Yellow_MOV_Sunshade_DIO11(0x4);
    setRem_LED_Blue_Blue_ForggerSpraySystem_DIO12(0x4);
    Val_Alert= false;
}
//*****
if (Val_Set_Mode== false){ //manual
    System_Temperature_Loop();
    System_Water_Loop();
    System_SunHade_Loop();
    System_AirRecycling_Loop();
    System_Alert_Loop();
    System_ForggerSpray_Loop();
    System_Light_Loop();
}else{ //auto
    style_controll_loop(client);
}

style_controllMode(client);
delay(10);
client.stop(); //sever client connection with server
}

//*****Start***Switch

void style_controllMode(EthernetClient cl){
    Serial.print("Auto");
    if (Val_Set_Mode){

```



```

        cl.println("<a class=""status2"" href=\"./?off1\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/off1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }else{
        cl.println("<a class=""status2"" href=\"./?on1\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/on1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }
}
void style_controll_loop(EthernetClient cl){
    Serial.print("Manual");
    if (Val_Fans){
        cl.println("<a class=""status3"" href=\"./?off2\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/off1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }else{
        cl.println("<a class=""status3"" href=\"./?on2\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/on1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }
    if (Val_Watering){
        cl.println("<a class=""status4"" href=\"./?off3\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/off1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }else{
        cl.println("<a class=""status4"" href=\"./?on3\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/on1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }
    if (Val_Light){
        cl.println("<a class=""status5"" href=\"./?off4\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/off1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }else{
        cl.println("<a class=""status5"" href=\"./?on4\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/on1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }
    if (Val_Alert){
        cl.println("<a class=""status6"" href=\"./?off5\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/off1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }else{
        cl.println("<a class=""status6"" href=\"./?on5\"><img
src=""http://192.168.1.12/db_allSensor/stylepage/Photos/on1.jpg""
alt=""off"" width=""90em"" height=""70em""</a>");
    }
}

void AJAX_request(EthernetClient client){
    client.print("Online");
}

void HTML_webpage(){
    //HTTP response header
    webPage = "HTTP/1.1 200 OK\n\rContent-Type:
text/html\n\r\n\r";//versions protocol->HTTP/1.1 or HTTP/2
    //-----
    //webpage - HTML + JavaScript with AJAX calls
    webPage += R"***(
<!DOCTYPE html>
<html lang="el">

```

```

<head>
  <TITLE>Control area</TITLE>
  <link rel='stylesheet' type='text/css'
href='http://192.168.1.12/db_allSensor/stylepage/StyleArduino.css' />
</head>
<html>
<body>
  <H1>Control area</H1>
  <div class=""menu-bar"">
    <ul>
      <li class="active"><a
href='https://192.168.1.12/db_allsensor/index.html'><i class="fa-solid
fa-house-blank"></i>HOME</a></li>
    </UL>
  </DIV>
  <div class="table-box">
    <div class="table-head">
      <div class="table-cell first-cell">
        <p>Devices</p>
      </div>
      <div class="table-cell last-cell">
        <p>Switch</p>
      </div>
    </div>
    <div class="table-row">
      <span style="color:rgb(196, 3, 3)" id="IdVals">N/A</span>
    </div>
    <div class="table-row">
      <p>Auto </p>
    </div>

    <div class="table-row">
      <p>Cooling Fan</p>
    </div>

    <div class="table-row">
      <p>Watering plants</p>
    </div>

    <div class="table-row">
      <p>Light</p>
    </div>

    <div class="table-row">
      <p>Alarm</p>
    </div>

  </div>
<!-------JavaScript----->
<script>
  setInterval(
    function(){
      getIdVals();
    }, 2000
  );

  function getIdVals(){
    var Request = new XMLHttpRequest();
    Request.onreadystatechange = function(){
      if(this.readyState == 4 && this.status == 200){
        document.getElementById("IdVals").innerHTML =
this.responseText;

```

```

    }
    };
    Request.open("GET", "read", true);
    Request.send();
}
</script>

</body>
</html>
)***";
}
////*****End***Switch
// Displays the value on the Serial Monitor
void Print_Lcd_loop() {
    // start to print at the first row
    lcd.setCursor(0, 0); lcd.write(1); lcd.print(Val_temp);
    lcd.print((char)223); lcd.print("C ");
    lcd.setCursor(9, 0); lcd.write(2); lcd.print(Val_humi); lcd.print("%");
    lcd.setCursor(1, 1); lcd.print("Hour:"); lcd.print(t.hour);
    lcd.print(":"); lcd.print(t.min); lcd.print(":"); lcd.println(t.sec);
}
//εμφάνιση τιμών στην σειριακή οθόνη
void Print_Serial_loop() {
    Serial.println(F("*****"));
    //Εμφάνιση θερμοκρασίας στην σειριακή οθόνη
    Serial.print(F("Temperature: "));
    Serial.print(Val_temp); Serial.println(F("°C"));
    // Εμφάνιση υγρασίας στην σειριακή οθόνη
    Serial.print(F("Humidity: "));
    Serial.print(Val_humi); Serial.println(F("%"));

    Serial.print(F("LDR OUT Value: "));
    Serial.print(Val_OutLdr); Serial.println(F("%"));

    Serial.print(F("Distance(%): "));
    Serial.print(Val_distancePerCent); Serial.println(F("%"));

    Serial.print(F("Distance(cm): "));
    Serial.print(Val_distance); Serial.println(F("cm"));

    Serial.print(F("Soil moisture: "));
    Serial.print(Val_SoilHygro); Serial.println(F("%"));

    Serial.print(F("Ldr IN Value: "));
    Serial.print(Val_InLdr); Serial.println(F("%"));

    Serial.print(F("Gas Value: "));
    Serial.print(Val_Mq07Gas); Serial.println(F("%"));

    Serial.print("Date : ");
    Serial.print(t.date); Serial.print("/"); Serial.print(t.mon);
    Serial.print("/"); Serial.print(t.year); Serial.print(" ");
    Serial.print("Hour : ");
    Serial.print(t.hour); Serial.print(":"); Serial.print(t.min);
    Serial.print(":"); Serial.println(t.sec);
}
void Print_DataBase_loop(){
    if (client.connect(serv, 80)) { //Connecting at the IP address and port
we saved before
        Serial.println("connected");
        client.print("GET /db_allSensor/data.php?"); //Connecting and Sending
values to database

```

```

//send the value to data.php
client.print("temperature="); client.print(Val_temp);
client.print("&humidity="); client.print(Val_humi);
client.print("&outLdr="); client.print(Val_OutLdr);
client.print("&tank="); client.print(Val_distancePerCent);
client.print("&soilhygro="); client.print(Val_SoilHygro);
client.print("&inLdr="); client.print(Val_InLdr);
client.print("&Mq07Gas="); client.print(Val_Mq07Gas);

client.println(" HTTP/1.1"); // Part of the GET request
client.println("Host: 192.168.1.12");
client.println(); // Empty line
client.println(); // Empty line
client.stop(); // Closing connection to server
// delay(900);
}else {
// if you didn't get a connection to the server:
Serial.println("connection failed");
}
}
void Print_ThingSpeak_loop(){
// μετατροπή της μεταβλητής Float σε String
String temp = String (Val_temp);
String hum = String (Val_humi);
String SoilHygro = String (Val_SoilHygro);
String InLdr = String (Val_InLdr);
String OutLdr = String (Val_OutLdr);
String Mq07Gas = String (Val_Mq07Gas);
String distance = String (Val_distancePerCent);

// Print Update Response to Serial Monitor
if (client.available()){
char c = client.read();
Serial.print(c);
}

// Disconnect from ThingSpeak
if (!client.connected() && lastConnected){
Serial.println("...disconnected");
Serial.println();
client.stop();
}

//Ενημέρωση τιμών
//chart sensor
if(!client.connected() && (millis() -
lastConnectionTime>updateThingSpeakInterval)){
updateThingSpeak("field1="+temp + "&field2="+hum +
"&field3="+SoilHygro + "&field4="+InLdr +
"&field5="+OutLdr+"&field6="+distance + "&field7="+ Mq07Gas);
}

// Check if Arduino Ethernet needs to be restarted
if (failedCounter > 3 ) {
startEthernet();
}

lastConnected = client.connected();
}

```

```

//---Read the value of Real Time Clock-----
void Read_Rtc_Loop() {
    t = rtc.getTime();
}
//---Read data Dht11 Data sensor
void Read_Sen_Dht11_Loop() {
    // read the humidity
    Val_humi = dht.readHumidity();
    // Read temperature as Celsius (the default)
    Val_temp = dht.readTemperature();
}
//---Read the distance at tank-----
void Read_Sen_UltraSonic_Loop(){
    Val_distance = hc.dist(); // Read Sensor ultrasonic data
    //Κάνουμε αντιστοίχιση των τιμών του αισθητήρα UltraSonic σε
    //εκατοστά σε εκατό τις εκατό για την καλύτεροί απεικόνιση της
    //στάθμης νερού στο δοχείο. Το δοχείο είναι 17 εκατοστά αρά
    //τότε θα έχει 0% νερό ενώ εάν είναι 0 εκ. τότε έχουμε 100%
    //νερό μέσα στο δοχείο.
    Val_distancePerCent = map(Val_distance, 17, 0, 0, 99);
}
//---Read the value of ldr out of greenhouse
void Read_Sen_OutLDR_Loop() {
    Val_OutLdr = analogRead(SEN_LdrOut); // read the analog data 0-1023 byte
    //Αντιστοιχίσει των Bytes σε επί τοις εκατό
    Val_OutLdr = map(Val_OutLdr, 0, 1023, 0, 100);
}
// πλαίσιο για των έλεγχο του μπλε LED που εξομοιώνει το σύστημα
//ψεκασμού ομίχλης νερού
void setRem_LED_Blue_Blue_ForggerSpraySytem_DIO12(char value) {
    Serial.write(0x7E);
    Serial.write(byte(0x0));
    Serial.write(0x10);
    Serial.write(0x17);
    Serial.write(byte(0x0));

    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(byte(0x0));
    Serial.write(0xFF);
    Serial.write(0xFF);
    Serial.write(0xFF);
    Serial.write(0xFE);

    Serial.write(0x02);
    Serial.write('P');
    Serial.write('2');
    Serial.write(value);
    long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '2' + value;
    Serial.write(0xFF - (sum & 0xFF));
}
//=====
//πλαίσιο για των ελέγχου του κίτρινου LED που εξομοιώνει το σύστημα
//σκίασης
void setRem_LED_Yellow_MOV_Sunshade_DIO11(char value) {
    Serial.write(0x7E); //start byte
    Serial.write(byte(0x0)); // high part of length (always zero)
    //low part of length (the number of bytes that follow, not including
    //check

```

```

Serial.write(0x10);
Serial.write(0x17); //0x17 is a remote AT command
Serial.write(byte(0x0)); //frame id set to zero for no reply
//Id of recipient, or use 0xFFFF for broadcast
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(byte(0x0));
Serial.write(0xFF); // 0xFF for broadcast
Serial.write(0xFF); // 0xFF for broadcast

//16 bit of recipient or 0xFFFE
Serial.write(0xFF);
Serial.write(0xFE);

Serial.write(0x02); // 0x02 to apply changes immediately on remote

//AT command name in ASCII character
Serial.write('P');
Serial.write('1');

//command data in as many bytes as needed
Serial.write(value);

//checksum is all bytes after length bytes
long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'P' + '1' + value;
Serial.write(0xFF - (sum & 0xFF)); // calculate the proper checksum
}
//=====
// create the frame to send to router
// πλαίσιο για των ελέγχου του κόκκινου LED και του buzzer που
//εξομοιώνει το σύστημα ειδοποίησης
void setRem_LED_RED_Org_ALERT_DIO4(char value) {
Serial.write(0x7E); //0
Serial.write(byte(0x0)); //1
Serial.write(0x10); //2
Serial.write(0x17); // 3
Serial.write(byte(0x0)); //4

Serial.write(byte(0x0)); //5
Serial.write(byte(0x0)); //6
Serial.write(byte(0x0)); //7
Serial.write(byte(0x0)); //8
Serial.write(byte(0x0)); //9
Serial.write(byte(0x0)); //10
Serial.write(0xFF); //12
Serial.write(0xFF); // 13

Serial.write(0xFF); //14
Serial.write(0xFE); //15

Serial.write(0x02); //16
Serial.write('D'); //17
Serial.write('4'); //18
Serial.write(value); // 19
long sum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'D' + '4' + value;
Serial.write(0xFF - (sum & 0xFF));
}

```

```

void startEthernet() {
  client.stop();
  Serial.println("Connecting Arduino to network...");
  Serial.println();
  // Connect to network and obtain an IP address using DHCP
  if (Ethernet.begin(mac) == 0) {
    Serial.println("DHCP Failed, reset Arduino to try again");
    Serial.println();
  } else {
    Serial.println("Arduino connected to network using DHCP");
    Serial.println();
  }
}

void updateThingSpeak(String tsData) {
  if (client.connect(thingSpeakAddress, 80)) {
    //Χρήση Http post για την ενημέρωση των δεδομένων
    //To Start line αποτελείται από
    //(method post/the protocol/έκδοση) αντίστοιχα
    client.print("POST /update HTTP/1.1\n");

    //Τα header για αναγνώριση μηνύματος post Request από το ThingSpeak
    //thingspeak Host
    client.print("Host: api.thingspeak.com\n");
    //Header Connection: το οποίο θα κλίσει την σύνδεση όταν στείλουμε κα
λάβουμε το αίτημα μας
    client.print("Connection: close\n");
    //Header X-THINGSPEAKAPIKEY που είναι το ApiKey
    client.print("X-THINGSPEAKAPIKEY: "+writeAPIKey+"\n");
    //Connection type: το οποίο υποδεικνύει το είδος αποστολής
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    //connection length
    client.print("Content-Length: ");

    client.print(tsData.length());
    client.print("\n\n");
    client.print(tsData);
    lastConnectionTime = millis();
    if (client.connected()) {
      Serial.println("Connecting to ThingSpeak...");
      Serial.println();
      failedCounter = 0;
    } else {
      failedCounter++;
      Serial.println("Connection to ThingSpeak failed
("+String(failedCounter, DEC)+"");
      Serial.println();
    }
  } else {
    failedCounter++;
    Serial.println("Connection to ThingSpeak Failed
("+String(failedCounter, DEC)+"");
    Serial.println();
    lastConnectionTime = millis();
  }
}

//control the Humidity System
void System_AirRecycling_Loop() {
  //εαν η τιμή της υγρασίας είναι πάνω από 90% ή
  if (Val_humi >= My_Val_Max_humi ||
  //η ποιότητα του αερα είναι πάνω από 70% τότε
  Val_Mq07Gas > My_Val_Max_Mq07Gas ) {

```



```

//Ο δεύτερος ανεμιστήρας που είναι για των εξαερισμό δουλεύει στις
μέγιστες στροφές
digitalWrite(MOS_AirRecyclingFan, HIGH);
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("Air Recycle Fan=Full"));
//εάν η υγρασία είναι πάνω από 65% τότε
}else if (Val_humi >= My_Val_Min_humi) {
// ο ανεμιστήρας θα δουλεύει στις μισές στροφές
analogWrite(MOS_AirRecyclingFan, 127); //PWM
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("Air Recycle=Medium"));
//αλλιώς εάν είναι χαμηλότερα από 65% τότε
}else {
// Ο ανεμιστήρας του εξαερισμού είναι σβηστός
digitalWrite(MOS_AirRecyclingFan, LOW);
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("Air Recycle=Stop"));
}
}
//control the Alert System ->Buzzer Red led
void System_Alert_Loop() {
//εάν η τιμή του αισθητήρα αεριού είναι πάνω από 70 ή
//η το επίπεδο του νερό ξεπεράσει το 90% τότε
if ((Val_Mq07Gas >= My_Val_Max_Mq07Gas) || (Val_distancePerCent >
My_Val_Max_distancePerCent)){
//κάνε το value=0x5 ώστε το Red Led και Buzzer να είναι ON
setRem_LED_RED_Org_ALERT_DIO4(0x5);
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("Alert=On"));
}
else{
//το value=0x4 ώστε το Red Led και Buzzer να είναι Off
setRem_LED_RED_Org_ALERT_DIO4(0x4);
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("Alert=Off"));
}
}
//control ForggerSpraySystem at the fire extinguishing system and humidity
void System_ForggerSpray_Loop() {
//Εάν η τιμή του αισθητήρα αεριού ή υγρασία είναι μικρότερο από 65% τότε
if (Val_Mq07Gas > My_Val_Max_Mq07Gas || Val_humi < My_Val_Min_humi)
{//High Humidity ->ForggerSpray system OFF
//το value=0x5 ώστε το Blue led να είναι ON
setRem_LED_Blue_Blue_ForggerSpraySystem_DIO12(0x5); // 0x5 -> on
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("blueLed=on"));
}else {
//το value=0x5 ώστε το Blue led να είναι OFF
setRem_LED_Blue_Blue_ForggerSpraySystem_DIO12(0x4); //0x4 -> off
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("blueLed=off"));
}
}
}
//control the inside Light System
void System_Light_Loop(){
//εάν η τιμή του εσωτερικού LDR είναι μικρότερη από 40 και
//η τιμή εξωτερικού είναι μικρότερη από 90 και
// η ώρα είναι πάνω από 6 και
// τα λεπτά είναι πάνω από 0 και
// η ώρα είναι κάτω από 22 και
// τα λεπτά είναι κάτω από 0 τότε

```

```

    if (Val_InLdr<=My_Val_InLdr && Val_OutLdr < My_Val_Max_OutLdr && t.hour
    >= My_Val_OnHour && t.min >= My_Val_OnMin && t.hour <= My_Val_OffHour
    && t.min <= My_Val_OffMin) {
        // οι Λεντ ταινίες είναι ON
        digitalWrite(MOS_RgbLeds, HIGH);
        //εμφάνιση κατάστασης στην σειριακή οθόνη
        Serial.println("RGB=ON");
    }else { //εάν δεν ισχύει κάτι από αυτά τότε
        // οι Λεντ ταινίες είναι OFF
        digitalWrite(MOS_RgbLeds, LOW);
        //εμφάνιση κατάστασης στην σειριακή οθόνη
        Serial.println(F("RGB=OFF"));
    }
}
//CONTROL THE SUNLIGHT
void System_SunSHade_Loop() {
//εάν η τιμή του LDR είναι μεγαλύτερη από 90% τότε
    if (Val_OutLdr > My_Val_Max_OutLdr) {
//Κάνε το value=0x5 ώστε το yellow led να είναι ON
        setRem_LED_Yellow_MOV_Sunshade_DIO11(0x5);
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
        Serial.println(F("Yellow Led=ON"));
    }else { // αλλιώς θα είναι κάτω από 90% και τότε
//Κάνε το value=0x4 ώστε το yellow led να είναι Off
        setRem_LED_Yellow_MOV_Sunshade_DIO11(0x4);
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
        Serial.println(F("Yellow Led=OFF"));
    }
}
// Control Temperature
void System_Temperature_Loop() {
//εάν η θερμοκρασία είναι πάνω από My_Val_Max_temp=30 τότε
    if (Val_temp > My_Val_Max_temp) {
//ο ανεμιστήρας να δουλέψει στις μέγιστες στροφές και
        digitalWrite(MOS_CoolingFan, HIGH);
//η λάμπα θερμοκρασίας να είναι σβηστή
        digitalWrite(RELAY_HeatingLight, HIGH);
//εμφάνισε στην σειριακή οθόνη την κατάσταση
        Serial.println(F("Full rpm Cool Fan temp is over 30 and
Heating=OFF"));
    }
//εάν η θερμοκρασία είναι πάνω από My_Val_Min_temp=26 τότε
    else if (Val_temp >= My_Val_Min_temp) {
//ο ανεμιστήρας να δουλέψει στις μισές στροφές μέσω της PWM και
        analogWrite(MOS_CoolingFan, 127);
        digitalWrite(RELAY_HeatingLight, HIGH); // η λάμπα να είναι σβηστή
// εμφάνισε στην σειριακή οθόνη την κατάσταση
        Serial.println(F("Medium rpm Cool Fan temp is under 30 and
Heating=OFF"));

        // εάν η θερμοκρασία δεν είναι πάνω από 30 αλλά ούτε πάνω από 26 τότε
    }else {
        digitalWrite(MOS_CoolingFan, LOW); // ο ανεμιστήρας να είναι σβηστός
και
        digitalWrite(RELAY_HeatingLight, LOW); //η λάμπα αναμμένη

        // εμφάνισε στην σειριακή οθόνη την κατάσταση
        Serial.println(F("Zero rpm Cool Fan temp is under 26 and
Heating=ON"));
    }
}
}
//-----

```

```

// Control the water system
void System_Water_Loop(){
    // εάν το επίπεδο νερού είναι χαμηλότερη από 80 τότε
    if (Val_distancePerCent < 80 ) {
// να ανάψει η αντλία 2 δηλαδή το LED για να γεμίσει το δοχείο με νερό
digitalWrite(LED_Pump2,HIGH );
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println("PUMP2(LED Blue)=ON - The water level is under 90%(Low
Level)");
    }else{ // εάν το επίπεδο είναι υψηλότερο από 80 τότε
// η αντλία που εξομοιώνετε με το μπλε led θα είναι σβηστή
digitalWrite(LED_Pump2, LOW);
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println(F("PUMP2(LED Blue)=OFF - The tank is Full is over
90%"));
    }
// εάν το επίπεδο του νερού είναι υψηλότερο από 60 και η τιμή της
//εξωτερική φωτεινότητας είναι υψηλότερη από 50 δηλαδή είναι μέρα τότε
//έλεγε εάν χεριάζετε νερό το φυτό
if(Val_distancePerCent > My_Val_Min_distancePerCent && Val_OutLdr >
My_Val_Min_OutLdr){
    Serial.println("The tank have water at tank, | The out light is over
50(Mera)");
//εάν η τιμή του αισθητήρα soil hygrometer είναι μεγαλύτερη από min=30
τότε το
//φυτό χρειάζεται νερό και
    if (Val_SoilHygro < My_Val_Min_SoilHygro){
digitalWrite(RELAY_Pump1, LOW); // η αντλία είναι on και
digitalWrite(RELAY_ValveWater, LOW); // η βάννα είναι on
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println("PUMP-1=ON | The plant wants water");

    }else{ // εάν είναι μικρότερη του 9 τότε το φυτό έχει νερό και
digitalWrite(RELAY_Pump1, HIGH); // η αντλία είναι off και
digitalWrite(RELAY_ValveWater, HIGH); // η βάννα είναι OFF
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println("PUMP-1=OFF | The plant has water");

    }
// εάν το επίπεδο του νερού είναι χαμηλότερη από 60 και η τιμή της
//εξωτερική φωτεινότητας είναι χαμηλότερη από 50 δηλαδή είναι βραδύ
τότε
    }else{
digitalWrite(RELAY_Pump1, HIGH); // η αντλία είναι κλεισεί OFF και
digitalWrite(RELAY_ValveWater, HIGH); // η βάννα κλεισεί
//Εμφάνιση της κατάστασης στην σειριακή οθόνη
Serial.println("PUMP-1=OFF / Είναι νύχτα ή δεν υπάρχει νερό στο
δοχείο. ");
    }
}
}

```

## Βιβλιογραφία

- [ T. E. o. E. Britannica, «greenhouse,» 39 May 2019. [Ηλεκτρονικό]. Available:  
1 <https://www.britannica.com/topic/greenhouse>.  
]
- [ Matt, «How Greenhouses Work,» swgreenhouses.co.uk, 6 January 2021. [Ηλεκτρονικό].  
2 Available: <https://www.swgreenhouses.co.uk/blog/how-greenhouses-work.html>.  
]
- [ ludvigsvensson, «The Basics of Greenhouse Climate Control,» ludvigsvensson,  
3 [Ηλεκτρονικό]. Available: [https://www.ludvigsvensson.com/en/climate-](https://www.ludvigsvensson.com/en/climate-screens/services-and-solutions/usage-guides/climate-control-basics/)  
] [screens/services-and-solutions/usage-guides/climate-control-basics/](https://www.ludvigsvensson.com/en/climate-screens/services-and-solutions/usage-guides/climate-control-basics/).
- [ drygair, «Greenhouse Climate Control – How to Improve Plant Growth,» drygair,  
4 [Ηλεκτρονικό]. Available: <https://drygair.com/blog/greenhouse-climate-control/>.  
]
- [ N. Vercelletto, «The ideal greenhouse temperature and humidity settings for every  
5 season,» happysprout, 31 August 2022. [Ηλεκτρονικό]. Available:  
] <https://www.happysprout.com/outdoor-living/greenhouse-temperature-humidity/>.
- [ D. J. L. a. D. X. Hao, «Do plants need sleep?,» greenhousecanada, 20 October 2020.  
6 [Ηλεκτρονικό]. Available: <https://www.greenhousecanada.com/do-plants-need-sleep/>.  
]
- [ autogrow, «automated-greenhouse,» autogrow, [Ηλεκτρονικό]. Available:  
7 <https://autogrow.com/your-growing-environment/automated-greenhouse>.  
]
- [ helbanna, «4 MAIN FEATURES OF IOT APPLICATION IN GREENHOUSE,»  
8 helbanna, [Ηλεκτρονικό]. Available: [https://helbanna.com/4-main-features-of-iot-](https://helbanna.com/4-main-features-of-iot-application-in-greenhouse/)  
] [application-in-greenhouse/](https://helbanna.com/4-main-features-of-iot-application-in-greenhouse/).
- [ <https://www.oracle.com/internet-of-things/what-is-iot/>, «What is IoT?,» oracle,  
9 [Ηλεκτρονικό]. Available: <https://www.oracle.com/internet-of-things/what-is-iot/>.  
]

[ helbanna, «4 MAIN FEATURES OF IOT APPLICATION IN GREENHOUSE,»  
1 helbanna, [Ηλεκτρονικό]. Available: <https://helbanna.com/4-main-features-of-iot-0-application-in-greenhouse/>.

]

[ B. Blog, «4 Benefits of Smart Greenhouses and How to Get Started,» BehrTech Blog,  
1 [Ηλεκτρονικό]. Available: <https://behrtech.com/blog/4-benefits-of-smart-greenhouses-1-and-how-to-get-started/>.

]

[ Δ. Τσελίκας, «Μονάδα ελέγχου περιβάλλοντος θερμοκηπίου,» repository library teiwest,  
1 2 2022. [Ηλεκτρονικό]. Available:  
2 <http://repository.library.teiwest.gr/xmlui/handle/123456789/9989>.

]

[ ViDes, «Automated Greenhouse,» instructables, 30 Μαΐου 2016. [Ηλεκτρονικό].  
1 Available: <https://www.instructables.com/Automated-Greenhouse/>.

3

]

[ B. Παναγιώτης, «Smart Greenhouse,» create.arduino.cc, 2021. [Ηλεκτρονικό].  
1 Available: <https://create.arduino.cc/projecthub/drpanosv/smart-greenhouse-fbc739>.

4

]

[ sensaphone, «Intuitive Web-Based Monitoring... Through the Cloud,» sensaphone,  
1 [Ηλεκτρονικό]. Available: <https://www.sensaphone.com/products/sensaphone-sentinel-5-monitoring-system>.

]

[ K. K. κ. K. Βασίλης, «Εφαρμογές του Internet of Things στην «έξυπνη γεωργία»,»  
1 repository.library.teimes.gr, Ιανουάριος 2021. [Ηλεκτρονικό]. Available:  
6 <http://repository.library.teimes.gr/xmlui/bitstream/handle/123456789/8913/%ce%a0%ce%a4%ce%a5%ce%a7%ce%99%ce%91%ce%9a%ce%97%20%ce%95%ce%a1%ce%93%ce%91%ce%a3%ce%99%ce%91.pdf?sequence=1&isAllowed=y>.

[ kionetworks, «Network Communication Protocols,» kionetworks, [Ηλεκτρονικό].  
1 Available: [https://www.kionetworks.com/en-us/blog/data-center/network-](https://www.kionetworks.com/en-us/blog/data-center/network-communication-protocols)  
7 communication-protocols.

]

[ J. Hopkins, «Understanding I2C and SPI Communication Protocols,» totalphase, 1  
1 December 2021. [Ηλεκτρονικό]. Available:  
8 [https://www.totalphase.com/blog/2021/12/i2c-vs-spi-vs-uart-introduction-and-](https://www.totalphase.com/blog/2021/12/i2c-vs-spi-vs-uart-introduction-and-comparison-similarities-differences/)  
] comparison-similarities-differences/.

[ A. I. J. B. Wesley Chai, «Ethernet,» techtarget, [Ηλεκτρονικό]. Available:  
1 <https://www.techtarget.com/searchnetworking/definition/Ethernet>.

9

]

[ eveloper.mozilla.org, «An overview of HTTP,» eveloper.mozilla.org, [Ηλεκτρονικό].  
2 Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>.

0

]

[ geeksforgeeks, «Difference between GET and POST request in Vanilla JavaScript,»  
2 geeksforgeeks, 13 June 2022. [Ηλεκτρονικό]. Available:  
1 [https://www.geeksforgeeks.org/difference-between-get-and-post-request-in-vanilla-](https://www.geeksforgeeks.org/difference-between-get-and-post-request-in-vanilla-javascript/)  
] javascript/.

[ developer.mozilla.org, «GET,» developer.mozilla.org, [Ηλεκτρονικό]. Available:  
2 <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/GET>.

2

]

[ developer.mozilla.org, «POST,» developer.mozilla.org, [Ηλεκτρονικό]. Available:  
2 <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/POST>.

3

]

[ wikipedia, «IEEE 802.15.4,» wikipedia, [Ηλεκτρονικό]. Available:  
2 [https://en.wikipedia.org/wiki/IEEE\\_802.15.4](https://en.wikipedia.org/wiki/IEEE_802.15.4).

4

]

[ A. electronicshub, «electronicshub,» electronicshub, 17 November 2017. [Ηλεκτρονικό].

2 Available: <https://www.electronicshub.org/zigbee-technology-architecture-applications/>.

5

]

[ cytron.io, «Is XBee ZigBee?,» tutorial.cytron.io, [Ηλεκτρονικό]. Available:

2 <https://tutorial.cytron.io/2011/03/06/is-xbee-zigbee/>.

6

]

[ en.wikipedia.org, «Zigbee,» en.wikipedia.org, [Ηλεκτρονικό]. Available:

2 <https://en.wikipedia.org/wiki/Zigbee>.

7

]

[ development.libelium.com, «Node Parameters,» development.libelium.com, 2020.

2 [Ηλεκτρονικό]. Available: [https://development.libelium.com/zigbee-networking-](https://development.libelium.com/zigbee-networking-guide/node-parameters)

8 [guide/node-parameters](https://development.libelium.com/zigbee-networking-guide/node-parameters).

]

[ mouser.com, «XBee Module,» electronicwings, [Ηλεκτρονικό]. Available:

2 <https://www.electronicwings.com/sensors-modules/xbee-module>.

9

]

[ C. Hope, «Star topology,» computerhope, 03 12 2022. [Ηλεκτρονικό]. Available:

3 <https://www.computerhope.com/jargon/s/startopo.htm>.

0

]

[ javatpoint, «What is Tree Topology?,» javatpoint, [Ηλεκτρονικό]. Available:

3 <https://www.javatpoint.com/what-is-tree-topology>.

1

]



[ C. Hope, «Mesh topology,» computerhope, 13 11 2018. [Ηλεκτρονικό]. Available:  
3 <https://www.computerhope.com/jargon/m/mesh.htm>.

2

]

[ docs.arduino.cc, «Arduino Integrated Development Environment (IDE) v1,»  
3 docs.arduino.cc, [Ηλεκτρονικό]. Available: [https://docs.arduino.cc/software/ide-](https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics)  
3 v1/tutorials/arduino-ide-v1-basics.

]

[ Eletrogate, «Τι είναι το Arduino: Σε τι χρησιμεύει, πλεονεκτήματα και πώς να το  
3 χρησιμοποιήσετε,» blog.eletrogate., 14 Ιουνίου 2022. [Ηλεκτρονικό]. Available:  
4 <https://blog.eletrogate.com/o-que-e-arduino-para-que-serve-vantagens-e-como-utilizar/>.

]

[ w3schools, «w3schools,» w3schools, [Ηλεκτρονικό]. Available:  
3 [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp).

5

]

[ w3schools, «w3schools,» w3schools, [Ηλεκτρονικό]. Available:  
3 <https://www.w3schools.com/Css/>.

6

]

[ freepnglogos, «JavaScript PNG, Transparent JS Logo Free Download,» freepnglogos,  
3 [Ηλεκτρονικό]. Available: <https://www.freepnglogos.com/pics/javascript>.

7

]

[ php.net, «php.net,» php.net, [Ηλεκτρονικό]. Available:  
3 <https://www.php.net/manual/en/intro-what-is.php>.

8

]

[ S. Gupta, «What Is SQL & How Does It Work? A Guide to Structured Query Language,»  
3 springboard, 6 October 2021. [Ηλεκτρονικό]. Available:  
9 <https://www.springboard.com/blog/data-analytics/what-is-sql/>.

]

[ talend, «What is MySQL? Everything You Need to Know,» talend, [Ηλεκτρονικό].  
4 Available: <https://www.talend.com/resources/what-is-mysql/>.

0

]

[ digi, «XCTU,» digi, [Ηλεκτρονικό]. Available:  
4 <https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu>.

1

]

[ digi, «digi,» digi, [Ηλεκτρονικό]. Available:  
4 <https://www.digi.com/resources/documentation/digidocs/PDFs/90001458-13.pdf>.

2

]

[ P. Pedamkar, «What is Visual Studio Code?,» educba, [Ηλεκτρονικό]. Available:  
4 <https://www.educba.com/what-is-visual-studio-code/>.

3

]

[ techopedia, «Intellisense,» techopedia, [Ηλεκτρονικό]. Available:  
4 <https://www.techopedia.com/definition/24580/intellisense>.

4

]

[ A. K, «What is XAMPP? and How to Install XAMPP?,» devopsschool, 11 May 2021.  
4 [Ηλεκτρονικό]. Available: [https://www.devopsschool.com/blog/what-is-xampp-and-](https://www.devopsschool.com/blog/what-is-xampp-and-how-to-install-xampp/)  
5 [how-to-install-xampp/](https://www.devopsschool.com/blog/what-is-xampp-and-how-to-install-xampp/).

]

[ R. Giaquinto, «What is Apache and What Does it Do for Website Development?,»  
4 es.greengeeks, [Ηλεκτρονικό]. Available: <https://es.greengeeks.com/blog/what-is-6 apache/>.

]

[ kinsta, «What Is Apache Web Server? A Basic Look at What It Is and How It Works,»  
4 kinsta, 28 May 2021. [Ηλεκτρονικό]. Available: <https://kinsta.com/knowledgebase/what-7 is-apache/>.

]

[ developers.google, «Using Google Charts,» developers.google, 23 September 2019.  
4 [Ηλεκτρονικό]. Available: <https://developers.google.com/chart/interactive/docs.8>

8

]

[ tutorialspoint, «Google Charts - Overview,» tutorialspoint, [Ηλεκτρονικό]. Available:  
4 [https://www.tutorialspoint.com/googlecharts/googlecharts\\_overview.htm.9](https://www.tutorialspoint.com/googlecharts/googlecharts_overview.htm.9)

9

]

[ J. A. M. a. M. Finnegan, «What is IFTTT? How to use If This, Then That services,»  
5 computerworld, 25 September 2020. [Ηλεκτρονικό]. Available:  
0 <https://www.computerworld.com/article/3239304/what-is-ifttt-how-to-use-if-this-then-1 that-services.html.2>

2

[ A. Σ. Β. Ε. Γ. Φαραντάτος, «ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ Μελέτη του Διαδικτύου των  
5 Πραγμάτων: Αρχιτεκτονική, Εφαρμογές και Μελλοντικό όραμα,» okeanis.lib.puas.gr,  
1 Νοέμβριος 2016. [Ηλεκτρονικό]. Available:  
] <http://okeanis.lib.puas.gr/xmlui/handle/123456789/3207.2>

[ thingspeak, «Learn More About ThingSpeak,» thingspeak, [Ηλεκτρονικό]. Available:  
5 [https://thingspeak.com/pages/learn\\_more.2](https://thingspeak.com/pages/learn_more.2)

2

]

[ A. Bhatt, «Arduino : What is Arduino,» engineersgarage, [Ηλεκτρονικό]. Available:  
5 <https://www.engineersgarage.com/arduino-what-is-arduino/>.

3

]

[ arduino, «What is Arduino?,» arduino, [Ηλεκτρονικό]. Available:  
5 <https://www.arduino.cc/en/Guide/Introduction>.

4

]

[ Κ. Κ. κ. Κ. Βασίλης, «ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ Εφαρμογές του Internet of Things στην  
5 «έξυπνη γεωργία»,» repository.library.teimes.gr, Ιανουάριος 2021. [Ηλεκτρονικό].  
5 Available: <http://repository.library.teiwest.gr/xmlui/handle/123456789/8913>.

]

[ Arduino, «Arduino UNO R3,» Arduino, [Ηλεκτρονικό]. Available:  
5 <https://docs.arduino.cc/hardware/uno-rev3>.

6

]

[ P. Gupta, «Difference Between Arduino Due vs Mega,» educba, [Ηλεκτρονικό].  
5 Available: <https://www.educba.com/arduino-due-vs-mega/>.

7

]

[ arduino, «Arduino Mega 2560 Rev3,» arduino, [Ηλεκτρονικό]. Available:  
5 <https://docs.arduino.cc/hardware/mega-2560>.

8

]

[ arduino, «PWM,» arduino, [Ηλεκτρονικό]. Available:  
5 <https://www.arduino.cc/en/Tutorial/Foundations/PWM>.

9

]

[ arduinogetstarted, «arduino shield,» arduinogetstarted, [Ηλεκτρονικό]. Available:  
6 <https://arduinogetstarted.com/tutorials/arduino-shield>.

0

]

[ sites.google.com, «sites.google.com,» sites.google.com, [Ηλεκτρονικό]. Available:  
6 <https://sites.google.com/site/eisagogestadiktyaypologiston1/home>.

1

]

[ sites.google.com, «sites.google.com,» sites.google.com, [Ηλεκτρονικό]. Available:  
6 <https://sites.google.com/site/eisagogestadiktyaypologiston1/home/basika-stoicheia->

2 [ylopoieses-diktyon](https://sites.google.com/site/eisagogestadiktyaypologiston1/home/basika-stoicheia-ylopoieses-diktyon).

]

[ docs.arduino.cc, «docs.arduino.cc,» docs.arduino.cc, 31 Αυγούστου 2022.

6 [Ηλεκτρονικό]. Available: [https://docs.arduino.cc/retired/shields/arduino-ethernet-](https://docs.arduino.cc/retired/shields/arduino-ethernet-shield-without-poe-module)  
3 [shield-without-poe-module](https://docs.arduino.cc/retired/shields/arduino-ethernet-shield-without-poe-module).

]

[ Chris, «Controlling A Solenoid Valve With Arduino,» bc-robotics, 2 July 2015.

6 [Ηλεκτρονικό]. Available: [https://bc-robotics.com/tutorials/controlling-a-solenoid-](https://bc-robotics.com/tutorials/controlling-a-solenoid-valve-with-arduino/)  
4 [valve-with-arduino/](https://bc-robotics.com/tutorials/controlling-a-solenoid-valve-with-arduino/).

]

[ ledsmagazine, «What is an LED?,» ledsmagazine, 2 Σεπτεμβρίου 2014. [Ηλεκτρονικό].

6 Available: [https://www.ledsmagazine.com/leds-ssl-](https://www.ledsmagazine.com/leds-ssl-design/materials/article/16701292/what-is-an-led)  
5 [design/materials/article/16701292/what-is-an-led](https://www.ledsmagazine.com/leds-ssl-design/materials/article/16701292/what-is-an-led).

]

[ B. Check, «Switched On: MOSFET Selection Guide,» boulderres, 9 Φεβρουαρίου 2017.

6 [Ηλεκτρονικό]. Available: [https://www.boulderres.com/resource-library/switched-on-](https://www.boulderres.com/resource-library/switched-on-mosfet-selection-guide)  
6 [mosfet-selection-guide](https://www.boulderres.com/resource-library/switched-on-mosfet-selection-guide).

]

[ L. Prasad, «What is Relay? How it Works? Types, Applications, Testing,»

6 electronicsHub, 28 February 2022. [Ηλεκτρονικό]. Available:  
7 <https://www.electronicshub.org/what-is-relay-and-how-it-works/>.

]

[ elprocus, «DHT11 Sensor and Its Working,» elprocus, [Ηλεκτρονικό]. Available:

6 <https://www.elprocus.com/a-brief-on-dht11-sensor/>.

8

]

[ K. Louisville, «Discussion on Humidity,» weather.gov, [Ηλεκτρονικό]. Available:  
6 <https://www.weather.gov/lmk/humidity>.

9

]

[ A. K. Ravi, 24 April 2020. [Ηλεκτρονικό]. Available: [https://www.c-](https://www.sharpcorner.com/UploadFile/167ad2/how-to-use-ultrasonic-sensor-hc-sr04-in-arduino/)  
7 [sharpcorner.com/UploadFile/167ad2/how-to-use-ultrasonic-sensor-hc-sr04-in-arduino/](https://www.sharpcorner.com/UploadFile/167ad2/how-to-use-ultrasonic-sensor-hc-sr04-in-arduino/).

0

]

[ L. REESE, «The working principle, applications and limitations of ultrasonic sensors,»  
7 microcontrollertips, 6 August 2019. [Ηλεκτρονικό]. Available:

1 <https://www.microcontrollertips.com/principle-applications-limitations-ultrasonic->

] [sensors-faq/](https://www.microcontrollertips.com/principle-applications-limitations-ultrasonic-sensors-faq/).

[ Electrical4U, «Light Dependent Resistor (LDR) or Photoresistor (What is it?),»  
7 Electrical4U, 24 January 2021. [Ηλεκτρονικό]. Available:

2 <https://www.electrical4u.com/light-dependent-resistor-ldr-working-principle-of-ldr/>.

]

[ electronics-notes, «Light Dependent Resistor LDR: Photoresistor,» electronics-notes,  
7 [Ηλεκτρονικό]. Available: [https://www.electronic-](https://www.electronic-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php)

3 [notes.com/articles/electronic\\_components/resistors/light-dependent-resistor-ldr.php](https://www.electronic-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php).

]

[ waveshare, «MQ-7 Gas Sensor,» waveshare, [Ηλεκτρονικό]. Available:  
7 [https://www.waveshare.com/wiki/MQ-7\\_Gas\\_Sensor](https://www.waveshare.com/wiki/MQ-7_Gas_Sensor).

4

]

[ A. Mukherjee, «Smoke Detection using MQ-2 Gas Sensor,» reate.arduino.cc, 6 May  
7 2016. [Ηλεκτρονικό]. Available: <https://create.arduino.cc/projecthub/Aritro/smoke->

5 [detection-using-mq-2-gas-sensor-79c54a](https://create.arduino.cc/projecthub/Aritro/smoke-detection-using-mq-2-gas-sensor-79c54a).

]

[ lastminuteengineers, «How Soil Moisture Sensor Works and Interface it with Arduino,»  
7 lastminuteengineers, [Ηλεκτρονικό]. Available: <https://lastminuteengineers.com/soil-6 moisture-sensor-arduino-tutorial/>.

]

[ Dejan, «Arduino and DS3231 Real Time Clock Tutorial,» howtomechatronics,  
7 [Ηλεκτρονικό]. Available: <https://howtomechatronics.com/tutorials/arduino/arduino-7 ds3231-real-time-clock-tutorial/>.

]

[ lastminuteengineers, «Interface DS3231 Precision RTC Module with Arduino,»  
7 lastminuteengineers, [Ηλεκτρονικό]. Available:  
8 <https://lastminuteengineers.com/ds3231-rtc-arduino-tutorial/>.

]

[ lastminuteengineers, «Interfacing 16×2 Character LCD Module with Arduino,»  
7 lastminuteengineers, [Ηλεκτρονικό]. Available:  
9 <https://lastminuteengineers.com/arduino-1602-character-lcd-tutorial/>.

]

[ lastminuteengineers, «Interface an I2C LCD with Arduino,» lastminuteengineers,  
8 [Ηλεκτρονικό]. Available: <https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/>.

0

]

[ G. Lazaridis, «How PC Fans Work,» pcbheaven, 29 Μαρτίου 2010. [Ηλεκτρονικό].  
8 Available: [https://www.pcbheaven.com/wikipages/How\\_PC\\_Fans\\_Work/](https://www.pcbheaven.com/wikipages/How_PC_Fans_Work/).

1

]

[ G. Lazaridis, «How Brushless Motors Work (BLDC Motors),» pcbheaven, 27 March  
8 2010. [Ηλεκτρονικό]. Available:  
2 [http://pcbheaven.com/wikipages/How\\_Brushless\\_Motors\\_Work/](http://pcbheaven.com/wikipages/How_Brushless_Motors_Work/).

]



[ X. K. M. B. Θ. Γ. ΒΑΣΙΛΕΙΟΣ ΔΗΜΗΤΡΟΠΟΥΛΟΣ, «Ειδικές Ηλεκτρικές  
8 Εγκαταστάσεις Α' Τεύχος,» σε *Ειδικές Ηλεκτρικές Εγκαταστάσεις Α' Τεύχος*, Αθήνα,  
3 2011, p. 56.

]

[ mechstudies, «what is pump basics parts types,» mechstudies, [Ηλεκτρονικό]. Available:  
8 <https://www.mechstudies.com/what-is-pump-basics-parts-types/>.

4

]

[ Neha, «PC to PC Communication over Zigbee Protocol using Xbee and XCTU,»  
8 engineersgarage, [Ηλεκτρονικό]. Available: <https://www.engineersgarage.com/pc-to-pc-communication-over-zigbee-protocol-using-xbee-and-xctu-3/>.

5

[ Z. Dannelly, «AT vs API (What, Why, How),» os.mbed, 15 February 2015.  
8 [Ηλεκτρονικό]. Available: <https://os.mbed.com/users/dannellyz/notebook/at-vs-api-when-why-how/>.

6

[ sparkfun, «XBee Buying Guide,» sparkfun, [Ηλεκτρονικό]. Available:  
8 [https://www.sparkfun.com/pages/xbee\\_guide](https://www.sparkfun.com/pages/xbee_guide).

7

]

[ sparkfun, «sparkfun,» sparkfun, [Ηλεκτρονικό]. Available:  
8 <https://www.sparkfun.com/products/11812>.

8

]

[ agroselida, «Πότισμα φυτών – Πότε και πως ποτίζουμε τα φυτά στον κήπο μας,» 10  
8 Ιουλίου 2016. [Ηλεκτρονικό]. Available: [https://agroselida.blogspot.com/2016/07/blog-post\\_10.html](https://agroselida.blogspot.com/2016/07/blog-post_10.html).

9

[ thingspeak, «ThingHTTP App,» mathworks, [Ηλεκτρονικό]. Available:  
9 <https://www.mathworks.com/help/thingspeak/thinghttp-app.html>.

0

]

[ thingspeak, «React App,» mathworks, [Ηλεκτρονικό]. Available:  
9 <https://www.mathworks.com/help/thingspeak/react-app.html>.

1

]

[ M. Olivotto, «Focus on greenhouse construction and safe operation,» 24 4 2020.  
9 [Ηλεκτρονικό]. Available: <https://ausveg.com.au/articles/focus-on-greenhouse-2-construction-and-safe-operation/>.

]

[ iotconnect, «Avnet's Smart Greenhouse Monitoring Solution,» iotconnect,  
9 [Ηλεκτρονικό]. Available: <https://www.iotconnect.io/smart-greenhouse-solution.html>.

3

]

[ R. Keim, «Back to Basics: The Universal Asynchronous Receiver/Transmitter (UART),»  
9 20 12 2016. [Ηλεκτρονικό]. Available: <https://www.allaboutcircuits.com/technical-4-articles/back-to-basics-the-universal-asynchronous-receiver-transmitter-uart/>.

]

[ D. LTD, «DLN-Series Interface AdaptersC/C++ API Manual Ver.2,» σε *DLN-Series*  
9 *Interface AdaptersC/C++ API Manual Ver.2*, [https://docplayer.net/101717366-Dln-5-series-interface-adapters-c-c-api-manual-supported-products-dln-1-dln-2-dl-4m-dln-4s-](https://docplayer.net/101717366-Dln-5-series-interface-adapters-c-c-api-manual-supported-products-dln-1-dln-2-dl-4m-dln-4s-version-november-2015.html)  
] [version-november-2015.html](https://docplayer.net/101717366-Dln-5-series-interface-adapters-c-c-api-manual-supported-products-dln-1-dln-2-dl-4m-dln-4s-version-november-2015.html), 2015, p. 58.

[ S. Afzal, «I2C Primer: What is I2C? (Part 1),» 1 1 2022. [Ηλεκτρονικό]. Available:  
9 [https://www.analog.com/en/technical-articles/i2c-primer-what-is-i2c-part-](https://www.analog.com/en/technical-articles/i2c-primer-what-is-i2c-part-6-1.html#author)  
6 [1.html#author](https://www.analog.com/en/technical-articles/i2c-primer-what-is-i2c-part-6-1.html#author).

]

[ Bosch, «About Zigbee technology,» 1 1 2022. [Ηλεκτρονικό]. Available: <https://bosch-9-iot-suite.com/iot-devices-zigbee/>.

7

]

[ w3, «w3,» w3, [Ηλεκτρονικό]. Available: <https://www.w3.org/html/logo/index.html>.

9

8

]

[ w3schools, «w3schools,» w3schools, [Ηλεκτρονικό]. Available:

9 [https://www.w3schools.com/whatis/whatis\\_html.asp](https://www.w3schools.com/whatis/whatis_html.asp).

9

]

[ S. Abe, «brandlogos,» brandlogos, [Ηλεκτρονικό]. Available:

1 <https://brandlogos.net/css3-eps-5904.html>.

0

0

]

[ php.net, «php.net,» php.net, [Ηλεκτρονικό]. Available: [https://www.php.net/download-](https://www.php.net/download-logos.php)

1 [logos.php](https://www.php.net/download-logos.php).

0

1

]

[ wikipedia, «File:MySQL logo.svg,» wikipedia, 11 May 2021. [Ηλεκτρονικό]. Available:

1 [https://en.wikipedia.org/wiki/File:MySQL\\_logo.svg](https://en.wikipedia.org/wiki/File:MySQL_logo.svg).

0

2

]

[ code.visualstudio, «Icons and names usage guidelines,» code.visualstudio,

1 [Ηλεκτρονικό]. Available: <https://code.visualstudio.com/brand>.

0

3

]

[ wikimedia, «File:Xampp logo.gif,» wikimedia, [Ηλεκτρονικό]. Available:

1 [https://commons.wikimedia.org/wiki/File:Xampp\\_logo.gif](https://commons.wikimedia.org/wiki/File:Xampp_logo.gif).

0

4

]

[ W. Stallings, ΕΠΙΚΟΙΝΩΝΙΕΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΔΕΔΟΜΕΝΩΝ 10Η ΕΚΔΟΣΗ,  
1 ΘΕΣΣΑΛΟΝΙΚΗ: ΤΖΙΟΛΑ, 2020.

0

5

]

[ Yida, «UART vs I2C vs SPI – Communication Protocols and Uses,» N/A N/A 2019.  
1 [Ηλεκτρονικό]. Available: [https://www.seeedstudio.com/blog/2019/09/25/uart-vs-i2c-](https://www.seeedstudio.com/blog/2019/09/25/uart-vs-i2c-vs-spi-communication-protocols-and-uses/)  
0 [vs-spi-communication-protocols-and-uses/](https://www.seeedstudio.com/blog/2019/09/25/uart-vs-i2c-vs-spi-communication-protocols-and-uses/).

6

]

[ Ovidiu, «Arduino Tiny Web Server - part 3,» 3 September 2011. [Ηλεκτρονικό].  
1 Available: <https://webweavertech.com/ovidiu/weblog/archives/000484.html>.

0

7

]

[ Vinod, «Light Emitting Diode (LED) and Photodiode,» 5 12 2021. [Ηλεκτρονικό].  
1 Available: <https://sciencevision.in/light-emitting-diode-led-and-photodiode/>.

0

8

]

[ Administrator, «How to Connect PCF8574 I2C LCD with Arduino?,» electronicshub, 22  
1 December 2019. [Ηλεκτρονικό]. Available: [https://www.electronicshub.org/pcf8574-](https://www.electronicshub.org/pcf8574-i2c-lcd-with-arduino/)  
0 [i2c-lcd-with-arduino/](https://www.electronicshub.org/pcf8574-i2c-lcd-with-arduino/).

9

]

[ osoyoo, «Micro bit Lesson — Using the Ultrasonic Module,» osoyoo, 18 9 2018.  
1 [Ηλεκτρονικό]. Available: [https://osoyoo.com/2018/09/18/micro-bit-lesson-using-the-](https://osoyoo.com/2018/09/18/micro-bit-lesson-using-the-ultrasonic-module/)  
1 [ultrasonic-module/](https://osoyoo.com/2018/09/18/micro-bit-lesson-using-the-ultrasonic-module/).

0

]

[ A. A. Jabbaar, «Ultrasonic Sensor HC-SR04 with Arduino Tutorial,» create.arduino.cc,  
1 17 September 2019. [Ηλεκτρονικό]. Available:  
1 [https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-](https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6)  
1 [arduino-tutorial-327ff6](https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6).

]

[ Waveshare, «MQ-7 Gas Sensor,» N/A N/A N/A. [Ηλεκτρονικό]. Available:  
1 <https://www.waveshare.com/mq-7-gas-sensor.htm>.

1

2

]

[ Grobotronics, «Soil Hygrometer / Moisture Detection Module,» 0 0 0. [Ηλεκτρονικό].  
1 Available: <https://grobotronics.com/soil-humidity-sensor.html>.

1

3

]

[ Amazon.in, «Electrobot DS3231 Real Time Clock (RTC) Module for Arduino & Rpi,»  
1 [Ηλεκτρονικό]. Available: [https://www.amazon.in/Electrobot-DS3231-Clock-Module-](https://www.amazon.in/Electrobot-DS3231-Clock-Module-Arduino/dp/B07NRV51KG)  
1 [Arduino/dp/B07NRV51KG](https://www.amazon.in/Electrobot-DS3231-Clock-Module-Arduino/dp/B07NRV51KG).

4

]

[ Digi, «About Digi,» [Ηλεκτρονικό]. Available: <https://www.digi.com/about-digi>.

1

1

5

]

[ R. Faludi, «Getting Ready,» σε *Wireless Sensor Networks*, United States of America,  
1 o'reilly, 2011, p. 15.

1

6

]

[ alselectro, «XBee S2–Digital sampling with API frame data,» alselectro, September 7  
1 2014. [Ηλεκτρονικό]. Available: <https://alselectro.wordpress.com/tag/xbee-s2-api-frames/>.

7

]

[ SparkFun, «SparkFun XBee Explorer USB,» [Ηλεκτρονικό]. Available:  
1 <https://www.sparkfun.com/products/11812>.

1

8

]

[ developer.mozilla.org, «An overview of HTTP,» developer.mozilla.org, [Ηλεκτρονικό].  
1 Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>.

1

9

]

[ P. Gupta, «Arduino Due vs Mega,» educba, [Ηλεκτρονικό]. Available:  
1 <https://www.educba.com/arduino-due-vs-mega/>.

2

0

]

[ WorldData.info, «WorldData.info,» 21 7 2021. [Ηλεκτρονικό]. Available:  
1 <https://www.worlddata.info/europe/greece/sunset.php>.

2

1

]

[ C. McGrady, «When you hear the word “snubber”, what does it bring to mind?,» arrow,  
1 19 October 2016. [Ηλεκτρονικό]. Available: [https://www.arrow.com/en/research-and-  
2 events/articles/what-is-a-snubber](https://www.arrow.com/en/research-and-events/articles/what-is-a-snubber).

2

]

[ W. Electrical, «What are Digital Sensors : Types and Their Uses,» watelectrical, 30  
1 Ιουνίου 2020. [Ηλεκτρονικό]. Available: <https://www.watelectrical.com/what-are-2-digital-sensors-types-and-their-uses/>.

3

]