



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ  
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

"Σχεδίαση και υλοποίηση συστημάτων αυτοματισμού για θερμοκήπιο με χρήση της πλατφόρμας Arduino και δυνατότητα επίβλεψης αποτελεσμάτων μέσω ασύρματης δικτύωσης αλλά και μέσω διαδικτύου."

Σωτήριος Δριμής (Α.Μ: 7068)

ΕΠΙΒΛΕΠΩΝ:  
Χαδέλλης Λουκάς  
Τοπάλης Ευάγγελος

ΠΑΤΡΑ 2024

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, Σεπτέμβριος 2024

#### ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Χαδέλλης Λουκάς
2. Τοπάλης Ευάγγελος
3. Καψάλης Βασίλειος

#### 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 Σωτηρίου Δριμή 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

Η παρούσα διπλωματική πραγματεύεται την υλοποίηση ενός συστήματος αυτοματισμών με στόχο τον έλεγχο των κλιματικών συνθηκών στο εσωτερικό ενός θερμοκηπίου. Η υλοποίηση στοχεύει στην αξιοποίηση τεχνολογιών όπως οι μικροελεγκτές, το Internet of Things (IoT), ενσύρματα και ασύρματα πρωτόκολλα επικοινωνίας, οι βάσεις δεδομένων, καθώς και ο προγραμματισμός για δημιουργία ιστότοπου. Η λειτουργία του προτεινόμενου συστήματος διαχείρισης του θερμοκηπίου στηρίζεται στην πλατφόρμα Arduino η οποία μέσω αισθητήρων θα διαχειρίζεται τις διάφορες εργασίες. Οι αισθητήρες παρέχουν λειτουργικότητα όπως ανίχνευση κίνησης στον χώρο, πληροφορίες που αφορούν τις συνθήκες του περιβάλλοντος στο θερμοκήπιο όπως η θερμοκρασία και η υγρασία, πυροπροστασία, πληροφορίες για την ποιότητα του χώματος, μεταβλητή σκίαση κατά τη διάρκεια της μέρας, πότισμα και έλεγχο του κλίματος στο θερμοκήπιο και κάποιον απλό έλεγχο πρόσβασης σε αυτό. Το Arduino επικοινωνεί με τους αισθητήρες μέσω του πρωτοκόλλου SPI όταν ο αισθητήρας είναι ψηφιακός ή διαβάζει απευθείας την αναλογική τάση στην έξοδό τους όταν ο αισθητήρας είναι αναλογικός. Παράλληλα μία συσκευή XBee επικοινωνεί με το Arduino ώστε να προωθήσει τα δεδομένα που αυτό συλλέγει, μέσω του πρωτοκόλλου ZigBee, σε ένα άλλο XBee, το οποίο ανεβάζει πληροφορίες σε μία βάση δεδομένων από την οποία θα τα αντλεί η ιστοσελίδα προς ενημέρωση του χρήστη. Η ανάπτυξη του ιστότοπου έγινε με HTML/CSS για το στατικό μέρος της σελίδας, ενώ χρησιμοποιήθηκαν και τεχνολογίες από τις Javascript και AJAX ώστε να προστεθεί λειτουργικότητα στον ιστότοπο ώστε αυτός να γίνει πιο φιλικός και εύχρηστος στον χρήστη. Η βάση δεδομένων χρησιμοποιεί το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων MySQL, ενώ ο ιστότοπος επικοινωνεί με το σύστημα αυτό μέσω συναρτήσεων υλοποιημένων σε PHP. Για την οπτική αναπαράσταση των δεδομένων χρησιμοποιήθηκαν οι πλατφόρμες ThingSpeak και Google Charts. Το προτεινόμενο σύστημα θα ενσωματωθεί σε μία κατασκευή η οποία θα περιλαμβάνει όλα τα σενάρια αυτοματισμών που θα αναπτυχθούν ώστε αφενός να αποδειχθεί η δυνατότητα υλοποίησης και αφετέρου να υπάρχει ένα πρότυπο για περαιτέρω ανάπτυξη.

**Λέξεις κλειδιά:** Θερμοκήπιο, Αισθητήρες, C++, Arduino, Ethernet shield, Xbee.

## **Abstract**

The goal of this thesis is to develop and implement an automated system for regulating the climatic conditions within a greenhouse. The implementation intends to utilize technologies like microcontrollers, the Internet of Things (IoT), methods of communication both wired and wireless, databases, and programming of websites. The functionality of the proposed greenhouse management system relies on the Arduino platform, which would oversee several operations using sensors. The sensors offer a range of capabilities including motion detection, temperature and humidity information, fire protection, soil quality data, adaptive shading throughout the day, greenhouse watering and climate control, and basic access control. In the case of digital sensors, the Arduino establishes communication with the sensors via the SPI protocol. Conversely, with analog sensors, the Arduino directly measures the analog voltage at their output. Concurrently, an XBee device establishes communication with the Arduino in order to transmit the data it gathers using the ZigBee protocol to another XBee device, which then uploads it to a website for the purpose of informing the user. The website was developed using HTML/CSS for the static components of the sites, and also incorporated Javascript and AJAX technologies to enhance its functionality and maximise user-friendliness. While the website interfaces with the MySQL relational database management system via PHP-implemented methods, the database itself utilizes this technology. The data visualisation was achieved using the ThingSpeak and GoogleCharts platforms. The technology will be included into a construction project that will encompass all the automation scenarios that will be created to showcase the practicality of implementation and to establish a blueprint for advanced development.

**Keywords:** Greenhouse, Sensors, C++, Arduino, Ethernet shield, Xbee.



## **Ευχαριστίες**

Πρώτα από όλα, θα ήθελα να ευχαριστήσω τους εισηγητές της διπλωματικής μου εργασίας, κ. Ευάγγελο Τοπάλη και κ. Λουκά Χαδέλλη, για την πολύτιμη βοήθεια, την καθοδήγηση και το ενδιαφέρον που έδειξαν καθ' όλη την διάρκεια εκπόνησης της διπλωματικής μου εργασίας. Τέλος θα ήθελα να εκφράσω θερμές ευχαριστίες προς την οικογένεια μου, για την αγάπη και την υποστήριξη τους όλα αυτά τα χρόνια. Αφιερώνω αυτή την διπλωματική εργασία στην οικογένεια μου.

# Πίνακας περιεχομένων

Περίληψη .....	iii
Abstract .....	iv
Ευχαριστίες .....	v
Πίνακας περιεχομένων .....	vi
Ευρετήριο εικόνων .....	ix
Ευρετήριο πινάκων .....	xiii
Συντομογραφίες & Ακρωνύμια.....	xiv
1 Εισαγωγή.....	1
<b>1.1</b> Γενικά στοιχεία.....	1
<b>1.2</b> Έξυπνα Θερμοκήπια.....	1
1.3 Σκοπός και στόχοι.....	3
1.4 Οργάνωση Διπλωματικής.....	4
2 Τεχνολογίες και Εργαλεία.....	6
2.1 Εισαγωγή.....	6
2.2 Πρωτόκολλα επικοινωνίας.....	6
2.2.1 Serial Peripheral Interface (SPI).....	6
2.2.2 Ethernet.....	7
2.2.3 Hyper Text Transfer Protocol (HTTP).....	7
2.2.4 ZigBee.....	8
2.3 Γλώσσες προγραμματισμού.....	9
2.3.1 Arduino IDE.....	9
2.3.2 Γλώσσες προγραμματισμού διαδικτύου.....	12
2.4 Εργαλεία και πλατφόρμες.....	18
2.4.1 Εισαγωγή.....	18
<b>2.4.2 Notepad++</b> .....	19
2.4.3 XAMPP.....	19
2.4.4 Fritzing.....	20
2.4.5 Google Charts.....	21
2.4.6 ThingSpeak.....	21
3 Υλικό μέρος συστήματος.....	23
3.1 Arduino.....	23
3.1.1 Γενικές πληροφορίες.....	23
<b>3.1.2</b> Εκδόσεις Arduino.....	26
3.1.3 Arduino shields.....	27
3.2 Αισθητήρες και συσκευές δράσης.....	30
3.2.1 Πλακέτα ρελέ 2 καναλιών.....	30

3.2.2	Αισθητήρας κίνησης PIR.....	32
3.2.3	Αισθητήρας αερίου MQ-135 .....	34
3.2.4	Αισθητήρας θερμοκρασίας-υγρασίας DHT-11 .....	36
3.2.5	Αισθητήρας φλόγας KY-026.....	38
3.2.6	DS1302 Real time clock module .....	40
3.2.7	Πλακέτα φωτοαντίστασης KY-018 .....	43
3.2.8	Αισθητήρας θερμοκρασίας νερού DS18B20p.....	44
3.2.9	Αισθητήρας υγρασίας εδάφους Soil moisture detector .....	46
3.2.10	Αισθητήρας καθαρότητας νερού SEN0189 .....	48
3.2.11	Αισθητήρας κλίσης KY-017 .....	50
<b>3.2.12</b>	<b>Αισθητήρας στάθμης νερού .....</b>	<b>52</b>
3.2.13	Κιτ ασύρματου απομακρυσμένου ελέγχου μέσω IR.....	54
3.2.14	Αισθητήρας αφής .....	56
3.2.15	Πλακέτα buzzer KY-012.....	57
3.2.16	Αντλία νερού JT-180.....	59
3.2.17	Ανεμιστήρας EB60251S1 SUNON.....	60
3.2.18	XBee.....	62
4	Σενάρια αυτοματισμών και κύκλωμα .....	68
4.1	Εισαγωγή .....	68
4.2	Σενάριο ελέγχου ποιότητας νερού δεξαμενής .....	68
4.3	Σενάριο αυτόματου ποτίσματος.....	70
4.4	Σενάριο ελέγχου εξαερισμού και ποιότητας αέρα .....	72
4.5	Σενάριο ανίχνευσης φωτιάς και πλημμύρας .....	75
4.6	Σενάριο ασφάλειας εξωτερικής πόρτας .....	77
4.7	Σενάριο ελέγχου παντζουριών και εσωτερικού φωτισμού .....	79
4.8	Σενάριο ελέγχου σκιάστρου.....	81
4.9	Τελικό κύκλωμα .....	84
5	Υλοποίηση Διασύνδεσης .....	92
5.1	Εισαγωγή .....	92
5.2	Σύνδεση στο Διαδίκτυο .....	92
5.3	Βάση Δεδομένων .....	93
5.4	Εμφάνιση και έλεγχος E/E με AJAX μέσω κάρτας SD.....	94
5.5	IoT Πλατφόρμα.....	95
5.6	Google Charts .....	98
5.6.1	Εβδομαδιαία στατιστικά.....	98
5.6.2	Μηνιαία στατιστικά.....	99
5.6.3	Ιστογράμματα .....	99

5.6.4	Συνολικά στατιστικά .....	100
5.6.5	Gauges .....	101
5.6.6	Προβολή δεδομένων για συγκεκριμένο χρονικό διάστημα.....	101
5.6.7	Προβολή alarms σε πραγματικό χρόνο .....	102
5.6.8	Συγκριτικά γραφήματα.....	102
5.6.9	Προβολή Historic Alarms .....	103
6	Συμπεράσματα .....	105
6.1	Εισαγωγή .....	105
6.2	Προβλήματα και αντιμετώπιση αυτών .....	105
6.3	Συμπεράσματα και Μελλοντικές επεκτάσεις.....	106
	Βιβλιογραφία.....	108
	Παράρτημα Α: Κώδικας Software Υλοποίηση Ιστότοπου, Ιστοσελίδων, Γραφικών Ιστογραμμάτων	111
	Παράρτημα Β: Κώδικας Software Υλοποίηση Arduino .....	208

## Ευρετήριο εικόνων

Εικόνα 2.1: Απεικόνιση συνδεσμολογίας SPI.....	6
Εικόνα 2.2: : Παράδειγμα πρωτοκόλλου HTTP στην γραμμή αναζήτησης ενός φυλλομετρητή. ....	7
Εικόνα 2.3: Παράδειγμα πρωτοκόλλου HTTP στην γραμμή αναζήτησης ενός φυλλομετρητή. ....	7
Εικόνα 2.4: Λογότυπο ZigBee. ....	8
Εικόνα 2.5: Λογότυπο Arduino IDE. ....	9
Εικόνα 2.6: Περιβάλλον Arduino IDE. ....	9
Εικόνα 2.7: Οδηγίες για επιλογή πλακέτας στο Arduino IDE. ....	10
Εικόνα 2.8: Οδηγίες για επιλογή θύρας στο Arduino IDE.....	10
Εικόνα 2.9: Οδηγίες για έλεγχο λαθών στο Arduino IDE.....	11
Εικόνα 2.10: Οδηγίες για φόρτωση του κώδικα στο Arduino IDE.....	11
Εικόνα 2.11: Ενδείξεις σφαλμάτων στο Arduino IDE.....	11
Εικόνα 2.12: Οδηγίες για άνοιγμα σειριακής οθόνης στο Arduino IDE.....	11
Εικόνα 2.13: Σειριακή οθόνη. ....	11
Εικόνα 2.14: Οδηγίες για άνοιγμα Serial Plotter στο Arduino IDE.....	12
Εικόνα 2.15: Serial Plotter .....	12
Εικόνα 2.16: Λογότυπο Hyper Text Markup Language. ....	12
(HTML   w3, 2024) .....	12
(HTML Basics   mason, 2024) .....	13
Εικόνα 2.17: Λογότυπο Cascading Style Sheets.....	13
(CSS  w3, 2024) .....	14
Εικόνα 2.18: Λογότυπο JavaScript. ....	14
(Javascript   w3, 2024) .....	14
Εικόνα 2.19: Λογότυπο PHP.....	15
(php   Wikipedia, 2024).....	15
Εικόνα 2.20: Παράδειγμα αρχιτεκτονικής PHP. ....	16
(What is PHP  easytolearning, 2024) .....	16
Εικόνα 2.21: Λογότυπο AJAX.....	16
(What is AJAX?   w3schools, 2024) .....	16
Εικόνα 2.22: Αρχιτεκτονική AJAX. ....	17
(How AJAX works?   javatpoint, 2024).....	17
Εικόνα 2.23: Λογότυπο MySQL.....	17
(MySQL   seeklogo, 2024) .....	17
Εικόνα 2.24: Λογότυπο Notepad++.....	19
(How to Use Notepad++  arrow, 2020).....	19
Εικόνα 2.25: Λογότυπο XAMPP. ....	19
(XAMPP Tutorial  javatpoint, 2024).....	19
Εικόνα 2.26: Λογότυπο fritzing. ....	20
(Fritzing, all you need to know   automatismosmundo, 2024) .....	20
Εικόνα 2.27: Περιβάλλον fritzing. ....	21
(Fritzing electronics   fritzingonline, 2024).....	20
Εικόνα 2.28: Λογότυπο Google Charts.....	21
(Google Chart Tools Review  financesonline, 2024) .....	21
Εικόνα 2.29: Λογότυπο ThingSpeak.....	21
(Thingspeak   iotone, 2024).....	21
Εικόνα 3.1: Τα κύρια εξαρτήματα της πλακέτας Arduino. ....	23
(Arduino   Arduino, 2024).....	23
Εικόνα 3.2: Ροή ισχύος σε ένα Arduino UNO. ....	24
(Everything you need to know about the Arduino Hardware   Circuit Digest, 2022).....	24
Εικόνα 3.3: Τοποθεσία των LEDs σε ένα Arduino UNO. ....	25
(Everything you need to know about the Arduino Hardware   Circuit Digest, 2022).....	25
Εικόνα 3.4: Arduino UNO R3.....	26
(Arduino Uno Rev3  Grobotronics, 2024).....	26
Εικόνα 3.5: Arduino Mega 2560 R3 .....	26

(Arduino Mega   Grobotronics, 2024).....	26
Εικόνα 3.6: Arduino Nano .....	27
(Arduino Nano   Grobotronics, 2024) .....	27
Εικόνα 3.7: Σύνδεση shield με Arduino.....	28
(Arduino ethernet shield   Arduinobots, 2015).....	28
Εικόνα 3.8: Σύνδεση Ethernet shield με το διαδίκτυο μέσω RJ45 καλωδίου. ....	28
(Interfacing ethernet shield with arduino   microcontrollerslab, 2021). ....	28
Εικόνα 3.9: : Πλακέτα ρελέ 2 καναλιών. ....	30
(Interface Two Channel Relay Module with Arduino   lastminuteengineers, 2024).....	30
Εικόνα 3.10: Αισθητήρας κίνησης PIR.....	32
(Mini Pyroelectric Pir Sensor Module Manual Motion Infrared Ir Detector  fruugoschweiz, 2024)....	32
Εικόνα 3.11: Συνδεσμολογία αισθητήρα κίνησης PIR .....	33
Εικόνα 3.12: Αισθητήρας αερίου MQ-135. ....	34
(MQ-135 Gas Sensor module   robocraze, 2024).....	34
Εικόνα 3.13: Συνδεσμολογία αισθητήρα MQ-135.....	35
Εικόνα 3.14: Αισθητήρας θερμοκρασίας DHT11.....	36
(Αισθητήρας Υγρασίας & Θερμοκρασίας DHT11   grobotronics, 2024) .....	36
Εικόνα 3.15: Συνδεσμολογία αισθητήρα DHT11 .....	37
Εικόνα 3.16: Αισθητήρας φλόγας KY-026. ....	38
(KY-026 Flame Sensor Module   componentstree, 2024).....	38
Εικόνα 3.17: Συνδεσμολογία αισθητήρα KY-026 .....	39
Εικόνα 3.18: Ρολόι πραγματικού χρόνου DS1302.....	40
(DS1302 Real Time Clock-RTC-Module with Battery   sharvielectronics, 2024) .....	40
Εικόνα 3.19: Συνδεσμολογία DS1302 module .....	41
Εικόνα 3.20: Πλακέτα φωτοαντίστασης KY-018. ....	43
(LDR Module KY-018 (Analog) Photosensitive Resistor   bec-lb, 2024) .....	43
Εικόνα 3.21: Συνδεσμολογία αισθητήρα KY-018 .....	44
Εικόνα 3.22: Αισθητήρας θερμοκρασίας νερού DS18B20p.....	44
(Guide for DS18B20 Temperature Sensor with Arduino   randomnerdtutorials, 2024) .....	45
Εικόνα 3.23: Συνδεσμολογία αισθητήρα DS18B20p.....	45
Εικόνα 3.24: Αισθητήρας υγρασίας εδάφους.....	46
(KTC CONS Labs Soil Soil Moisture Sensor Module   desertcart, 2024).....	47
Εικόνα 3.25: Συνδεσμολογία soil moisture sensor.....	47
Εικόνα 3.26: Αισθητήρας καθαρότητας νερού. ....	48
(Turbidity sensor SKU SEN0189  dfrobot, 2024).....	48
Εικόνα 3.27: Συνδεσμολογία αισθητήρα SEN0189 .....	49
Εικόνα 3.28: Αισθητήρας κλίσης. ....	50
(KY-017 Mercury Tilt Switch Module   arduinomodules, 2024).....	50
Εικόνα 3.29: Συνδεσμολογία αισθητήρα KY-017 .....	51
Εικόνα 3.30: Αισθητήρας στάθμης νερού.....	52
(Robodo SEN18 Water Level Sensor   amazon, 2024) .....	52
Εικόνα 3.31: Συνδεσμολογία αισθητήρα στάθμης νερού .....	53
Εικόνα 3.32: Κιτ ασύρματου απομακρυσμένου ελέγχου μέσω IR. ....	54
(Infrared IR Receiver Module  topelectronics, 2024).....	54
Εικόνα 3.33: Συνδεσμολογία κιτ ασύρματου απομακρυσμένου ελέγχου .....	71
Εικόνα 3.34: Αισθητήρας αφής.....	56
(Gravity: Digital Capacitive Touch Sensor for ArduinoModule   dfrobot, 2024).....	56
Εικόνα 3.35: Συνδεσμολογία αισθητήρα αφής .....	57
Εικόνα 3.36: Πλακέτα buzzer KY-012. ....	57
(KY-012 Active Piezo-Buzzer Module   thegeekpub, 2024) .....	58
Εικόνα 3.37: Συνδεσμολογία πλακέτας buzzer KY-012.....	58
Εικόνα 3.38: Αντλία νερού JT-180. ....	59
(DC Mini Immersible Water Pump   diyables, 2024).....	59
Εικόνα 3.39: Συνδεσμολογία Αντλίας νερού JT-180.....	57

Εικόνα 3.40: Ανεμιστήρας EB60251S1 SUNON.....	61
(Ανεμιστήρας 12V DC   diyables, 2024).....	61
Εικόνα 3.41: Συνδεσμολογία Ανεμιστήρα EB60251S1 SUNON.....	57
Εικόνα 3.42: Αναζήτηση των συνδεδεμένων RF συσκευών.....	63
Εικόνα 3.43: Επιλογή σειριακής θύρας.....	63
Εικόνα 3.44: Ρυθμίσεις σειριακής επικοινωνίας.....	64
Εικόνα 3.45: Επιλογή συσκευής XBee για σύνδεση στο XCTU.....	64
Εικόνα 3.46: Ρύθμιση XBee module και ανανέωση firmware.....	80
Εικόνα 3.47: Επιλογές ανανέωσης firmware.....	80
Εικόνα 3.48: Ορισμός ID και διευθύνσεων στην πλευρά του Coordinator.....	81
Εικόνα 3.49: Συνδεσμολογία XBee στο Arduino.....	81
Εικόνα 4.1: : Κύκλωμα σεναρίου ελέγχου ποιότητας νερού δεξαμενής.....	84
Εικόνα 4.2: Αισθητήρας ελέγχου ποιότητας νερού δεξαμενής.....	85
Εικόνα 4.3: Κύκλωμα σεναρίου αυτόματου ποτίσματος.....	86
Εικόνα 4.4: Αισθητήρας ελέγχου αυτόματου ποτίσματος.....	87
Εικόνα 4.5: : Κύκλωμα σεναρίου εξαερισμού.....	89
Εικόνα 4.6: Αισθητήρας ελέγχου εξαερισμού.....	89
Εικόνα 4.7: Κύκλωμα σεναρίου ανίχνευσης φωτιάς και πλημμύρας.....	91
Εικόνα 4.8: Αισθητήρας ανίχνευσης φωτιάς και πλημμύρας.....	91
Εικόνα 4.9: Κύκλωμα σεναρίου ασφάλειας εξωτερικής πόρτας.....	93
Εικόνα 4.10: Αισθητήρας ασφάλειας εξωτερικής πόρτας.....	94
Εικόνα 4.11: Κύκλωμα σεναρίου ελέγχου παντζουριών και εσωτερικού φωτισμού.....	95
Εικόνα 4.12: Αισθητήρας ελέγχου παντζουριών και εσωτερικού φωτισμού.....	95
Εικόνα 4.13: Κύκλωμα σεναρίου ελέγχου σκιάστρου.....	96
Εικόνα 4.14: Αισθητήρας ελέγχου σκιάστρου.....	97
Εικόνα 4.15: Τελικό κύκλωμα συστήματος.....	98
Εικόνα 4.16: Τελικό υλοποιημένο κύκλωμα συστήματος.....	99
Εικόνα 4.17: Βόρεια πλευρά συστήματος.....	90
Εικόνα 4.18: Ανατολική πλευρά συστήματος.....	100
Εικόνα 4.19: Νότια πλευρά συστήματος.....	100
Εικόνα 4.20: Εσωτερική δυτική πλευρά συστήματος.....	101
Εικόνα 4.21: Εσωτερική νότια πλευρά συστήματος.....	101
Εικόνα 4.22: Δυτική πλευρά συστήματος.....	102
Εικόνα 4.23: Εσωτερική βόρεια πλευρά συστήματος.....	102
Εικόνα 4.24: Κάτοψη συστήματος.....	103
Εικόνα 4.25: Εσωτερική ανατολική πλευρά συστήματος.....	103
Εικόνα 4.26: Εσωτερική κάτοψη συστήματος.....	104
Εικόνα 4.27: Τελική μακέτα συστήματος(1).....	104
Εικόνα 4.28: Τελική μακέτα συστήματος(2).....	105
Εικόνα 5.1: Απεικόνιση του Homepage της ιστοσελίδας.....	107
Εικόνα 5.2: Δημιουργία βάσης δεδομένων μέσω του phpMyAdmin.....	108
Εικόνα 5.3: Δομή βάσης δεδομένων.....	109
Εικόνα 5.4: Τα δεδομένα της βάσης στην ιστοσελίδα.....	109
Εικόνα 5.5: Ιστοσελίδα για εμφάνιση και έλεγχο E/E.....	110
Εικόνα 5.6: Πεδία μεταβλητών στο ThingSpeak.....	111
Εικόνα 5.7: Κλειδιά API για εγγραφή ή ανάγνωση δεδομένων.....	111
Εικόνα 5.8: Καρτέλα “Sharing” στο ThingSpeak.....	112
Εικόνα 5.9: Το διάγραμμα θερμοκρασίας όπως φαίνεται στην ιστοσελίδα.....	112
Εικόνα 5.10: Το διάγραμμα για την πλημμύρα όπως φαίνεται στην ιστοσελίδα.....	112
Εικόνα 5.11: Εβδομαδιαία στατιστικά θερμοκρασίας.....	114
Εικόνα 5.12: Μηνιαία στατιστικά θερμοκρασίας.....	114
Εικόνα 5.13: Ιστόγραμμα δεδομένων.....	115
Εικόνα 5.14: Συνολικά στατιστικά θερμοκρασίας.....	115
Εικόνα 5.15: Γραφική απεικόνιση με Google charts.....	115
Εικόνα 5.16: Προβολή δεδομένων για συγκεκριμένο χρονικό διάστημα.....	116

Εικόνα 5.17: Προβολή Live Alarms .....	117
Εικόνα 5.18: Συγκριτικά γραφήματα .....	117
Εικόνα 5.19: Ιστορικά Alarms .....	118



## Ευρετήριο πινάκων

Πίνακας 2.1: Βασικές εντολές HTML .....	13
Πίνακας 2.2: Βασικές εντολές SQL .....	18
Πίνακας 3.1: Περιγραφή ακίδων πλακέτας Arduino Uno.....	25
Πίνακας 3.2: Κώδικας αρχικοποίησης επικοινωνία με το W5100 shield .....	29
Πίνακας 3.3: Περιγραφή LEDs κατάστασης στο Ethernet shield.....	30
Πίνακας 3.4: Χαρακτηριστικά πλακέτας ρελέ δύο καναλιών.....	32
Πίνακας 3.5: Χαρακτηριστικά αισθητήρα κίνησης PIR.....	33
Πίνακας 3.6: Ενδεικτικός κώδικας για την διεπαφή με τον αισθητήρα PIR.....	34
Πίνακας 3.7: Χαρακτηριστικά αισθητήρα αερίου MQ-135.....	36
Πίνακας 3.8: Ενδεικτικός κώδικας για την διεπαφή με τον αισθητήρα αερίου MQ-135.....	37
Πίνακας 3.9: Χαρακτηριστικά αισθητήρα θερμοκρασίας DHT11.....	37
Πίνακας 3.10: Ενδεικτικός κώδικας αισθητήρα θερμοκρασίας DHT11.....	37
Πίνακας 3.11: Χαρακτηριστικά αισθητήρα αίσθησης φλόγας KY-026.....	37
Πίνακας 3.12: Ενδεικτικός κώδικας αισθητήρα φλόγας KY-026.....	37
Πίνακας 3.13: Χαρακτηριστικά ρολογιού πραγματικού χρόνου DS1302.....	37
Πίνακας 3.14: Ενδεικτικός κώδικας ρολογιού πραγματικού χρόνου DS1302.....	37
Πίνακας 3.15: Χαρακτηριστικά πλακέτας φωτοαντίστασης KY-018.....	37
Πίνακας 3.16: Ενδεικτικός κώδικας πλακέτας φωτοαντίστασης KY-018.....	60
Πίνακας 3.17: Χαρακτηριστικά αισθητήρα θερμοκρασίας νερού DS18B20p.....	61
Πίνακας 3.18: Ενδεικτικός κώδικας αισθητήρα θερμοκρασίας νερού DS18B20p.....	62
Πίνακας 3.19: Χαρακτηριστικά αισθητήρα υγρασίας εδάφους.....	63
Πίνακας 3.20: Ενδεικτικός κώδικας αισθητήρα υγρασίας εδάφους.....	64
Πίνακας 3.21: Χαρακτηριστικά αισθητήρα καθαρότητας νερού SEN0189.....	65
Πίνακας 3.22: Ενδεικτικός κώδικας αισθητήρα θερμοκρασίας νερού DS18B20p.....	65
Πίνακας 3.23: Χαρακτηριστικά αισθητήρα κλίσης KY-017.....	66
Πίνακας 3.24: Ενδεικτικός κώδικας αισθητήρα κλίσης KY-017.....	67
Πίνακας 3.25: Χαρακτηριστικά αισθητήρα στάθμης νερού.....	68
Πίνακας 3.26: Ενδεικτικός κώδικας αισθητήρα στάθμης νερού.....	69
Πίνακας 3.27: Χαρακτηριστικά kit ασύρματου απομακρυσμένου ελέγχου μέσω IR.....	70
Πίνακας 3.28: Ενδεικτικός κώδικας kit ασύρματου απομακρυσμένου ελέγχου μέσω IR.....	71
Πίνακας 3.29: Χαρακτηριστικά αισθητήρα αφής.....	72
Πίνακας 3.30: Ενδεικτικός κώδικας για τον αισθητήρα αφής.....	73
Πίνακας 3.31: Χαρακτηριστικά της πλακέτας buzzer KY-012.....	74
Πίνακας 3.32: Ενδεικτικός κώδικας για το buzzer KY-012.....	74
Πίνακας 3.33: Χαρακτηριστικά της αντλίας νερού JT-180.....	75
Πίνακας 3.34: Ενδεικτικός κώδικας της αντλίας νερού JT-180.....	76
Πίνακας 3.35: Χαρακτηριστικά ανεμιστήρα EB60251S1 SUNON.....	59
Πίνακας 3.36: Ενδεικτικός κώδικας ανεμιστήρα EB60251S1 SUNON.....	60
Πίνακας 3.37: Ενδεικτικός κώδικας για την λειτουργία του XBee.....	83

## Συντομογραφίες & Ακρωνύμια

AP	Access Point
API	Application Programming Interface
ANFIS	Adaptive Neuro-Fuzzy Interface System
AVR	Alf and Vegard's RISC processor
CSS	Cascading Style Sheets
DOM	Document Object Model
GHz	Giga Hertz
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
I2C	Inter-IC bus
ICSP	In Circuit Serial Programming
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
IR	Infrared Radiation
JSON	JavaScript Object Notation
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LSB	Least Significant Bit
MAC	Media Access Control address
Mbps	Megabits per second
MISO	Master In Slave Out
MOSI	Master Out Slave In

MSB	Most Significant Bit
PCB	Printed Circuit Board
PHP	Hypertext Preprocessor
PWM	Pulse Width Modulation
RF	Radio Frequency
RFID	Radio Frequency Identification
RGB	Red, Green, and Blue
RX	Receive Cross
SCK	Serial Clock
SCL	Serial CLock
SDA	Serial DATA
SMD	Surface Mounted Diode
SMTP	Simple Mail Transfer Protocol
SPI	Serial Peripheral Interface
SQL	Structured Query Language
SVG	Scalable Vector Graphics
TX	Transmit Cross
UART	Universal Asynchronous Receiver/Transmitter
URL	Uniform Resource Locator
USB	Universal Serial Bus
UTF	Unicode Transformation Format
W3C	World Wide Web Consortium
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network
WSDL	Web Service Description Language
WSN	Wireless Sensor Network

WWW	World Wide Web
XAMPP	X(cross-platform) Apache Http, MySQL, Php Perl
XCTU	XBee Configuration and Test Utility
XML	Extensible Markup Language
ZDO	ZigBee Device Objects

# 1 Εισαγωγή

## 1.1 Γενικά στοιχεία

Ο πρωτογενής τομέας αποτελεί έναν βασικό πυλώνα της οικονομίας κάθε κράτους και ιδιαίτερα της Ελλάδας καθώς ένα μεγάλο ποσοστό των εξαγωγών αποτελεί προϊόν αγροτικής δραστηριότητας. Στα πλαίσια αυτής, η χρήση θερμοκηπίων είναι εκτεταμένη σε μια προσπάθεια να προστατευτεί η παραγωγή από τις καιρικές συνθήκες, ειδικά τα τελευταία χρόνια που η κλιματική αλλαγή έχει κάνει συνήθη την εκδήλωση ακραίων καιρικών φαινομένων όλες τις εποχές του χρόνου. Παράλληλα, η εξέλιξη της τεχνολογίας είναι τόσο ραγδαία που ολοένα και εισέρχεται σε διάφορες πτυχές της καθημερινής ζωής των ανθρώπων με τον πρωτογενή τομέα να μην αποτελεί εξαίρεση. Για να λειτουργήσει σωστά ένα θερμοκήπιο χρειάζεται έλεγχο διάφορων συνθηκών όπως η θερμοκρασία, η υγρασία, ο αερισμός, η σκίαση, το πότισμα κ. α. ανάλογα με τις ανάγκες της κάθε καλλιέργειας. Δεν είναι λίγοι οι αγρότες που προσπαθούν να αυτοματοποιήσουν όλες αυτές τις διαδικασίες με σκοπό την αύξηση της παραγωγικότητας στη δουλειά τους. (Ntousakis, 2021)

## 1.2 Έξυπνα Θερμοκήπια

Η αυξανόμενη ζήτηση τροφίμων, οδηγούμενη από την ραγδαία αύξηση του παγκόσμιου πληθυσμού τις τελευταίες δεκαετίες και τις προσπάθειες για την εξάλειψη της παγκόσμιας πείνας, σε συνδυασμό με τις πιέσεις της αγοράς για ποιοτικότερα και φρέσκα γεωργικά προϊόντα όλο το χρόνο, οδήγησε στην αύξηση της καλλιέργειας σε θερμοκήπια. Ωστόσο, τα θερμοκήπια έχουν υψηλές απαιτήσεις σε πόρους όπως το νερό, τα λιπάσματα, τα φυτοφάρμακα και φυσικά την ενέργεια. Τα έξυπνα θερμοκήπια ελαχιστοποιούν αυτές τις απαιτήσεις χρησιμοποιώντας σύγχρονα υπολογιστικά συστήματα και, πιο συγκεκριμένα, το Διαδίκτυο των Πραγμάτων (IoT). Εξοπλισμένα με διάφορους αισθητήρες, μπορούν να παρακολουθούν και να διατηρούν το περιβάλλον τους, εξοικονομώντας παράλληλα πόρους. Λειτουργίες όπως η αυτοματοποιημένη και ελεγχόμενη άρδευση, η διατήρηση της βέλτιστης υγρασίας και θερμοκρασίας, καθώς και η παρακολούθηση της οξύτητας του εδάφους είναι δυνατές. Πολλά έξυπνα θερμοκήπια ενσωματώνουν επίσης πρόσθετα μέτρα ασφαλείας, όπως πυροπροστασία και συστήματα παρακολούθησης με κάμερες σε πραγματικό χρόνο. (Kodali, 2016)

Οι Pradyumna Tripathy, Aditi Agarwal, Ajaya Tripathy και Saraju Mohanty (Tripathy, 2021) παρουσιάζουν τις δυνατότητες του Διαδικτύου των Πραγμάτων στον τομέα της

καλλιέργειας σε θερμοκήπια, στηρίζοντας την έξυπνη γεωργία. Διάφορες παράμετροι, όπως η υγρασία, το επίπεδο θρεπτικών συστατικών στο νερό, η τιμή pH και η ηλεκτρική αγωγιμότητα, η θερμοκρασία, η ένταση του υπεριώδους φωτός, το επίπεδο CO<sub>2</sub>, η ομίχλη και η ποσότητα εντομοκτόνων ή παρασιτοκτόνων, παρακολουθούνται μέσω διαφόρων αισθητήρων, επιτρέποντας τη συλλογή σημαντικών γνώσεων και την έγκαιρη ανίχνευση και διάγνωση σφαλμάτων. Ένα σύστημα υποστήριξης αποφάσεων αποτελεί το κεντρικό λειτουργικό μέρος που διέπει και συντονίζει όλες τις δραστηριότητες. Η εφαρμογή αυτή λαμβάνει επίσης υπόψη τις διάφορες προκλήσεις της καλλιέργειας τριαντάφυλλων σε θερμοκήπια και προτείνει μια νέα, βιώσιμη λύση βασισμένη στο IoT.

Οι Soheli, Jahan, Hossain και Adhikary (Soheli, 2022) παρουσίασαν ένα σύστημα παρακολούθησης έξυπνου θερμοκηπίου βασισμένο στο IoT και ένα προσαρμοστικό ασαφές νευρωνικό δίκτυο (ANFIS) το οποίο παρακολουθεί και ελέγχει τις τρέχουσες περιβαλλοντικές συνθήκες μέσα στο θερμοκήπιο, όπως η θερμοκρασία, η υγρασία και η υγρασία του εδάφους, χρησιμοποιώντας αισθητήρες που είναι συνδεδεμένοι μέσω ενός ασύρματου δικτύου αισθητήρων. Όλα αυτά τα δεδομένα αποστέλλονται συλλογικά στον αγρότη χρησιμοποιώντας μία εφαρμογή όπου ανέπτυξαν οι ίδιοι οι ερευνητές. Αυτό το σύστημα χρησιμοποιεί fuzzy logic για να προβλέψει τις βέλτιστες τιμές για τις επιθυμητές συνθήκες εντός του θερμοκηπίου. Τα αποτελέσματα δείχνουν την ακρίβεια του μοντέλου στη λήψη αποφάσεων και παρέχουν γρήγορη πρόσβαση στην ανάπτυξη των καλλιεργειών για την επίτευξη ικανοποιητικής απόδοσης. Ανάλογα με τις απαιτήσεις των καλλιεργειών, ορίζεται μια τιμή κατωφλίου και εάν οι περιβαλλοντικές συνθήκες, πέσουν κάτω ή ανέβουν πάνω από την τιμή κατωφλίου, τότε ο αισθητήρας ανιχνεύει την αλλαγή στις παραμέτρους και όλα τα δεδομένα μεταδίδονται στους αγρότες. Στη συνέχεια, οι αγρότες λαμβάνουν την απόφαση ελέγχου και την στέλνουν στο σύστημα.

Σε μία ακόμα υλοποίηση, οι Nitesh Kumar, Barnali Dey, Chandan Chetri, Amrita Agarwal και Aritri Debnath (Kumar, 2022) προτείνουν την υλοποίηση ενός συστήματος όπου αποτελείται από διάφορους αισθητήρες, όπως θερμοκρασίας, υγρασίας κ.λπ., που παρέχουν τα δεδομένα τους σε έναν μικροελεγκτή βασισμένο στο Arduino και στη συνέχεια αυτός προωθεί τις πληροφορίες σε ένα κινητό τηλέφωνο ως SMS μέσω μιας μονάδας Wi-Fi. Εάν οι μετρήσεις που λαμβάνει το σύστημα υπερβούν κάποιες προκαθορισμένες τιμές που ορίζει ο χρήστης, τότε αποστέλλεται μια ειδοποίηση στην συσκευή του.

### 1.3 Σκοπός και στόχοι

Με γνώμονα τα παραπάνω, η παρούσα διπλωματική πραγματεύεται την υλοποίηση αυτών των αυτοματισμών με στόχο τον έλεγχο των κλιματικών συνθηκών στο εσωτερικό ενός θερμοκηπίου. Γίνεται υλοποίηση διάφορων σεναρίων αυτοματισμού, όπως έλεγχος αυτόματου ποτίσματος και έλεγχος σκίαστρων, τα οποία θα αναλυθούν στη συνέχεια. Η υλοποίηση στοχεύει στην κατανόηση εννοιών όπως οι μικροελεγκτές, το Internet of Things (IoT), πρωτόκολλα επικοινωνίας (ενσύρματα και ασύρματα), οι βάσεις δεδομένων, καθώς και ο προγραμματισμός για δημιουργία ενός ιστότοπου. Πιο αναλυτικά, η λειτουργία του θερμοκηπίου θα στηρίζεται στην πλατφόρμα Arduino η οποία μέσω διάφορων αισθητήρων θα διαχειρίζεται τις διάφορες εργασίες, ενώ παράλληλα θα υπάρχει μία συσκευή XBee η οποία θα επικοινωνεί με το Arduino ώστε να λάβει τα δεδομένα και να τα στέλνει ασύρματα μέσω του πρωτοκόλλου ZigBee σε ένα άλλο XBee το οποίο θα τα ανεβάζει σε μία ιστοσελίδα προς ενημέρωση του χρήστη. Όπως αναφέρθηκε, το σύστημα στοχεύει στην αυτοματοποίηση κάποιων διαδικασιών έτσι ώστε να ελέγχονται οι κλιματικές συνθήκες στο εσωτερικό του θερμοκηπίου για βελτιστοποίηση της καλλιέργειας.

Ένα από τα πιο χρήσιμα σενάρια είναι το αυτόματο πότισμα των φυτών. Με τη χρήση ενός αισθητήρα μέτρησης της υγρασίας του εδάφους, θα ενεργοποιείται αυτόματα η αντλία νερού η οποία μέσω μίας δεξαμενής νερού, θα μπορεί να ποτίζει τα φυτά. Παράλληλα, ο αγρότης θα αποδεσμευτεί αφενός από την ανάγκη για συνεχή αξιολόγηση της κατάστασης του φυτού όσον αφορά την υγρασία στη ρίζα του και αφετέρου από την ίδια τη διαδικασία του ποτίσματος.

Σε επέκταση του προηγούμενου, το νερό που θα περιέχεται στην δεξαμενή από την οποία θα αντλείται για το πότισμα θα πρέπει να βρίσκεται σε κατάλληλες συνθήκες. Αυτό σημαίνει πως θα πρέπει να βρίσκεται σε θερμοκρασία ίση με τη θερμοκρασία του περιβάλλοντος, ενώ παράλληλα θα πρέπει να είναι απαλλαγμένο από μηχανικές, χημικές ή άλλες ακαθαρσίες. Έτσι, στο «έξυπνο θερμοκήπιο» θα υπάρχει ένας αισθητήρας θερμοκρασίας καθώς και ένας αισθητήρας μέτρησης της καθαρότητας του νερού ώστε να ειδοποιείται ο αγρότης αν κάποια από αυτές τις δύο τιμές βρίσκεται εκτός των επιθυμητών ορίων και ο αγρότης θα εξετάζει και θα επιλύει το ζήτημα.

Εξίσου σημαντικό ρόλο με την ποιότητα του νερού, παίζει και η ποιότητα του αέρα. Ο εξαερισμός είναι πρώτιστης σημασίας σε ένα θερμοκήπιο καθώς βελτιώνει την κυκλοφορία του αέρα αλλά βοηθά επίσης και στον έλεγχο της υγρασίας. Γι' αυτό το λόγο, το «έξυπνο θερμοκήπιο» θα είναι εξοπλισμένο με αισθητήρες μέτρησης της θερμοκρασίας του αέρα αλλά και της ποιότητάς του ώστε να ενεργοποιηθεί ο ανεμιστήρας που χρησιμοποιείται ως σύστημα

εξαερισμού. Έτσι, μετά την άρδευση θα εξατμιστεί η υπερβολική υγρασία οδηγώντας την σε ιδανικά επίπεδα για την λειτουργία του θερμοκηπίου (50% για τον αέρα).

Μία άλλη λειτουργία του «έξυπνου θερμοκηπίου» θα είναι ο έλεγχος του εσωτερικού φωτισμού μέσω ειδικών λαμπτήρων. Με στόχο την βελτίωση των συνθηκών φωτοσύνθεσης των φυτών, θα υπάρχει η δυνατότητα τις περιόδους που δεν είναι επαρκής ο φυσικός φωτισμός, να λειτουργεί συμπληρωματικά τεχνητός φωτισμός ώστε να συμπληρώνονται 12-16 ώρες φωτός την ημέρα που θεωρούνται ιδανικές για τη βέλτιστη ανάπτυξη των φυτών. Ένας αισθητήρας φωτοαντίστασης θα μετράει την ύπαρξη φυσικού φωτισμού και κατά τη δύση του ηλίου που αυτή πέφτει, θα ενεργοποιούνται οι λαμπτήρες μέχρι να συμπληρωθεί ο απαιτούμενος αριθμός ωρών.

Επιπλέον, η ανίχνευση φωτιάς αποτελεί θεμελιώδης προδιαγραφή ασφαλείας για οποιαδήποτε στεγασμένη κατασκευή, με το θερμοκήπιο να μην αποτελεί εξαίρεση. Έτσι, το θερμοκήπιο θα είναι εξοπλισμένο με αισθητήρα εντοπισμού φλόγας, ενώ παράλληλα στα πλαίσια ασφάλειας με την χρήση του αισθητήρα πλημμύρας, θα είναι σε θέση να εντοπίζει υπερχειλίση της δεξαμενής. Και στις δύο περιπτώσεις θα ενεργοποιείται το σύστημα συναγερμού, ειδοποιώντας τον χρήστη.

Η είσοδος στο θερμοκήπιο θα γίνεται με ασφάλεια και ευκολία καθώς η εξωτερική πόρτα θα ξεκλειδώνει με έναν αισθητήρα αφής σε συνδυασμό με την πληκτρολόγηση ενός μυστικού κωδικού.

Κλείνοντας με τα συστήματα αυτοματισμών, μέσω ενός αισθητήρα κλίσης θα μπορεί να ελέγχεται το σκίαστρο που θα υπάρχει στην οροφή του θερμοκηπίου ώστε να επιτευχθεί μείωση της εισερχόμενης ηλιακής ακτινοβολίας.

## 1.4 Οργάνωση Διπλωματικής

Η διπλωματική εργασία οργανώνεται ως εξής:

Στο δεύτερο κεφάλαιο γίνεται μία σύντομη παρουσίαση και περιγραφή των τεχνολογιών και προτύπων που χρησιμοποιούνται για την υλοποίηση του προτεινόμενου συστήματος. Πιο συγκεκριμένα, γίνεται αναφορά στα πρωτόκολλα SPI, Ethernet, HTTP και ZigBee, στις γλώσσες προγραμματισμού C++, HTML/CSS, Javascript/AJAX, PHP και MySQL που χρησιμοποιήθηκαν και τέλος στα εργαλεία Notepad++, XAMPP, Fritzting, Google Charts και ThingSpeak.

Στο τρίτο κεφάλαιο παρουσιάζεται η πλατφόρμα ανάπτυξης Arduino και οι αισθητήρες που χρησιμοποιούνται στο σύστημα. Επιγραμματικά, γίνεται αναφορά στο Arduino και τις διάφορες εκδόσεις του και παρατίθενται λεπτομερώς όλες οι πληροφορίες για τον κάθε



αισθητήρα και περιφερειακό υλικό (hardware) που επικοινωνεί με αυτό. Αναλύονται επίσης ο τρόπος με τον οποίο συνδέονται μεταξύ τους, αλλά και ο κώδικας που αναπτύχθηκε για την ανταλλαγή δεδομένων μεταξύ αυτών.

Στο τέταρτο κεφάλαιο αναλύονται τα σενάρια τα οποία αναπτύχθηκαν στην παρούσα διπλωματική εργασία, ενώ στο πέμπτο παρουσιάζεται ο ιστότοπος που αναπτύχθηκε και περιγράφεται λεπτομερώς η διαδικασία ανάπτυξής του. Γίνεται αναφορά στην βάση δεδομένων που χρησιμοποιείται για την αποθήκευση και επεξεργασία των δεδομένων που συλλέγονται από το σύστημα, αποφάσεις που ελήφθησαν για την παρουσίαση των δεδομένων στον χρήστη και ο κώδικας που αναπτύχθηκε για την επίτευξη αυτού του στόχου.

Στο έκτο κεφάλαιο παρατίθενται τα συμπεράσματα που εξήχθησαν από την θεωρητική μελέτη αλλά και την υλοποίηση του θερμοκηπίου, καθώς και τα προβλήματα που παρουσιάστηκαν κατά την κατασκευή και η αντιμετώπιση αυτών. Τέλος, αναφέρονται μελλοντικές προεκτάσεις που έχουν σκοπό την εξέλιξη της λειτουργικότητας του συστήματος.

Τέλος, στο παράρτημα Α παρουσιάζεται ο κώδικας που αναπτύχθηκε για την υλοποίηση της λειτουργικότητας του Arduino, τον σχεδιασμό της βάσης δεδομένων και την επικοινωνία της με τον ιστότοπο και την λειτουργικότητα του ίδιου του ιστότοπου.

## 2 Τεχνολογίες και Εργαλεία

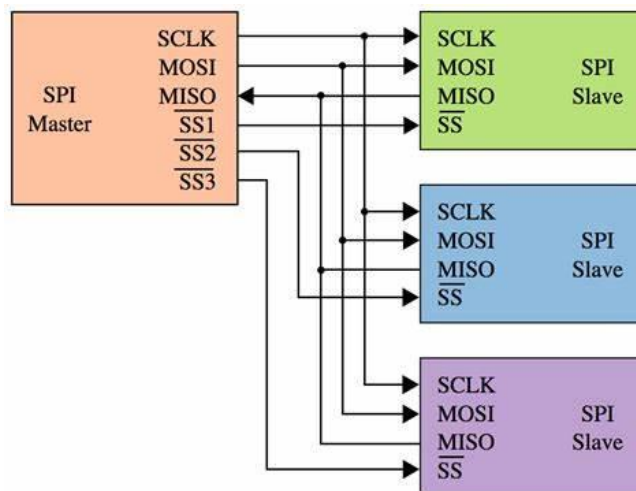
### 2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει μία επεξήγηση των τεχνολογιών αλλά και των εργαλείων που χρησιμοποιήθηκαν. Αρχικά θα επεξηγηθούν τα διαφορετικά πρωτόκολλα επικοινωνίας, στη συνέχεια θα γίνει μια αναφορά στις διαφορετικές γλώσσες προγραμματισμού, ενώ στο τέλος θα γίνει μία παρουσίαση των εργαλείων ανάπτυξης λογισμικού, τόσο σχετικά με τον μικροελεγκτή όσο και με τον προγραμματισμό διαδικτύου.

### 2.2 Πρωτόκολλα επικοινωνίας

Για την επικοινωνία δύο ή περισσότερων υπολογιστών ή κόμβων απαιτείται ένα σύνολο κανόνων που να περιγράφει το πως θα γίνεται αυτή η επικοινωνία. Αυτό το σύνολο, λοιπόν, ονομάζεται πρωτόκολλο επικοινωνίας και οι κανόνες που περιλαμβάνει, ουσιαστικά διαχωρίζουν την πληροφορία σε πακέτα έτοιμα προς μετάδοση, διεξάγουν ελέγχους ακεραιότητας της πληροφορίας, επαληθεύουν την λήψη της πληροφορίας από το δέκτη και ανακτούν ή επαναλαμβάνουν την αποστολή μηνυμάτων τα οποία είτε δεν έφτασαν, είτε καταστράφηκαν στην πορεία. Επικρατούν διάφορα πρωτόκολλα επικοινωνίας ανάλογα με την εφαρμογή, τόσο ενσύρματα όσο και ασύρματα. Παρακάτω θα αναλυθούν τέσσερα από αυτά. (kionetworks, 2021)

#### 2.2.1 Serial Peripheral Interface (SPI)



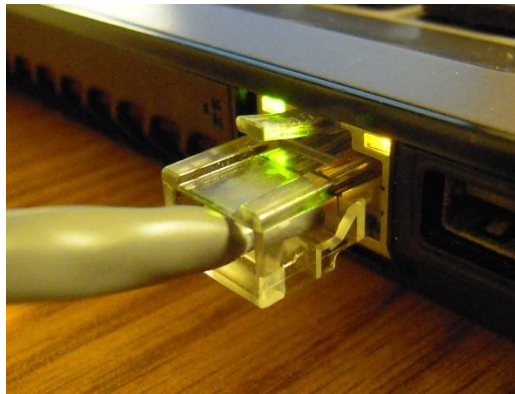
Εικόνα 2.1: Απεικόνιση συνδεσμολογίας SPI.

(SPI | FPGA Key, 2024)

Το SPI είναι ένα ενσύρματο πρωτόκολλο σειριακής επικοινωνίας το οποίο χρησιμοποιείται συχνά για επικοινωνία μεταξύ ενός κύριου (Main) μικροελεγκτή και άλλων περιφερειακών (Peripherals). Αποτελείται από τέσσερις δίαυλους επικοινωνίας: το SCK (Serial Clock) που

είναι το ρολόι για συγχρονισμό της επικοινωνίας, το MOSI (Main Out, Sub In) για αποστολή δεδομένων από τον μικροελεγκτή στα περιφερειακά, το MISO (Main In, Sub Out) για λήψη δεδομένων και το CS (Chip Select) για την επιλογή του περιφερειακού με το οποίο θα διεξαχθεί η επικοινωνία. Η επικοινωνία μέσω SPI είναι full-duplex υποδηλώνοντας πως ο μικροελεγκτής είναι σε θέση και να στέλνει αλλά και να λαμβάνει δεδομένα ταυτόχρονα. Στην παρούσα εργασία θα χρησιμοποιηθεί το SPI για την επικοινωνία του Arduino με το Ethernet shield. (Hopkins, 2021)

## 2.2.2 Ethernet



*Εικόνα 2.2: : Παράδειγμα πρωτοκόλλου HTTP στην γραμμή αναζήτησης ενός φυλλομετρητή.  
(Ethernet | Wikipedia, 2024)*

Το Ethernet είναι ένα πρωτόκολλο ενσύρματης επικοινωνίας, τυποποιημένο ως IEEE 802.3, το οποίο δίνει τη δυνατότητα σύνδεσης στο διαδίκτυο. Αποτελεί ένα ευρέως χρησιμοποιούμενο πρωτόκολλο καθώς προσφέρει ισορροπία μεταξύ ταχύτητας, κόστους και ευκολίας στη χρήση. (developer.mozilla.org, 2024)

## 2.2.3 Hyper Text Transfer Protocol (HTTP)



*Εικόνα 2.3: Παράδειγμα πρωτοκόλλου HTTP στην γραμμή αναζήτησης ενός φυλλομετρητή.  
(Hypertext Transfer Protocol | Khan Academy, 2024)*

Το HTTP αποτελεί το κύριο πρωτόκολλο επικοινωνίας που χρησιμοποιείται στους φυλλομετρητές του διαδικτύου για τη μεταφορά δεδομένων μεταξύ ενός διακομιστή (server) και ενός πελάτη (client). Ο πελάτης, ο οποίος συνήθως είναι ένα πρόγραμμα περιήγησης ιστού, εκτελεί ένα αίτημα HTTP στον διακομιστή ο οποίος απαντάει με το έγγραφο HTML που

ζητήθηκε. Υπάρχει η δυνατότητα μεταφοράς οποιασδήποτε πληροφορίας μεταφέρεται μέσω του διαδικτύου, π.χ. κείμενο, ήχος, εικόνες, βίντεο και άλλα. (kionetworks, 2021)

#### 2.2.4 ZigBee



Εικόνα 2.4: Λογότυπο ZigBee.

(Zigbee | w3.org, 2024)

Το ZigBee είναι ένα πρωτόκολλο επικοινωνίας βασισμένο στην τυποποίηση IEEE 802.15.4, το οποίο είναι σχεδιασμένο για τη δημιουργία ενός οικιακού δικτύου (PAN - Personal Area Network), δηλαδή για ασύρματη δικτύωση συσκευών με ραδιοσυχνότητες που απαιτούν χαμηλό ρυθμό δεδομένων και χαμηλή ισχύ. Πιο συγκεκριμένα, προορίζεται για εφαρμογές μικρής κλίμακας που απαιτούν ασύρματη δικτύωση οι οποίες χαρακτηρίζονται από μικρό εύρος ζώνης (bandwidth) και βρίσκονται σε κοντινές αποστάσεις (από 10 έως 100 μέτρα με οπτική επαφή). Τα πλεονεκτήματά του είναι πως εξασφαλίζει χαμηλή κατανάλωση ενέργειας, είναι εύκολο τόσο στη χρήση όσο και στην επεκτασιμότητα με νέα εξαρτήματα, προσφέρει ασφαλή δικτύωση και έχει αρκετά προσιτή τιμή. Η ταχύτητα αποστολής δεδομένων φτάνει τα 250 kbit/s, ούσα αρκετή για τη λειτουργία της αποστολής δεδομένων από αισθητήρες σε έναν κεντρικό κόμβο.

Υπάρχουν τρεις κατηγορίες συσκευών ZigBee: Συντονιστής (coordinator), δρομολογητής (router) και τερματικές συσκευές (end devices). Πιο αναλυτικά:

- Συντονιστής ZigBee: Ο συντονιστής (coordinator) αποτελεί τη βάση του δικτύου. Υπάρχει ακριβώς ένας συντονιστής σε κάθε δίκτυο ZigBee καθώς είναι η συσκευή που ξεκίνησε αρχικά το δίκτυο. Αποθηκεύει πληροφορίες σχετικά με το δίκτυο, ενώ λειτουργεί και σαν αποθετήριο των κλειδιών ασφαλείας.
- Δρομολογητής ZigBee: Ο δρομολογητής (router) πέρα από την εκτέλεση μιας λειτουργίας της εφαρμογής, δρα και ως ενδιάμεσος δρομολογητής μεταφέροντας δεδομένα σε άλλες συσκευές. Συνήθως τροφοδοτούνται από το ηλεκτρικό δίκτυο ώστε να είναι πάντα διαθέσιμα.
- Τερματικές συσκευές ZigBee: Οι τερματικές συσκευές (end devices) έχουν μόνο τη λειτουργικότητα της επικοινωνίας με τον γονικό κόμβο (είτε συντονιστή, είτε δρομολογητή), ενώ δεν μπορούν να αναμεταδίδουν δεδομένα από άλλες συσκευές. Με αυτόν τον τρόπο δίνεται η δυνατότητα στον κόμβο να τίθεται σε αδράνεια ένα

σημαντικό μέρος του χρόνου, εξασφαλίζοντας μεγάλη διάρκεια ζωής, μιας και συνήθως είναι τροφοδοτούμενος από μπαταρία. (cytron.io, 2011)

## 2.3 Γλώσσες προγραμματισμού

### 2.3.1 Arduino IDE

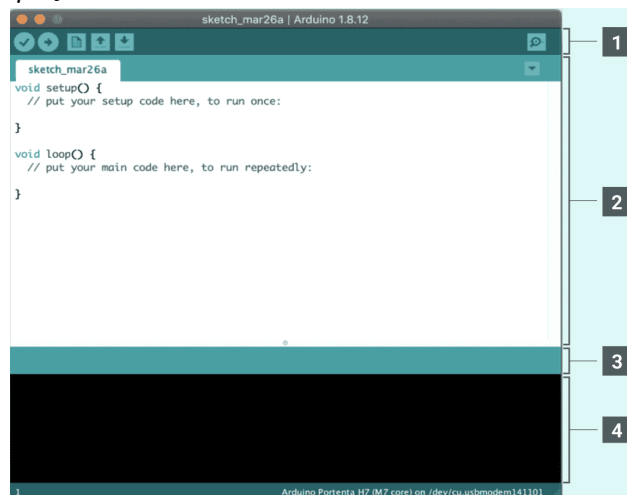


Εικόνα 2.5: Λογότυπο Arduino IDE.

(Arduino | Arduino, 2024)

Το λογισμικό Arduino IDE (Integrated Development Environment) είναι μία πλατφόρμα ανοιχτού κώδικα (open-source) σχεδιασμένη να υποστηρίζει μικροελεγκτές τύπου Arduino, δίνοντας τη δυνατότητα συγγραφής κώδικα, μεταγλώττισης και φόρτωσης του προγράμματος στην πλακέτα. Επίσης, επιτρέπει και την αλληλεπίδραση με το πρόγραμμα κατά την εκτέλεσή του μέσω μιας σειριακής οθόνης, όμοιας με το command line ενός υπολογιστή. Η ανάπτυξη του κώδικα γίνεται σε C++ εμπλουτισμένη με ειδικές βιβλιοθήκες, μεθόδους και συναρτήσεις οι οποίες βοηθούν τον προγραμματιστή στην αλληλεπίδραση με το Arduino. Το Arduino IDE είναι ένα αρκετά φιλικό προς το χρήστη περιβάλλον τόσο στην εγκατάσταση όσο και στη συγγραφή και εκτέλεση του κώδικα. (Arduino, 2024)

#### 2.3.1.1 Προγραμματισμός Arduino IDE



Εικόνα 2.6: Περιβάλλον Arduino IDE.

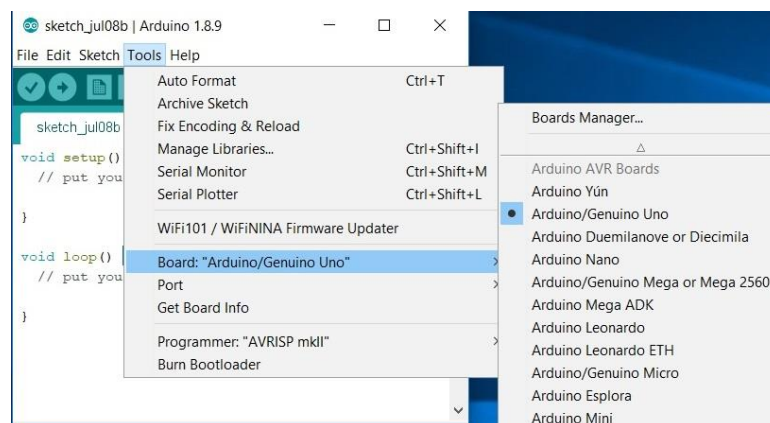
(Arduino | Arduino, 2024)

Το περιβάλλον του Arduino IDE αποτελείται από τέσσερα κομμάτια:

1. Μία γραμμή εργαλείων με κουμπιά για συνηθισμένες συναρτήσεις και μια σειρά από μενού. Τα κουμπιά της γραμμής εργαλείων επιτρέπουν τον έλεγχο για συντακτικά λάθη και τη φόρτωση του προγράμματος, τη δημιουργία, το άνοιγμα και την αποθήκευση προγραμμάτων (sketches) και το άνοιγμα της σειριακής οθόνης.
2. Τον επεξεργαστή κειμένου στον οποίο γίνεται η συγγραφή του κώδικα.
3. Την περιοχή μηνυμάτων η οποία παρέχει ανατροφοδότηση κατά την αποθήκευση και εξαγωγή του προγράμματος και επίσης εμφανίζει τα σφάλματα.
4. Την κονσόλα κειμένου η οποία εμφανίζει το κείμενο που εξάγεται από το λογισμικό Arduino IDE, συμπεριλαμβανομένων πλήρων μηνυμάτων σφάλματος και άλλων πληροφοριών.

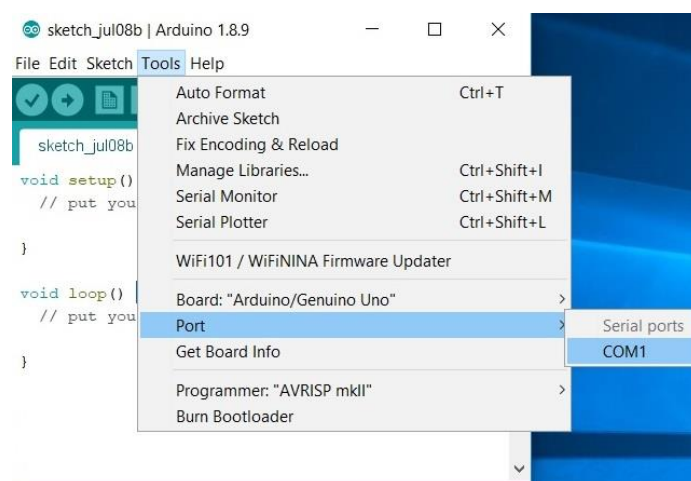
Τα βήματα για την επιτυχή εκτέλεση ενός προγράμματος σε Arduino είναι τα εξής:

1. Επιλογή πλακέτας Arduino:



Εικόνα 2.7: Οδηγίες για επιλογή πλακέτας στο Arduino IDE.

2. Επιλογή θύρας USB στην οποία είναι συνδεδεμένη η πλακέτα Arduino:



Εικόνα 2.8: Οδηγίες για επιλογή θύρας στο Arduino IDE.

3. Έλεγχος του προγράμματος για συντακτικά λάθη:



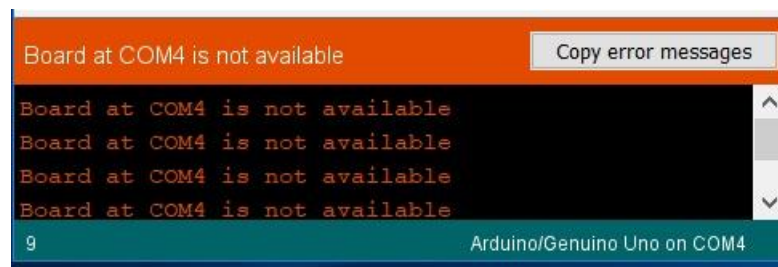
Εικόνα 2.9: Οδηγίες για έλεγχο λαθών στο Arduino IDE.

4. Φόρτωση του κώδικα στην πλακέτα Arduino:



Εικόνα 2.10: Οδηγίες για φόρτωση του κώδικα στο Arduino IDE.

5. Αντιμέτωπιση πιθανών σφαλμάτων που μπορεί να προκύψουν κατά τη μεταγλώττιση ή κατά το ανέβασμα:

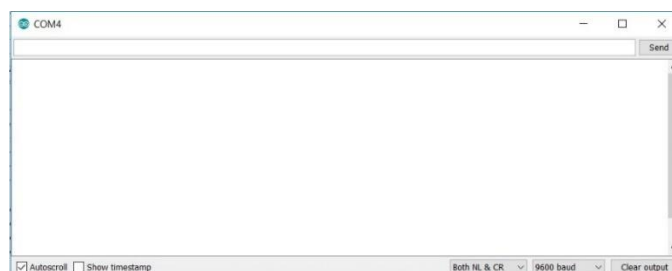


Εικόνα 2.11: Ενδείξεις σφαλμάτων στο Arduino IDE.

6. Άνοιγμα σειριακής οθόνης:



Εικόνα 2.12: Οδηγίες για άνοιγμα σειριακής οθόνης στο Arduino IDE.



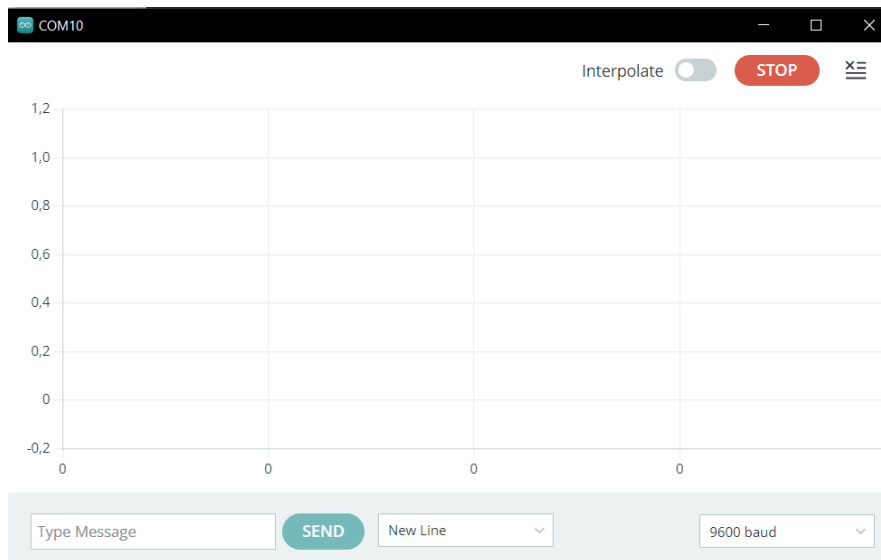
Εικόνα 2.13: Σειριακή οθόνη.

Η σειριακή οθόνη είναι ένα πολύ χρήσιμο εργαλείο τόσο για την αλληλεπίδραση με το πρόγραμμα όσο και για την αποσφαλμάτωσή του. Η εκτύπωση των αποτελεσμάτων στην σειριακή οθόνη είναι η πιο απλή έξοδος που μπορεί να έχει ένα πρόγραμμα. Πιο σύνθετες μέθοδοι περιλαμβάνουν την σύνδεση μιας LCD οθόνης ή το «ανέβασμα» των πληροφοριών σε μια ιστοσελίδα ή σε κάποια εφαρμογή.

## 7. Άνοιγμα Serial Plotter:



Εικόνα 2.14: Οδηγίες για άνοιγμα Serial Plotter στο Arduino IDE.



Εικόνα 2.15: Serial Plotter

Το Serial Plotter είναι ένα εξίσου χρήσιμο εργαλείο το οποίο επιτρέπει την οπτικοποίηση των τιμών κάποιας μεταβλητής μέσω ενός διαγράμματος. Ακόμη, υπάρχει η δυνατότητα για ταυτόχρονη παρατήρηση πολλαπλών μεταβλητών. (docs.arduino, 2024)

## 2.3.2 Γλώσσες προγραμματισμού διαδικτύου

### 2.3.2.1 Hyper Text Markup Language (HTML)



Εικόνα 2.16: Λογότυπο Hyper Text Markup Language.

(HTML | w3, 2024)



Η HTML (Hyper Text Markup Language) είναι η καθιερωμένη γλώσσα σήμανσης για ιστοσελίδες. Ο φυλλομετρητής (web browser) διαβάζει το αρχείο και ανάλογα με τη σήμανση, αναγνωρίζει την μορφή με την οποία θα παρουσιάσει το κείμενο, τις εικόνες, τους συνδέσμους και άλλα. Επιπλέον, η HTML επιτρέπει την ενσωμάτωση εντολών και σε άλλες γλώσσες όπως η CSS και η JavaScript εμπλουτίζοντας τη μορφή της ιστοσελίδας κάνοντάς την πιο διαδραστική. Η ανάπτυξη προγραμμάτων μπορεί να γίνει γράφοντας σε οποιοδήποτε αρχείο κειμένου με κατάληξη *.html* ή *.htm*. (w3schools, HTML Introduction, 2024)

Τα στοιχεία της HTML ονομάζονται ετικέτες (tags) και αποτελούν τα βασικά δομικά στοιχεία μιας ιστοσελίδας. Οι ετικέτες λειτουργούν ανά ζεύγη και ο τρόπος γραφής τους είναι μέσα στα σύμβολα `<...>` για την ετικέτα έναρξης (π.χ. `<head>`) και `</...>` για την ετικέτα λήξης (π.χ. `</head>`). Έπειτα ανάλογα με την επιθυμητή μορφή της ιστοσελίδας μπορεί να περιλαμβάνονται και άλλες εντολές, η καθεμία από τις οποίες εισάγει και ένα νέο στοιχείο στην ιστοσελίδα. Κάποιες από τις βασικές εντολές συνοψίζονται στον παρακάτω πίνακα:

Ετικέτα	Περιγραφή
<code>&lt;!DOCTYPE&gt;</code>	Δήλωση τύπου εγγράφου
<code>&lt;html&gt; ... &lt;/html&gt;</code>	Έναρξη κώδικα HTML
<code>&lt;head&gt; ... &lt;/head&gt;</code>	Καθορίζει την επικεφαλίδα της σελίδας
<code>&lt;title&gt; ... &lt;/title&gt;</code>	Τίτλος σελίδας
<code>&lt;body&gt; ... &lt;/body&gt;</code>	Καθορίζει το κύριο σώμα της σελίδας
<code>&lt;h1&gt; ... &lt;/h1&gt;</code>	Κεφαλίδα (h1 μέχρι h6)
<code>&lt;p&gt; ... &lt;/p&gt;</code>	Αρχή παραγράφου
<code>&lt;img src="..."&gt;</code>	Εισαγωγή εικόνας
<code>&lt;a href="..."&gt; ... &lt;/a&gt;</code>	Εισαγωγή υπερσυνδέσμου

Πίνακας 2.1: Βασικές εντολές HTML

(HTML Basics | mason, 2024)

### 2.3.2.2 Cascading Style Sheets (CSS)



Εικόνα 2.17: Λογότυπο Cascading Style Sheets.

(CSS | w3, 2024)

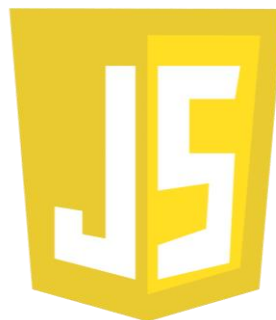
Το CSS (Cascading Style Sheets) αποτελεί συμπλήρωμα της HTML, καθώς χρησιμοποιείται για να περιγράψει την εμφάνιση και τη μορφοποίηση των ιστοσελίδων. Δίνει τη δυνατότητα επεξεργασίας χρωμάτων, γραμματοσειρών, περιθωρίων κ.λπ. συνεισφέροντας στην στυλιστική ανάπτυξη της ιστοσελίδας. Για να αναγνωριστούν από τον φυλλομετρητή τα αρχεία γραμμένα σε CSS πρέπει να έχουν την κατάληξη `.css`. Η χρησιμότητα της CSS έγκειται στο γεγονός πως διαχωρίζει το περιεχόμενο της ιστοσελίδας από την εμφάνισή της, βελτιώνοντας την προσβασιμότητα και παρέχοντας περαιτέρω ευελιξία και έλεγχο στα εμφανισιακά χαρακτηριστικά της ιστοσελίδας. Ο κώδικας σε CSS ουσιαστικά περιλαμβάνει ως κλάσεις τις ετικέτες της HTML και τους δίνει κάποια επιπλέον χαρακτηριστικά. Για παράδειγμα για τις ετικέτες «body», «h1» και «p» μπορούμε να γράψουμε το παρακάτω:

```
body {  
  background-color: lightblue;  
}  
h1 {  
  color: white;  
  text-align: center;  
}  
p {  
  font-family: verdana;  
  font-size: 20px;  
}
```

Σε αυτό το κομμάτι κώδικα καθορίζονται τα χρώματα, η στοίχιση, η γραμματοσειρά και το μέγεθός της για τις διάφορες ετικέτες HTML. (w3schools, CSS Introduction, 2024)

### 2.3.2.3 *JavaScript (JS)*

# JavaScript



Εικόνα 2.18: Λογότυπο JavaScript.

(Javascript | w3, 2024)

Η JavaScript είναι μία γλώσσα προγραμματισμού η οποία επιτρέπει την υλοποίηση σύνθετων λειτουργιών σε ιστοσελίδες καθιστώντας τες δυναμικές. Μερικά παραδείγματα τέτοιων λειτουργιών είναι η εμφάνιση περιοδικών ανανεώσεων του περιεχομένου, οι διαδραστικοί χάρτες, κινούμενα γραφικά 2D/3D και άλλα, ενώ επιπλέον με τη χρήση κάποιων προγραμματιστικών τεχνικών μπορεί να γίνει ανανέωση του περιεχομένου της σελίδας χωρίς ανανέωση ολόκληρης της σελίδας AJAX (Asynchronous JavaScript And XML). Η JavaScript μπορεί να χρησιμοποιηθεί είτε δημιουργώντας ξεχωριστό αρχείο με κατάληξη `.js`, είτε να ενσωματωθεί στην html με τη χρήση των ετικετών `<script type="text/javascript"> ... </script>`.

Ακόμη, οι δυνατότητες της JavaScript ανεβαίνουν ραγδαία με την ενσωμάτωση των APIs (Application Programming Interfaces). Τα APIs είναι έτοιμα κομμάτια κώδικα τα οποία κάνουν μία συγκεκριμένη εργασία και μπορούν να ενσωματωθούν στο πρόγραμμα της JavaScript για να εμπλουτίσουν τις λειτουργίες της ιστοσελίδας. Κάποια γνωστά APIs είναι το DOM (Document Object Model) API το οποίο επιτρέπει την δυναμική ανανέωση της μορφής της ιστοσελίδας (π.χ. pop-up windows), όπως επίσης και το Geolocation API το οποίο ανακτά τις γεωγραφικές πληροφορίες (π.χ. Google Maps).

Τέλος, ιδιαίτερη εξέλιξη υπάρχει και στις βιβλιοθήκες οι οποίες είναι βασισμένες σε JavaScript. Για παράδειγμα η React.js είναι μία ανοιχτού κώδικα βιβλιοθήκη η οποία στοχεύει στη δημιουργία διεπαφών χρήστη (User Interfaces). Υπάρχουν δεκάδες παρόμοιες βιβλιοθήκες ή πλαίσια (frameworks) όπως ονομάζονται, με τα πιο γνωστά να είναι το Node.js, το Angular και το Next.js. (docs, 2024)

#### 2.3.2.4 PHP

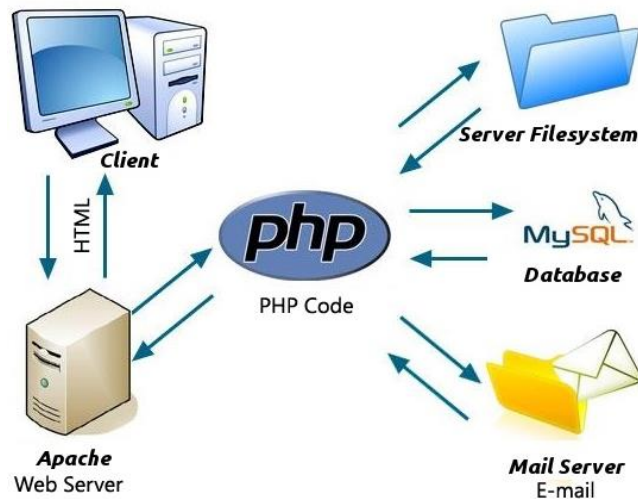


Εικόνα 2.19: Λογότυπο PHP.

(php | Wikipedia, 2024)

Η PHP (Hypertext Preprocessor) είναι μία γλώσσα προγραμματισμού που χρησιμοποιείται κυρίως για τη δημιουργία δυναμικών ιστοσελίδων, προσδίδοντάς τους χαρακτηριστικά όπως φόρμες επικοινωνίας, συστήματα σύνδεσης χρηστών και άλλα. Για να το πετύχει αυτό είναι σχεδιασμένη να τρέχει στην πλευρά του διακομιστή (server-side). Έτσι, μπορεί να επικοινωνεί με βάσεις δεδομένων για την ανάκτηση και εγγραφή δεδομένων, με κάποιο σύστημα διαχείρισης αρχείων ή κάποιο σύστημα αλληλογραφίας χωρίς να έχει πρόσβαση ο χρήστης-

πελάτης (client) στις πληροφορίες αυτές. Αυτή είναι και η ειδοποιός διαφορά της από τη JavaScript, η οποία επίσης χρησιμοποιείται για τη δημιουργία δυναμικών ιστοσελίδων ωστόσο λειτουργεί στην πλευρά του πελάτη (client-side). Ο κώδικας PHP μπορεί να γραφτεί είτε σε αρχείο με κατάλληλη επέκταση *.php* ή *.php4* ή *.phtml*, ή να ενσωματωθεί σε αρχείο HTML χρησιμοποιώντας τις ετικέτες `<?php ... ?>`. (php.net, 2020)



Εικόνα 2.20: Παράδειγμα αρχιτεκτονικής PHP.

(What is PHP| easytolearning, 2024)

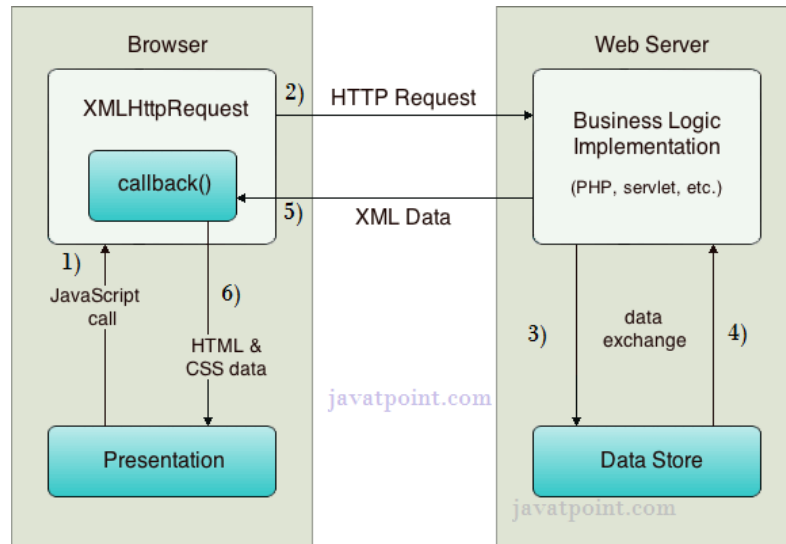
### 2.3.2.5 Asynchronous JavaScript and XML (AJAX)



Εικόνα 2.21: Λογότυπο AJAX.

(What is AJAX? | w3schools, 2024)

Το AJAX (Asynchronous JavaScript And XML) είναι ένα σύνολο τεχνικών προγραμματισμού το οποίο επιτρέπει την ανανέωση ενός τμήματος της σελίδας χωρίς την ανανέωση ολόκληρης της σελίδας. Οι διαδικτυακές εφαρμογές μπορούν να στέλνουν και να λαμβάνουν δεδομένα από έναν διακομιστή (server) με ασύγχρονο τρόπο χωρίς να επιδρούν στην υπάρχουσα σελίδα. Στις σύγχρονες εφαρμογές, χρησιμοποιείται κυρίως η μορφή δεδομένων JSON (JavaScript Object Notation) έναντι της XML λόγω των πλεονεκτημάτων που συνοδεύονται από τη JavaScript. (IBM, 2021)



Εικόνα 2.22: Αρχιτεκτονική AJAX.

(How AJAX works? | javatpoint, 2024)

Το AJAX χρησιμοποιεί το αντικείμενο XMLHttpRequest για να επικοινωνήσει με τον διακομιστή. Τα βήματα που ακολουθούνται είναι τα εξής:

1. Ο χρήστης στέλνει ένα αίτημα από το UI (User Interface) και μία κλήση μέσω JavaScript στέλνεται στο αντικείμενο XMLHttpRequest.
2. Το αίτημα HTTP στέλνεται στο διακομιστή από το αντικείμενο XMLHttpRequest.
3. Ο διακομιστής αλληλεπιδρά με τη βάση δεδομένων χρησιμοποιώντας μια σειρά τεχνολογιών όπως PHP, Servlet κ.λπ.
4. Γίνεται ανάκτηση των δεδομένων
5. Ο διακομιστής στέλνει είτε δεδομένα XML είτε δεδομένα JSON στη συνάρτηση ανάκλησης (callback) του XMLHttpRequest.
6. Τα δεδομένα HTML και CSS εμφανίζονται στο πρόγραμμα περιήγησης.

### 2.3.2.6 MySQL



Εικόνα 2.23: Λογότυπο MySQL.

(MySQL | seeklogo, 2024)

Η MySQL (Michael Widenius Structured Query Language) είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων το οποίο χρησιμοποιείται για τη διαχείριση δεδομένων σε μία βάση δεδομένων. Είναι αρκετά δημοφιλές σύστημα διαχείρισης βάσεων δεδομένων και χρησιμοποιείται ευρέως στις εφαρμογές λογισμικού. (Gurta S. , 2021)

Τα δεδομένα στη MySQL αποθηκεύονται σε πίνακες, οι οποίοι αποτελούνται από γραμμές και στήλες. Η διαχείρισή τους γίνεται μέσω των SQL εντολών με την πιο σημαντική κατηγορία εντολών να είναι τα ερωτήματα (queries) με τα οποία γίνεται ανάκτηση ή ενημέρωση των δεδομένων.

Οι κύριες εντολές της MySQL φαίνονται στον παρακάτω πίνακα:

Εντολή	Περιγραφή
SELECT	Εξαγωγή δεδομένων
UPDATE	Ανανέωση δεδομένων
DELETE	Διαγραφή δεδομένων
INSERT INTO	Εισαγωγή νέων δεδομένων
CREATE DATABASE	Δημιουργία νέας βάσης δεδομένων
ALTER DATABASE	Τροποποίηση βάσης δεδομένων
CREATE TABLE	Δημιουργία νέου πίνακα
ALTER TABLE	Τροποποίηση πίνακα
DROP TABLE	Διαγραφή πίνακα
CREATE INDEX	Δημιουργία κλειδιού αναζήτησης
DROP INDEX	Διαγραφή κλειδιού αναζήτησης

Πίνακας 2.2: Βασικές εντολές SQL.

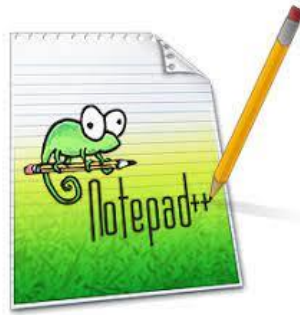
## 2.4 Εργαλεία και πλατφόρμες

### 2.4.1 Εισαγωγή

Στη συνέχεια θα γίνει μια αναφορά στα διάφορα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν στα πλαίσια της διπλωματικής. Καθώς όλα τα λογισμικά που αξιοποιήθηκαν ήταν ανοιχτού κώδικα, η παρούσα εφαρμογή χαρακτηρίζεται από χαμηλό κόστος υλοποίησης, χωρίς ωστόσο να υπολείπεται σε χρήσιμα και ενδιαφέροντα χαρακτηριστικά.

Τα προγράμματα που θα παρουσιαστούν παρακάτω αξιοποιήθηκαν για τη σχεδίαση των απαραίτητων κυκλωμάτων, τη συγγραφή του κώδικα της ιστοσελίδας, την παρουσίαση των αποτελεσμάτων σε κατάλληλα διαγράμματα καθώς επίσης και την αποστολή ειδοποιήσεων στον χρήστη.

## 2.4.2 Notepad++



Εικόνα 2.24: Λογότυπο Notepad++.

(How to Use Notepad++ | arrow, 2020)

Το Notepad++ είναι ένα open-source πρόγραμμα επεξεργασίας κειμένου και πηγαιού κώδικα για το λειτουργικό σύστημα Microsoft Windows. Είναι από τα πιο δημοφιλή προγράμματα επεξεργασίας κειμένου λόγω του ότι είναι ελαφρύ, γρήγορο και εύκολα επεκτάσιμο. Παρέχει syntax highlighting χρωματίζοντας τον κώδικα ανάλογα με τη δομή της κάθε γλώσσας προγραμματισμού που χρησιμοποιείται ενώ το περιβάλλον είναι αρκετά απλό και κατανοητό. Στα πλαίσια της εργασίας χρησιμοποιήθηκε για την συγγραφή των scripts σε PHP, JavaScript, HTML, CSS, AJAX και MySQL. (notepad-plus-plus.org, 2020)

## 2.4.3 XAMPP



Εικόνα 2.25: Λογότυπο XAMPP.

(XAMPP Tutorial | javatpoint, 2024)

Το XAMPP είναι πακέτο προγραμμάτων λογισμικού ανοιχτού κώδικα που χρησιμοποιείται για την σχεδίαση και ανάπτυξη ιστοσελίδων δοκιμαστικά χωρίς την σύνδεση στο διαδίκτυο. Περιέχει τον εξυπηρετητή ιστοσελίδων Apache HTTP, την βάση δεδομένων MySQL και έναν μεταγλωττιστή για κώδικα γραμμένο σε PHP και Perl. Το όνομα XAMPP προκύπτει όπως φαίνεται παρακάτω, υποδηλώνοντας τις τεχνολογίες που περιέχονται στο λογισμικό:

- X (“cross platform”, λογισμικό ανεξαρτήτως πλατφόρμας)
- Apache HTTP Server

- MySQL
- PHP
- Perl

Βασίζεται στον Apache HTTP Server ο οποίος αποτελεί το ενδιάμεσο στάδιο μεταξύ της ιστοσελίδας που θέλει να ανοίξει ο χρήστης και του προγράμματος πλοήγησης (browser). Ουσιαστικά, κάθε φορά που ο χρήστης επισκέπτεται έναν ιστότοπο, το πρόγραμμα πλοήγησης επικοινωνεί με έναν διακομιστή (εδώ: Apache) μέσω του πρωτοκόλλου HTTP και εκείνος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης.

Με το XAMPP ο προγραμματιστής μπορεί να καθορίσει τη μορφή και τις λειτουργίες της ιστοσελίδας πριν την θέσει σε εφαρμογή, δηλαδή τοπικά στον υπολογιστή χωρίς να είναι απαραίτητη η σύνδεση στο διαδίκτυο. Υποστηρίζει PHP και MySQL προσφέροντας τη δημιουργία και ανάπτυξη βάσεων δεδομένων καθώς και τον σχεδιασμό δυναμικών χαρακτηριστικών της ιστοσελίδας. (K, 2021)

#### 2.4.4 Fritzing



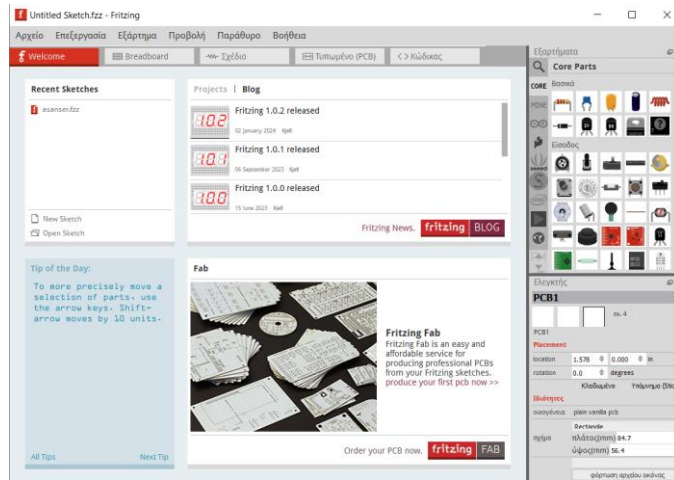
*Εικόνα 2.26: Λογότυπο fritzing.*

(Fritzing, all you need to know | automatismosmundo, 2024)

Το Fritzing είναι ένα open-source πρόγραμμα σχεδίασης ηλεκτρονικών κυκλωμάτων. Δίνει τη δυνατότητα στο χρήστη να σχεδιάσει πρωτότυπα κυκλώματα με βάση την πλατφόρμα Arduino και στη συνέχεια παρέχει τόσο τη σχηματική περιγραφή του κυκλώματος όσο και το σχέδιο σε PCB (Printed Circuit Board). Αποτελεί έναν εύχρηστο και δωρεάν τρόπο σχεδίασης κυκλωμάτων τυπωμένων σε πλακέτα και γι' αυτό θεωρείται ιδανική επιλογή για ερασιτέχνες.

Ακόμη, διαθέτει μια μεγάλη συλλογή ηλεκτρονικών στοιχείων τα οποία μπορεί να χρησιμοποιήσει ο χρήστης, καθώς το πρόγραμμα εμπλουτίζεται με ελεύθερες βιβλιοθήκες που παρέχουν διάφορες εταιρείες κατασκευής ηλεκτρονικών. Παράλληλα, ο χρήστης μπορεί και ο ίδιος να σχεδιάσει κάποιον αισθητήρα ή κάποιο εξάρτημα. (en.wikipedia.org, n.d.)





Εικόνα 2.27: Περιβάλλον fritzing.  
(Fritzing electronics| fritzingonline, 2024)

## 2.4.5 Google Charts



Εικόνα 2.28: Λογότυπο Google Charts.  
(Google Chart Tools Review| financesonline, 2024)

Το Google Charts είναι μία διαδικτυακή υπηρεσία που παρέχει δωρεάν βιβλιοθήκες βασισμένες σε JavaScript για την οπτικοποίηση δεδομένων. Υποστηρίζει διάφορα διαγράμματα όπως ιστόγραμμα, διάγραμμα «πίτα», διάγραμμα διασποράς και άλλα ενώ οι πληροφορίες μπορούν να εισαχθούν απευθείας από μία βάση δεδομένων. Προσφέρει έναν εύκολο τρόπο οπτικοποίησης δεδομένων αποτελώντας χρήσιμο εργαλείο κατά την ενασχόληση με την επιστήμη δεδομένων. (developers.google, 2019)

## 2.4.6 ThingSpeak



Εικόνα 2.29: Λογότυπο ThingSpeak.  
(Thingspeak | iotone, 2024)

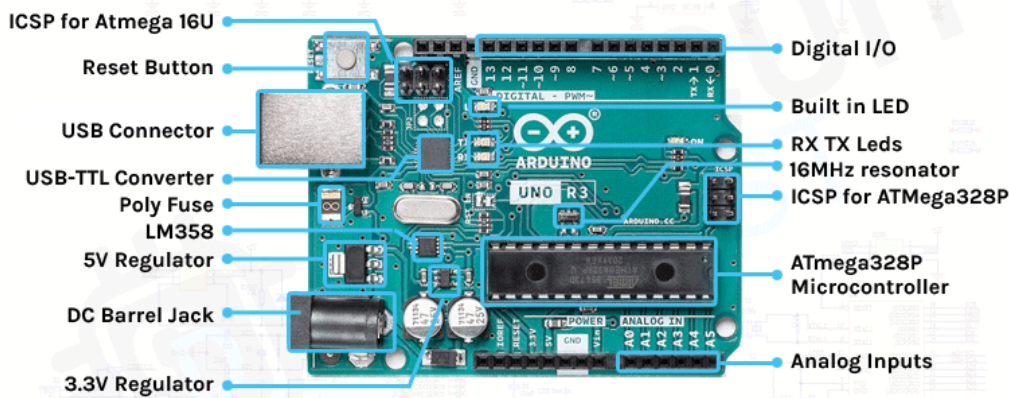
Το ThingSpeak είναι άλλο ένα open-source λογισμικό το οποίο επιτρέπει στον χρήστη την επικοινωνία με συσκευές-μέλη ενός Internet of Things (IoT) οικοσυστήματος. Δίνει τη δυνατότητα για ανάκτηση, οπτικοποίηση και επεξεργασία δεδομένων σε πραγματικό χρόνο στο cloud. Μπορεί να λάβει τα δεδομένα από τις συσκευές και είτε να τα προβάλλει σε πραγματικό

χρόνο είτε να στείλει ειδοποιήσεις. Επίσης, υποστηρίζει το λογισμικό MATLAB της MathWorks δίνοντας τη δυνατότητα για ανάλυση και οπτικοποίηση δεδομένων με τη χρήση των μεθόδων του MATLAB. (thingspeak, n.d.)

## 3 Υλικό μέρος συστήματος

### 3.1 Arduino

#### 3.1.1 Γενικές πληροφορίες



Εικόνα 3.1: Τα κύρια εξαρτήματα της πλακέτας Arduino.

(Arduino | Arduino, 2024)

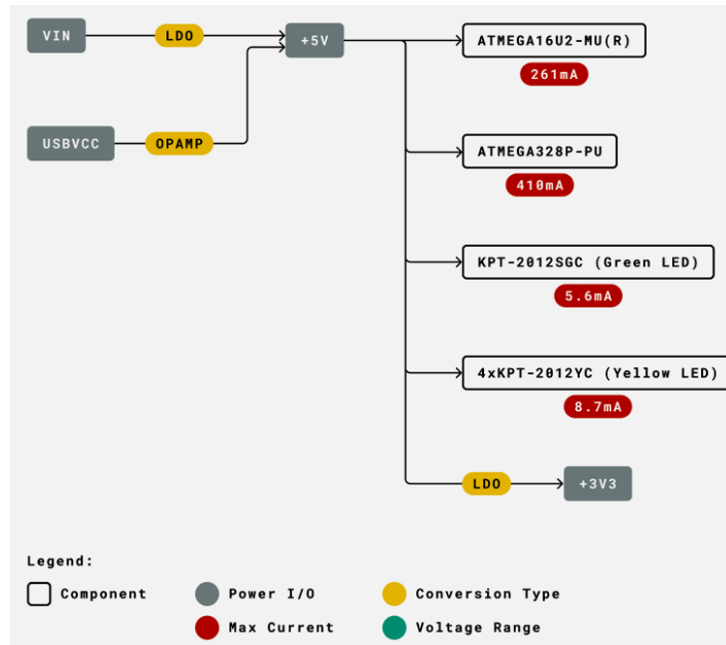
Όπως έχει αναφερθεί, ο κώδικας ο οποίος υλοποιεί τις λειτουργίες της συλλογής δεδομένων από αισθητήρες και η διαβίβασή τους στην βάση δεδομένων, εκτελείται εξ' ολοκλήρου στο Arduino. Το Arduino αποτελείται από τον κύριο μικροελεγκτή και από άλλα συμπληρωματικά κυκλώματα απαραίτητα για την κατάλληλη τροφοδοσία του και συνδεσιμότητά του τόσο με τον υπολογιστή όσο και με τα περιφερειακά που χρησιμοποιούνται στην εκάστοτε εφαρμογή.

Πιο αναλυτικά, το τσιπάκι που παίζει το ρόλο του μικροελεγκτή ενός Arduino UNO είναι το ATmega328P. Εκεί εκτελούνται όλες οι διεργασίες οι οποίες απαιτούνται κατά την εκτέλεση του προγράμματος. Κοντά του βρίσκεται ένας ταλαντωτής συχνότητας 16MHz ο οποίος απαιτείται για την λειτουργία του ρολογιού (clock) που βρίσκεται μέσα στο ATmega328P. Ακόμη, υπάρχει και ένας ICSP (In-Circuit Serial Programming) connector μέσω του οποίου γίνεται η φόρτωση του bootloader. Ο bootloader εκτελείται πριν το κύριο πρόγραμμα και είναι υπεύθυνος για την αρχικοποίηση των καταχωρητών του μικροελεγκτή, ώστε να είναι σε θέση να φορτωθεί το κύριο πρόγραμμα. (Arduino, 2024)

##### 3.1.1.1 Τροφοδοσία

Η τροφοδοσία του Arduino μπορεί να γίνει με δύο τρόπους. Ο πρώτος είναι ο USB – B socket connector ο οποίος χρησιμοποιείται τόσο για την τροφοδοσία της πλακέτας μέσω του υπολογιστή όσο και για την επικοινωνία του υπολογιστή με την πλακέτα μέσω USB αλλά και τον προγραμματισμό της. Ο δεύτερος τρόπος τροφοδοσίας είναι το DC Barrel Jack μέσω του οποίου δίνεται η δυνατότητα για παροχή εξωτερικής τροφοδοσίας 9V-12V.

Μετά την παροχή τάσης στην πλακέτα ακολουθεί το κύκλωμα τροφοδοσίας το οποίο είναι υπεύθυνο για τον υποβιβασμό της τάσης σε 5V και 3.3V η οποία είναι και η τάση λειτουργίας του μικροελεγκτή. Στην εικόνα παρακάτω φαίνεται ολοκληρωμένη η ροή της ισχύος από την είσοδο μέχρι τα εκάστοτε αντικείμενα μαζί με την μέγιστη ανοχή τους σε ένταση ρεύματος. (circuitdigest, 2022)



Εικόνα 3.2: Ροή ισχύος σε ένα Arduino UNO.

(Everything you need to know about the Arduino Hardware | Circuit Digest, 2022)

### 3.1.1.2 Υποδοχές – Ακίδες (Headers)

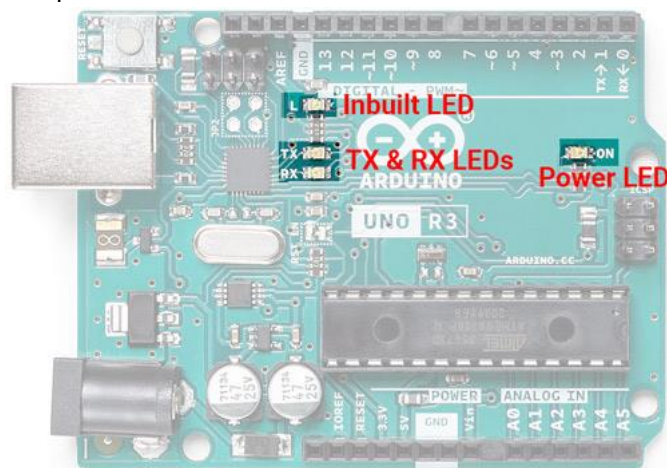
Οι υποδοχές στο Arduino είναι τα σημεία στα οποία συνδέονται τα καλώδια ώστε να είναι εφικτή η κατασκευή ενός κυκλώματος. Αυτές παρουσιάζονται στον παρακάτω πίνακα συνοδευόμενες από μια μικρή περιγραφή.

Header	Description
GND	Γείωση. Μπορεί να χρησιμοποιηθεί ως γείωση ολόκληρου του κυκλώματος.
5V & 3.3V	Τροφοδοσία 5V και 3.3V. Από αυτές τις ακίδες παρέχεται τάση η οποία μπορεί να χρησιμοποιηθεί για την τροφοδοσία του κυκλώματος.
Αναλογικές εισοδοι	Analog In (pins A0-A5). Μέσω αυτών των ακίδων μπορεί να γίνει ανάγνωση ενός αναλογικού σήματος από τον μικροελεγκτή, το

	οποίο στη συνέχεια μπορεί να μετατραπεί σε ψηφιακό.
Ψηφιακές Είσοδοι/Εξοδοι (E/E)	Digital IN/OUT (pins 0-13). Αυτές οι ακίδες μπορούν να χρησιμοποιηθούν τόσο για είσοδο όσο και για έξοδο ψηφιακών σημάτων (λογικό 0 ή 1)
PWM (~)	Ακίδες (3, 5, 6, 9, 10 και 11 στο UNO). Αυτές οι ακίδες χρησιμοποιούνται τόσο σαν ψηφιακές E/E αλλά και για διαμόρφωση πλάτους παλμών (Pulse Width Modulation).
AREF	Υποστηρίζει αναλογική αναφορά. Χρησιμοποιείται για τη ρύθμιση μιας εξωτερικής αναφοράς (0-5V) ως το ανώτερο όριο για τους ακροδέκτες αναλογικής εισόδου.
Κουμπί Reset	Με τη χρήση του γίνεται επανεκκίνηση του προγράμματος που «τρέχει» στο Arduino

Πίνακας 3.1: Περιγραφή ακίδων πλακέτας Arduino Uno (docs.arduino.cc, 2022)

### 3.1.1.3 LEDs κατάστασης



Εικόνα 3.3: Τοποθεσία των LEDs σε ένα Arduino UNO.

(Everything you need to know about the Arduino Hardware | Circuit Digest, 2022)

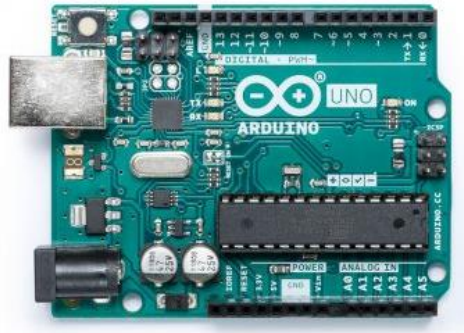
Υπάρχουν 4 LED στο Arduino. Ένα χρησιμοποιείται ως ένδειξη τροφοδοσίας και δύο (RX-TX) χρησιμοποιούνται για ένδειξη μεταφοράς δεδομένων από και προς το Arduino (όπως γίνεται κατά τη φόρτωση ενός νέου προγράμματος). Τέλος υπάρχει ένα LED συνδεδεμένο με την ψηφιακή ακίδα 13 η οποία μπορεί να χρησιμοποιηθεί από τον χρήστη σαν ένδειξη ως μέρος κάποιας λειτουργίας του προγράμματος.

### 3.1.2 Εκδόσεις Arduino

Υπάρχουν διάφορες εκδόσεις πλακετών Arduino με βάση τις λειτουργίες που προσφέρουν. Οι πιο διαδεδομένοι τύποι είναι οι Uno, Mega και Nano, ενώ υπάρχουν επίσης πλακέτες με έξτρα λειτουργικότητες όπως η υποστήριξη πρωτοκόλλων όπως Bluetooth, Ethernet και WiFi, η υποστήριξη κάρτας μνήμης για αποθήκευση δεδομένων ή ακόμη και η ενσωμάτωση FPGA chip για ισχυρή υπολογιστική δύναμη.

Παρακάτω θα αναφερθούν κάποια χρήσιμα χαρακτηριστικά για τους τρεις βασικούς τύπους πλακετών Arduino.

#### 3.1.2.1 *Arduino Uno*



Εικόνα 3.4: *Arduino UNO R3*

(Arduino Uno Rev3 | Grobotronics, 2024)

Το Arduino Uno αποτελεί την βασική πλακέτα Arduino και προτείνεται για να ξεκινήσει κάποιος την εκμάθηση προγραμματισμού και την εξοικείωση με τα ηλεκτρονικά. Είναι η πιο διαδεδομένη πλακέτα, ενώ είναι συμβατή με πλήθος αισθητήρων και επεκτάσεων.

Αναλυτικά, η πλακέτα διαθέτει 14 ψηφιακές εισόδους/εξόδους με τις 6 από αυτές να μπορούν να χρησιμοποιηθούν ως PWM έξοδοι, 6 αναλογικές εισόδους, 1 είσοδο τροφοδοσίας, 1 θύρα USB τύπου B, 1 υποδοχή ICSP και ένα κουμπί Reset. Η μνήμη Flash στην οποία αποθηκεύονται τα προγράμματα είναι χωρητικότητας 32KB καθιστώντας την ικανή να δεχτεί τα περισσότερα απλά προγράμματα.

#### 3.1.2.2 *Arduino Mega*



Εικόνα 3.5: *Arduino Mega 2560 R3*

(Arduino Mega | Grobotronics, 2024)



Το Arduino Mega είναι πιο εξελιγμένη πλακέτα της τεχνολογίας Arduino και προτείνεται για περίπλοκες κατασκευές που απαιτούν μεγαλύτερη μνήμη και περισσότερες εισόδους/εξόδους. Βασίζεται στην ίδια αρχιτεκτονική με την πλακέτα Uno με βασική διαφορά την χωρητικότητα της μνήμης και το πλήθος των εισόδων/εξόδων.

Η πλακέτα βασίζεται στον μικροελεγκτή ATmega2560 και διαθέτει 54 ψηφιακές εισόδους/εξόδους, με τις 15 να μπορούν να χρησιμοποιηθούν για PWM, 16 αναλογικές εισόδους, 4 σειριακές θύρες ενώ όλα τα υπόλοιπα χαρακτηριστικά είναι ίδια με αυτά της πλακέτας Uno. Η χωρητικότητα της μνήμης Flash είναι στα 256KB έναντι των 32KB του Arduino Uno. (Gupta P. , 2024)

### 3.1.2.3 *Arduino Nano*



*Εικόνα 3.6: Arduino Nano*

(Arduino Nano | Grobotronics, 2024)

Το Arduino Nano αποτελεί την πιο μικρή πλακέτα της τεχνολογίας Arduino καθώς είναι μία μικρή έκδοχή του Arduino Uno. Είναι ιδανική για απλές εφαρμογές στις οποίες η χωροταξία της κατασκευής απαιτείται να είναι σχετικά μικρή.

Το πλήθος εισόδων/εξόδων είναι ίδιο με αυτό του Uno, δηλαδή υπάρχουν 14 ψηφιακές με τις 6 να υποστηρίζουν PWM, 6 αναλογικές εισοδοί, ενώ η διαφορά είναι στην τροφοδοσία. Το Arduino Nano διαθέτει μία θύρα Mini-B USB για τον προγραμματισμό και την τροφοδοσία, ενώ δεν διαθέτει ξεχωριστή υποδοχή τροφοδοσίας τύπου DC Barrel Jack. Τέλος, η χωρητικότητα μνήμης του Nano είναι κι αυτή στα 32KB.

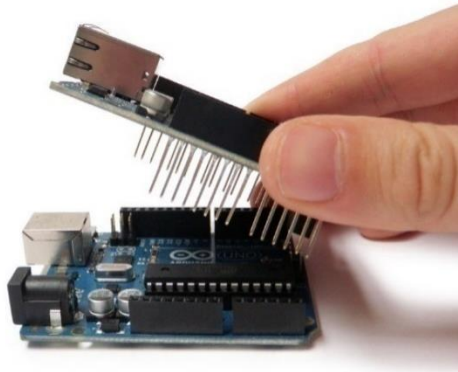
## 3.1.3 **Arduino shields**

### 3.1.3.1 *Εισαγωγή*

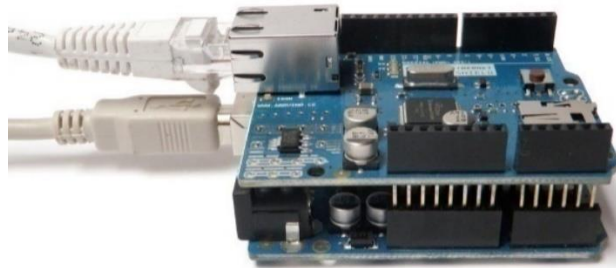
Τα Arduino shields είναι πλακέτες κυκλωμάτων οι οποίες εκτελούν μία συγκεκριμένη λειτουργία και μπορούν να συνδεθούν πάνω σε κανονικές πλακέτες Arduino για να τους επεκτείνουν τη λειτουργικότητα. Υπάρχουν διάφορα shields τα οποία παρέχουν ένα πλήθος δυνατοτήτων όπως σύνδεση στο διαδίκτυο μέσω Ethernet (Ethernet shield), έλεγχος κινητήρων (Motor shield), ενσωμάτωση Inertial Measurement Unit (IMU shield) και άλλα.

Στην παρούσα διπλωματική θα χρησιμοποιηθεί το Arduino Ethernet shield με σκοπό την ενσύρματη σύνδεση στο διαδίκτυο για την λειτουργία της ιστοσελίδας. (arduinogetstarted, 2024)

### 3.1.3.2 Ethernet shield



Εικόνα 3.7: Σύνδεση shield με Arduino.  
(Arduino ethernet shield | Arduinobots, 2015)



Εικόνα 3.8: Σύνδεση Ethernet shield με το διαδίκτυο μέσω RJ45 καλωδίου.  
(Interfacing ethernet shield with arduino | microcontrollerslab, 2021)

Το Ethernet shield βασίζεται στο Wiznet W5100 chip το οποίο προσφέρει ενσύρματη σύνδεση στο διαδίκτυο μέσω Ethernet υποστηρίζοντας ταχύτητες 10Mbps και 100Mbps. Η χρήση του γίνεται αφενός συνδέοντας την πλακέτα πάνω στο Arduino όπως φαίνεται στην εικόνα και αφετέρου ενσωματώνοντας στον κώδικα την βιβλιοθήκη *Ethernet.h* του Arduino.

Ακόμη, στο Ethernet shield υπάρχει και μία θύρα για κάρτα SD η οποία μπορεί να χρησιμοποιηθεί για την αποθήκευση αρχείων. Η πρόσβαση στη θύρα αυτή επιτυγχάνεται μέσω της βιβλιοθήκης *SD.h*.

Παρακάτω παρατίθεται το τμήμα κώδικα στο οποίο αρχικοποιούνται οι απαραίτητες δομές δεδομένων για την επικοινωνία με την SD κάρτα και την σύνδεση στο διαδίκτυο.

```
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>

EthernetClient client;
void setup() {
  SD.begin(4);
  Ethernet.init(10);
  Serial.println("Initialize Ethernet with DHCP:");
```



```

if (Ethernet.begin(mac) == 0) {
  Serial.println("Failed to configure Ethernet using DHCP");
  // Check for Ethernet hardware present
  if (Ethernet.hardwareStatus() == EthernetNoHardware) {
    Serial.println("Ethernet shield was not found.  Sorry,
can't run without hardware.  :(");
    // do nothing, no point running without Ethernet hardware
    while (true) delay(1);
  }
  if (Ethernet.linkStatus() == LinkOFF) {
    Serial.println("Ethernet cable is not connected.");
  }
  // try to configure using IP address instead of DHCP:
  Ethernet.begin(mac, ip, serv );
} else {
  Serial.print("DHCP assigned IP");
  Serial.println(Ethernet.localIP());
}
}

```

*Πίνακας 3.2: Κώδικας αρχικοποίησης επικοινωνία με το W5100 shield*

Η επικοινωνία του Arduino τόσο με το W5100 chip όσο και με την SD κάρτα γίνεται μέσω του πρωτοκόλλου SPI και συγκεκριμένα από τις ψηφιακές εισόδους/εξόδους 11, 12 και 13 στο Uno και 50, 51 και 52 στο Mega. Και στις δύο πλακέτες ο ακροδέκτης 10 χρησιμοποιείται για την επιλογή του W5100 και ο ακροδέκτης 4 για την SD κάρτα, ενώ οι δύο ακροδέκτες δεν μπορούν να χρησιμοποιηθούν πλέον ως γενικές εισοδοί/έξοδοι. Ακόμη, επειδή το W5100 και η SD κάρτα μοιράζονται τον ίδιο διάυλο επικοινωνίας στο SPI, μόνο ένα από τα δύο μπορεί να χρησιμοποιηθεί κάθε φορά. Αυτό σημαίνει πως για να χρησιμοποιηθούν και τα δύο θα πρέπει να δοθεί ιδιαίτερη έμφαση στη σειρά με την οποία θα εκτελεστούν οι εντολές στο πρόγραμμα, ώστε να μην απαιτείται επικοινωνία και με την SD κάρτα και με το W5100 ταυτόχρονα. (docs.arduino.cc, 2022)

Τέλος, το Ethernet shield διαθέτει ένα πλήθος LED τα οποία έχουν σκοπό να ενημερώσουν το χρήστη για διάφορες καταστάσεις. Αυτά παρουσιάζονται στον παρακάτω πίνακα:

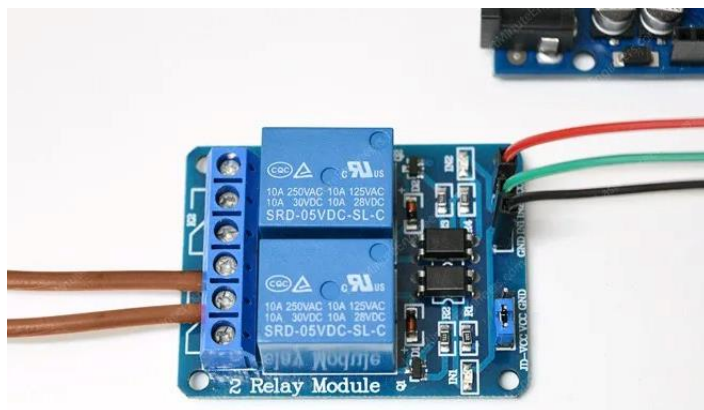
LED	Περιγραφή
PWR	Ένδειξη παροχής ισχύος στην πλακέτα και στο shield
LINK	Ένδειξη ύπαρξης δικτύου. Αναβοσβήνει κατά την μεταφορά δεδομένων.
FULLD	Ένδειξη ύπαρξης αμφίδρομης επικοινωνίας
100M	Ένδειξη ύπαρξης σύνδεσης 100Mbps
RX	Αναβοσβήνει όταν το shield λαμβάνει δεδομένα
TX	Αναβοσβήνει όταν το shield μεταδίδει δεδομένα
COLL	Αναβοσβήνει όταν εντοπιστεί σύγκρουση πακέτων στο δίκτυο

Πίνακας 3.3: Περιγραφή LEDs κατάστασης στο Ethernet shield. (docs.arduino.cc, 2022)

## 3.2 Αισθητήρες και συσκευές δράσης

Στα πλαίσια της παρούσας διπλωματικής χρησιμοποιήθηκαν διάφοροι αισθητήρες και ενεργοποιητές με σκοπό την υλοποίηση των διάφορων σεναρίων αυτοματισμού που θα περιλαμβάνει το «έξυπνο» θερμοκήπιο.

### 3.2.1 Πλακέτα ρελέ 2 καναλιών



Εικόνα 3.9: : Πλακέτα ρελέ 2 καναλιών.

(Interface Two Channel Relay Module with Arduino | lastminuteengineers, 2024)

Αυτή η συσκευή συνδυάζει 2 ρελέ σε μία πλακέτα, το καθένα με ξεχωριστή είσοδο και έξοδο. Το ρελέ περιέχει στον πυρήνα του ένα πηνίο το οποίο όταν διαρρέεται από ρεύμα λειτουργεί ως μαγνήτης. Αυτό το χαρακτηριστικό δίνει τη δυνατότητα τροφοδοσίας του πηνίου με ένα σχετικά μικρό ρεύμα έτσι ώστε να μαγνητιστεί η επαφή και να κλείσει ο διακόπτης ελέγχου ενός κυκλώματος το οποίο από τη φύση του χρειάζεται πολύ μεγαλύτερο ρεύμα για να λειτουργήσει.

Συνήθως η χρήση των ρελέ σε ηλεκτρονικά κυκλώματα γίνεται με σκοπό να τροφοδοτήσουν λάμπες, ανεμιστήρες ή άλλα στοιχεία τα οποία απαιτούν πολύ υψηλότερη ισχύ από αυτή που βγάζουν οι ακίδες 5V που παρέχει το Arduino.

Το συγκεκριμένο module περιέχει και μεθόδους ηλεκτρικής απομόνωσης μεταξύ του κυκλώματος ελέγχου και της εξόδου του ρελέ ώστε να προστατεύει τις συνδεδεμένες συσκευές σε περίπτωση κάποιου βραχυκυκλώματος. (lastminuteengineers, Interface Two Channel Relay Module with Arduino, 2024)

Για τον έλεγχο του ρελέ επιλέχθηκε η ακίδα 47, η οποία χρησιμοποιείται ως ψηφιακή.

Οι ακίδες της πλακέτας και η λειτουργία της κάθε μίας παρουσιάζονται στην παρακάτω πίνακα:

<b>Ακίδα</b>	<b>Λειτουργία</b>
1 (V <sub>CC</sub> )	Παρέχει τροφοδοσία στην πλακέτα. Συνδέεται στην έξοδο 5V του Arduino.
2 (IN <sub>2</sub> )	Ακίδες που ελέγχουν την λειτουργία των ρελέ. Συνδέονται στις αναλογικές ακίδες του Arduino.
3 (IN <sub>1</sub> )	
4 (GND)	Γείωση της πλακέτας. Συνδέεται στην ακίδα GND του Arduino.
5 (GND)	
6 (V <sub>CC</sub> )	Παρέχει τροφοδοσία στην πλακέτα. Συνδέεται στην έξοδο 5V του Arduino.
7 (JD-V <sub>CC</sub> )	Η ακίδα αυτή παρέχει ενέργεια στον ηλεκτρομαγνήτη των ρελέ. Είναι βραχυκυκλωμένο με τις ακίδες V <sub>cc</sub>
10, 13 NO	Η ακίδα αυτή είναι συνδεδεμένη με ακίδα COM όσο το ρελέ δεν είναι οπλισμένο.
9, 12 COM	Η ακίδα αυτή συνδέεται στην προς έλεγχο συσκευή

8, 11 NC	Η ακίδα αυτή δεν είναι συνδεδεμένη με ακίδα COM όσο το ρελέ δεν είναι οπλισμένο.
----------	--

Πίνακας 3.4: Χαρακτηριστικά πλακέτας ρελέ δύο καναλιών.

### 3.2.2 Αισθητήρας κίνησης PIR



Εικόνα 3.10: Αισθητήρας κίνησης PIR.

(Mini Pyroelectric Pir Sensor Module Manual Motion Infrared Ir Detector| fruugoschweiz, 2024)

Ο αισθητήρας κίνησης PIR (Passive Infrared) είναι υπεύθυνος για την ανίχνευση κίνησης εντός του οπτικού του πεδίου. Χρησιμοποιείται στο σενάριο κλειδώματος/ξεκλειδώματος της εξωτερικής πόρτας το οποίο θα εξηγηθεί στο επόμενο κεφάλαιο.

Η αρχή λειτουργίας του είναι ένας πυροηλεκτρικός αισθητήρας ο οποίος εντοπίζει υπέρυθη ακτινοβολία η οποία εκπέμπεται από όλα τα θερμά σώματα. Ουσιαστικά, όταν ένα θερμό σώμα (π.χ. άνθρωπος ή ζώο) μπει στο οπτικό πεδίο του αισθητήρα, εκείνος εντοπίζει την υπέρυθη ακτινοβολία του σώματος και παράγει ένα σήμα. Αυτό ενισχύεται και μεταφέρεται στην έξοδο του αισθητήρα η οποία σηματοδοτεί και την ύπαρξη κίνησης. Μετά την ανίχνευση κίνησης, η έξοδος παραμένει ενεργή μέχρι το πέρας μιας σύντομης περιόδου καθυστέρησης (2 δευτερολέπτων) και εφόσον έχει σταματήσει να ανιχνεύεται κίνηση.

Στον παρακάτω πίνακα παρουσιάζονται κάποια χρήσιμα τεχνικά χαρακτηριστικά του αισθητήρα (randomnerdtutorials, Arduino with PIR Motion Sensor, n.d.):

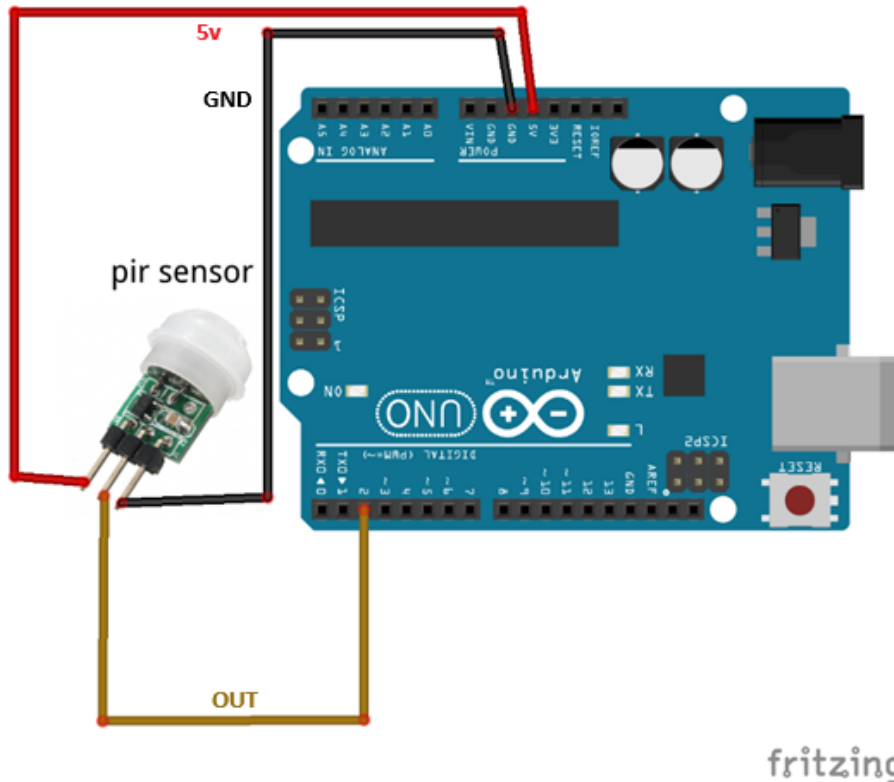
Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	DC 2.7V – 12V
Στατική κατανάλωση ενέργειας	<0.1mA
Χρόνος καθυστέρησης	2s
Χρόνος αποκλεισμού	2s
Trigger	Μπορεί να επαναληφθεί

Εύρος ανίχνευσης	≤ 100 ° γωνία κώνου, 3-5 μέτρα
Θερμοκρασία λειτουργίας	-20 °C έως +60 °C
Διαστάσεις PCB	10mm x 8mm
Συνολικό μέγεθος	Περίπου 12mm x 25mm

Πίνακας 3.5: Χαρακτηριστικά αισθητήρα κίνησης PIR.

(How Infrared motion detector components work| glolab, 2015)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του αισθητήρα κίνησης PIR με το Arduino Uno:



Εικόνα 3.11: Συνδεσμολογία αισθητήρα κίνησης PIR

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του αισθητήρα:

```
int led = 13; // the pin that the LED is attached to
int sensor = 2; // the pin that the sensor is attached to
int state = LOW; // by default, no motion detected
int val = 0; // variable to store the sensor status
(value)

void setup() {
  pinMode(led, OUTPUT); // initialize LED as an output
  pinMode(sensor, INPUT); // initialize sensor as an input
  Serial.begin(9600); // initialize serial
}

void loop(){
  val = digitalRead(sensor); // read sensor value
  if (val == HIGH) { // check if the sensor is HIGH
    digitalWrite(led, HIGH); // turn LED ON
    delay(100); // delay 100 milliseconds
  }
}
```

```

if (state == LOW) {
  Serial.println("Motion detected!");
  state = HIGH;      // update variable state to HIGH
}
}
else {
  digitalWrite(led, LOW); // turn LED OFF
  delay(200);           // delay 200 milliseconds

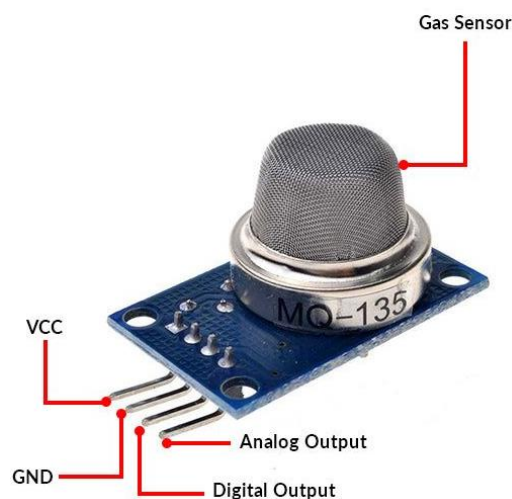
  if (state == HIGH){
    Serial.println("Motion stopped!");
    state = LOW;      // update variable state to LOW
  }
}
}
}

```

Πίνακας 3.6: Ενδεικτικός κώδικας για την διεπαφή με τον αισθητήρα PIR

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.3 Αισθητήρας αερίου MQ-135



Εικόνα 3.12: Αισθητήρας αερίου MQ-135.

(MQ-135 Gas Sensor module | robocraze, 2024)

Ο αισθητήρας αερίου MQ-135 χρησιμοποιείται για την ανίχνευση διάφορων επικίνδυνων ουσιών σε αέρια μορφή που μπορεί να υπάρχουν στην ατμόσφαιρα, δίνοντας μια ένδειξη της ποιότητας του αέρα στο χώρο. Βασίζεται σε έναν ημιαγωγό τύπου SnO<sub>2</sub> ο οποίος παρουσιάζει υψηλή αγωγιμότητα όταν εκτεθεί σε συγκεκριμένα αέρια. Αυτή η αλλαγή στην αγωγιμότητα μετατρέπεται σε ηλεκτρικό σήμα το οποίο μπορεί να μετρηθεί και να χρησιμοποιηθεί για την εξαγωγή συμπερασμάτων. Κάποια από τα αέρια τα οποία μπορεί να ενεργοποιήσουν τον αισθητήρα είναι NH<sub>3</sub>, NO<sub>x</sub>, αλκοόλη, βενζόλιο, καπνός, CO<sub>2</sub> και άλλα.

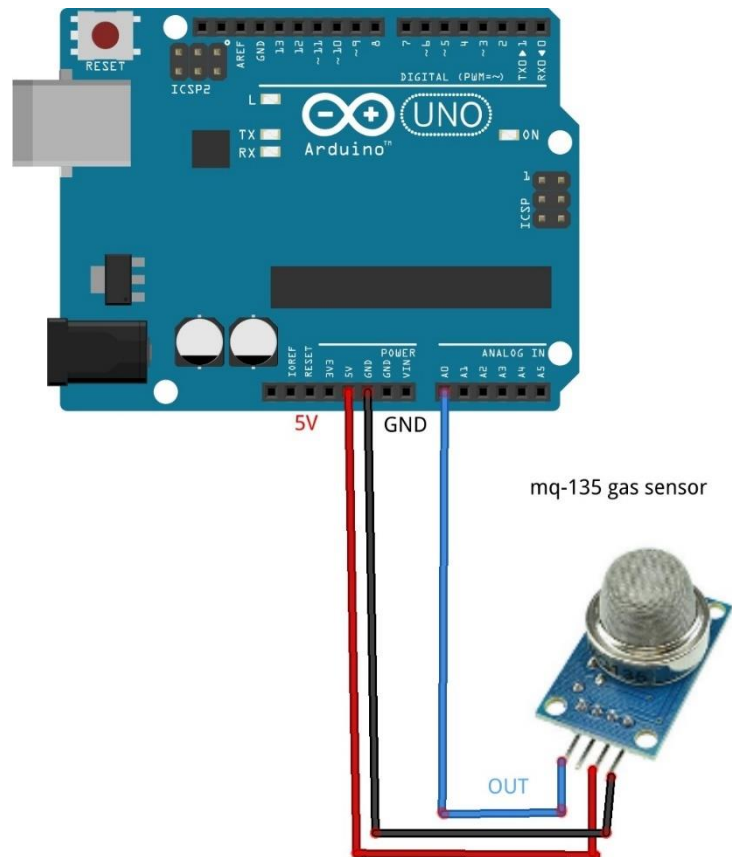
Στην παρούσα διπλωματική χρησιμοποιήθηκε για την μέτρηση της ποιότητας του αέρα στο εσωτερικό του θερμοκηπίου με σκοπό την ενεργοποίηση του συστήματος εξαερισμού του θερμοκηπίου. (waveshare, n.d.)

Κάποια τεχνικά χαρακτηριστικά παρουσιάζονται στον παρακάτω πίνακα:

Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	5V± 0.1
Μέγιστο ρεύμα λειτουργίας	2.5mA max
Πρότυπη ανίχνευση υγρασίας	65% /± 5%
Πρότυπη ανίχνευση θερμοκρασίας	0 - 20°C / ± 2°C
Κατανάλωση ισχύος	<800mw
Μέγεθος	20mm x 12mm x 10mm

Πίνακας 3.7: Χαρακτηριστικά αισθητήρα αερίου MQ-135  
(Gas Sensor Module| datasheets, 2008)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του αισθητήρα MQ-135 με το Arduino Uno:



Εικόνα 3.13: Συνδεσμολογία αισθητήρα MQ-135

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του αισθητήρα:

```
#include <LiquidCrystal.h>
int sensorValue;
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() { lcd.begin(16, 2);
               Serial.begin(9600); // sets the serial port to 9600
```

```

}

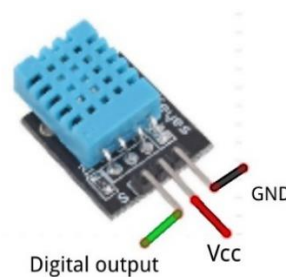
void loop(){sensorValue = analogRead(0); // read analog input pin 0
Serial.print("AirQua=");
Serial.print(sensorValue, DEC); // prints the value read
Serial.println(" PPM");
lcd.setCursor(0,0);
lcd.print("ArQ=");
lcd.print(sensorValue,DEC);
lcd.print(" PPM");
lcd.println(" ");
lcd.print(" ");
delay(100); // wait 100ms for next reading
}

```

Πίνακας 3.8: Ενδεικτικός κώδικας αισθητήρα αερίου MQ-135

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.4 Αισθητήρας θερμοκρασίας-υγρασίας DHT-11



Εικόνα 3.14: Αισθητήρας θερμοκρασίας DHT11.

(Αισθητήρας Υγρασίας & Θερμοκρασίας DHT11 | grobotronics, 2024)

Ο αισθητήρας DHT11 είναι μία συνήθης επιλογή όταν πρόκειται για τη μέτρηση θερμοκρασίας και υγρασίας ενός εσωτερικού χώρου. Για την λειτουργία του χρησιμοποιεί έναν χωρητικό αισθητήρα υγρασίας καθώς και ένα thermistor (μεταβλητή αντίσταση ανάλογα με τη θερμοκρασία) για τη μέτρηση υγρασίας και θερμοκρασίας αντίστοιχα, παράγοντας ένα ψηφιακό σήμα στην έξοδο. (elprocus, n.d.)

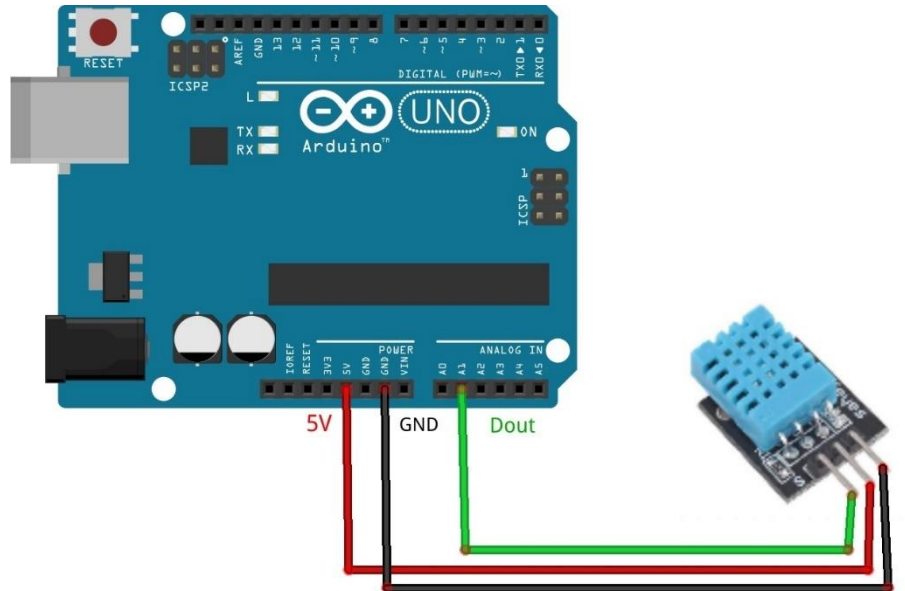
Κάποια τεχνικά χαρακτηριστικά παρουσιάζονται στον παρακάτω πίνακα:

Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	3V – 5V
Μέγιστο ρεύμα λειτουργίας	2.5mA max
Εύρος μέτρησης υγρασίας	20-80% / 5%
Εύρος μέτρησης θερμοκρασίας	0 - 50°C / ± 2°C
Ρυθμός δειγματοληψίας	1Hz
Μέγεθος	15.5mm x 12mm x 5.5mm



Πίνακας 3.9: Χαρακτηριστικά αισθητήρα θερμοκρασίας DHT11.  
(DHT11 Sensor Guide with Pinout| ElectronicWings, 2019)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του αισθητήρα DHT11 με το Arduino Uno:



Εικόνα 3.15: Συνδεσμολογία αισθητήρα DHT11

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του αισθητήρα:

```
//Libraries
#include <DHT.h>

//Constants
#define DHTPIN 2 // what pin we're connected to
#define DHTTYPE DHT11 // DHT 11 (AM2302)
// Initialize DHT sensor for normal 16mhz Arduino
DHT dht(DHTPIN, DHTTYPE);

//Variables
int chk;
float hum; //Stores humidity value
float temp; //Stores temperature value

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

void loop()
{
  //Read data and store it to variables hum and temp
  hum = dht.readHumidity();
  temp= dht.readTemperature();
  //Print temp and humidity values to serial monitor
  Serial.print("Humidity: ");
  Serial.print(hum);
  Serial.print(" %, Temp: ");
```

```
Serial.print(temp);  
Serial.println(" Celsius");  
delay(2000); //Delay 2 sec.  
}
```

Πίνακας 3.10: Ενδεικτικός κώδικας αισθητήρα θερμοκρασίας DHT11

Για τον παραπάνω κώδικα χρησιμοποιήθηκε η βιβλιοθήκη DHT11.

### 3.2.5 Αισθητήρας φλόγας KY-026



Εικόνα 3.16: Αισθητήρας φλόγας KY-026.

(KY-026 Flame Sensor Module | componentstree, 2024)

Ο αισθητήρας φλόγας KY-026 δίνει τη δυνατότητα ανίχνευσης υπέρυθρου φωτός το οποίο εκπέμπεται από τον περιβάλλοντα χώρο (π.χ. από κάποια εστία φωτιάς). Βασίζεται σε ένα φωτοτρανζίστορ το οποίο είναι ένα ηλεκτρονικό στοιχείο που έχει τη δυνατότητα να μετατρέπει την φωτεινή ακτινοβολία σε ρεύμα στη βάση ενός τυπικού διπολικού τρανζίστορ (BJT). Εκτός από τον αισθητήρα, η πλακέτα περιέχει και ένα ποτενσιόμετρο για την ρύθμιση της ευαισθησίας του αισθητήρα. Τέλος το πεδίο ανίχνευσης είναι στις 60 μοίρες περίπου. (electrothinks, n.d.)

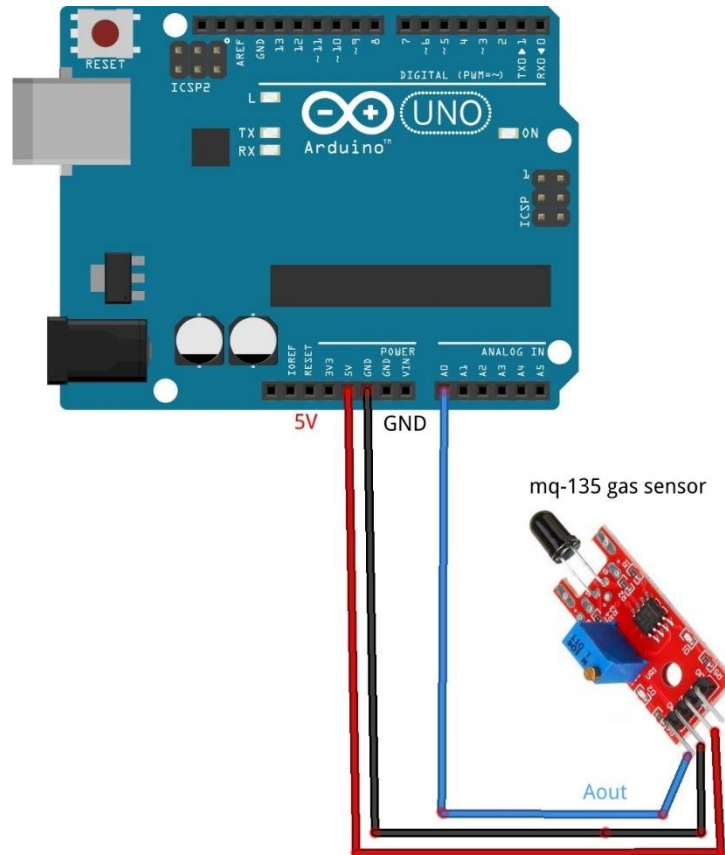
Κάποια τεχνικά χαρακτηριστικά παρουσιάζονται στον παρακάτω πίνακα:

Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	3.3V ~ 5.5V
Μέγιστο ρεύμα λειτουργίας	2.5mA max
Ανίχνευση υπέρυθρου μήκους κύματος	760nm ~ 1100nm
Γωνία ανίχνευσης αισθητήρα	60°
Ρυθμός δειγματοληψίας	1Hz
Μέγεθος	1.5cm x 3.6cm [0.6in x 1.4in]

Πίνακας 3.11: Χαρακτηριστικά αισθητήρα φλόγας KY-026.

(KY-026 Flame Sensor Module | ArduinoModuls, 2021)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του αισθητήρα KY-026 με το Arduino Uno:



Εικόνα 3.17: Συνδεσμολογία αισθητήρα KY-026

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του αισθητήρα:

```
int led = 13; // define the LED pin
int digitalPin = 2; // KY-026 digital interface
int analogPin = A0; // KY-026 analog interface
int digitalVal; // digital readings
int analogVal; //analog readings

void setup()
{
  pinMode(led, OUTPUT);
  pinMode(digitalPin, INPUT);
  //pinMode(analogPin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  // Read the digital interface
  digitalVal = digitalRead(digitalPin);
  if(digitalVal == HIGH) // if flame is detected
  {
    digitalWrite(led, HIGH); // turn ON Arduino's LED
  }
  else
  {
    digitalWrite(led, LOW); // turn OFF Arduino's LED
  }
}
```

```

}

// Read the analog interface
analogVal = analogRead(analogPin);
Serial.println(analogVal); // print analog value to serial

delay(100);
}

```

Πίνακας 3.12: Ενδεικτικός κώδικας αισθητήρα φλόγας KY-026

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.6 DS1302 Real time clock module



Εικόνα 3.18: Ρολόι πραγματικού χρόνου DS1302.

(DS1302 Real Time Clock-RTC-Module with Battery | sharvielectronics, 2024)

Το DS1302 module είναι ένα ρολόι πραγματικού χρόνου που παρέχει δευτερόλεπτα, λεπτά, ώρες, ημερομηνία, μέρα της εβδομάδας καθώς και χρονολογία μέχρι το έτος 2100. Για να επιστρέψει όλες αυτές τις πληροφορίες χρησιμοποιεί έναν κρύσταλλο 32kHz ο οποίος πραγματοποιεί την μέτρηση του χρόνου σε συνδυασμό με μία μικρή μπαταρία ώστε να διατηρεί την μέτρηση και μετά το πέρας της παροχής τροφοδοσίας. Η μπαταρία είναι τύπου CR2032 με τάση τροφοδοσίας 3V και ενέργεια 260mAh, μη επαναφορτιζόμενη. Είναι ικανή να διατηρήσει σε λειτουργία το ρολόι για περισσότερα από 10 χρόνια. (Dejan, n.d.)

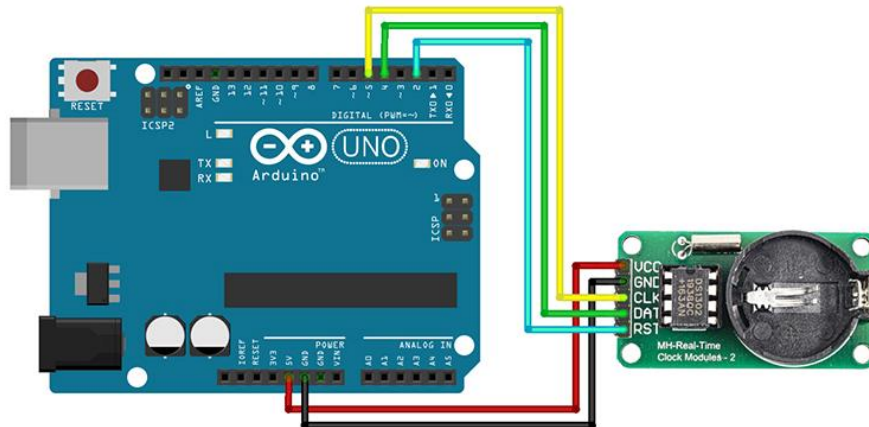
Κάποια τεχνικά χαρακτηριστικά παρουσιάζονται στον παρακάτω πίνακα:

Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	2V ~ 5.5V
Μέγιστο ρεύμα λειτουργίας	300 nA στα 2V
Αποθήκευση δεδομένων	31x8 RAM
Εύρος βιομηχανικών θερμοκρασιών	-°C έως +85 °C
Ρυθμός δειγματοληψίας	1Hz
Μέγεθος	5.22 mm

Πίνακας 3.13: Χαρακτηριστικά ρολογιού πραγματικού χρόνου DS1302 .

(DS1302 Datasheet - Dallas Semiconductor| Alldatasheet, 2024)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του DS1302 module με το Arduino Uno:



Εικόνα 3.19: Συνδεσμολογία DS1302 module

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του ρολογιού:

```
#include <ThreeWire.h>
#include <RtcDS1302.h>

ThreeWire myWire(4,5,2); // IO, SCLK, CE
RtcDS1302<ThreeWire> Rtc(myWire);

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

  Serial.print("compiled: ");
  Serial.print(__DATE__);
  Serial.println(__TIME__);

  Rtc.Begin();

  RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__);
  printDateTime(compiled);
  Serial.println();

  if (!Rtc.IsDateTimeValid())
  {
    // Common Causes:
    // 1) first time you ran and the device wasn't running yet
    // 2) the battery on the device is low or even missing

    Serial.println("RTC lost confidence in the DateTime!");
    Rtc.SetDateTime(compiled);
  }

  if (Rtc.GetIsWriteProtected())
  {
    Serial.println("RTC was write protected, enabling writing now");
    Rtc.SetIsWriteProtected(false);
  }

  if (!Rtc.GetIsRunning())
```

```

    {
        Serial.println("RTC was not actively running, starting now");
        Rtc.SetIsRunning(true);
    }

    RtcDateTime now = Rtc.GetDateTime();
    if (now < compiled)
    {
        Serial.println("RTC is older than compile time! (Updating
DateTime)");
        Rtc.SetDateTime(compiled);
    }
    else if (now > compiled)
    {
        Serial.println("RTC is newer than compile time. (this is
expected)");
    }
    else if (now == compiled)
    {
        Serial.println("RTC is the same as compile time! (not expected
but all is fine)");
    }
}

void loop ()
{
    RtcDateTime now = Rtc.GetDateTime();

    printDateTime(now);
    Serial.println();

    if (!now.IsValid())
    {
        // Common Causes:
        // 1) the battery on the device is low or even missing and the
power line was disconnected
        Serial.println("RTC lost confidence in the DateTime!");
    }

    delay(5000); // five seconds
}

#define countof(a) (sizeof(a) / sizeof(a[0]))

void printDateTime(const RtcDateTime& dt)
{
    char datestring[20];

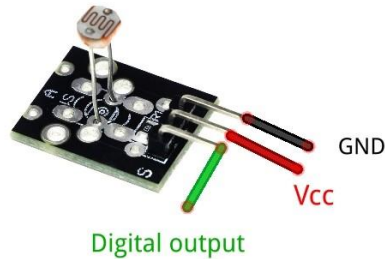
    snprintf_P(datestring,
countof(datestring),
PSTR("%02u/%02u/%04u %02u:%02u:%02u"),
dt.Month(),
dt.Day(),
dt.Year(),
dt.Hour(),
dt.Minute(),
dt.Second());
    Serial.print(datestring);
}

```

Πίνακας 3.14: Ενδεικτικός κώδικας clock module DS1302

Για τον παραπάνω κώδικα χρησιμοποιήθηκαν οι βιβλιοθήκες ThreeWire και RtcDS1302.

### 3.2.7 Πλακέτα φωτοαντίστασης KY-018



Εικόνα 3.20: Πλακέτα φωτοαντίστασης KY-018.

(LDR Module KY-018 (Analog) Photosensitive Resistor | bec-lb, 2024)

Η πλακέτα φωτοαντίστασης KY-018 χρησιμοποιείται για την μέτρηση έντασης του φωτός. Η φωτοαντίσταση είναι ένα παθητικό στοιχείο του οποίου η αντίσταση μεταβάλλεται ανάλογα με τη φωτεινότητα του φωτός μέσω της ταινίας χαλκού που υπάρχει στην κορυφή της. Στη συνέχεια, το Arduino μπορεί να διαβάσει την έξοδο του αισθητήρα μέσω μιας αναλογικής εισόδου. Υψηλές τιμές εξόδου αντιστοιχίζονται σε φωτεινό περιβάλλον, ενώ χαμηλές τιμές αποτυπώνουν σκοτεινό περιβάλλον. (grobotronics, n.d.)

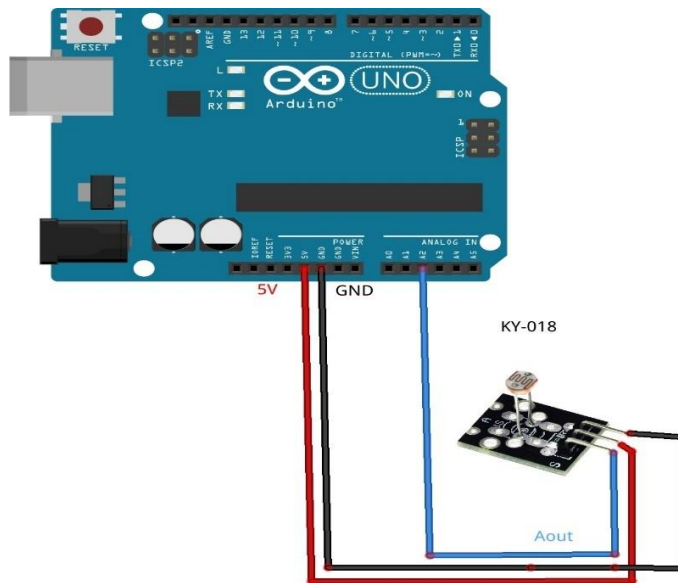
Κάποια τεχνικά χαρακτηριστικά παρουσιάζονται στον παρακάτω πίνακα:

Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	3.3V ~ 5.5V
Αντίσταση φωτορεζίστορ	10kΩ
Τύπος εξόδου	Αναλογικός

Πίνακας 3.15: Χαρακτηριστικά πλακέτας φωτοαντίστασης KY-018.

(KY-018 Photoresistor Module | ArduinoModuls, 2019)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του αισθητήρα KY-018 με το Arduino Uno:



Εικόνα 3.21: Συνδεσμολογία αισθητήρα KY-018

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του αισθητήρα:

```
int sensorPin = 2; //define analog pin 2
int value = 0;

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

void loop() {
  value = analogRead(sensorPin);
  Serial.println(value, DEC); // light intensity
                               // high values for bright environment
                               // low values for dark environment

  delay(100);
}
```

Πίνακας 3.16: Ενδεικτικός κώδικας KY-018

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.8 Αισθητήρας θερμοκρασίας νερού DS18B20p



Εικόνα 3.22: Αισθητήρας θερμοκρασίας νερού DS18B20p



(Guide for DS18B20 Temperature Sensor with Arduino | randomnerdtutorials, 2024)

Κατά τη διαδικασία του ποτίσματος είναι σημαντικός ο έλεγχος της θερμοκρασίας του νερού που διαχέεται στο κάθε φυτό. Γι' αυτό το λόγο έχει χρησιμοποιηθεί ένας αδιάβροχος αισθητήρας θερμοκρασίας ο οποίος είναι αρκετά απλός στη χρήση. Ακόμη, με τη χρήση της βιβλιοθήκης <OneWire.h> δίνει τη δυνατότητα να χρησιμοποιηθούν πολλοί όμοιοι αισθητήρες οι οποίοι συνδέονται σε μία μόνο ψηφιακή είσοδο του Arduino. Έπειτα, με τη συγγραφή του κατάλληλου κώδικα, ο μικροελεγκτής είναι σε θέση να αναγνωρίσει από ποιον αισθητήρα έρχεται η κάθε μέτρηση. (randomnerdtutorials, Guide for DS18B20 Temperature Sensor with Arduino, 2021)

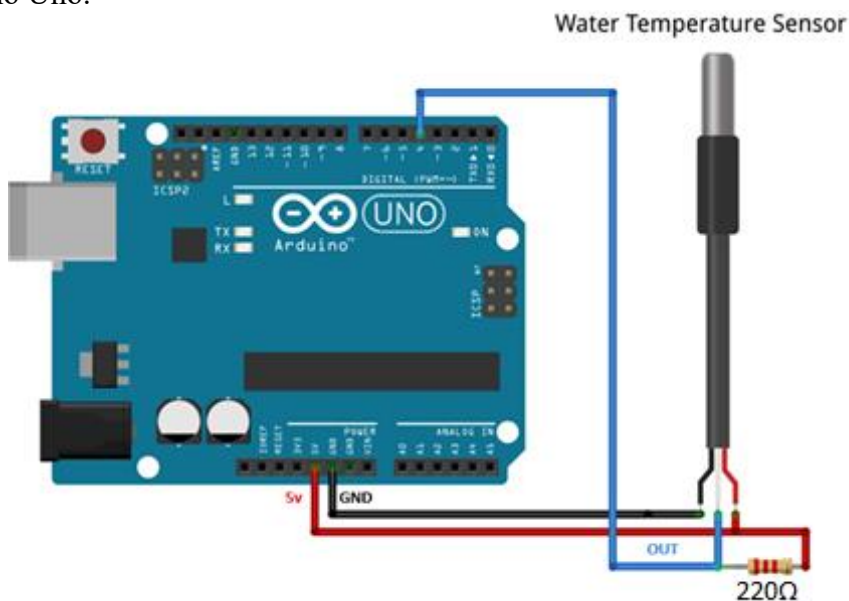
Κάποια τεχνικά χαρακτηριστικά παρουσιάζονται στον παρακάτω πίνακα:

Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	3V ~ 5.5V
Μέγιστο ρεύμα λειτουργίας	30 $\mu$ A στα 5V
Ακρίβεια	$\pm 0,5^{\circ}\text{C}$ από $-10^{\circ}\text{C}$ έως $+85^{\circ}\text{C}$
Εύρος βιομηχανικών θερμοκρασιών	$-55^{\circ}\text{C}$ έως $+125^{\circ}\text{C}$
Μετατροπέας θερμοκρασίας	12 bit σε 750 ms
Μέγεθος	20 mm

Πίνακας 3.17: Χαρακτηριστικά αισθητήρα θερμοκρασίας νερού DS18B20.

(DS18B20 Datasheet - Dallas Semiconductor | Alldatasheet, 2024)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του αισθητήρα DS18B20p με το Arduino Uno:



Εικόνα 3.23: Συνδεσμολογία αισθητήρα DS18B20p

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του αισθητήρα:

```

#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is conntec to the Arduino digital pin 4
#define ONE_WIRE_BUS 4

// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature sensor
DallasTemperature sensors(&oneWire);

void setup(void)
{
  // Start serial communication for debugging purposes
  Serial.begin(9600);
  // Start up the library
  sensors.begin();
}

void loop(void) {
  // Call sensors.requestTemperatures() to issue a global temperature and
  Requests to all devices on the bus
  sensors.requestTemperatures();

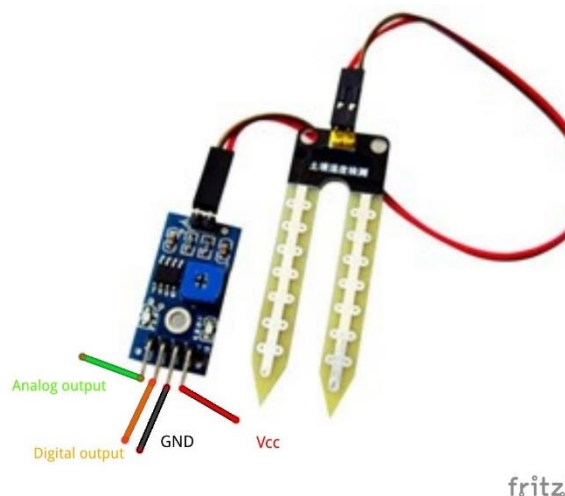
  Serial.print("Celsius temperature: ");
  // Why "byIndex"? You can have more than one IC on the same bus. 0
  refers to the first IC on the wire
  Serial.print(sensors.getTempCByIndex(0));
  Serial.print(" - Fahrenheit temperature: ");
  Serial.println(sensors.getTempFByIndex(0));
  delay(1000);
}

```

Πίνακας 3.18: Ενδεικτικός κώδικας αισθητήρα θερμοκρασίας νερού DS18B20

Για τον παραπάνω κώδικα χρησιμοποιήθηκαν οι βιβλιοθήκες OneWire και DallasTemperature.

### 3.2.9 Αισθητήρας υγρασίας εδάφους Soil moisture detector



Εικόνα 3.24: Αισθητήρας υγρασίας εδάφους.

Απαραίτητη προϋπόθεση για το σενάριο ποτίσματος είναι η ύπαρξη ενός αισθητήρα που θα μετράει την υγρασία του χώματος. Την μέτρηση αυτήν την παίρνει ο συγκεκριμένος αισθητήρας αξιοποιώντας χωρητικά φαινόμενα για να μετρήσει την διηλεκτρική αγωγιμότητα του νερού που περιέχεται μέσα στο χώμα.

Ο αισθητήρας συνοδεύεται από μία πλακέτα η οποία είναι υπεύθυνη για την μετατροπή της μέτρησης του αισθητήρα σε τάση, ώστε να μπορεί αυτή να διαβαστεί από το Arduino. Όταν ο αισθητήρας βρίσκεται σε υγρό χώμα, η τάση στην έξοδο της πλακέτας είναι χαμηλή, ενώ όταν βρίσκεται σε ξηρό χώμα, η τάση ανεβαίνει. Ακόμη, η πλακέτα περιέχει και ένα ποτενσιόμετρο για ρύθμιση της ευαισθησίας της μέτρησης. (lastminuteengineers, How Soil Moisture Sensor Works and Interface it with Arduino, n.d.)

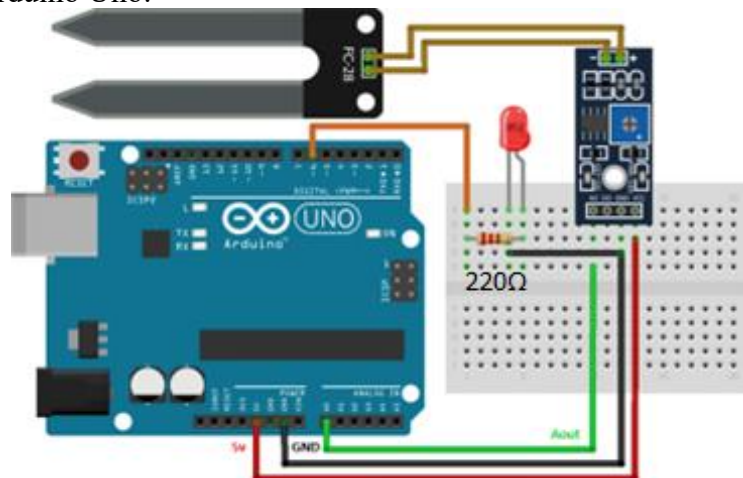
Κάποια τεχνικά χαρακτηριστικά παρουσιάζονται στον παρακάτω πίνακα:

Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	3.3V ~ 5.5V
Μέγιστο ρεύμα λειτουργίας	15 mA στα 5V
Ψηφιακή έξοδος	0 V έως 5 V
Αναλογική έξοδος	0 V έως 5 V
Ενδεικτικές λυχνίες	LED
Μέγεθος	3.2cm x 1.4cm

Πίνακας 3.19: Χαρακτηριστικά αισθητήρα υγρασίας εδάφους FC-28 Soil Moisture Sensor Module .

(FC-28 Soil Moisture Sensor Module| Alldatasheet, 2024)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του αισθητήρα μέτρησης υγρασίας χώματος με το Arduino Uno:



Εικόνα 3.25: Συνδεσμολογία soil moisture sensor

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του αισθητήρα:

```
#define ledPin 6
#define sensorPin A0

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
}

void loop() {
  Serial.print("Analog output: ");
  Serial.println(readSensor());
  delay(500);
}

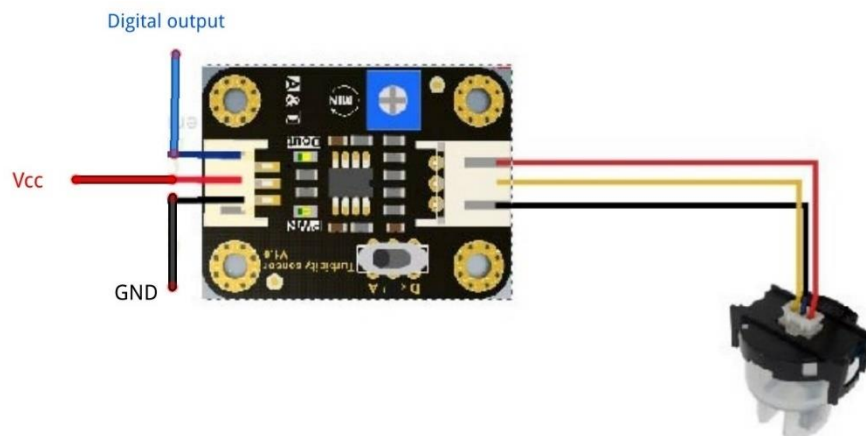
// This function returns the analog data to calling function

int readSensor() {
  int sensorValue = analogRead(sensorPin); // Read the analog value from
  sensor
  int outputValue = map(sensorValue, 0, 1023, 255, 0); // map the 10-bit
  data to 8-bit data
  analogWrite(ledPin, outputValue); // generate PWM signal
  return outputValue; // Return analog moisture value
}
```

Πίνακας 3.20: Ενδεικτικός κώδικας αισθητήρα υγρασίας εδάφους

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.10 Αισθητήρας καθαρότητας νερού SEN0189



Εικόνα 3.26: Αισθητήρας καθαρότητας νερού.

(Turbidity sensor SKU SEN0189| dfrobot, 2024)

Η καθαρότητα του νερού παίζει κι αυτή σημαντικό ρόλο κατά τη διαδικασία του ποτίσματος και για το λόγο αυτό χρησιμοποιήθηκε ο αισθητήρας καθαρότητας SEN0189.

Χρησιμοποιώντας φως για να εντοπίσει τη διάθλασή του στο νερό, μπορεί να ανιχνεύσει αν το νερό είναι διαφανές ή αν λόγω χρώματος και άλλων υλικών είναι πιο θολό.

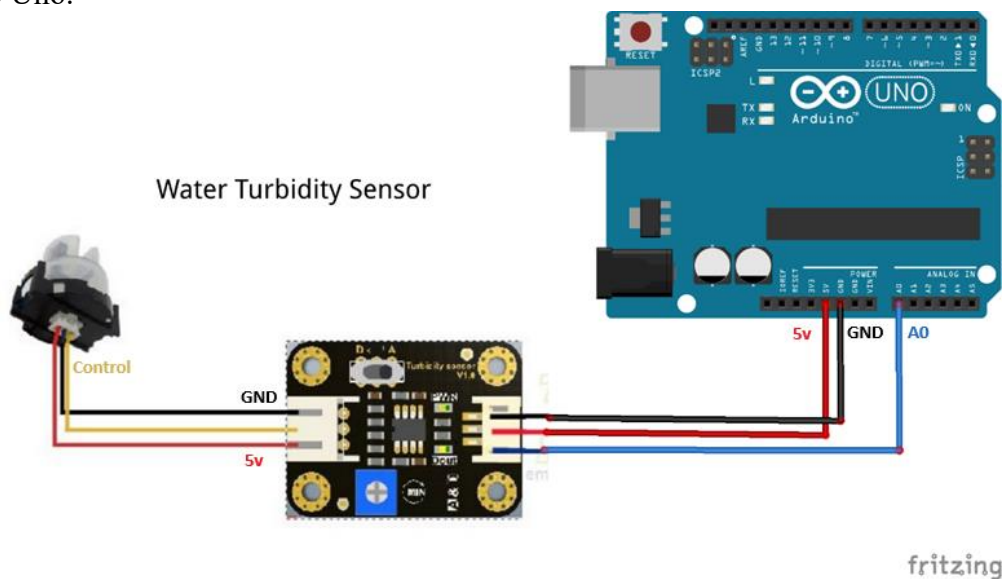
Η πλακέτα που συνοδεύει τον αισθητήρα παρέχει τόσο ψηφιακή όσο και αναλογική έξοδο δίνοντας την ελευθερία στον χρήστη να χρησιμοποιήσει την πιο χρήσιμη για την εφαρμογή του. Το κατώφλι στο οποίο θα αλλάξει η ψηφιακή έξοδος μπορεί να ρυθμιστεί μέσω ενός ποτενσιόμετρου που υπάρχει πάνω στην πλακέτα. (DFRobot, Turbidity\_sensor\_SKU\_\_SEN0189, n.d.)

Κάποια τεχνικά χαρακτηριστικά παρουσιάζονται στον παρακάτω πίνακα:

Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	5.5V
Μέγιστο ρεύμα λειτουργίας	40 mA στα 5V
Θερμοκρασία λειτουργίας	5°C ~90°C
Χρόνος απόκρισης	<500ms
Θερμοκρασία αποθήκευσης	-10°C ~90°C
Διαστάσεις προσαρμογέα	38mm*28mm*10mm/1.5inches * 1.1inches*0.4inches

Πίνακας 3.21: Χαρακτηριστικά αισθητήρα καθαρότητας νερού SEN0189.  
(Turbidity sensor SKU:SEN0189| DFROBOT, 2017)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του αισθητήρα SEN0189 με το Arduino Uno:



Εικόνα 3.27: Συνδεσμολογία αισθητήρα SEN0189

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του αισθητήρα:

```
void setup() {
```

```

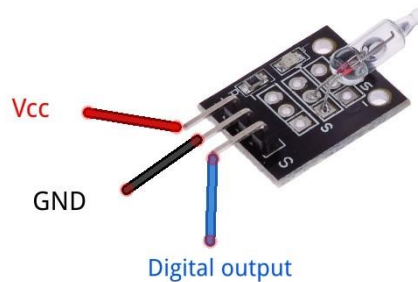
Serial.begin(9600); //Baud rate: 9600
}
void loop() {
  int sensorValue = analogRead(A0); // read the input on analog pin 0:
  float voltage = sensorValue * (5.0 / 1024.0); // Convert the analog
  reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  Serial.println(voltage); // print out the value you read:
  delay(500);
}

```

Πίνακας 3.22: Ενδεικτικός κώδικας αισθητήρα καθαρότητας νερού SEN0189

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.11 Αισθητήρας κλίσης KY-017



Εικόνα 3.28: Αισθητήρας κλίσης.

(KY-017 Mercury Tilt Switch Module | arduinomodules, 2024)

Ο αισθητήρας κλίσης KY-017 χρησιμοποιεί μία μπάλα υδραργύρου για να εντοπίσει την αλλαγή στην κλίση. Όταν εκείνη αλλάξει, η μπάλα κινείται και κλείνει το κύκλωμα οπότε η έξοδος παίρνει την τιμή λογικό «1». Επίσης, ανάβει και ένα προειδοποιητικό LED που είναι τοποθετημένο πάνω στην πλακέτα. Στην παρούσα υλοποίηση χρησιμοποιείται για την λειτουργία του σκιάστρου. (ArduinoModules, n.d.)

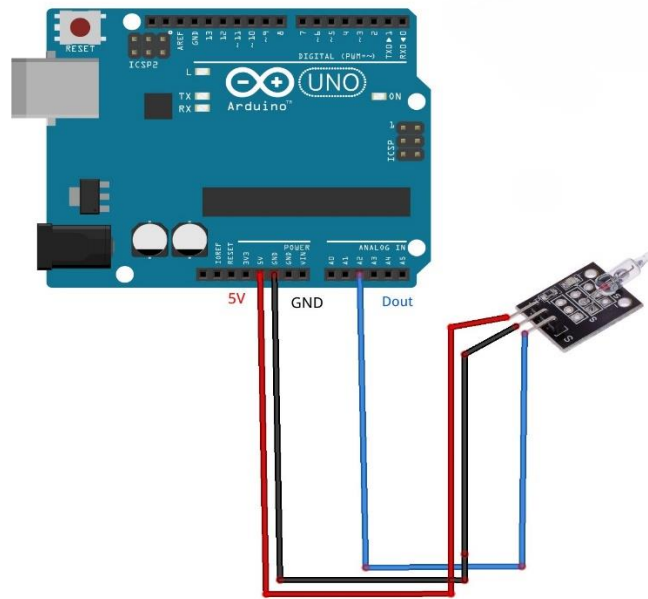
Κάποια τεχνικά χαρακτηριστικά παρουσιάζονται στον παρακάτω πίνακα:

Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	3.3V ~ 5.5V
Αντίσταση φωτορεζίστορ	680Ω
Τύπος εξόδου	Ψηφιακός

Πίνακας 3.23: Χαρακτηριστικά αισθητήρα κλίσης KY-017.

(KY-017 Mercury Tilt Switch Module|ArduinoModuls, 2021)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του αισθητήρα KY-017 με το Arduino Uno:



Εικόνα 3.29: Συνδεσμολογία αισθητήρα KY-017

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του αισθητήρα:

```
int led_pin = 13; // Define the LED interface
int switch_pin = 3; // Definition of mercury tilt switch sensor interface
int val; // Defines a numeric variable

void setup()
{
  pinMode(led_pin, OUTPUT);
  pinMode(switch_pin, INPUT);
}

void loop()
{
  val = digitalRead(switch_pin); // check mercury switch state
  if(val == HIGH)
  {
    digitalWrite(led_pin, HIGH);
  }
  else
  {
    digitalWrite(led_pin, LOW);
  }
}
```

Πίνακας 3.24: Ενδεικτικός κώδικας αισθητήρα κλίσης KY-017

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.12 Αισθητήρας στάθμης νερού



Εικόνα 3.30: Αισθητήρας στάθμης νερού.

(Robodo SEN18 Water Level Sensor | amazon, 2024)

Η μέτρηση της στάθμης του νερού είναι πολύ σημαντική σε εφαρμογές που περιλαμβάνουν δεξαμενές αποθήκευσης όπως είναι και το αντικείμενο αυτής της διπλωματικής. Για να αποφευχθεί, λοιπόν, κάποιο συμβάν υπερχειλίσης, χρησιμοποιούμε έναν αισθητήρα μέτρησης της στάθμης. Αυτός έχει εκτεθειμένους στην επιφάνειά του τους αγώγιμους δρόμους του σήματος και της τροφοδοσίας ώστε καθώς βυθίζεται στο νερό, τα δύο σήματα να βραχυκυκλώνουν εν μέρει σχηματίζοντας μία μεταβλητή αντίσταση προσομοιώνοντας τη λειτουργία ενός ποτενσιόμετρου. Έτσι, όταν ο αισθητήρας είναι ελάχιστα βυθισμένος στο νερό, η αντίσταση που βγάζει στην έξοδο είναι μεγάλη, ενώ όσο βυθίζεται περαιτέρω εκείνη μειώνεται ως αποτέλεσμα της βραχυκύκλωσης εξαιτίας του νερού. Η ανάγνωση της μέτρησης γίνεται από το Arduino μέσω μιας απλής αναλογικής εισόδου. (Robodo, n.d.)

Κάποια τεχνικά χαρακτηριστικά παρουσιάζονται στον παρακάτω πίνακα:

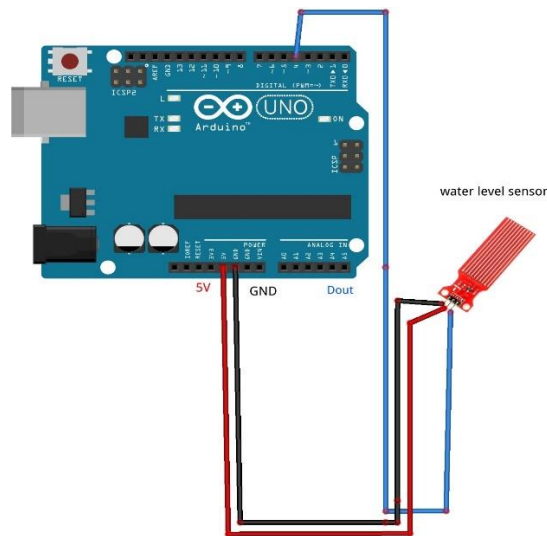
Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	3-5V
Μέγιστο ρεύμα λειτουργίας	<20 mA
Θερμοκρασία λειτουργίας	10°C-30°C Humidity: 10% -90% non-condensing
Τύπος αισθητήρα	Αναλογικός
Περιοχή ανίχνευσης	40mmx16mm
Διαστάσεις	60mmx20mm

Πίνακας 3.25: Χαρακτηριστικά αισθητήρα στάθμης νερού.

(Arduino Water Level Sensor|datasheethub, 2023)



Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του αισθητήρα στάθμης νερού με το Arduino Uno:



Εικόνα 3.31: Συνδεσμολογία αισθητήρα στάθμης νερού

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του αισθητήρα:

```
// Sensor pins
#define sensorPower 7
#define sensorPin A0

// Value for storing water level
int val = 0;

void setup() {
  // Set D7 as an OUTPUT
  pinMode(sensorPower, OUTPUT);

  // Set to LOW so no power flows through the sensor
  digitalWrite(sensorPower, LOW);

  Serial.begin(9600);
}

void loop() {
  //get the reading from the function below and print it
  int level = readSensor();

  Serial.print("Water level: ");
  Serial.println(level);

  delay(1000);
}

//This is a function used to get the reading
int readSensor() {
  digitalWrite(sensorPower, HIGH); // Turn the sensor ON
  delay(10); // wait 10 milliseconds
  val = analogRead(sensorPin); // Read the analog value form
  sensor
  digitalWrite(sensorPower, LOW); // Turn the sensor OFF
  return val; // send current reading
}
```

Πίνακας 3.26: Ενδεικτικός κώδικας αισθητήρα στάθμης νερού

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.13 Κιτ ασύρματου απομακρυσμένου ελέγχου μέσω IR



Εικόνα 3.32: Κιτ ασύρματου απομακρυσμένου ελέγχου μέσω IR.

(Infrared IR Receiver Module | topelectronics, 2024)

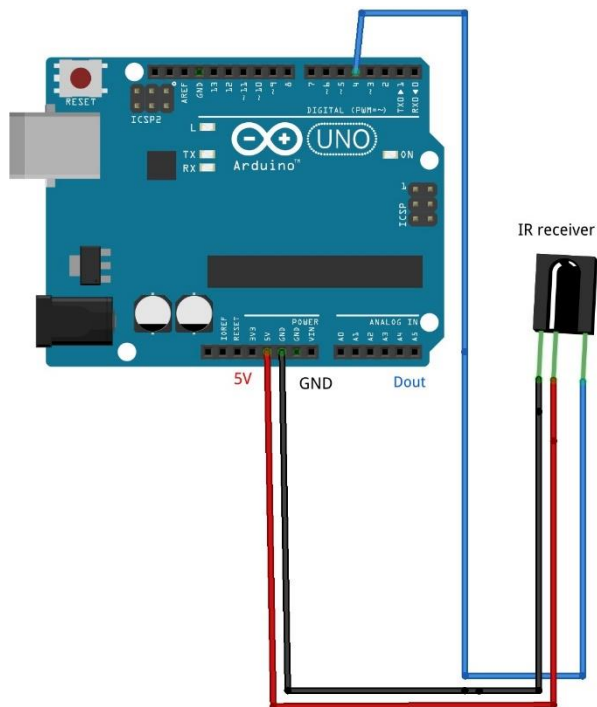
Το κιτ ασύρματου απομακρυσμένου ελέγχου για Arduino αποτελείται από ένα τηλεχειριστήριο υπερύθρων και έναν δέκτη υπερύθρων. Το τηλεχειριστήριο περιλαμβάνει 20 πλήκτρα λειτουργιών μεταξύ των οποίων νούμερα, βελάκια και ένα κουμπί επιβεβαίωσης «OK». Αποτελεί ιδανική επιλογή για το χειρισμό συσκευών σε εσωτερικούς χώρους καθώς διαθέτει απόσταση μετάδοσης έως και 8 μέτρα. Κάποια χρήσιμα χαρακτηριστικά για το κιτ φαίνονται στον παρακάτω πίνακα (topelectronics, n.d.):

Χαρακτηριστικό	Περιγραφή
Απόσταση μετάδοσης	~ 8m
Γωνία μετάδοσης	60°
Μπαταρία	CR2025
Χωρητικότητα μπαταρίας	160mAh
Χρόνος ζωής	20000 χρήσεις
Ρεύμα λειτουργίας	3μΑ – 5μΑ

Πίνακας 3.27: Χαρακτηριστικά του κιτ ασύρματου απομακρυσμένου ελέγχου μέσω IR.

(Arduino IR Receiver Module | datasheethub, 2023)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του κιτ ασύρματου απομακρυσμένου ελέγχου με το Arduino Uno:



Εικόνα 3.33: Συνδεσμολογία κιτ ασύρματου απομακρυσμένου ελέγχου

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του κιτ:

```
int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;

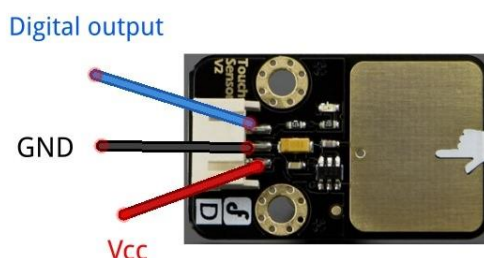
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}

void loop()
{
  if (irrecv.decode(&results))
  {
    Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
  }
}
```

Πίνακας 3.28: Ενδεικτικός κώδικας κιτ ασύρματου απομακρυσμένου ελέγχου μέσω IR

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.14 Αισθητήρας αφής



Εικόνα 3.34: Αισθητήρας αφής

(Gravity: Digital Capacitive Touch Sensor for ArduinoModule | dfrobot, 2024)

Ο αισθητήρας αφής χρησιμοποιείται στο σενάριο κλειδώματος/ξεκλειδώματος της εξωτερικής πόρτας, σαν άλλη μία δικλείδα ασφαλείας. Χρησιμοποιεί κυκλώματα πυκνωτών των οποίων η χωρητικότητα μεταβάλλεται καθώς ακουμπάει το ανθρώπινο δάχτυλο χάρη στην επιπλέον χωρητικότητα που υπάρχει μεταξύ του ανθρώπου και της γης. Με αυτόν τον τρόπο μπορεί να στέλνει στην έξοδό του σήμα 1 ή 0 ανάλογα με το αν έχει υπάρξει επαφή ή όχι αντίστοιχα. Μάλιστα διαθέτει πολύ μεγάλη ευαισθησία καθώς είναι ικανός να ανιχνεύσει αφή ακόμη και μέσα από κάποιο ύφασμα. (DFRobot, Gravity: Digital Capacitive Touch Sensor for Arduino, 2022)

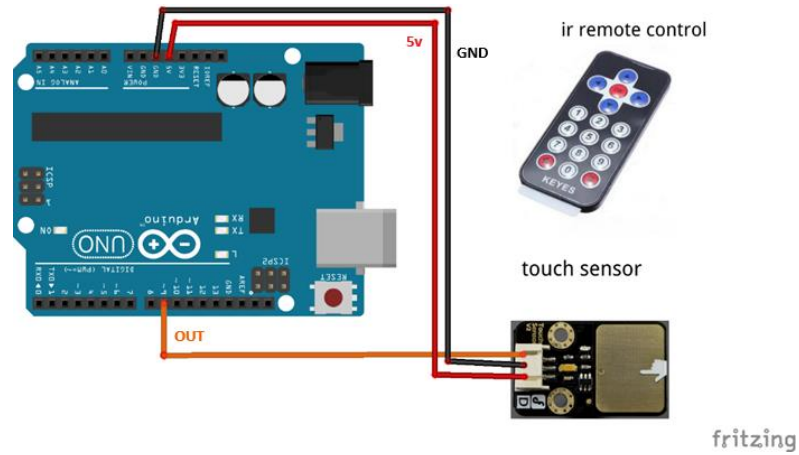
Κάποια τεχνικά χαρακτηριστικά παρουσιάζονται στον παρακάτω πίνακα:

Χαρακτηριστικό	Περιγραφή
Τάση λειτουργίας	2-5V
Χρόνος απόκρισης	60ms
Ρεύμα λειτουργίας	3μΑ – 7μΑ

Πίνακας 3.29: Χαρακτηριστικά αισθητήρα αφής.

(Digital Touch Sensor Module|osoyoo, 2018)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του αισθητήρα αφής με το Arduino Uno:



Εικόνα 3.35: Συνδεσμολογία αισθητήρα αφής

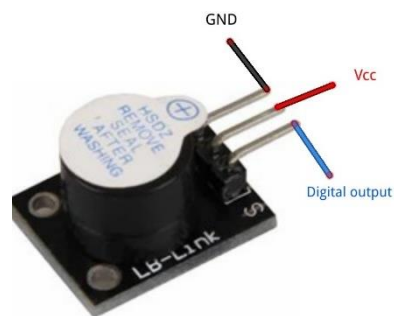
Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του αισθητήρα:

```
#define BUTTON_PIN 4
struct touch {
    byte wasPressed = LOW;
    byte isPressed = LOW;
};
touch touch;
void setup()
{
    pinMode(BUTTON_PIN, INPUT);
    Serial.begin(115200);
}
void loop()
{
    touch.isPressed = isTouchPressed(BUTTON_PIN);
    if (touch.wasPressed != touch.isPressed) {
        Serial.println("Touch pressed");
    }
    touch.wasPressed = touch.isPressed;
}
bool isTouchPressed(int pin)
{
    return digitalRead(pin) == HIGH;
}
```

Πίνακας 3.30: Ενδεικτικός κώδικας για τον αισθητήρα αφής

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.15 Πλακέτα buzzer KY-012



Εικόνα 3.36: Πλακέτα buzzer KY-012.

(KY-012 Active Piezo-Buzzer Module | thegeekpub, 2024)

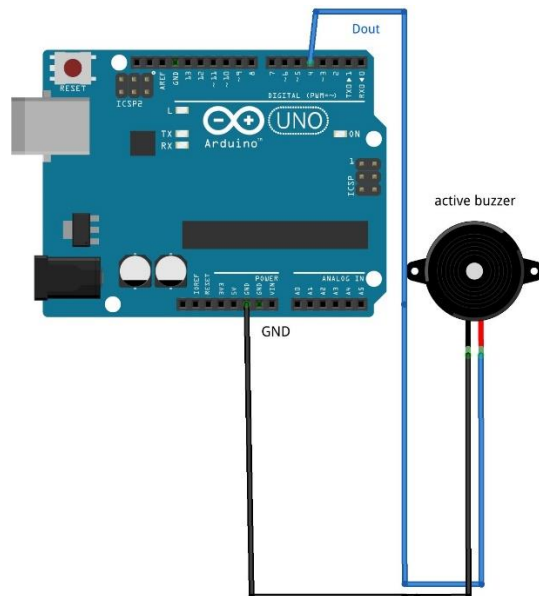
Η πλακέτα buzzer περιέχει ενσωματωμένο ένα ενεργό πιεζοηλεκτρικό βομβητή ο οποίος παράγει έναν ήχο στα 2.5kHz χωρίς να είναι απαραίτητη η χρήση κάποιας παλμοσειράς μέσω PWM παρά μόνο στέλνοντας ένα σήμα λογικό 1 στον ακροδέκτη σήματός του. Κάποια τεχνικά χαρακτηριστικά για τον αισθητήρα παρουσιάζονται στον παρακάτω πίνακα (thegeekpub, n.d.):

Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	+3.3V - +5V
Μέγιστο ρεύμα	30mA
Συχνότητα συντονισμού	2500Hz ± 300Hz συνεχής
Ελάχιστη ένταση ήχου εξόδου	85dB @ 10cm
Θερμοκρασία αποθήκευσης	-30°C - 105°C
Θερμοκρασία λειτουργίας	-20°C - 70°C
Διαστάσεις	18.5mm x 15mm

Πίνακας 3.31: Χαρακτηριστικά της πλακέτας buzzer KY-012.

(KY-012 Active Piezo-Buzzer Module | osoyoo, 2020)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία της πλακέτας buzzer KY-012 με το Arduino Uno:



Εικόνα 3.37: Συνδεσμολογία πλακέτας buzzer KY-012

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία της πλακέτας:

```
int buzzerPin = 8;

void setup ()
{
```

```

pinMode (buzzerPin, OUTPUT);
}

void loop ()
{
digitalWrite (buzzerPin, HIGH);
delay (500);
digitalWrite (buzzerPin, LOW);
delay (500);
}

```

Πίνακας 3.32: Ενδεικτικός κώδικας για το buzzer

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.16 Αντλία νερού JT-180



Εικόνα 3.38: Αντλία νερού JT-180.  
(DC Mini Immersible Water Pump | diyables, 2024)

Για την άντληση του νερού από τη δεξαμενή και την παροχή στο σύστημα ποτίσματος χρησιμοποιήθηκε η αντλία νερού JT-180 η οποία έχει αρκετά μεγάλη διάρκεια ζωής καθώς ενδείκνυται για παραπάνω από 10000 ώρες συνεχούς χρήσης. Καθώς απαιτεί τροφοδοσία 12V, είναι απαραίτητη η χρήση του ρελέ που εξηγήθηκε παραπάνω. Στον παρακάτω πίνακα παρουσιάζονται κάποια τεχνικά χαρακτηριστικά της αντλίας (Research, n.d.):

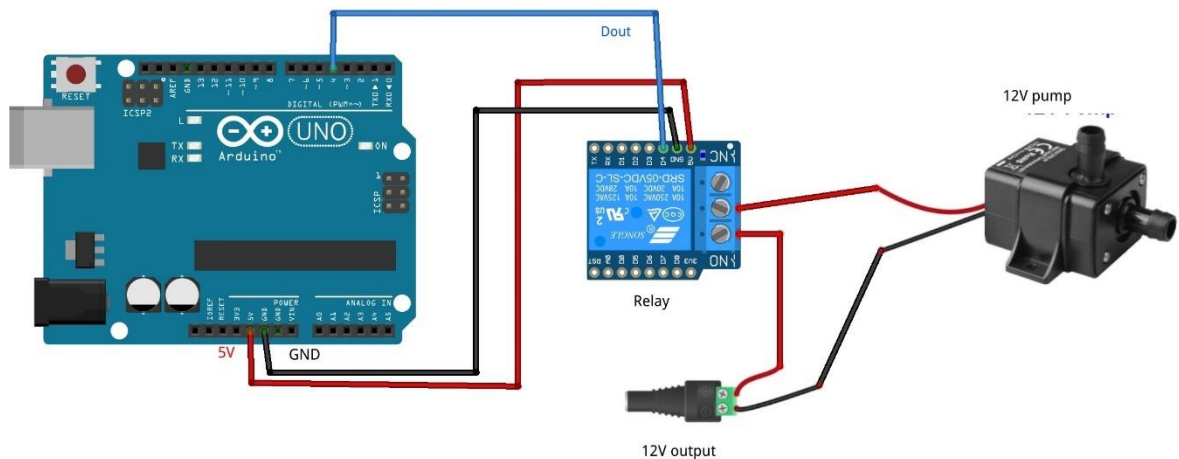
Χαρακτηριστικό	Τιμή
Θερμοκρασία περιβάλλοντος	-20°C - 60°C
Υγρασία λειτουργίας	<90%

Θερμοκρασία ρευστού	-20°C - 60°C
Προτεινόμενο ρευστό	Νερό ή άλλο μη εύφλεκτο ρευστό
Ταχύτητα ροής	370 L/H (liters per hour)

Πίνακας 3.33: Χαρακτηριστικά της αντλίας νερού JT-180

(Arduino control pump|Arduino get started, 2018

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία της αντλίας νερού με το Arduino Uno:



Εικόνα 3.39: Συνδεσμολογία αντλίας νερού JT-180.

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία της αντλίας:

```
const int pump = 8;
void setup ()
{
  pinMode (pump, OUTPUT) ;}

void loop ()
{
  digitalWrite (pump, HIGH) ;
  delay (500) ;
  digitalWrite (pump, LOW) ;
  delay (500) ;
}
```

Πίνακας 3.34: Ενδεικτικός κώδικας για την αντλία

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.17 Ανεμιστήρας EB60251S1 SUNON





Εικόνα 3.40: Ανεμιστήρας EB60251S1 SUNON.

(Ανεμιστήρας 12V DC | diyables, 2024)

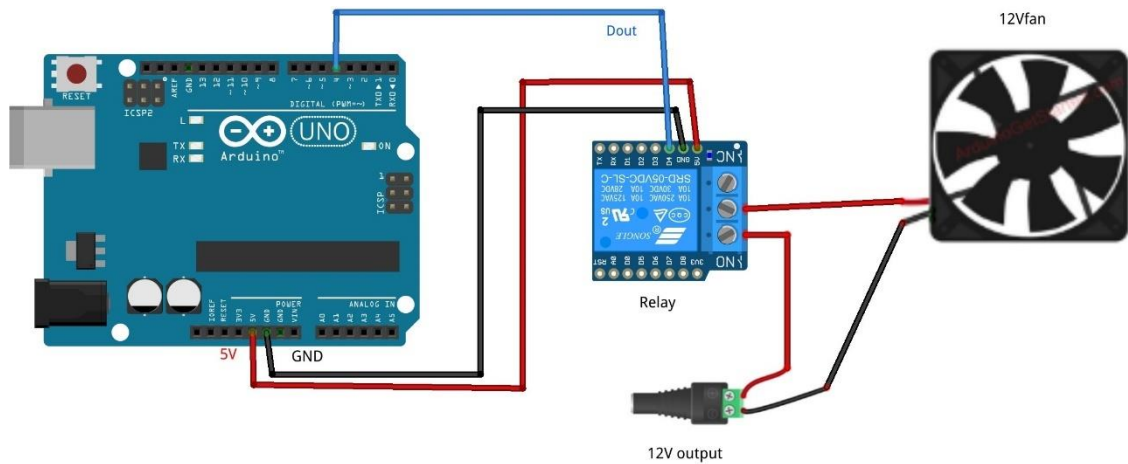
Για το σύστημα εξαερισμού χρησιμοποιήθηκε ένας χαμηλής ισχύος ανεμιστήρας μεγέθους 60mm x 60mm x 25mm με τροφοδοσία 12V. Επομένως, όπως και στην αντλία νερού, θα χρειαστεί η παρεμβολή και ενός ρελέ για την ασφαλή τροφοδοσία του. Η κατανάλωση ισχύος του είναι 1.62W με ρεύμα λειτουργίας τα 0.125A. Με αυτά τα δεδομένα μπορεί να φτάσει ταχύτητα περιστροφής έως και 4500 RPM παρέχοντας μέγιστο ρυθμό ροής αέρα 32.7 m<sup>3</sup>/h. Στον παρακάτω πίνακα παρουσιάζονται κάποια τεχνικά χαρακτηριστικά της αντλίας: (Net, n.d.)

Χαρακτηριστικό	Τιμή
Τάση λειτουργίας	12V
Ρεύμα λειτουργίας	135mA
Στροφές	4500 rpm
Κατανάλωση ενέργειας	1.62 W
Διαστάσεις	60mm x 60mm x 25mm

Πίνακας 3.35: Χαρακτηριστικά Ανεμιστήρα EB60251S1 SUNON

(Arduino fan EB60251S1 SUNON | sunon, 2024)

Στο παρακάτω κύκλωμα φαίνεται η συνδεσμολογία του ανεμιστήρα με το Arduino Uno:



Εικόνα 3.41: Συνδεσμολογία ανεμιστήρα EB60251S1 SUNON.

Ακόμη, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία της αντλίας:

```
const int fan = 4;
void setup ()
{
  pinMode (fan, OUTPUT);
}

void loop ()
{
  digitalWrite (fan, HIGH);
  delay (500);
  digitalWrite (fan, LOW);
  delay (500);
}
```

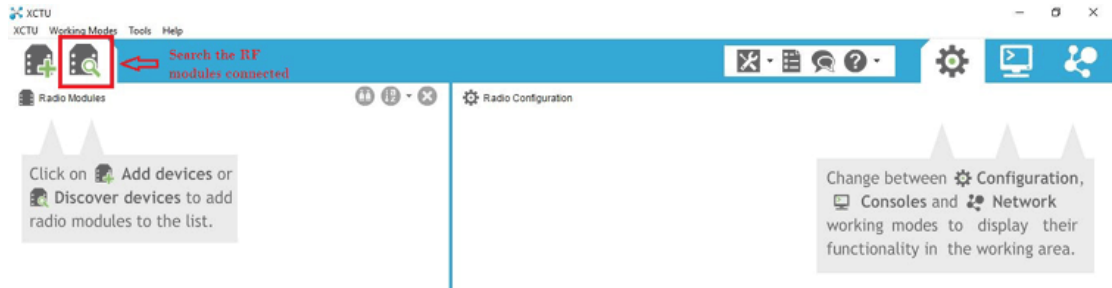
Πίνακας 3.36: Ενδεικτικός κώδικας για τον ανεμιστήρα

Για τον παραπάνω κώδικα δεν χρησιμοποιήθηκε κάποια βιβλιοθήκη.

### 3.2.18 XBee

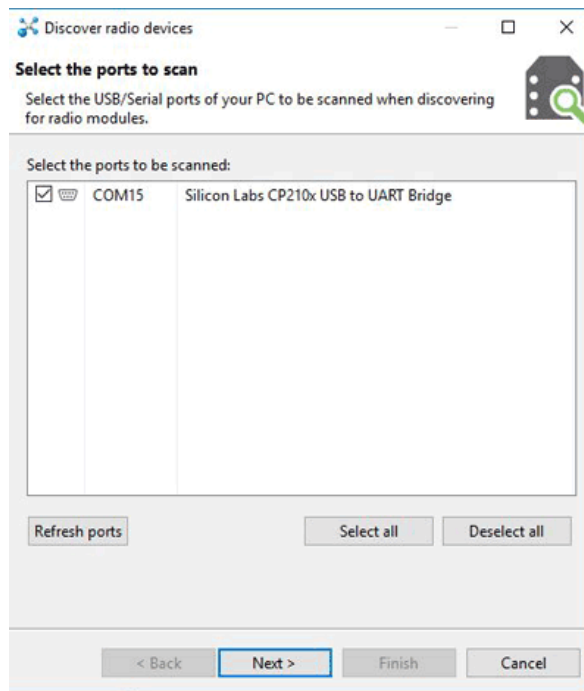
Στο σενάριο ελέγχου ποιότητας νερού δεξαμενής υλοποιήθηκε ασύρματη αποστολή των τιμών της καθαρότητας του νερού όπως επίσης και της θερμοκρασίας του με τη χρήση δύο XBee modules. Όπως έχει αναφερθεί στο προηγούμενο κεφάλαιο, ένα XBee module μπορεί να λειτουργήσει ως συντονιστής (coordinator), δρομολογητής (router) ή τερματική συσκευή (end device). Ο ρόλος του καθενός καθώς επίσης και οι παράμετροι της επικοινωνίας ορίζονται μέσω του προγράμματος XCTU. Μετά τη σύνδεση του XBee στον Η/Υ μέσω USB, τα βήματα για την αρχικοποίησή του φαίνονται παρακάτω.

1. Επιλογή κουμπιού αναζήτησης. Αυτό εμφανίζει όλες τις συσκευές RF που είναι συνδεδεμένες με τον Η/Υ. Στην συγκεκριμένη περίπτωση θα εμφανιστεί μόνο μία μονάδα XBee.



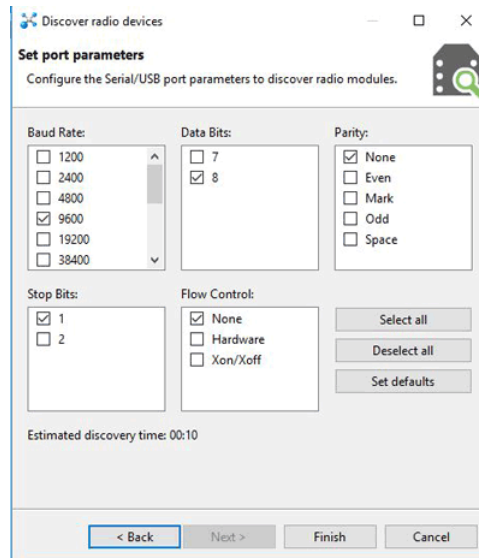
Εικόνα 3.42: Αναζήτηση των συνδεδεμένων RF συσκευών.

2. Επιλογή της σειριακής θύρας και επιλογή κουμπιού «Next».



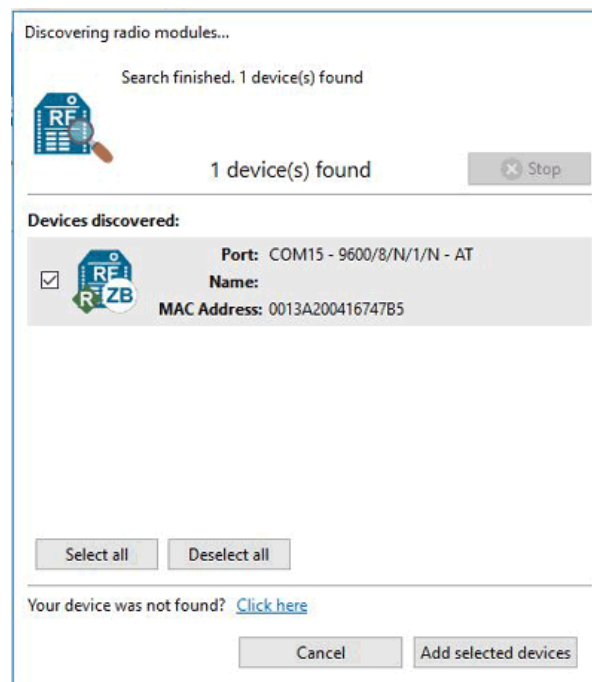
Εικόνα 3.43: Επιλογή σειριακής θύρας

3. Ορισμός ρυθμίσεων σειριακής επικοινωνίας.



Εικόνα 3.44: Ρυθμίσεις σειριακής επικοινωνίας

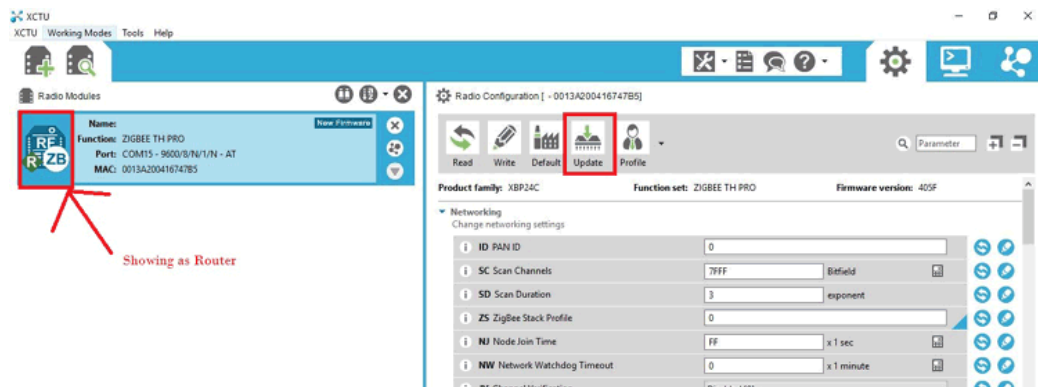
4. Επιλογή συσκευής. Με αυτή τη διαδικασία συνδέεται το XBee module στο περιβάλλον XCTU.



Εικόνα 3.45: Επιλογή συσκευής XBee για σύνδεση στο XCTU

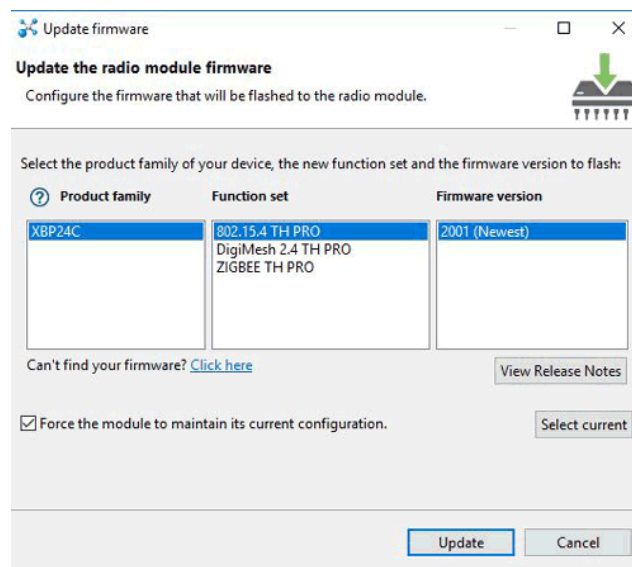
5. Ρύθμιση XBee module είτε με χρήση AT commands είτε χειροκίνητης τοποθέτησης δεδομένων. Στο αριστερό πλαίσιο εμφανίζεται το R που σημαίνει ότι το XBee βρίσκεται σε λειτουργία Router (δρομολογητή). Για την αποστολή δεδομένων θα πρέπει να

αλλάξει σε Coordinator. Αρχικά γίνεται ανανέωση του Firmware από την αντίστοιχη επιλογή.



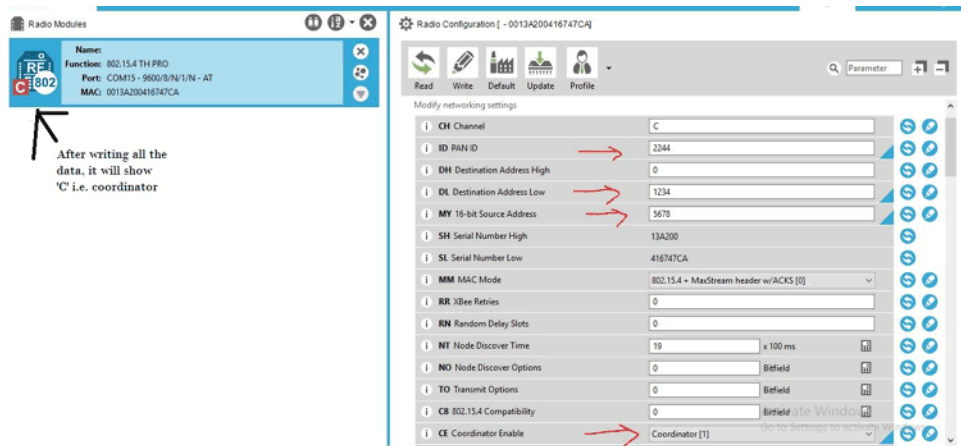
Εικόνα 3.46: Ρύθμιση XBee module και ανανέωση firmware

6. Επιλογή Product family, Function set και Firmware version όπως φαίνεται παρακάτω.



Εικόνα 3.47: Επιλογές ανανέωσης firmware

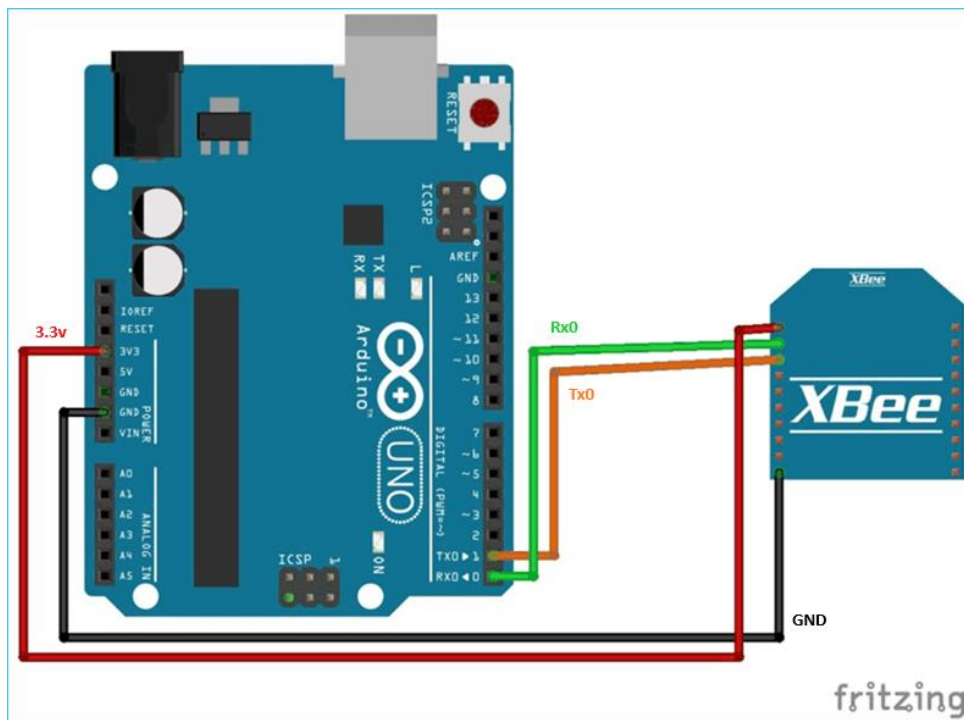
7. Για τη σύνδεση με άλλα XBee απαιτείται η συμπλήρωση των ID, MY και DL. Πιο συγκεκριμένα, το ID είναι το ίδιο και για τις δύο συσκευές. Μόνο οι τιμές των διευθύνσεων MY και DL ανταλλάσσονται, δηλαδή το MY για το XBee του δέκτη γίνεται DL στο XBee του Coordinator και αντιστρόφως. Επίσης πρέπει να οριστεί το CE ως Coordinator και στη συνέχεια να επιλεγεί το κουμπί Write. Όλες οι παραπάνω ρυθμίσεις φαίνονται παρακάτω:



Εικόνα 3.48: Ορισμός ID και διευθύνσεων στην πλευρά του Coordinator

8. Μετά την εγγραφή των παραπάνω δεδομένων στον Coordinator, γίνεται σύνδεση του δεύτερου XBee module και επαναλαμβάνεται η ίδια διαδικασία ρύθμισης των ID, DL, MY και CE με τη μόνη διαφορά να είναι στο CE η επιλογή “End device”. Στο τέλος πάλι επιλέγεται το κουμπί “Write”.
9. Σε αυτό το σημείο τα XBee modules είναι έτοιμα για σύνδεση στο Arduino. Γίνεται σύνδεση του Coordinator στον H/Y και του End device στο Arduino. Έπειτα είναι δυνατή η αποστολή εντολών στο End device μέσω του H/Y.

Η συνδεσμολογία του XBee στο Arduino γίνεται με βάση το παρακάτω σχήμα:



Εικόνα 3.49: Συνδεσμολογία XBee στο Arduino

Οι συνδέσεις που γίνονται είναι οι εξής:

1. Tx (pin2) του XBee -> Rx του Arduino

2. Rx (pin3) του Xbee -> Tx του Arduino
3. Gnd (pin10) του Xbee -> GND του Arduino
4. Vcc (pin1) του Xbee -> 3.3V του Arduino

Τέλος, παρακάτω φαίνεται ένας ενδεικτικός κώδικας για τη λειτουργία του XBee. Για να λειτουργήσει αρκεί να σταλεί ένα γράμμα “a” ή “b” από την κονσόλα του XCTU:

```
int led = 13;
int received = 0;
int i;

void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT);
}

void loop() {
  if (Serial.available() > 0) {
    received = Serial.read();
    if (received == 'a') {
      digitalWrite(led, HIGH);
      delay(2000);
      digitalWrite(led, LOW);
    }
    else if (received == 'b') {

      for (i = 0; i < 5; i++) {
        digitalWrite(led, HIGH);
        delay(1000);
        digitalWrite(led, LOW);
        delay(1000);
      }
    }
  }
}
```

Πίνακας 3.37: Ενδεικτικός κώδικας για την λειτουργία του XBee

## 4 Σενάρια αυτοματισμών και κύκλωμα

### 4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα αναλυθούν όλα τα σενάρια αυτοματισμών που υλοποιήθηκαν. Για κάθε σενάριο ξεχωριστά θα γίνει μία μικρή επεξήγηση του, θα παρουσιαστούν οι συσκευές που χρειάστηκαν για την υλοποίησή του, ενώ θα περιέχεται το κύκλωμα καθώς και ο κώδικας σε Arduino που γράφτηκε για το συγκεκριμένο σενάριο. (Σ. Δρίμης, 2029)

Τα σενάρια που υλοποιήθηκαν είναι τα παρακάτω:

- Σενάριο ελέγχου ποιότητας νερού δεξαμενής.
- Σενάριο αυτόματου ποτίσματος.
- Σενάριο ελέγχου εξαερισμού και ποιότητας αέρα.
- Σενάριο ανίχνευσης φωτιάς και πλημμύρας.
- Σενάριο ασφάλειας εξωτερικής πόρτας.
- Σενάριο ελέγχου παντζουριών και εσωτερικού φωτισμού.
- Σενάριο ελέγχου σκιάστρου.

### 4.2 Σενάριο ελέγχου ποιότητας νερού δεξαμενής

Το σενάριο ελέγχου της ποιότητας του νερού που περιέχεται στη δεξαμενή υλοποιήθηκε ώστε να διασφαλιστεί το γεγονός πως τα φυτά θα ποτίζονται με καθαρό νερό στη σωστή θερμοκρασία.

Τα ηλεκτρονικά στοιχεία που χρησιμοποιήθηκαν στο σενάριο είναι τα εξής:

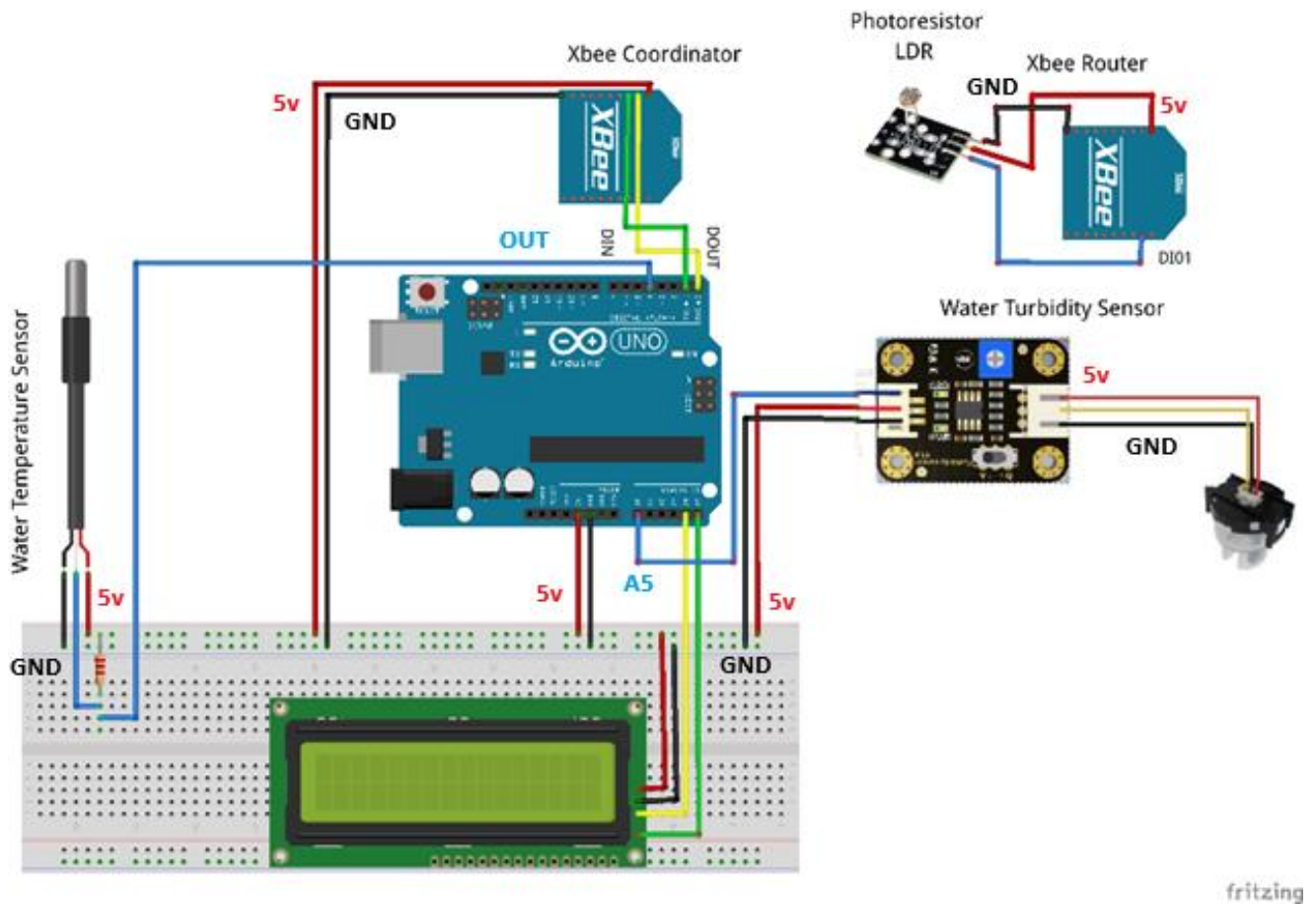
1. SEN0189 – Water turbidity sensor
2. DS18B20p – Water temperature sensor
3. LDR photoresistor- Light value sensor
4. 1 XBee στο ρόλο Coordinator το οποίο επικοινωνεί σειριακά με το Arduino. Η συνδεσμολογία είναι η εξής: ο DIN του XBee συνδέεται στο D0-TX του Arduino και το DOUT του XBee συνδέεται στο D1-RX του Arduino
5. 1 XBee στο ρόλο Router το οποίο επικοινωνεί με τους αισθητήρες. Η συνδεσμολογία είναι η εξής: ο αισθητήρας θερμοκρασίας νερού συνδέεται στο DIO1 του XBee και ο αισθητήρας καθαρότητας συνδέεται στο DIO2 του XBee.
6. 4.7k Ohm resistor
7. 16x2 I2C LCD Display

Η υλοποίηση του σεναρίου παρατίθεται στο Παράρτημα Β.1 και περιλαμβάνει 3 διαφορετικές επιλογές φυτών (ντομάτα, αγγούρι, μαρούλι) καθώς το καθένα απαιτεί

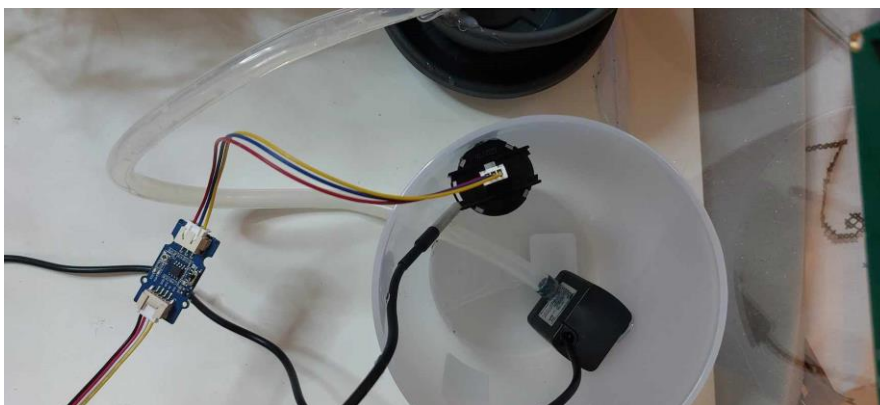


διαφορετική θερμοκρασία νερού ποτίσματος. Ο χρήστης μπορεί να πλοηγηθεί στις επιλογές με τη χρήση των δύο Buttons. Έστω ότι έχει επιλεγεί το φυτό 1 (ντομάτα). Αν η θερμοκρασία είναι εκτός των ορίων που έχουν οριστεί ( $21\text{ }^{\circ}\text{C}$  -  $27\text{ }^{\circ}\text{C}$ ) ή το νερό είναι αρκετά βρώμικο ανάλογα με την μέτρηση του αισθητήρα θολότητας ή η φωτεινότητα είναι πάνω από 60%, τότε διακόπτεται η λειτουργία της αντλίας και άρα η παροχή νερού στα φυτά. Ο χρήστης ενημερώνεται για το συμβάν και παραπέμπεται στο να ελέγξει την κατάσταση του νερού μέσα στη δεξαμενή.

Έπειτα στην σειριακή οθόνη τυπώνονται η τιμή του αισθητήρα θολότητας του νερού, καθώς και η θερμοκρασία του. Για τη λειτουργία του σεναρίου απαιτείται η ενσωμάτωση στον κώδικα του Arduino των βιβλιοθηκών OneWire.h, DallasTemperature.h, Wire.h και LiquidCrystal.h.



Εικόνα 4.1: : Κύκλωμα σεναρίου ελέγχου ποιότητας νερού δεξαμενής



*Εικόνα 4.2: Αισθητήρας ελέγχου ποιότητας νερού δεξαμενής*

Για την υλοποίηση του κώδικα, απαραίτητη προϋπόθεση είναι η εισαγωγή βιβλιοθηκών (Παράρτημα B.3.1.1). Πιο συγκεκριμένα απαιτούνται οι βιβλιοθήκες για την επικοινωνία του Ethernet shield με το Arduino Uno, η βιβλιοθήκη για την επικοινωνία με το Thingspeak, η βιβλιοθήκη για την επικοινωνία με τον αισθητήρα θερμοκρασίας και η βιβλιοθήκη για την επικοινωνία με την οθόνη. Στο παράρτημα B.3.1.2 γίνεται η αρχικοποίηση των μεταβλητών δικτύου. Η αρχικοποίηση pin των κουμπιών, του αισθητήρα θερμοκρασίας, του αισθητήρα καθαρότητας, της φωτοαντίστασης και του Xbee παρουσιάζεται στο παράρτημα B.3.1.3. Η αρχικοποίηση της οθόνης παρουσιάζεται στο παράρτημα B.3.1.5. Στο παράρτημα B.3.1.8 φαίνεται η λειτουργία της συνάρτησης που υλοποιήθηκε για το σενάριο ποιότητας δεξαμενής νερού. Τέλος στο παράρτημα B.3.1.9 και B.3.1.10 παρουσιάζονται αντίστοιχα οι λειτουργίες της οθόνης και του Xbee.

### **4.3 Σενάριο αυτόματου ποτίσματος**

Το σενάριο αυτόματου ποτίσματος βασίζεται στη μέτρηση της υγρασίας του χώματος καθώς και στην μέτρηση της φωτεινής ακτινοβολίας. Έτσι, η ενεργοποίησή του στοχεύει τόσο στην διατήρηση της υγρασίας του χώματος σε επιθυμητά επίπεδα, όσο και στο πότισμα τις ώρες που δεν υπάρχει έντονη ηλιοφάνεια με σκοπό την προστασία των φυτών από ασθένειες.

Οι αισθητήρες που χρησιμοποιήθηκαν για την υλοποίηση του σεναρίου αυτόματου ποτίσματος είναι οι παρακάτω:

1. Soil moisture detector module
2. KY-018 PHOTORESISTOR MODULE
3. 1 x LED

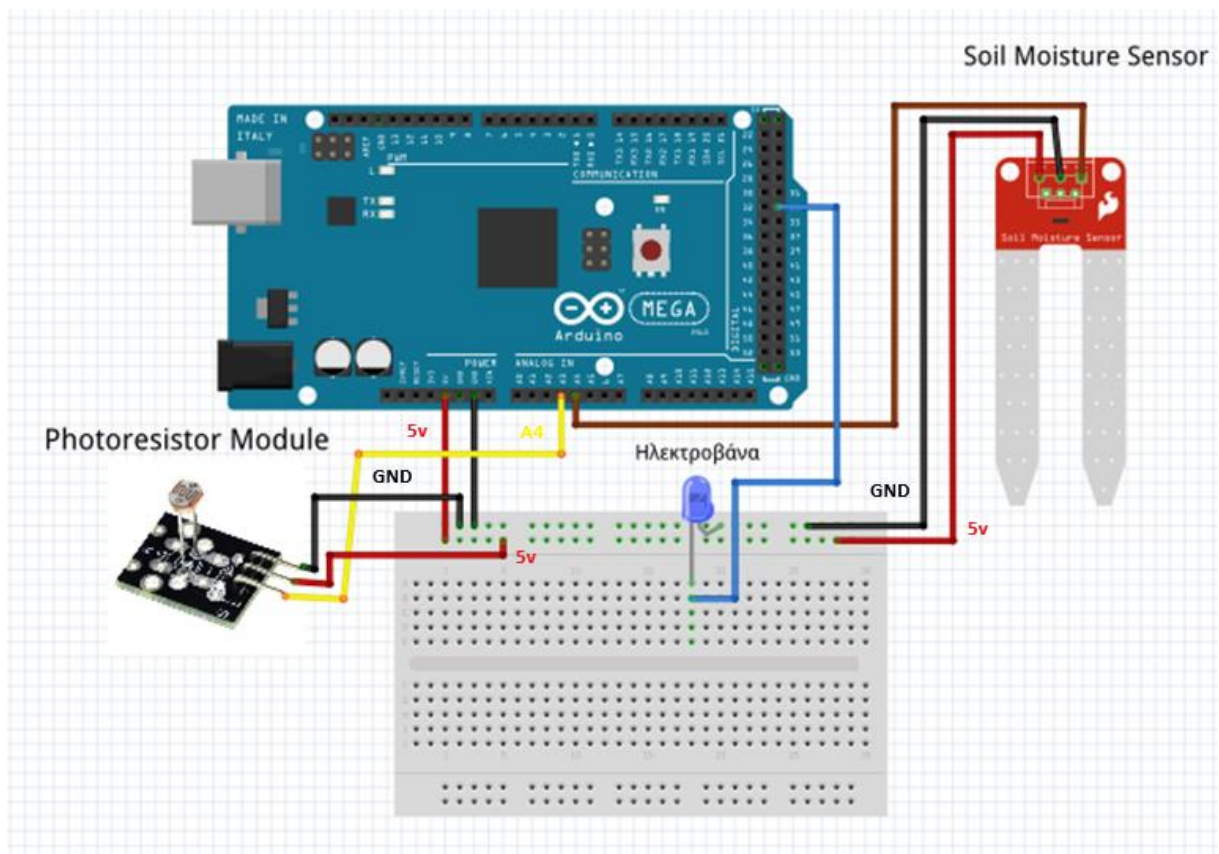
Η υλοποίηση του σεναρίου παρατίθεται στο Παράρτημα Β.1. Χρησιμοποιήθηκε ένα μπλε LED το οποίο υποδηλώνει την ηλεκτρική βαλβίδα η οποία ενεργοποιεί την παροχή του νερού προς τα φυτά. Για την ενεργοποίηση, λοιπόν, του αυτόματου ποτίσματος χρειάζεται να ισχύουν δύο συνθήκες (λογικό «και»):

Αρχικά, διαβάζεται η τιμή της υγρασίας του χώματος από τον αντίστοιχο αισθητήρα. Αν το ποσοστό υγρασίας είναι μικρότερο του 40% ή μικρότερο του 80%, τότε ενεργοποιείται η πρώτη συνθήκη. Με αυτή τη λογική στοχεύουμε στη διατήρηση την υγρασίας σε επιθυμητά επίπεδα (ούτε πολύ υγρό, ούτε πολύ ξηρό).

Παράλληλα, διαβάζεται η τιμή του φωτός μέσω της φωτοαντίστασης. Η τιμή αυτή πρέπει να είναι μικρότερη από 60% για να ισχύει η δεύτερη συνθήκη. Έτσι αποτρέπεται το πότισμα τις ώρες έντονης ηλιοφάνειας, γεγονός το οποίο μπορεί να προκαλέσει ασθένειες στα φυτά.

Για τη λειτουργία του σεναρίου δεν απαιτείται κάποια βιβλιοθήκη.

Παρακάτω φαίνεται το κύκλωμα που υλοποιήθηκε για το σύστημα αυτόματου ποτίσματος:



Εικόνα 4.3: Κύκλωμα σεναρίου αυτόματου ποτίσματος



*Εικόνα 4.4: Αισθητήρας ελέγχου αυτόματου ποτίσματος*

Για την υλοποίηση του κώδικα, δεν είναι απαραίτητη κάποια βιβλιοθήκη. Πιο συγκεκριμένα το μόνο που απαιτείται είναι η καταχώρηση των μεταβλητών που παρουσιάζονται στο παράρτημα B.1.1.4 . Στο παράρτημα B.1.1.3 και B.1.1.6 γίνεται η αρχικοποίηση των μεταβλητών δικτύου. Επίσης, στο παράρτημα B.1.1.5 γίνεται η αρχικοποίηση μεταβλητών της κάρτας μνήμης του Ethernet shield που λειτουργεί ως web server. Στο παράρτημα B.1.2.1 φαίνεται η σύνδεση του Ethernet shield με το διαδίκτυο . Στο παράρτημα B.1.2.2 διακρίνονται οι εισοδοί/έξοδοι των αισθητήρων. Στο παράρτημα B.1.3.2 φαίνεται η λειτουργία της συνάρτησης που υλοποιήθηκε για το σενάριο αυτόματου ποτίσματος. Έπειτα, στο παράρτημα B.1.3.6 και B.1.3.7 φαίνεται ο υπολογισμός της συνάρτησης Thingspeak και η επικοινωνία του Arduino με το διαδίκτυο. Τέλος στο παράρτημα B.1.3.8 και B.1.3.9 παρουσιάζονται αντίστοιχα η επικοινωνία του Arduino με την βάση δεδομένων και την κάρτα μνήμης του Ethernet shield.

#### **4.4 Σενάριο ελέγχου εξαερισμού και ποιότητας αέρα**

Το συγκεκριμένο σενάριο πραγματοποιήθηκε για τον έλεγχο της θερμοκρασίας μέσα στο θερμοκήπιο καθώς επίσης και για τον έλεγχο της ποιότητας του αέρα. Πιο συγκεκριμένα είναι επιθυμητό να υπάρχει ένα συγκεκριμένο εύρος θερμοκρασίας για καλύτερες συνθήκες ανάπτυξης των φυτών ώστε να μεγιστοποιηθεί η παραγωγή. Ακόμη, σημαντική είναι και διατήρηση καλής ποιότητας του αέρα απαλλαγμένου από επιβλαβείς αέρια και άλλες επικίνδυνες ουσίες που προκύπτουν από αγροτικές εργασίες (π.χ. καπνός από καύση ξηρών χόρτων). Η υλοποίηση του σεναρίου παρατίθεται στο Παράρτημα B.1. Για την επίτευξη όλων των παραπάνω, υλοποιήθηκε ένα αυτόματο σύστημα εξαερισμού με τη χρήση ενός

ανεμιστήρα. Για τη λειτουργία του σεναρίου απαιτείται η ενσωμάτωση στον κώδικα του Arduino της βιβλιοθήκης *dht.h*.

Οι αισθητήρες που χρησιμοποιήθηκαν για το σύστημα εξαερισμού και ελέγχου της ποιότητας του αέρα είναι οι παρακάτω:

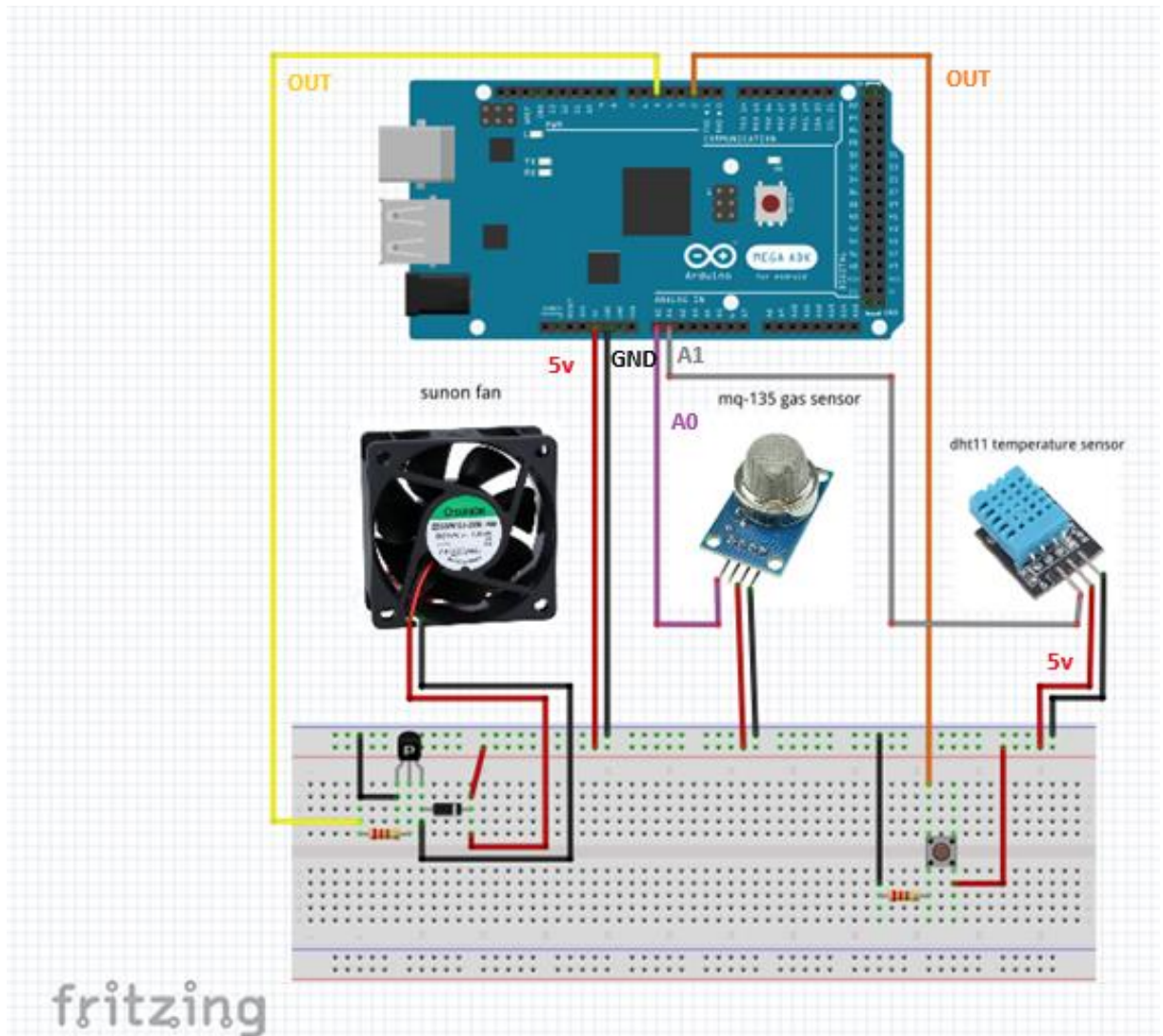
1. MQ-135 Gas Sensor Module
2. Temperature and humidity sensor module
3. 16 Fan 60X60X25 EB60251S1 SUNON
4. 1x PN2222 Transistor
5. 1x N4001 diode
6. 2x 270 Ω Resistor
7. 1 x Button
8. 1 x LED (κλιματιστικό)

Για το σύστημα εξαερισμού υπάρχουν τα εξής σενάρια λειτουργίας αναλόγως τις συνθήκες:

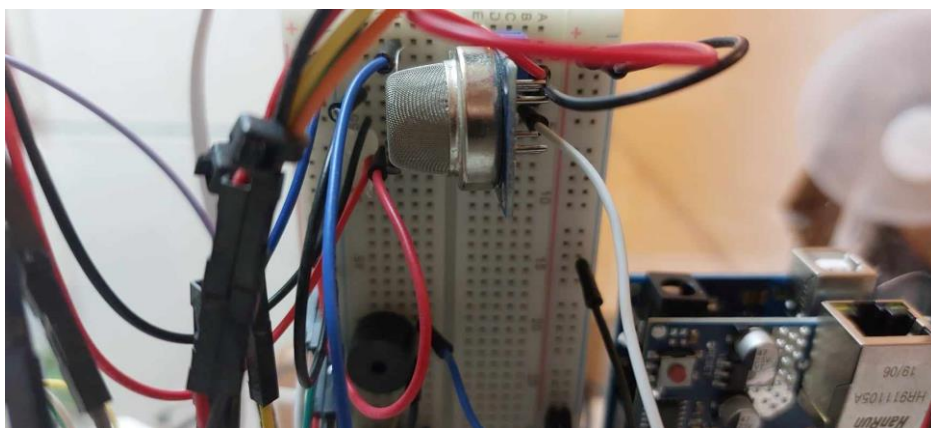
1. Όταν ο αισθητήρας ανίχνευσης επιβλαβών ουσιών εντοπίσει συγκέντρωση αερίων μεγαλύτερη της τιμής που έχουμε ορίσει (αναλογική τιμή 400), τότε θα ενεργοποιηθεί σύστημα εξαερισμού μέχρι να καθαρίσει η ατμόσφαιρα και η τιμή να πέσει κάτω από 400.
2. Αν η θερμοκρασία είναι μέσα στο εύρος από 21 μέχρι και 24 βαθμούς κελσίου τότε θερμοκρασία θεωρείται φυσιολογική και ο χρήστης μπορεί να ενεργοποιήσει τον ανεμιστήρα χειροκίνητα με τη βοήθεια του button αν αυτό είναι επιθυμητό.
3. Αν η θερμοκρασία είναι μεγαλύτερη από 24 βαθμούς όπως συμβαίνει τους καλοκαιρινούς μήνες τότε το σύστημα εξαερισμού θα ενεργοποιείται αυτόματα μέχρι η θερμοκρασία να φτάσει κάτω από 24 βαθμούς.
4. Αν η υγρασία είναι μεγαλύτερη από 50% όπως συμβαίνει συνήθως μετά την άρδευση, τότε το σύστημα εξαερισμού θα ενεργοποιείται αυτόματα μέχρι εκείνη να πέσει κάτω από αυτό το όριο.

Το κύκλωμα που υλοποιήθηκε για το σενάριο ελέγχου εξαερισμού και ελέγχου ποιότητας αέρα φαίνεται παρακάτω:





Εικόνα 4.5: : Κύκλωμα σεναρίου εξαερισμού



Εικόνα 4.6: Αισθητήρας ελέγχου εξαερισμού

Για την υλοποίηση του κώδικα, είναι απαραίτητη η βιβλιοθήκη θερμοκρασίας. Πιο συγκεκριμένα η βιβλιοθήκη θερμοκρασίας dht παρουσιάζεται στο παράρτημα B.1.1.2 . Στο παράρτημα B.1.1.3 και B.1.1.6 γίνεται η αρχικοποίηση των μεταβλητών δικτύου. Επίσης, Στο

παράρτημα B.1.1.5 γίνεται η αρχικοποίηση μεταβλητών της κάρτας μνήμης του Ethernet shield που λειτουργεί ως web server. Στο παράρτημα B.1.2.1 φαίνεται η σύνδεση του Ethernet shield με το διαδίκτυο. Στο παράρτημα B.1.2.2 διακρίνονται οι είσοδοι/έξοδοι των αισθητήρων. Στο παράρτημα B.1.3.3 φαίνεται η λειτουργία της συνάρτησης που υλοποιήθηκε για το σενάριο ελέγχου εξαερισμού και ποιότητας αέρα. Έπειτα, στο παράρτημα B.1.3.6 και B.1.3.7 φαίνεται ο υπολογισμός της συνάρτησης Thingspeak και η επικοινωνία του Arduino με το διαδίκτυο. Τέλος στο παράρτημα B.1.3.8 και B.1.3.9 παρουσιάζονται αντίστοιχα η επικοινωνία του Arduino με την βάση δεδομένων και την κάρτα μνήμης του Ethernet shield.

## 4.5 Σενάριο ανίχνευσης φωτιάς και πλημμύρας

Η ανίχνευση της φωτιάς και της πλημμύρας είναι πολύ σημαντική στις μέρες μας καθώς δίνεται η δυνατότητα στο χρήστη να ενημερώνεται άμεσα για τις συνθήκες που επικρατούν στο θερμοκήπιο, ώστε να εξασφαλιστεί η έγκαιρη επέμβαση όποτε αυτή κριθεί αναγκαία.

Οι αισθητήρες που χρησιμοποιήθηκαν για την ανίχνευση της φωτιάς και της πλημμύρας είναι οι παρακάτω:

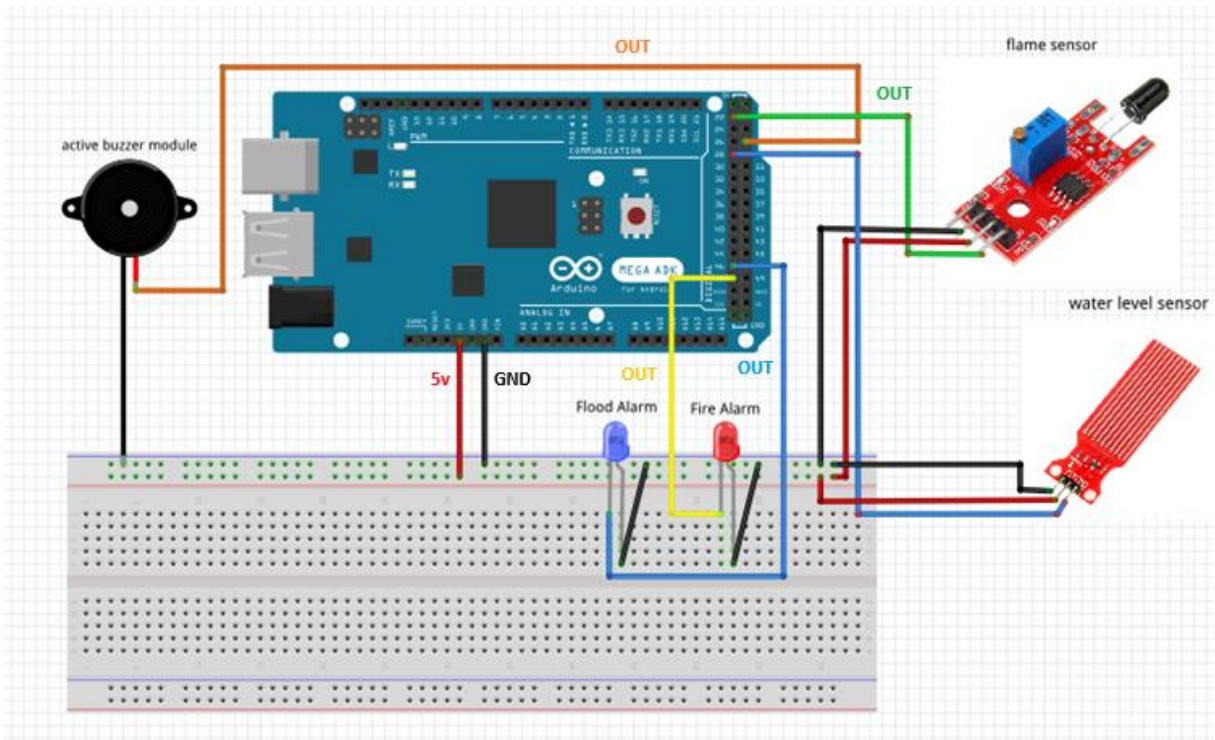
1. Flame Sensor Module IR Sensor Detector
2. Water Level Sensor
3. 1x Buzzer
4. 2x LEDs (fire, flood)

Η υλοποίηση του σεναρίου παρατίθεται στο Παράρτημα B.1. Το σενάριο ανίχνευσης της φωτιάς και της πλημμύρας λειτουργεί ως εξής:

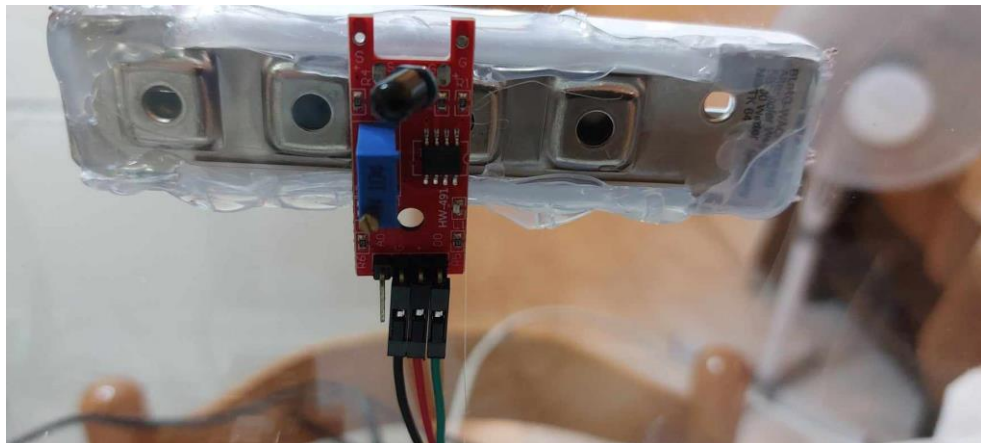
1. Όταν ο αισθητήρας στάθμης νερού ανιχνεύσει υπερχειλίση της δεξαμενής νερού, τότε θα πάρει την τιμή λογικό 1, θα ανάψει το μπλε LED «Flood Alarm» και θα ενεργοποιηθεί η ηχητική ένδειξη (Buzzer) με καθορισμένη ηχητική σήμανση.
2. Όταν ο αισθητήρας φωτιάς ανιχνεύσει φλόγα, τότε θα πάρει την τιμή λογικό 1, θα ανάψει το κόκκινο LED «Fire Alarm» και θα ενεργοποιηθεί η ηχητική ένδειξη (Buzzer) με καθορισμένη ηχητική σήμανση.

Σε περίπτωση που και οι δύο αισθητήρες ανιχνεύσουν την ίδια στιγμή πλημμύρα και φλόγα αντίστοιχα, τότε θα πάρουν την τιμή λογικό 1, θα ανάψουν τα δύο LED Flood & Fire Alarm ταυτόχρονα και θα ενεργοποιηθεί η ηχητική ένδειξη (Buzzer) με καθορισμένη ηχητική σήμανση. Για τη λειτουργία του σεναρίου δεν απαιτείται κάποια βιβλιοθήκη.

Το κύκλωμα που υλοποιήθηκε για το σενάριο ανίχνευσης φωτιάς και πλημμύρας φαίνεται παρακάτω:



Εικόνα 4.7: Κύκλωμα σεναρίου ανίχνευσης φωτιάς και πλημμύρας



Εικόνα 4.8: Αισθητήρας ανίχνευσης φωτιάς

Για την υλοποίηση του κώδικα, δεν είναι απαραίτητη κάποια βιβλιοθήκη. Πιο συγκεκριμένα το μόνο που απαιτείται είναι η καταχώρηση των μεταβλητών που παρουσιάζονται στο παράρτημα B.1.1.8 . Στο παράρτημα B.1.1.3 και B.1.1.6 γίνεται η αρχικοποίηση των μεταβλητών δικτύου. Επίσης, στο παράρτημα B.1.1.5 γίνεται η αρχικοποίηση μεταβλητών της κάρτας μνήμης του Ethernet shield που λειτουργεί ως web server. Στο παράρτημα B.1.2.1 φαίνεται η σύνδεση του Ethernet shield με το διαδίκτυο . Στο παράρτημα B.1.2.2 διακρίνονται οι είσοδοι/έξοδοι των αισθητήρων. Στο παράρτημα B.1.3.5 φαίνεται η λειτουργία της συνάρτησης που υλοποιήθηκε για το σενάριο ανίχνευσης φωτιάς και πλημμύρας. Έπειτα, στο παράρτημα B.1.3.6 και B.1.3.7 φαίνεται ο υπολογισμός της



συνάρτησης Thingspeak και η επικοινωνία του Arduino με το διαδίκτυο. Τέλος στο παράρτημα B.1.3.8 και B.1.3.9 παρουσιάζονται αντίστοιχα η επικοινωνία του Arduino με την βάση δεδομένων και την κάρτα μνήμης του Ethernet shield.

## 4.6 Σενάριο ασφάλειας εξωτερικής πόρτας

Η ασφάλεια της εξωτερικής πόρτας πραγματοποιήθηκε με τον συνδυασμό μυστικού κωδικού και με το πάτημα προκαθορισμένης χρονικής διάρκειας του αισθητήρα αφής. Το σενάριο αυτό υλοποιήθηκε για την διευκόλυνση του χρήστη στο ξεκλείδωμα της εξωτερικής πόρτας χωρίς κλειδιά. Για τη λειτουργία του σεναρίου απαιτείται να ενσωματωθεί στον κώδικα του Arduino η βιβλιοθήκη *IRremote.h*. Η υλοποίηση του σεναρίου παρατίθεται στο Παράρτημα B.2.

Οι αισθητήρες που χρησιμοποιήθηκαν για την ασφάλεια εξωτερικής πόρτας είναι οι παρακάτω:

1. Mini IR Infrared Pyroelectric PIR Body Motion Human Sensor Detector Module
2. Infrared IR Receiver Module Wireless Remote-Control Kit For Arduino
3. DFRobot Capacitive Touch Sensor
4. 1 x Button
5. 4 x LEDs (φωτισμού, αισθητήρα αφής, κωδικού, πόρτας)

Αρχικά έχει χρησιμοποιηθεί ο αισθητήρας ανίχνευσης κίνησης για την διευκόλυνση του χρήστη σε συνθήκες έλλειψης φωτισμού στον χώρο της εξωτερικής πόρτας. Όταν ο αισθητήρας εντοπίσει κίνηση ανάβει το LED φωτισμού και παραμένει αναμμένο για όση ώρα βλέπει κίνηση. Με βάση τον τρόπο λειτουργίας του αισθητήρα, το LED σβήνει δύο δευτερόλεπτα αφού σταματήσει η κίνηση στον χώρο.

Έπειτα έχει τοποθετηθεί το Button για να υποδηλώνει το αν είναι ανοιχτή ή κλειστή η πόρτα. Όταν το button είναι πατημένο η πόρτα είναι κλειστή. Για να λειτουργήσει το αυτόματο σενάριο θα πρέπει η εξωτερική πόρτα να είναι κλειστή. Προκειμένου να ξεκλειδώσει η πόρτα πρέπει να ισχύουν συγχρόνως δύο συνθήκες (λογικό «και»).

1. Ο χρήστης πρέπει να πατήσει τον αισθητήρα αφής για ένα συγκεκριμένο χρονικό διάστημα (π.χ. 2 δευτερόλεπτα) ώστε να ενεργοποιηθεί η πρώτη συνθήκη. Σε περίπτωση που ο αισθητήρας πατηθεί για λιγότερο ή περισσότερο χρόνο, η συνθήκη απενεργοποιείται. Για την διευκόλυνση του πειράματος έχει τοποθετηθεί ενδεικτικά ένα κόκκινο LED το οποίο δείχνει πότε επιτυχάνουμε τον χρόνο. Αυτό το LED έχει τοποθετηθεί μόνο για επεξηγηματικούς σκοπούς μιας και δεν θα υπάρχει στην πραγματική κατασκευή.





*Εικόνα 4.10: Αισθητήρας ασφάλειας εξωτερικής πόρτας*

Για την υλοποίηση του κώδικα, απαραίτητη προϋπόθεση είναι η εισαγωγή βιβλιοθηκών (Παράρτημα B.2.1.1). Πιο συγκεκριμένα απαιτούνται οι βιβλιοθήκες για την επικοινωνία του Ethernet shield με το Arduino Uno, η βιβλιοθήκη για την επικοινωνία με το Thingspeak και η βιβλιοθήκη για την επικοινωνία με τον αισθητήρα απομακρυσμένου ελέγχου IR . Στο παράρτημα B.2.1.2 γίνεται η αρχικοποίηση των μεταβλητών δικτύου. Η αρχικοποίηση των μεταβλητών που αφορούν τα κουμπιά, τα led και τον δέκτη του αισθητήρα απομακρυσμένου ελέγχου IR παρουσιάζεται στο παράρτημα B.2.1.3. Στο παράρτημα B.2.1.5 παρουσιάζονται οι είσοδοι/ έξοδοι των μεταβλητών. Στο παράρτημα B.2.1.7 φαίνεται η λειτουργία της συνάρτησης που υλοποιήθηκε για το σενάριο ασφάλειας εξωτερικής πόρτας. Τέλος στο παράρτημα B.2.1.8 παρουσιάζεται αντίστοιχα η λειτουργία του αισθητήρα απομακρυσμένου ελέγχου.

#### **4.7 Σενάριο ελέγχου παντζουριών και εσωτερικού φωτισμού**

Στο σενάριο αυτό έχει αυτοματοποιηθεί η λειτουργία του εσωτερικού φωτισμού ανάλογα με την ώρα της ημέρας, καθώς και ο έλεγχος των παντζουριών. Ο έλεγχος του εσωτερικού φωτισμού μπορεί να ενισχύσει την παραγωγή τις ώρες που δεν υπάρχει επαρκής φυσικός φωτισμός και ο έλεγχος των παντζουριών προσφέρει προστασία από το κρύο ή τον άνεμο. Για την λειτουργία του σεναρίου απαιτείται η ενσωμάτωση στον κώδικα του Arduino της βιβλιοθήκης *DS1302.h*. Η υλοποίηση του σεναρίου παρατίθεται στο Παράρτημα B.1.

Οι αισθητήρες που χρησιμοποιήθηκαν για τον έλεγχο παντζουριών και εσωτερικού φωτισμού είναι οι παρακάτω:

1. DS1302 Real Time Clock Module
2. 1x Photoresistor
3. 2x LEDs (παντζούρια, εσωτερικός φωτισμός)

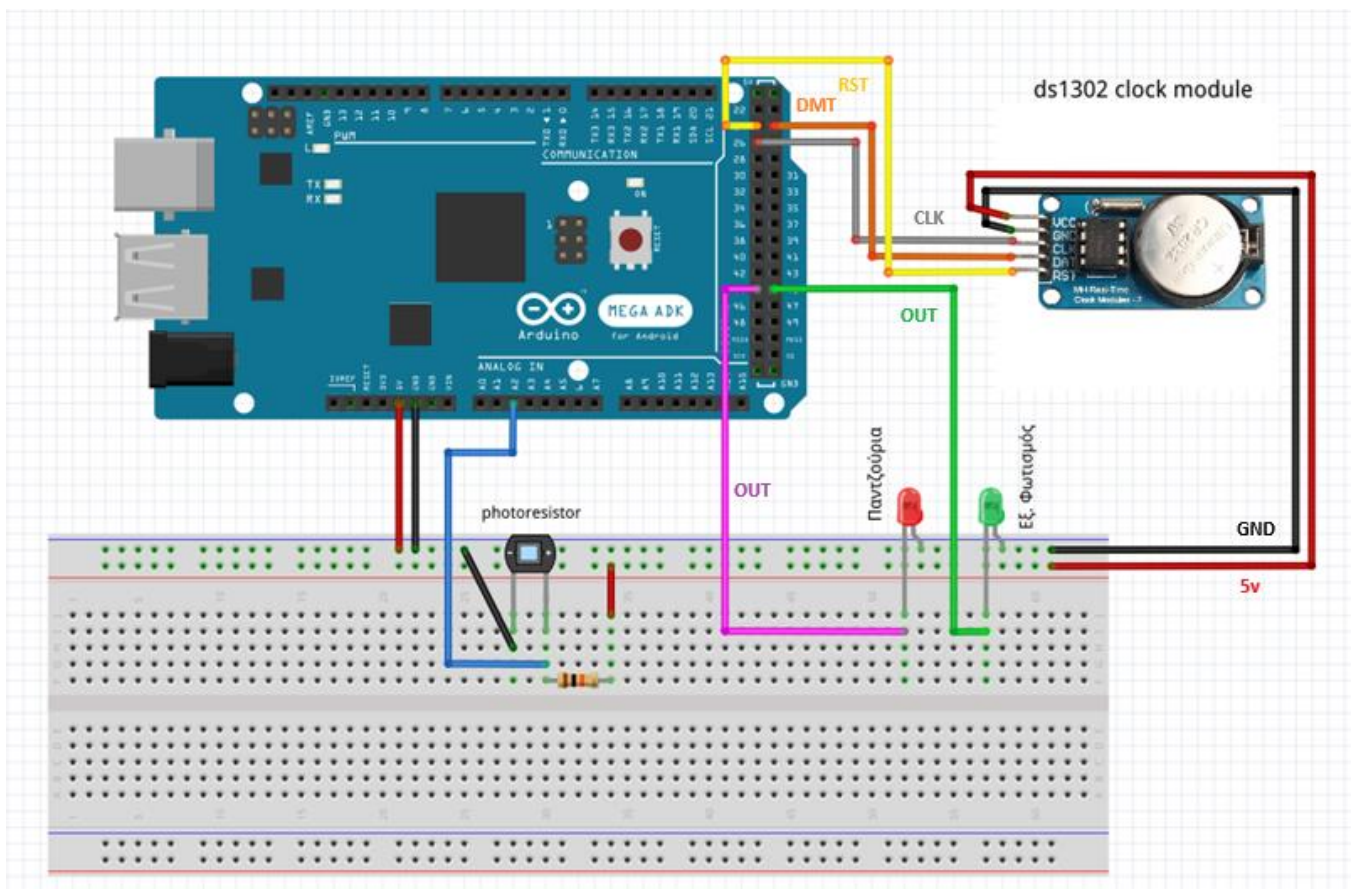
#### 4. 1x Resistor 10kΩ

Η λειτουργία του σεναρίου γίνεται με τον εξής τρόπο. Όταν ο αισθητήρας φωτοαντίστασης εντοπίσει τιμή φυσικού φωτισμού κάτω από 50% (δύση ηλίου) τότε ενεργοποιούνται τα εσωτερικά φώτα (πράσινο LED στο κύκλωμα). Αντίστοιχα θα σβήσουν όταν ο φωτισμός ξεπεράσει το 40% κατά το ξημέρωμα.

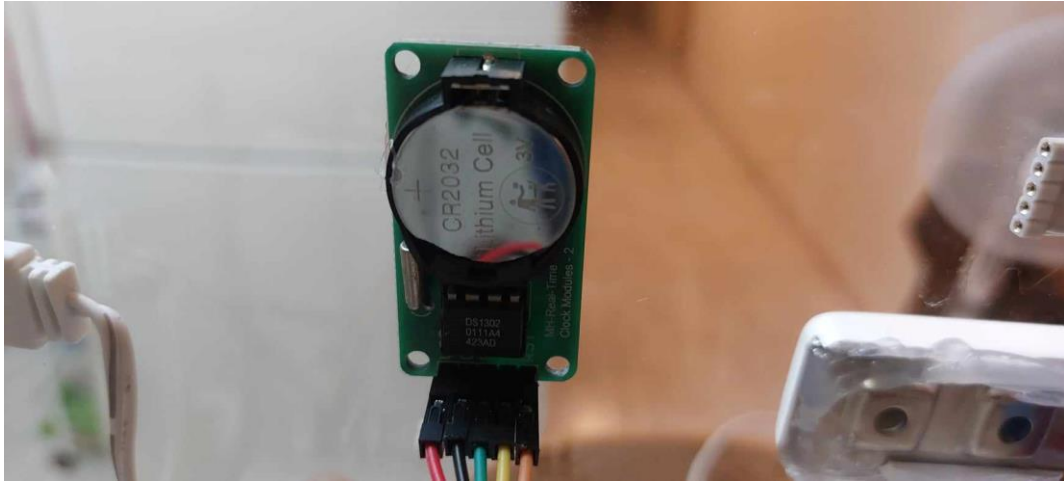
Για τα παντζούρια (κόκκινο LED στο κύκλωμα), εκτός από την παράμετρο του εσωτερικού φωτισμού, έχει οριστεί και η παράμετρος της ώρας της ημέρας. Αυτό έχει πραγματοποιηθεί ώστε το βράδυ που πέφτει η θερμοκρασία, να μπορεί να οριστεί η ώρα κλεισίματος των παντζουριών ανάλογα με τις ανάγκες των φυτών. Η ώρα διαβάζεται με την βοήθεια του αισθητήρα DS1302 Real Time Clock Module.

Επομένως, αν η φωτεινότητα πέσει κάτω από το 50% και η ώρα είναι αυτή που έχει οριστεί (π.χ. 21:00μμ) τότε τα παντζούρια κλείνουν. Αντίστοιχα, αν η φωτεινότητα ξεπεράσει το 40% και η ώρα είναι η καθορισμένη (π.χ. 7:00πμ) τότε τα παντζούρια ανοίγουν.

Το κύκλωμα του σεναρίου ελέγχου παντζουριών και εσωτερικού φωτισμού φαίνεται παρακάτω:



Εικόνα 4.11: Κύκλωμα σεναρίου ελέγχου παντζουριών και εσωτερικού φωτισμού



Εικόνα 4.12: Αισθητήρας ελέγχου παντζουριών και εσωτερικού φωτισμού

Για την υλοποίηση του κώδικα, είναι απαραίτητη η βιβλιοθήκη θερμοκρασίας. Πιο συγκεκριμένα η βιβλιοθήκη θερμοκρασίας ds1302 παρουσιάζεται στο παράρτημα B.1.1.2 . Στο παράρτημα B.1.1.3 και B.1.1.6 γίνεται η αρχικοποίηση των μεταβλητών δικτύου. Επίσης, στο παράρτημα B.1.1.5 γίνεται η αρχικοποίηση μεταβλητών της κάρτας μνήμης του Ethernet shield που λειτουργεί ως web server. Στο παράρτημα B.1.2.1 φαίνεται η σύνδεση του Ethernet shield με το διαδίκτυο . Στο παράρτημα B.1.2.2 διακρίνονται οι εισοδοί/έξοδοι των αισθητήρων. Στο παράρτημα B.1.3.4 φαίνεται η λειτουργία της συνάρτησης που υλοποιήθηκε για το σενάριο ελέγχου παντζουριών και εσωτερικού φωτισμού . Έπειτα, στο παράρτημα B.1.3.6 και B.1.3.7 φαίνεται ο υπολογισμός της συνάρτησης Thingspeak και η επικοινωνία του Arduino με το διαδίκτυο. Τέλος στο παράρτημα B.1.3.8 και B.1.3.9 παρουσιάζονται αντίστοιχα η επικοινωνία του Arduino με την βάση δεδομένων και την κάρτα μνήμης του Ethernet shield.

## 4.8 Σενάριο ελέγχου σκιάστρου

Το σενάριο αυτό υλοποιήθηκε με τη χρήση του αισθητήρα κλίσης (υδραργύρου) και σκοπός του είναι ένδειξη της κατάστασης του σκιάστρου (ανοιχτό ή κλειστό) ώστε ο χρήστης να μπορεί να το παρακολουθεί μέσω της ιστοσελίδας. Η υλοποίηση του σεναρίου παρατίθεται στο Παράρτημα B.1.

Οι αισθητήρες και τα ηλεκτρονικά στοιχεία που χρησιμοποιήθηκαν για τον έλεγχο του σκιάστρου είναι τα παρακάτω:

1. KY-017 Mercury Tilt Switch Module
2. 2x LEDs (ένδειξη ανοιχτού/κλειστού σκιάστρου)
3. 1x Buzzer

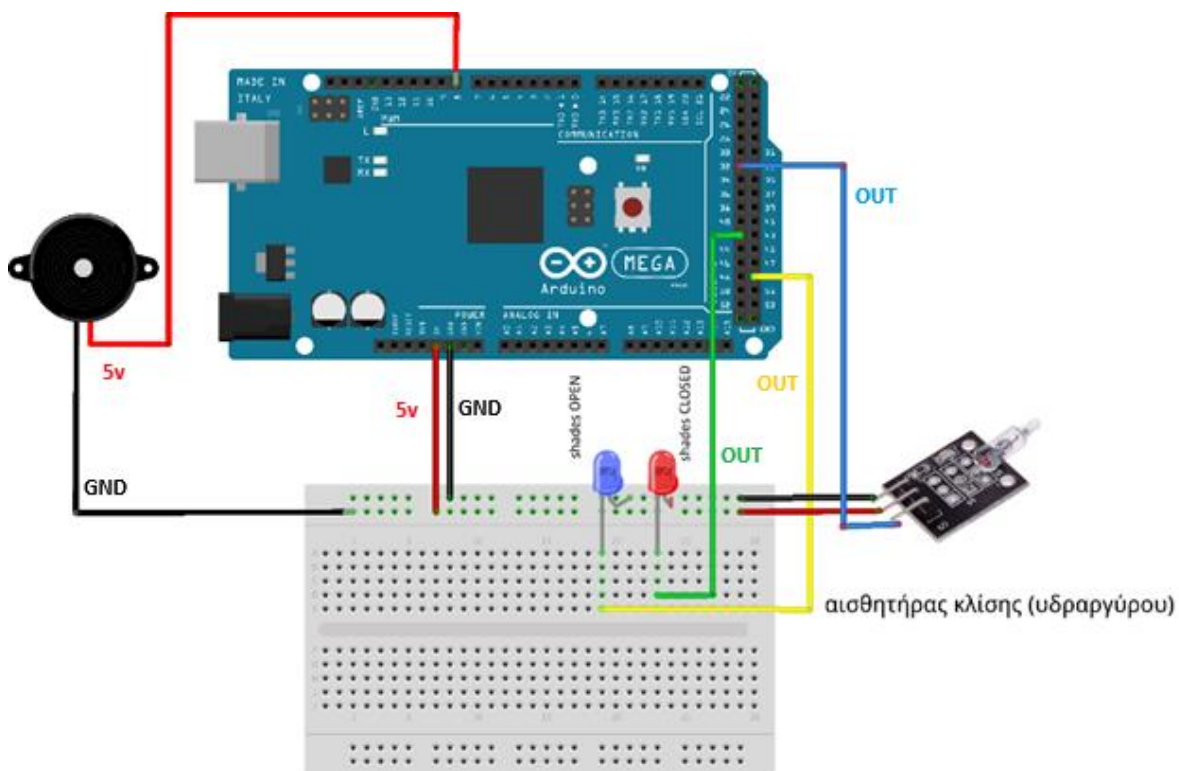


Η αρχή λειτουργίας είναι πολύ απλή εφόσον ο αισθητήρας κλίσης παρέχει ψηφιακή έξοδο ανάλογα με το αν βρίσκεται σε κλίση ή όχι. Έτσι υπάρχουν δύο καταστάσεις:

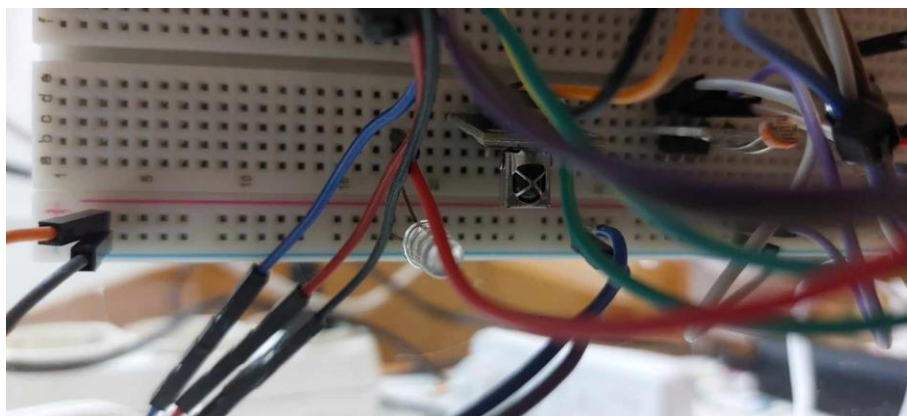
1. Όταν το σκιάστρο είναι κλειστό, τότε ο αισθητήρας παίρνει την τιμή λογικό 0, ανάβει το κόκκινο LED και το buzzer «χτυπάει» περιοδικά.
2. Όταν το σκιάστρο είναι ανοιχτό, τότε ο αισθητήρας παίρνει την τιμή λογικό 1 και ανάβει το μπλε LED

Για τη λειτουργία του σεναρίου δεν απαιτείται κάποια βιβλιοθήκη.

Το κύκλωμα του σεναρίου σκιάστρου φαίνεται παρακάτω:



Εικόνα 4.13: Κύκλωμα σεναρίου ελέγχου σκιάστρου

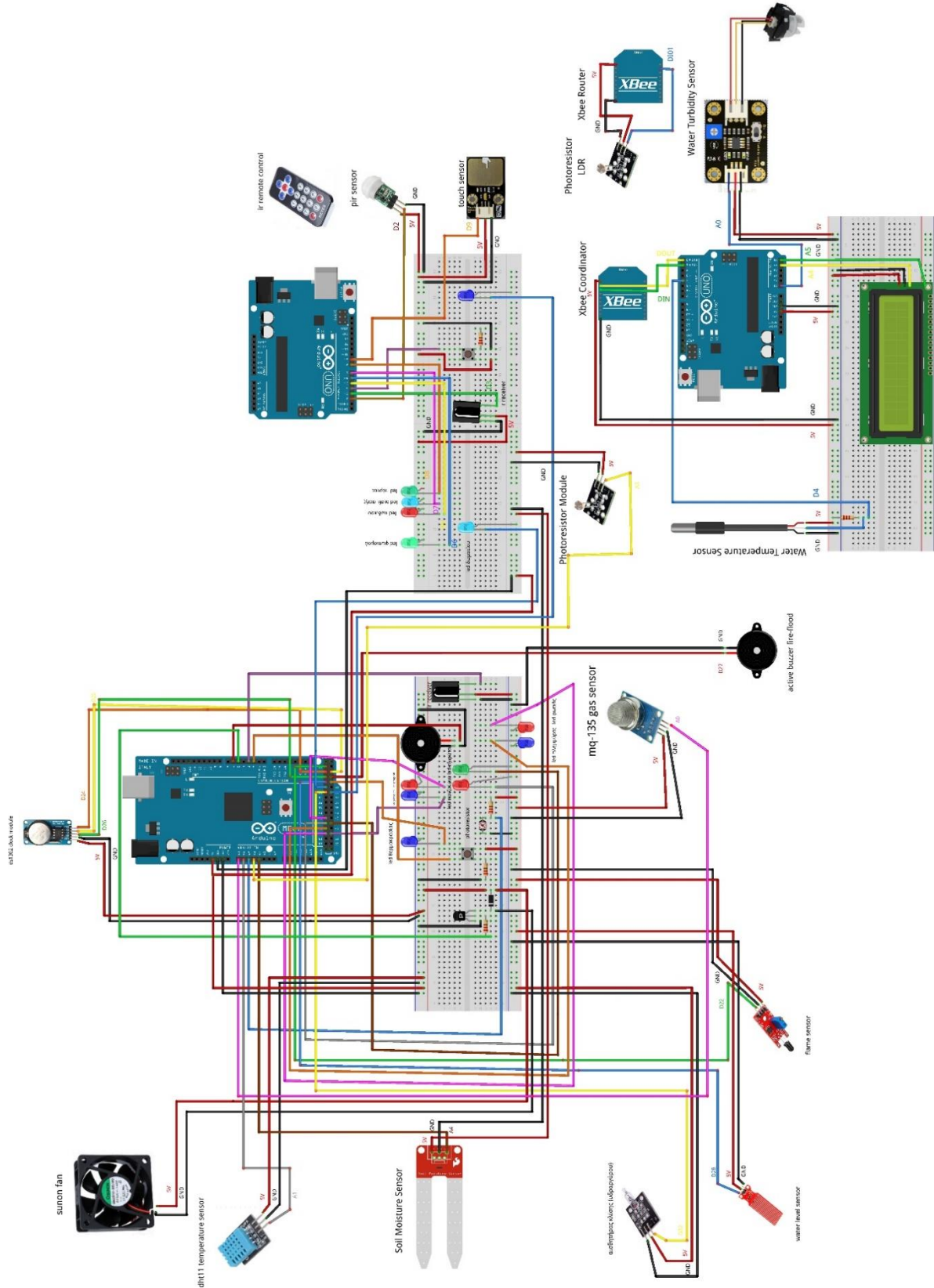


Εικόνα 4.14: Αισθητήρας ελέγχου σκιάστρου

Για την υλοποίηση του κώδικα, δεν είναι απαραίτητη κάποια βιβλιοθήκη. Πιο συγκεκριμένα το μόνο που απαιτείται είναι η καταχώρηση των μεταβλητών που παρουσιάζονται στο παράρτημα B.1.1.7 . Στο παράρτημα B.1.1.3 και B.1.1.6 γίνεται η αρχικοποίηση των μεταβλητών δικτύου. Επίσης, στο παράρτημα B.1.1.5 γίνεται η αρχικοποίηση μεταβλητών της κάρτας μνήμης του Ethernet shield που λειτουργεί ως web server. Στο παράρτημα B.1.2.1 φαίνεται η σύνδεση του Ethernet shield με το διαδίκτυο . Στο παράρτημα B.1.2.2 διακρίνονται οι είσοδοι/έξοδοι των αισθητήρων. Στο παράρτημα B.1.3.2 φαίνεται η λειτουργία της συνάρτησης που υλοποιήθηκε για το σενάριο ελέγχου σκιάστρου. Έπειτα, στο παράρτημα B.1.3.6 και B.1.3.7 φαίνεται ο υπολογισμός της συνάρτησης Thingspeak και η επικοινωνία του Arduino με το διαδίκτυο. Τέλος στο παράρτημα B.1.3.8 και B.1.3.9 παρουσιάζονται αντίστοιχα η επικοινωνία του Arduino με την βάση δεδομένων και την κάρτα μνήμης του Ethernet shield.

## 4.9 Τελικό κύκλωμα

Το τελικό κύκλωμα του συστήματος φαίνεται παρακάτω:

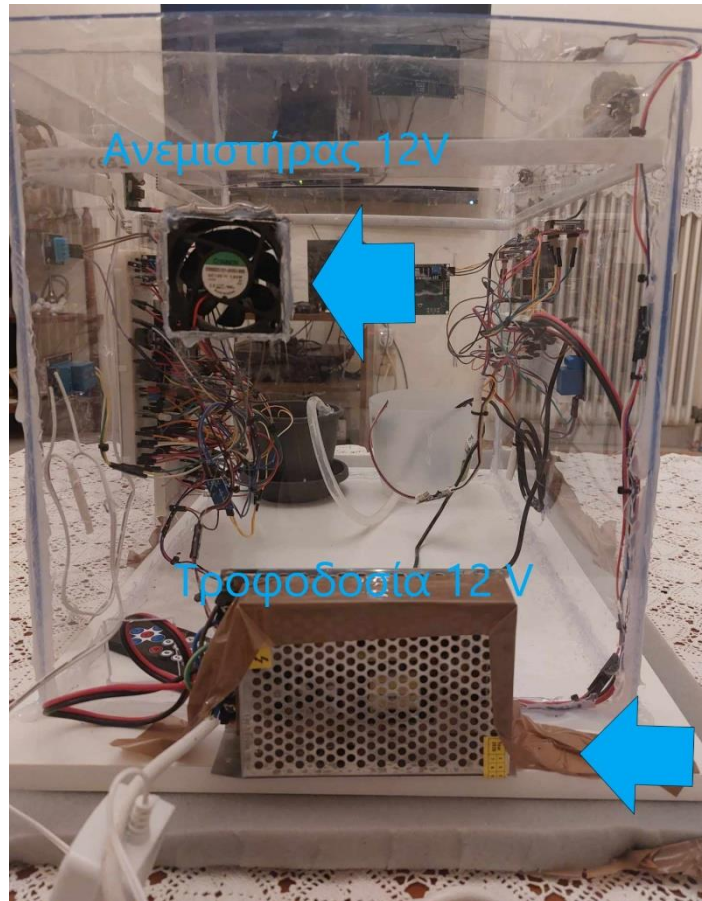


Εικόνα 4.15: Τελικό κύκλωμα συστήματος

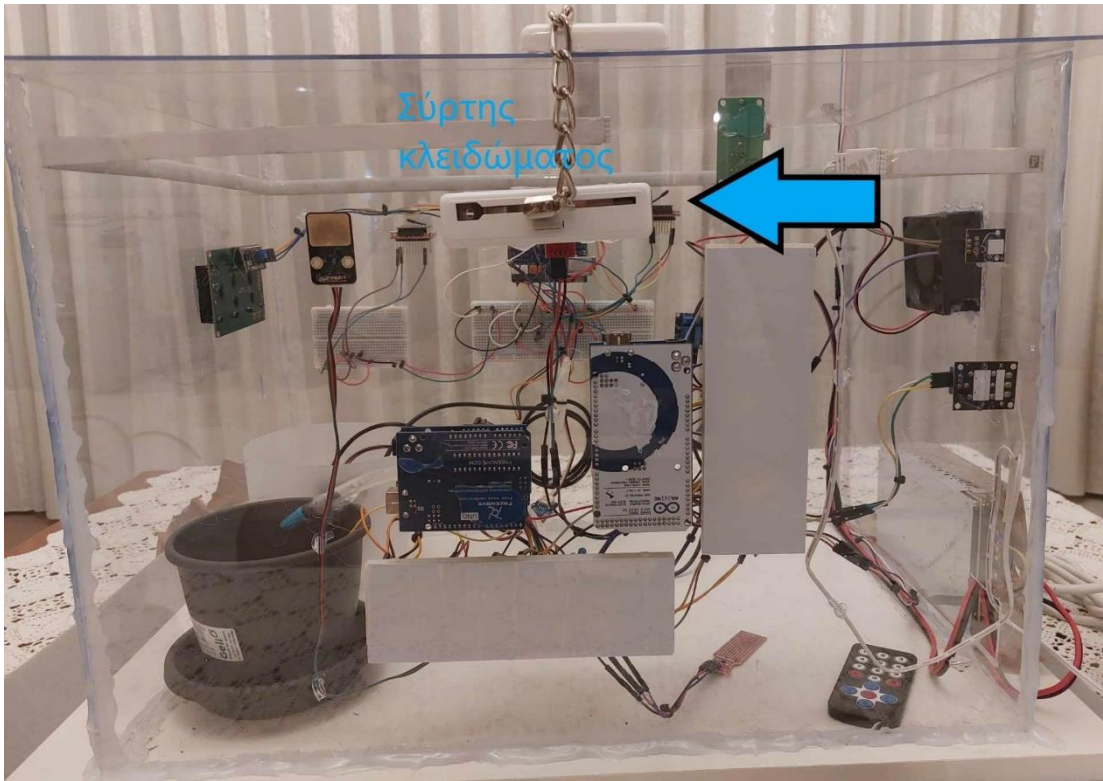




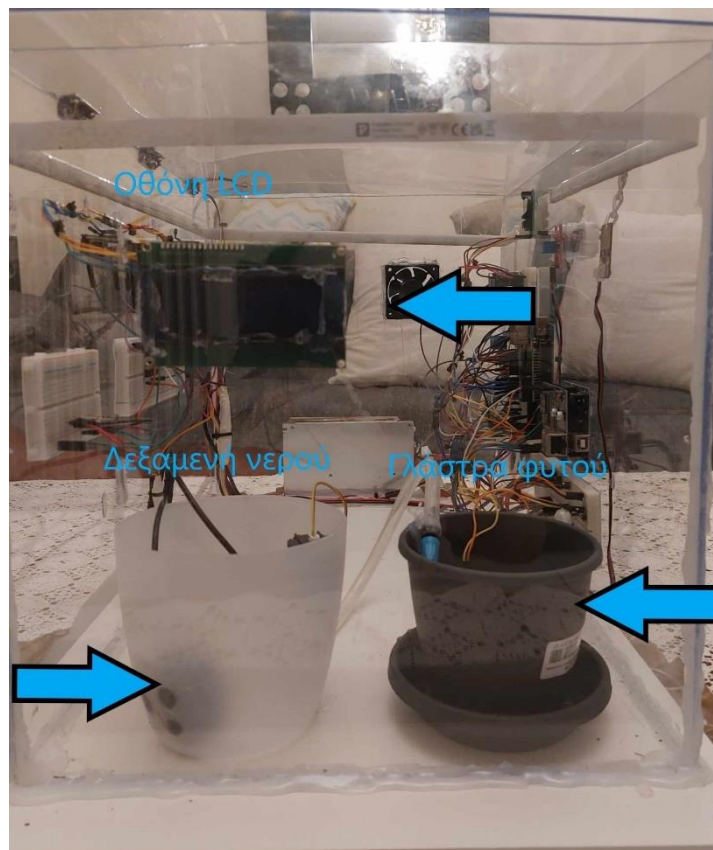
Εικόνα 4.16: Τελικό υλοποιημένο κύκλωμα συστήματος



Εικόνα 4.17: Βόρεια πλευρά συστήματος

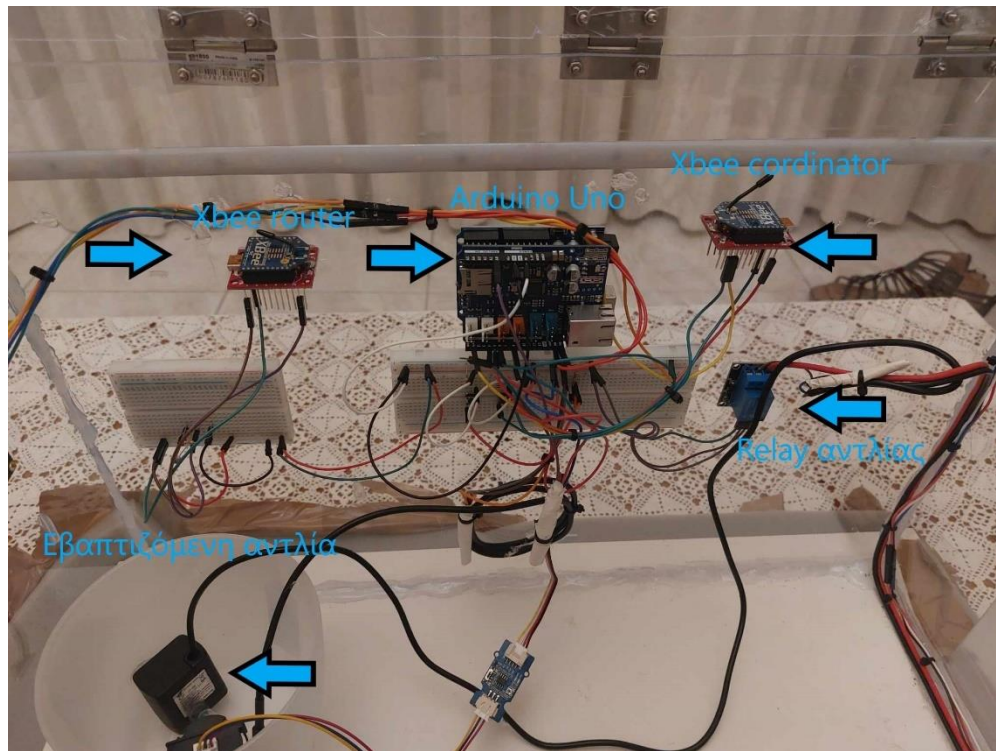


Εικόνα 4.18: Ανατολική πλευρά συστήματος

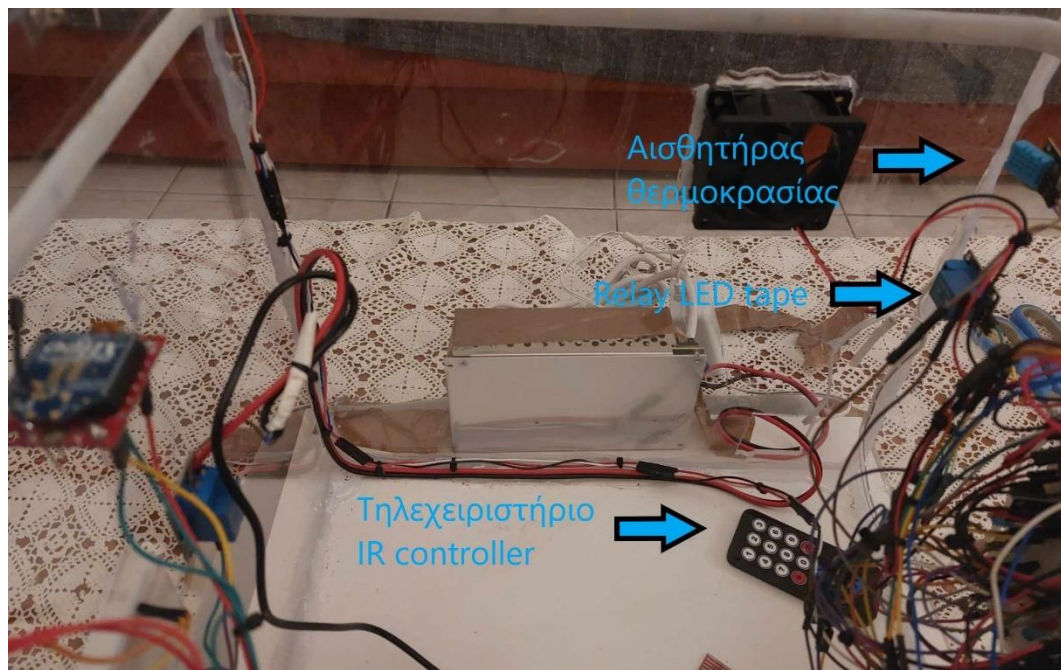


Εικόνα 4.19: Νότια πλευρά συστήματος

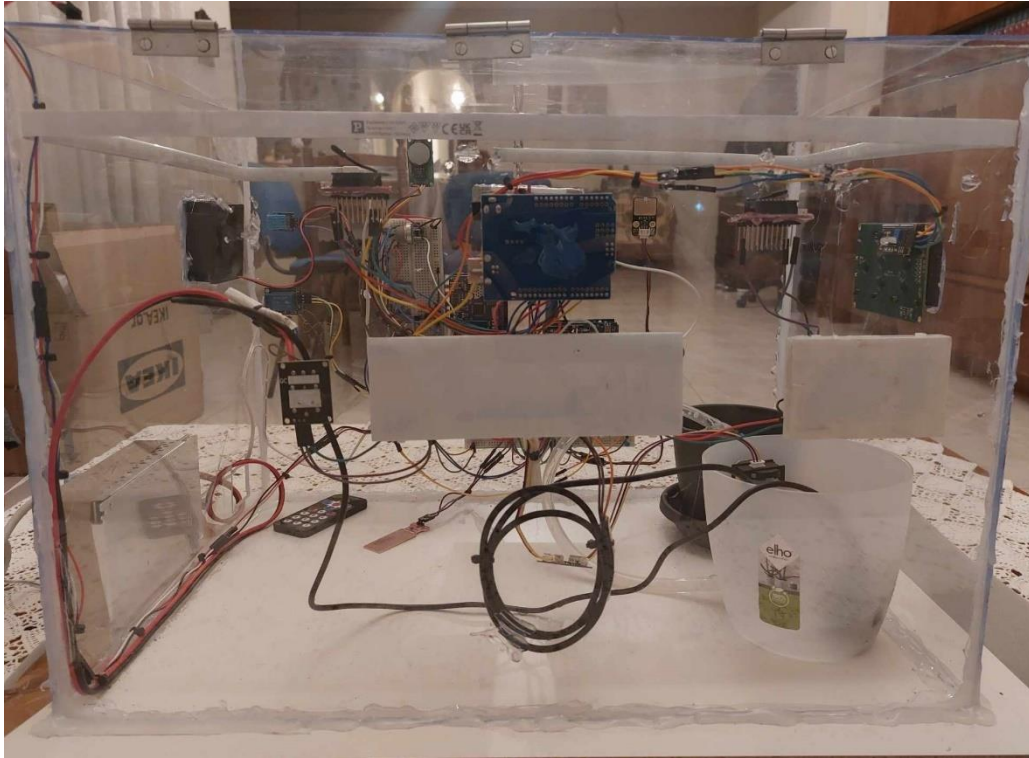




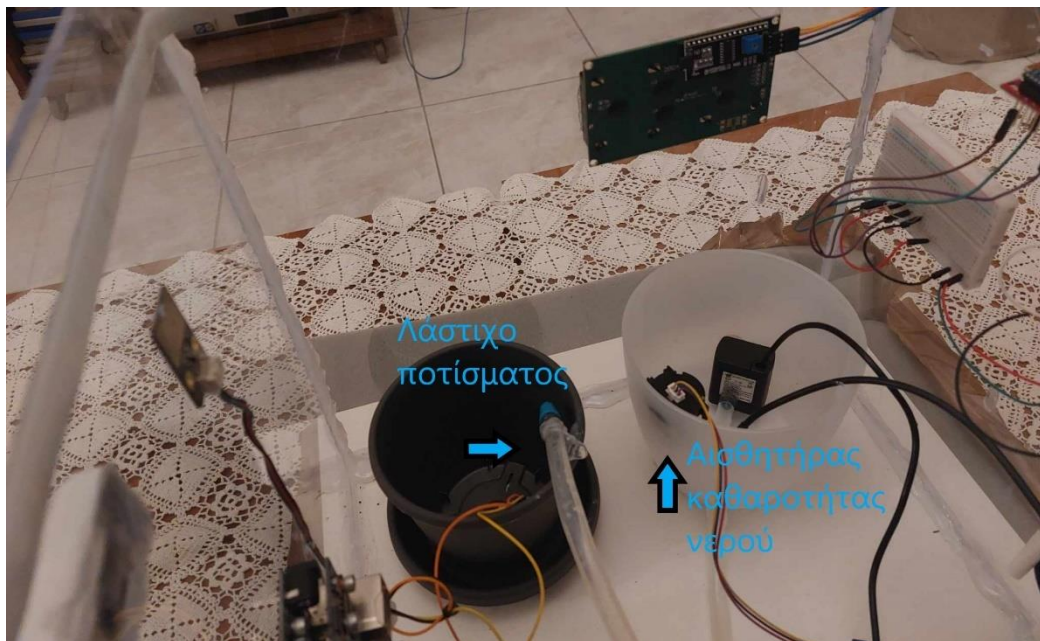
Εικόνα 4.20: Εσωτερική δυτική πλευρά συστήματος



Εικόνα 4.21: Εσωτερική νότια πλευρά συστήματος



Εικόνα 4.22: Δυτική πλευρά συστήματος

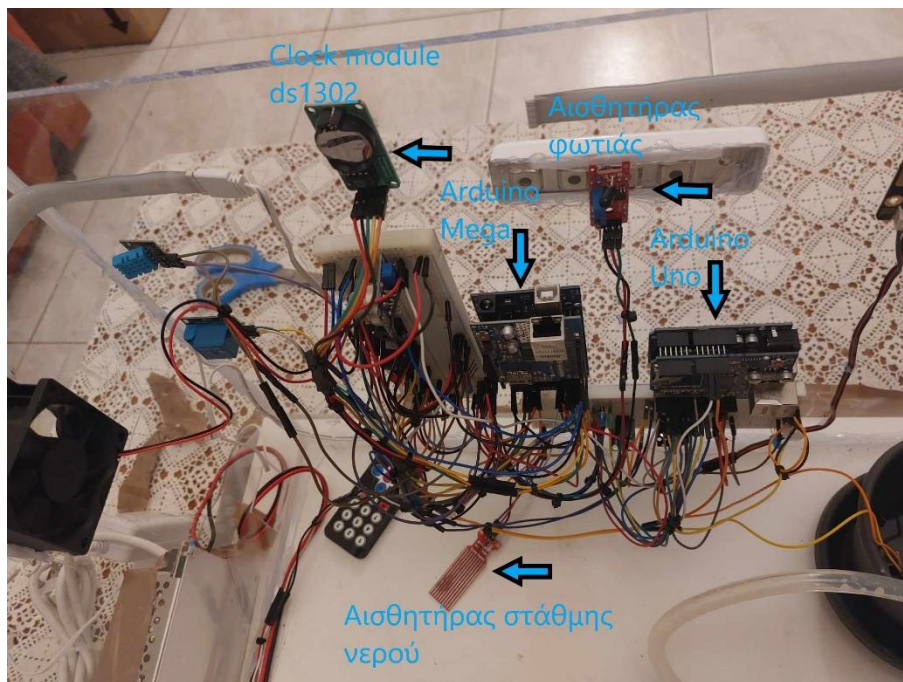


Εικόνα 4.23: Εσωτερική βόρεια πλευρά συστήματος



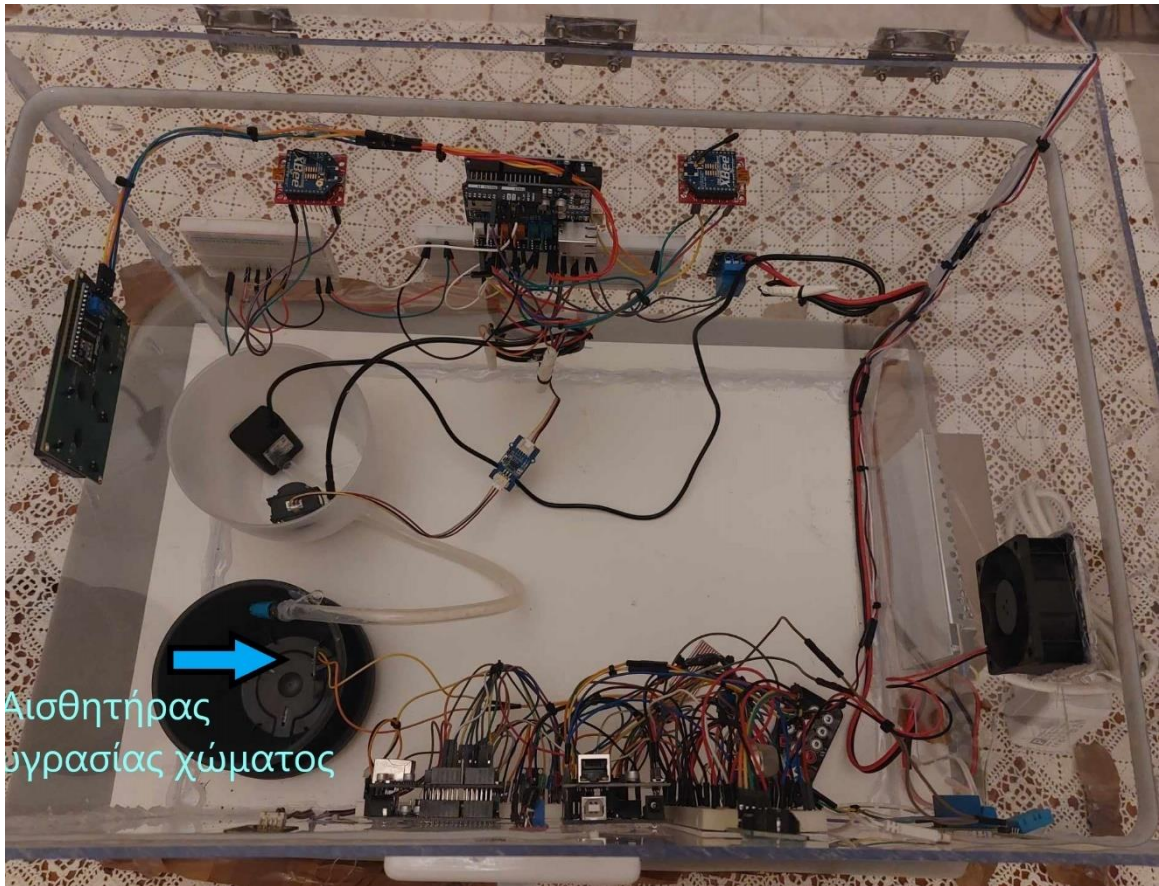


Εικόνα 4.24: Κάτοψη συστήματος



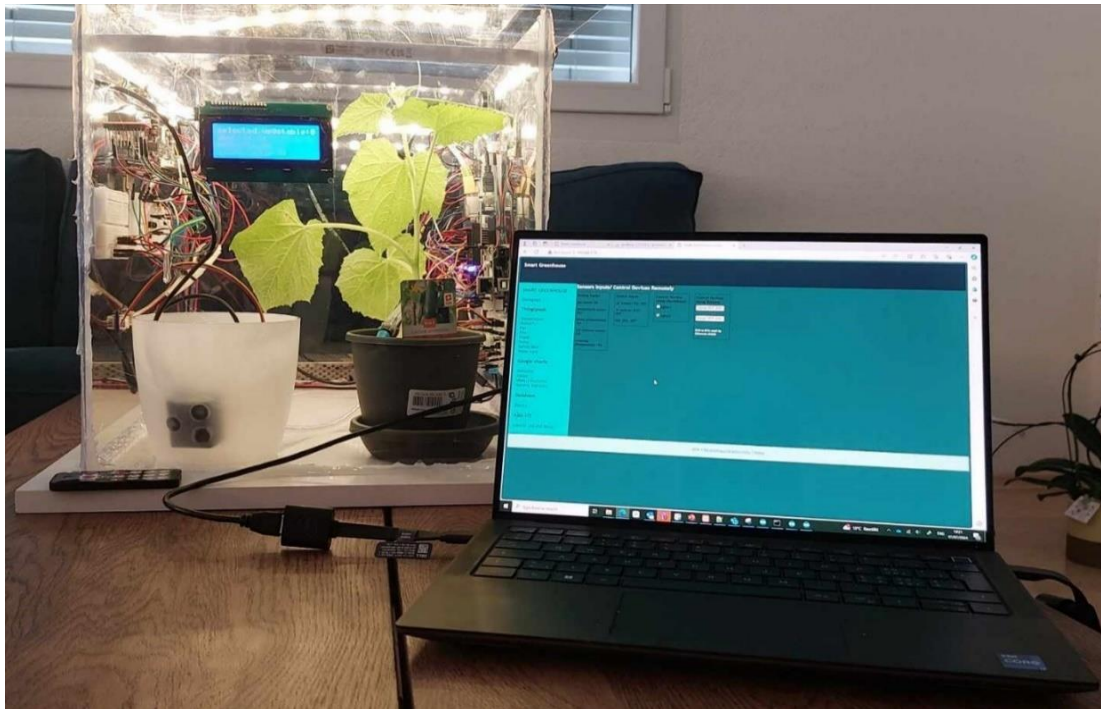
Εικόνα 4.25: Εσωτερική ανατολική πλευρά συστήματος



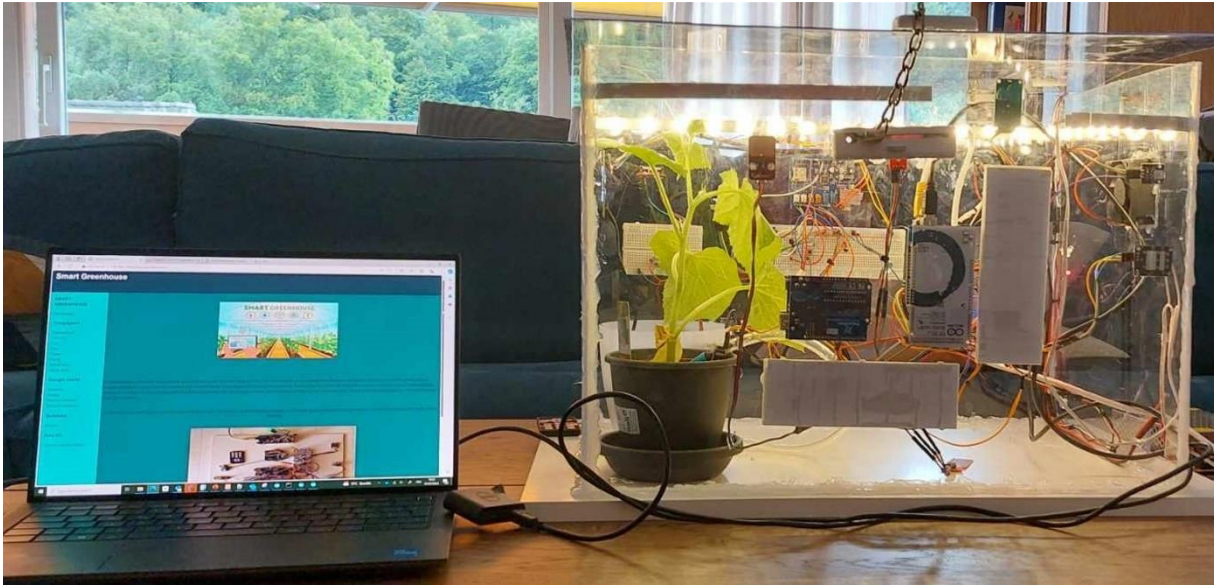


Εικόνα 4.26: Εσωτερική κάτοψη συστήματος

Το τελικό κύκλωμα της μακέτας σε λειτουργία φαίνεται παρακάτω:



Εικόνα 4.27: Τελική μακέτα συστήματος(1)

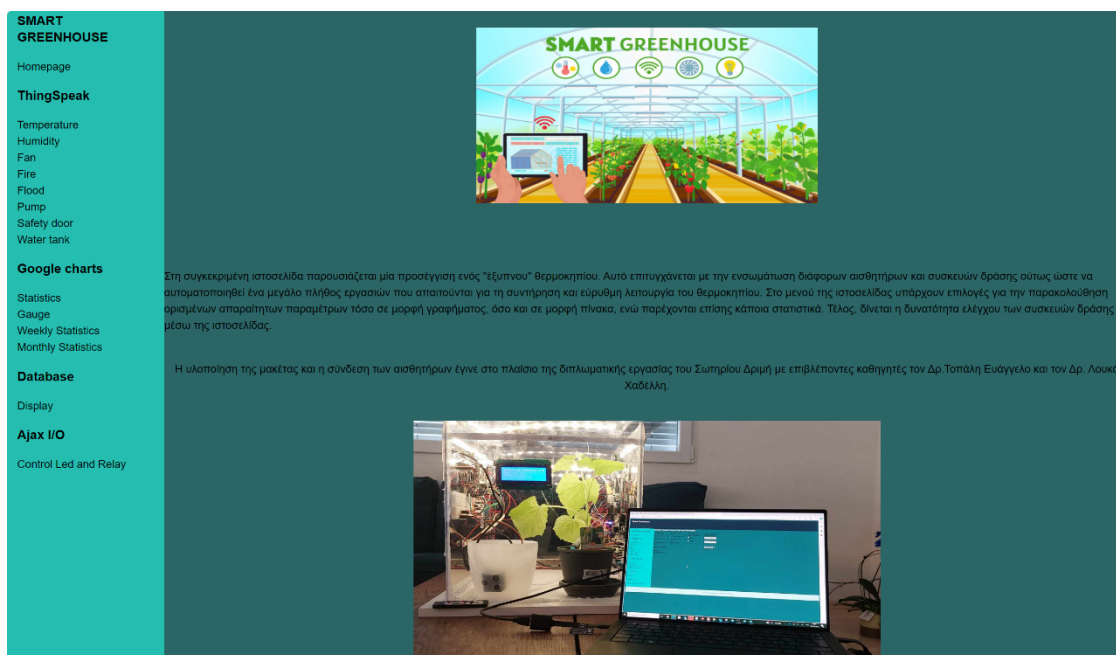


*Εικόνα 4.28: Τελική μακέτα συστήματος(2)*

## 5 Υλοποίηση Διασύνδεσης

### 5.1 Εισαγωγή

Στα πλαίσια της παρούσας διπλωματικής δημιουργήθηκε ένας ιστότοπος ο οποίος δίνει τη δυνατότητα στον χρήστη να παρακολουθεί την κατάσταση του θερμοκηπίου μέσω των αυτοματισμών που αναφέρθηκαν παραπάνω. Αυτό έγινε χρησιμοποιώντας το Ethernet shield σε συνδυασμό με το Arduino, ενώ δημιουργήθηκε και μία Βάση Δεδομένων για την αποθήκευση όλων των μετρήσεων των αισθητήρων. Για την προβολή της ιστοσελίδας δημιουργήθηκε το αρχείο *index.html*, που βρίσκεται στο παράρτημα A16, το οποίο περιέχεται στον φάκελο htdocs του XAMPP μαζί με τα υπόλοιπα αρχεία *.php* και *.js* καθώς και τις εικόνες που απαιτούνται για την ορθή απεικόνιση των περιεχομένων της ιστοσελίδας. Για τη δημιουργία της ιστοσελίδας χρησιμοποιήθηκε το διαδικτυακό πρότυπο από το *quackit.com*. Η κύρια σελίδα (Homepage) φαίνεται παρακάτω:



Εικόνα 5.1: Απεικόνιση του Homepage της ιστοσελίδας

### 5.2 Σύνδεση στο Διαδίκτυο

Η πρόσβαση του Arduino στο διαδίκτυο εξασφαλίζεται μέσω του Ethernet shield και του κατάλληλου κώδικα στο Arduino. Για την σύνδεση απαιτείται η διεύθυνση IP του server και η διεύθυνση IP του Arduino. Τέλος, θα χρειαστεί και το MAC address του Arduino.

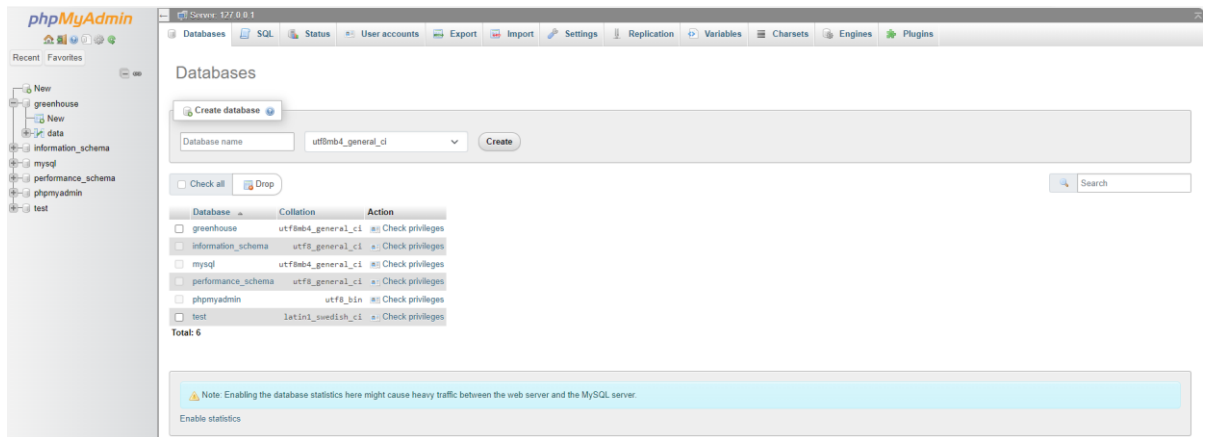
Ακόμη, μέσω του Ethernet τα δεδομένα που συλλέγει το Arduino καταγράφονται σε μία Βάση Δεδομένων για περαιτέρω αξιοποίηση από το χρήστη. Η βάση αυτή δημιουργήθηκε με τη βοήθεια του Apache XAMPP και είναι μία ελεύθερη βάση δεδομένων MySQL. Η καταγραφή των μετρήσεων στη βάση γίνεται μέσω της εντολής *client.print()* για κάθε στοιχείο



που θέλουμε να καταγράψουμε. Επειδή θέλουμε η καταγραφή να είναι συνεχής, η συνάρτηση που περιέχει τις εντολές αυτές «τρέχει» μέσα σε ατέρμονο βρόχο *while*.

### 5.3 Βάση Δεδομένων

Για τη δημιουργία της βάσης ανοίγουμε τη σελίδα *phpMyAdmin* και φτιάχνουμε μία νέα βάση δεδομένων, δίνοντάς της το όνομα *greenhouse*, όπως φαίνεται παρακάτω:



Εικόνα 5.2: Δημιουργία βάσης δεδομένων μέσω του *phpMyAdmin*

Στη συνέχεια δημιουργείται ο πίνακας *data* σε SQL ο οποίος περιέχει όλες τις μεταβλητές οι οποίες πρόκειται να καταγραφούν καθώς επίσης και κάποιες άλλες βοηθητικές για την ομαλή λειτουργία της βάσης. Αναλυτικά τα πεδία του πίνακα *data* φαίνονται παρακάτω:

- **id:** Αποτελεί το αναγνωριστικό κάθε εγγραφής. Σε κάθε νέα εγγραφή που πραγματοποιείται έχει προγραμματιστεί να αυξάνεται κατά 1 (Πρόσθετη επιλογή *AUTO\_INCREMENT*). Είναι ακέραιος έως 11 ψηφίων.
- **event:** Γίνεται εγγραφή της ακριβής ημερομηνίας και ώρας (π.χ. 18/07/2019 20:38:12) ώστε ο χρήστης να γνωρίζει ακριβώς πότε έγινε η κάθε καταγραφή δεδομένων.
- **Temperature:** Αποθηκεύει την τιμή της θερμοκρασίας περιβάλλοντος.
- **Humidity:** Αποθηκεύει την τιμή της υγρασίας περιβάλλοντος.
- **Fan:** Αποθηκεύει την τιμή του εξαερισμού.
- **Fire:** Αποθηκεύει την τιμή ανίχνευσης της φωτιάς.
- **Flood:** Αποθηκεύει την τιμή ανίχνευσης της πλημμύρας.
- **Pump:** Αποθηκεύει την τιμή της αντλίας ποτίσματος όταν λειτουργεί ή δεν λειτουργεί.
- **Shades:** Αποθηκεύει την τιμή του σκιάστρου.
- **Safety door:** Αποθηκεύει την τιμή της εξωτερικής πόρτας όταν αυτή είναι ανοιχτή ή κλειστή.
- **Water tank:** Αποθηκεύει την τιμή της κατάστασης της δεξαμενής νερού

Στην παρακάτω εικόνα φαίνεται η δομή του πίνακα data όπως φαίνεται στο phpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	event	timestamp			No	current_timestamp()			Change Drop More
3	temperature	float			No	None			Change Drop More
4	humidity	float			No	None			Change Drop More
5	fan	int(11)			No	None			Change Drop More
6	fire	int(11)			No	None			Change Drop More
7	flood	int(11)			No	None			Change Drop More
8	pump	int(11)			No	None			Change Drop More
9	shades	int(11)			No	None			Change Drop More
10	safety_door	int(11)			No	None			Change Drop More
11	water_tank	int(11)			No	None			Change Drop More

Εικόνα 5.3: Δομή βάσης δεδομένων

Για την εμφάνιση των περιεχομένων της βάσης στην ιστοσελίδα δημιουργούνται τρία αρχεία PHP τα οποία αποθηκεύονται στον φάκελο *htdocs* της XAMPP. Αρχικά απαιτείται η διασύνδεση μεταξύ PHP και MySQL η οποία γίνεται μέσω του αρχείου *connection.php* το οποίο πραγματοποιεί επίσης και τη μεταφορά δεδομένων. Έπειτα οι τιμές των αισθητήρων από το Arduino φορτώνονται στη βάση δεδομένων μέσω του αρχείου *data.php*, ενώ τέλος γίνεται οπτικοποίηση των δεδομένων στην ιστοσελίδα με τη βοήθεια του αρχείου *display.php* όπως φαίνεται στην παρακάτω εικόνα. Ο κώδικας του αρχείου *connection.php*, *data.php* και του *display.php* βρίσκονται στο παράρτημα A1, A2, A3 αντίστοιχα.

ID	Date and Time	Temperature	Humidity	Fan	Fire	Flood	Pump	Safety door	Water tank
14960	2024-06-27 22:12:40 25	72	0	0	0	1	0	0	0
14959	2024-06-27 22:12:38 25	72	0	0	0	1	0	0	0
14958	2024-06-27 22:12:37 25	72	0	0	0	1	0	0	0
14957	2024-06-27 22:12:35 25	72	0	0	0	1	0	0	0
14956	2024-06-27 22:12:33 25	72	0	0	0	1	0	0	0
14955	2024-06-27 22:12:31 25	72	0	0	0	1	0	0	0
14954	2024-06-27 22:12:29 25	72	0	0	0	1	0	0	0
14953	2024-06-27 22:12:27 25	72	0	0	0	1	0	0	0
14952	2024-06-27 22:12:25 25	72	0	0	0	1	0	0	0
14951	2024-06-27 22:12:23 25	72	0	0	0	1	0	0	0
14950	2024-06-27 22:12:21 25	72	0	0	0	1	0	0	0
14949	2024-06-27 22:12:19 25	72	0	0	0	1	0	0	0
14948	2024-06-27 22:12:16 25	72	0	0	0	1	0	0	0
14947	2024-06-27 22:12:12 25	72	0	0	0	1	0	0	0
14946	2024-06-27 22:12:11 25	72	0	0	0	1	0	0	0
14945	2024-06-27 22:12:09 25	72	0	0	0	1	0	0	0
14944	2024-06-27 22:12:06 25	72	0	0	0	1	0	0	0
14943	2024-06-27 22:12:04 25	72	0	0	0	1	0	0	0
14942	2024-06-27 22:11:58 25	72	0	0	0	1	0	0	0
14941	2024-06-27 22:11:56 25	72	0	0	0	1	0	0	0
14940	2024-06-27 22:11:54 25	72	0	0	0	1	0	0	0
14939	2024-06-27 22:11:53 25	72	0	0	0	1	0	0	0
14938	2024-06-27 22:11:51 25	72	0	0	0	1	0	0	0

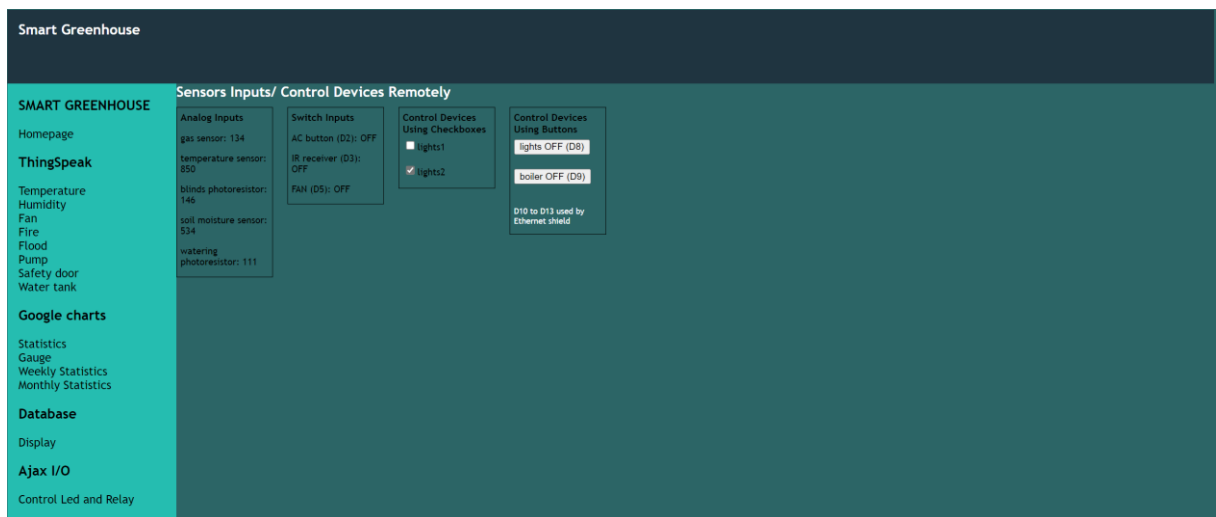
Εικόνα 5.4: Τα δεδομένα της βάσης στην ιστοσελίδα

## 5.4 Εμφάνιση και έλεγχος Ε/Ε με AJAX μέσω κάρτας SD

Μία από τις λειτουργικότητες της ιστοσελίδας είναι η εμφάνιση αναλογικών και ψηφιακών εισόδων του Arduino και η δυνατότητα απομακρυσμένου ελέγχου συσκευών μέσω ψηφιακών εξόδων. Αυτό γίνεται μέσω ιστοσελίδας (αρχείο *index1.htm*) η οποία είναι αποθηκευμένη στην κάρτα SD και διαβάζεται από το Ethernet shield. Η ιστοσελίδα εμφανίζει τέσσερις τιμές

αναλογικών εισόδων και την κατάσταση τριών διακοπών, ενώ επιτρέπει τον έλεγχο φωτισμού και αντλίας ποτίσματος μέσω checkboxes.

Όταν ο χρήστης πατήσει το αντίστοιχο πεδίο στο μενού επιλογών στην αριστερή περιοχή της σελίδας, το πρόγραμμα περιήγησης φορτώνει την ιστοσελίδα από την κάρτα SD. Έπειτα, η σελίδα στέλνει ένα αίτημα AJAX για λήψη δεδομένων από το Arduino κάθε δευτερόλεπτο. Το Arduino ανταποκρίνεται σε κάθε αίτημα AJAX στέλνοντας ένα αρχείο XML πίσω στη σελίδα, το οποίο περιέχει τις τιμές πέντε αναλογικών εισόδων του Arduino (A0 έως A4), την κατάσταση τριών ψηφιακών εξόδων (2, 3, 5) καθώς και την κατάσταση του φωτισμού και της αντλίας. Όταν ο χρήστης ενεργοποιήσει ή απενεργοποιήσει τον φωτισμό ή την αντλία μέσω των checkboxes της ιστοσελίδας, τότε η σελίδα θα στείλει την κατάσταση του checkbox με το επόμενο αίτημα AJAX. Η αντίστοιχη σελίδα φαίνεται στην παρακάτω εικόνα, ενώ ο κώδικας υπάρχει στο παράρτημα A17.



Εικόνα 5.5: Ιστοσελίδα για εμφάνιση και έλεγχο E/E

## 5.5 ΙοT Πλατφόρμα

Ένα από τα βασικότερα χαρακτηριστικά του ThingSpeak είναι το ThingSpeak Channel, το οποίο δίνει τη δυνατότητα αποθήκευσης και οπτικοποίησης έως και 8 μεταβλητών. Για να γίνει αυτό πρέπει να προηγηθεί η δημιουργία λογαριασμού και η δημιουργία ενός νέου καναλιού. Έπειτα συμπληρώνονται οι μεταβλητές για τις οποίες θα εμφανίζονται διαγράμματα. Για το έξυπνο θερμοκήπιο δημιουργήθηκε το κανάλι Smart Greenhouse και συμπληρώθηκαν τα εξής πεδία:

1. Temperature (float)
2. Humidity (float)
3. Fan (0 ή 1)
4. Fire (0 ή 1)

5. Flood (0 ή 1)
6. Pump (0 ή 1)
7. Safety\_door (0 ή 1)
8. Water\_tank (0 ή 1)

<b>Name</b>	<input type="text" value="Smart Greenhouse"/>
<b>Description</b>	<input type="text" value="Master thesis in design and manufacturing a smart greenhouse equipped with several automations."/>
Field 1	<input type="text" value="temperature"/> <input checked="" type="checkbox"/>
Field 2	<input type="text" value="humidity"/> <input checked="" type="checkbox"/>
Field 3	<input type="text" value="fan"/> <input checked="" type="checkbox"/>
Field 4	<input type="text" value="fire"/> <input checked="" type="checkbox"/>
Field 5	<input type="text" value="flood"/> <input checked="" type="checkbox"/>
Field 6	<input type="text" value="pump"/> <input checked="" type="checkbox"/>
Field 7	<input type="text" value="safety_door"/> <input checked="" type="checkbox"/>
Field 8	<input type="text" value="water_tank"/> <input checked="" type="checkbox"/>

*Εικόνα 5.6: Πεδία μεταβλητών στο ThingSpeak*

Για να γίνει εγγραφή δεδομένων σε ένα κανάλι ή την ανάγνωση δεδομένων από αυτό χρησιμοποιούνται τα κλειδιά API, τα οποία δημιουργούνται αυτόματα με τη δημιουργία ενός νέου καναλιού. Στο πεδίο κάτω από το “Write API Key” περιέχεται το κλειδί εγγραφής, ενώ στο πεδίο κάτω από το “Read API Key” περιέχεται το κλειδί για ανάγνωση δεδομένων.

Write API Key

Key

Read API Keys

Key

Note

---

*Εικόνα 5.7: Κλειδιά API για εγγραφή ή ανάγνωση δεδομένων*

Ακόμη, από την καρτέλα “Sharing” καθορίζεται το visibility (public/private) των διαγραμμάτων. Για τη δημιουργία της ιστοσελίδας επιλέχτηκε το “Share channel with everyone” ώστε να είναι ορατά τα διαγράμματα.

## Channel Sharing Settings

- Keep channel view private
- Share channel view with everyone
- Share channel view only with the following users:

Email Address

Enter email here

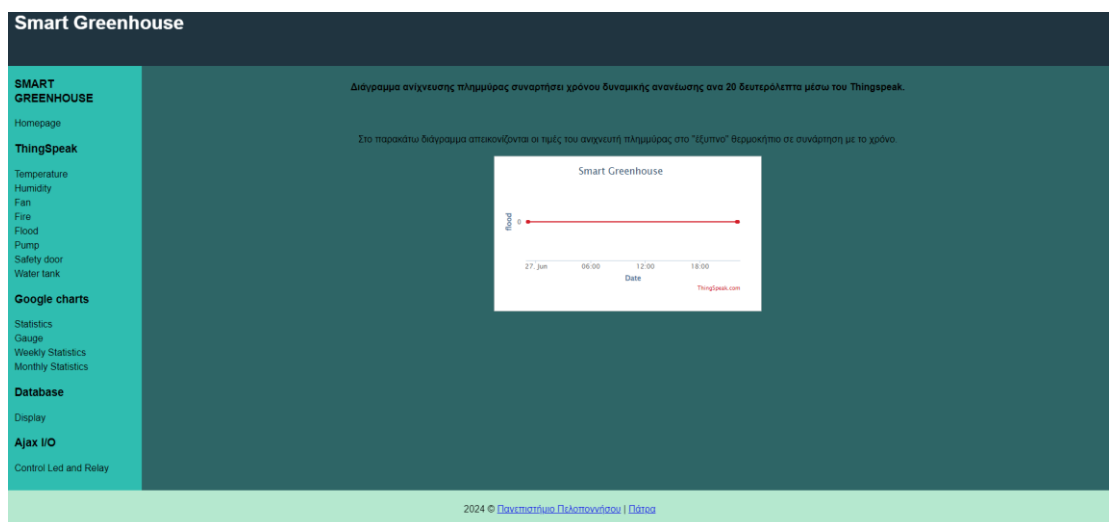
Add User

Εικόνα 5.8: Καρτέλα “Sharing” στο ThingSpeak

Τελικά, παρακάτω φαίνονται ενδεικτικά κάποια από τα διαγράμματα που υλοποιήθηκαν, όπως εμφανίζονται στην ιστοσελίδα:



Εικόνα 5.9: Το διάγραμμα θερμοκρασίας όπως φαίνεται στην ιστοσελίδα



Εικόνα 5.10: Το διάγραμμα για την πλημμύρα όπως φαίνεται στην ιστοσελίδα

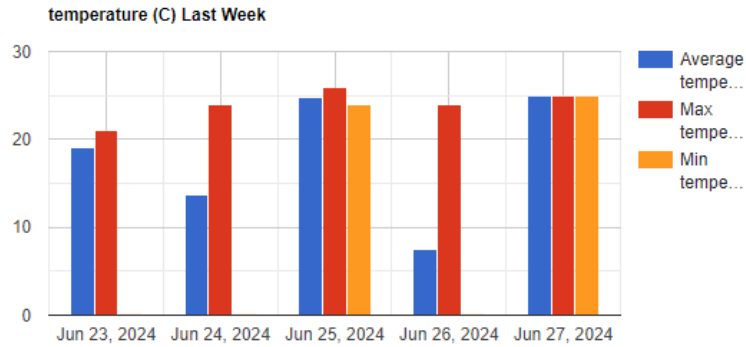
Επιλέγοντας θερμοκρασία και υγρασία από το πλαϊνό μενού επιλογών, οδηγούμαστε στην σελίδα προβολής γραφημάτων των δεδομένων που έχουν αποσταλεί στην IoT πλατφόρμα Thingspeak και αφορούν στις μετρήσεις των αισθητήρων (η υλοποίηση της πλατφόρμας αναλύεται στο Κεφάλαιο 5.5). Οι σελίδες αποτελούνται από ενσωματωμένα γραφήματα(<iframe>) , το καθένα από τα οποία δείχνει ένα δυναμικό γραμμικό γράφημα διαφορετικής σειράς δεδομένων για τις συνθήκες της θερμοκρασίας, πλημμύρας κ.α., κάθε 20 δευτερόλεπτα. Το δημόσιο κανάλι με αναγνωριστικό 2521847 είναι η πηγή που αντλούνται τα δεδομένα . Μέσω του υπό-μενού επιλογών «temperature», «flood», «fire», «safety\_door», δίνεται η δυνατότητα εναλλαγής της προβολής σε μορφή widget του Thingspeak (εικόνα 5.9,5.10 ). Η υλοποίηση της προβολής πραγματοποιείται από τα αρχεία «temperature.php», «flood.php», «fire.php» και «safety\_door.php», ο κώδικας των οποίων είναι παρόμοιος και παρουσιάζεται στα Τμήματα A.13, A.7, A.6 και A.11 του Παραρτήματος Α.

## 5.6 Google Charts

Στα πλαίσια της παρούσας διπλωματικής υλοποιήθηκαν και κάποια έξτρα αρχεία σε PHP στα οποία ενσωματώθηκαν διαγράμματα με χρήση Google charts. Τα διαγράμματα αντλούν στοιχεία από τη βάση δεδομένων και παράγουν κάποια στατιστικά εβδομαδιαία, μηνιαία και συνολικά. Τέλος, υπάρχει και ένα παράθυρο που απεικονίζει γραφικά τις τιμές των δεδομένων που αποθηκεύονται στη βάση.

### 5.6.1 Εβδομαδιαία στατιστικά

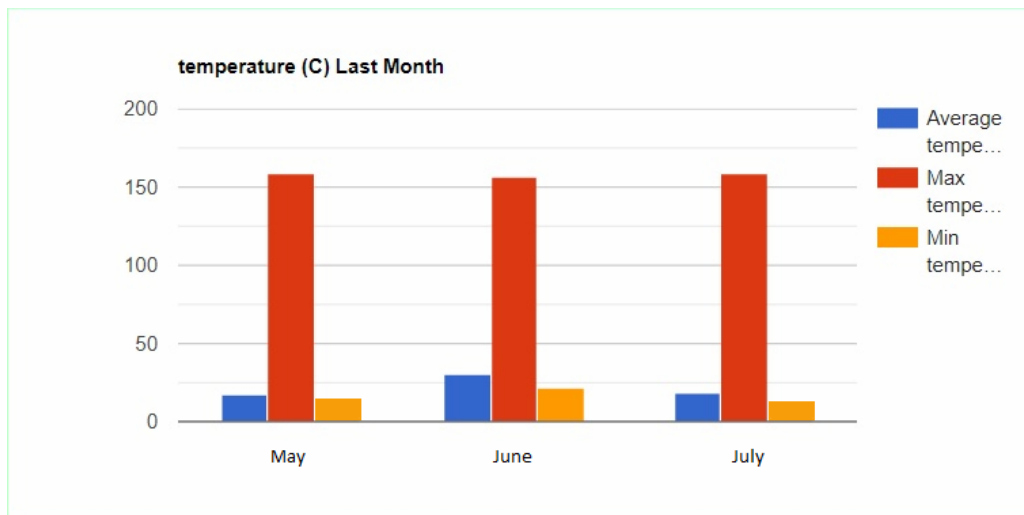
Εάν επιλέξουμε τα εβδομαδιαία στατιστικά από το πλαϊνό μενού επιλογών οδηγούμαστε στην σελίδα προβολής γραφημάτων στήλης (εικόνα 5.11). Η υλοποίηση γίνεται από το αρχείο «weekly\_statistics.php» και ανακτώνται οι εβδομαδιαίες τιμές των αισθητήρων . Επίσης, δημιουργείται από τον κατασκευαστή ένα αντικείμενο τύπου "`new google.visualization.DataView(data)`" και το γράφημα προβάλλεται μέσω του στοιχείου `div HTML` που βρίσκεται στο τμήμα A15 Παράρτημα Α.



Εικόνα 5.11: Εβδομαδιαία στατιστικά θερμοκρασίας

### 5.6.2 Μηνιαία στατιστικά

Εάν επιλέξουμε τα μηνιαία στατιστικά από το πλαϊνό μενού επιλογών οδηγούμαστε στην σελίδα προβολής γραφημάτων στήλης (εικόνα 5.12). Η υλοποίηση γίνεται από το αρχείο «monthly\_statistics.php» και ανακτώνται οι μηνιαίες τιμές των αισθητήρων. Επίσης, δημιουργείται από τον κατασκευαστή ένα αντικείμενο τύπου "`new google.visualization.DataView(data)`" και το γράφημα προβάλλεται μέσω του στοιχείου `div HTML` που βρίσκεται στο τμήμα A10 Παράρτημα Α.



Εικόνα 5.12: Μηνιαία στατιστικά θερμοκρασίας

### 5.6.3 Ιστογράμματα

Εάν επιλέξουμε τα ιστογράμματα από το πλαϊνό μενού επιλογών οδηγούμαστε στην σελίδα προβολής γραφημάτων στήλης (εικόνα 5.13). Η υλοποίηση γίνεται από το αρχείο «histogramms.php», καθώς ανακτάτε από την βάση η μέγιστη, ελάχιστη και ο μέσος όρος της επιλεγμένης μέτρησης για το επιθυμητό διάστημα. Επίσης τώρα δημιουργείται με τον

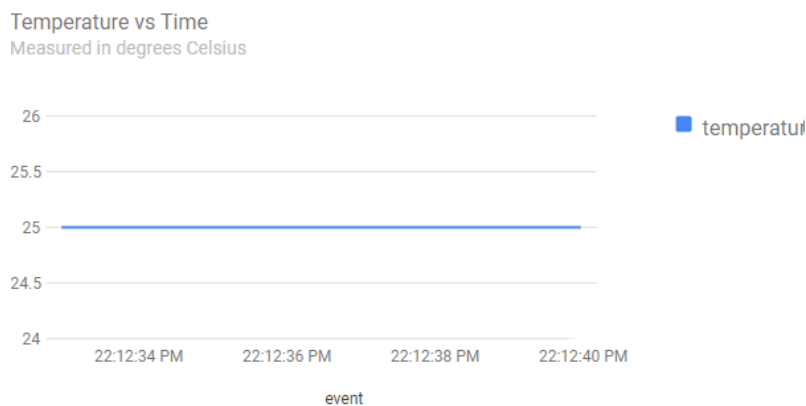
κατασκευαστή ένα αντικείμενο τύπου "Bar" και το γράφημα προβάλλεται μέσω του στοιχείου div HTML και παρουσιάζεται στο τμήμα A18 του Παραρτήματος Α.



Εικόνα 5.13: Ιστόγραμμα δεδομένων

### 5.6.4 Συνολικά στατιστικά

Εάν επιλέξουμε τα συνολικά στατιστικά από το πλαϊνό μενού επιλογών οδηγούμαστε στην σελίδα προβολής γραφημάτων στήλης (εικόνα 5.14). Η υλοποίηση γίνεται από το αρχείο «statistics.php» και ανακτάται η τιμή της επιλεγμένης μέτρησης ζωντανά από τους αισθητήρες. Επίσης, δημιουργείται από τον κατασκευαστή ένα αντικείμενο τύπου "newgoogle.visualization.DataView(data)" και το γράφημα προβάλλεται μέσω του στοιχείου div HTML που βρίσκεται στο τμήμα A12 Παράρτημα Α.

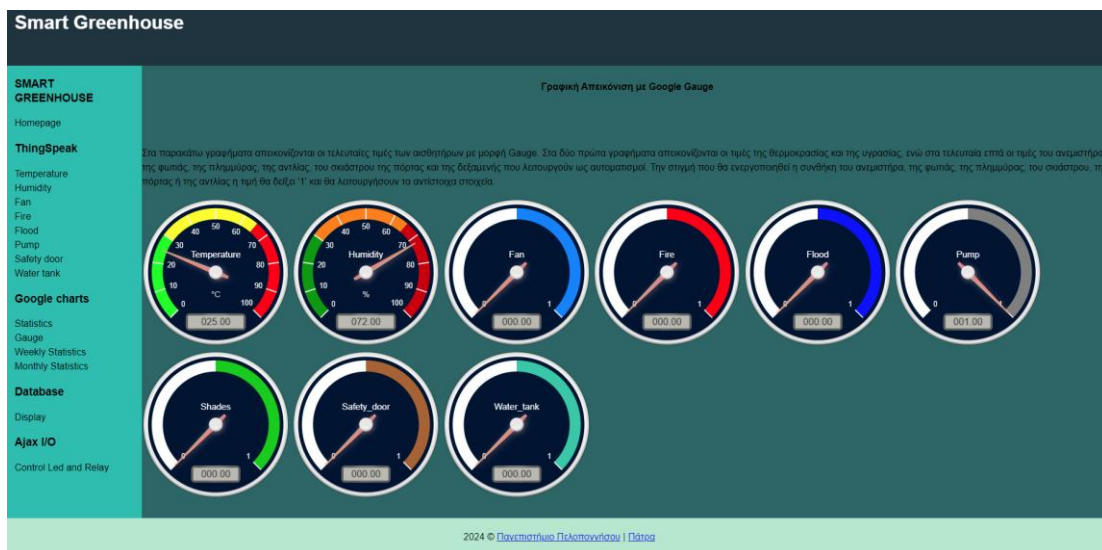


Εικόνα 5.14: Συνολικά στατιστικά θερμοκρασίας



## 5.6.5 Gauges

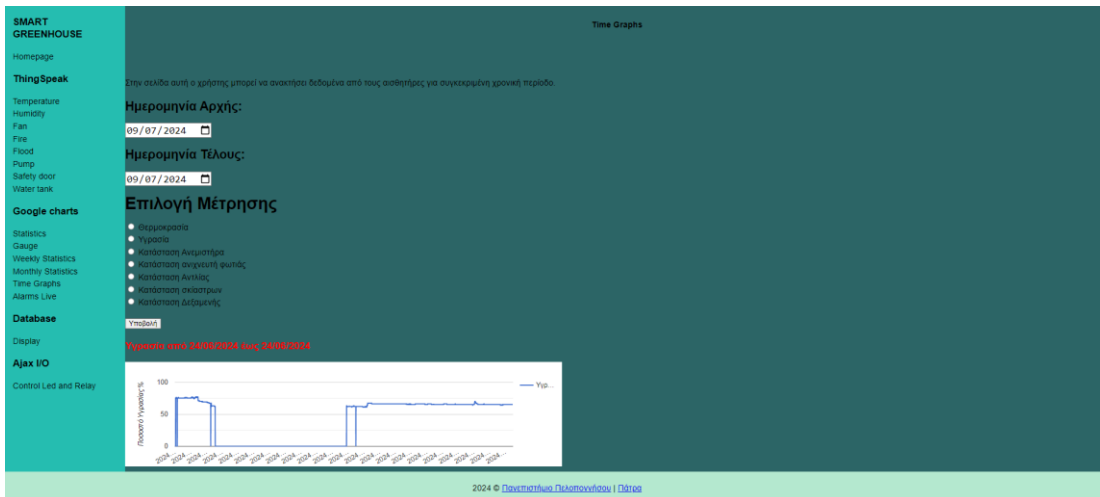
Εάν επιλέξουμε τα Gauge από το πλαϊνό μενού επιλογών οδηγούμαστε στην σελίδα προβολής γραφημάτων Gauge, των τελευταίων μετρήσεων υπό κλίμακα (εικόνα 5.15). Η υλοποίηση γίνεται από το αρχείο «gauge.php» που βρίσκεται στο τμήμα A.8 Παράρτημα A. Η σελίδα ενημερώνεται δυναμικά κάθε πέντε δευτερόλεπτα "window.setTimeout(drawChart, 5000);". Η συνάρτηση "drawChart()" εκτελεί ένα αίτημα AJAX για ανάκτηση των τελευταίων δεδομένων από το αρχείο "gauge.js". Αν το αίτημα είναι επιτυχές, επιστρέφεται ένα αντικείμενο JSON το οποίο μεταβιβάζεται στη μέθοδο done() ως παράμετρος jsonData. Τα δεδομένα, μετατρέπονται σε Data Table χρησιμοποιώντας μέθοδο "google.visualization.arrayToDataTable()" και παρομοίως κατασκευάζονται αντικείμενα τύπου "Gauge" που προβάλλονται μέσω των στοιχείων div HTML. Η ανάκτηση των μετρήσεων υλοποιείτε στο "gauge.min" καθώς στέλνεται ερώτημα στη βάση και ανακτάται η εγγραφή από τον πίνακα της βάσης. Η τιμή κάθε πεδίου αποθηκεύεται σε πίνακα βασισμένο στη μονάδα μέτρησης και την τιμή του αισθητήρα.



Εικόνα 5.15: Γραφική απεικόνιση με Google charts

## 5.6.6 Προβολή δεδομένων για συγκεκριμένο χρονικό διάστημα

Εάν επιλέξουμε τα διαγράμματα χρόνου από το πλαϊνό μενού επιλογών οδηγούμαστε στην σελίδα προβολής γραφημάτων των δεδομένων που έχουν αποθηκευτεί στην βάση (εικόνα 5.16). Η υλοποίηση γίνεται από το αρχείο «time\_graphs.php» τμήμα A.20 Παράρτημα A και για την ανάλυση των δεδομένων χρησιμοποιούνται τα Google Charts. Αν επιλεγεί το χρονικό διάστημα και το είδος μέτρησης αισθητήρα από την φόρμα επιλογής, τότε εκτελείται ερώτημα ανάκτησης των επιλεγμένων δεδομένων από τον πίνακα της βάσης.



Εικόνα 5.16: Προβολή δεδομένων για συγκεκριμένο χρονικό διάστημα

### 5.6.7 Προβολή alarms σε πραγματικό χρόνο

Εάν επιλέξουμε τα Alarms Live από το παλιό μενού επιλογών οδηγούμαστε στην σελίδα προβολής γραφημάτων Gauge, των τελευταίων μετρήσεων υπό κλίμακα (εικόνα 5.17).

Η υλοποίηση γίνεται στο αρχείο «alarms\_live.php» τμήμα A.19 Παράρτημα Α, είναι ίδια με αυτή του αρχείου «gauge.php», με την διαφορά ότι οι τιμές των αισθητήρων αποστέλλονται από το αρχείο "gauge\_data.php" με τιμή "0" ή "1". Η τιμή "1" αντιστοιχεί σε τιμή των αισθητήρων που έχουν υπερβεί τα όρια ασφαλείας που τους έχουν τεθεί. Τέλος, έχει γίνει μορφοποίηση της κλίμακας για την καλύτερη προσομοίωση των συναγεμών.

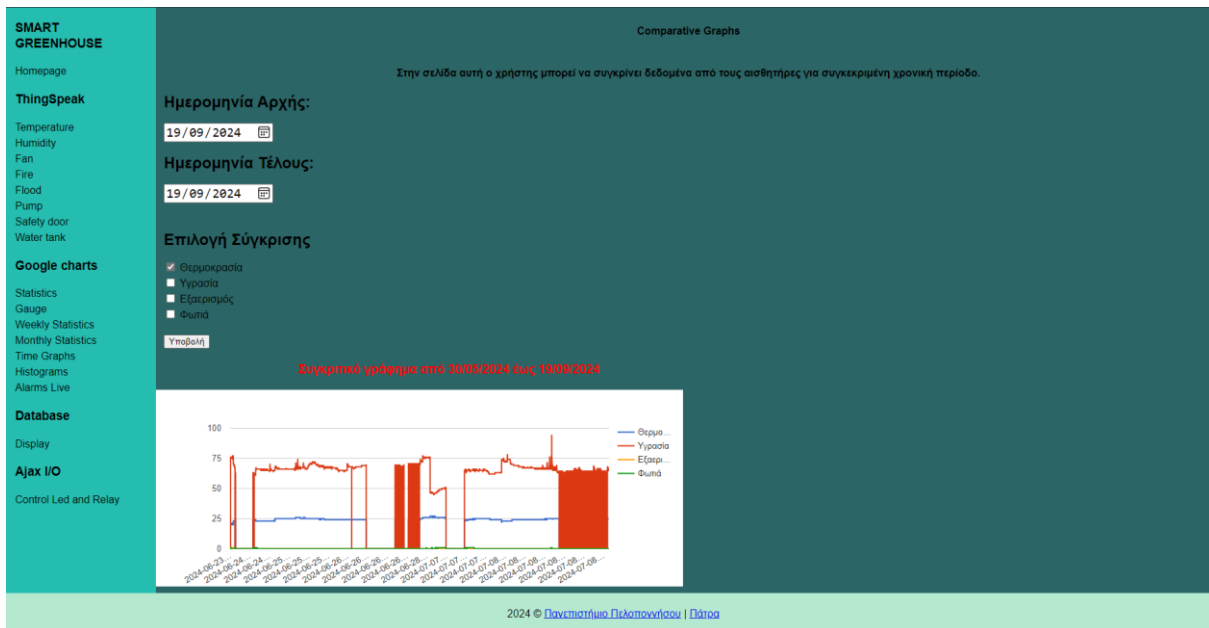


Εικόνα 5.17: Προβολή Live Alarms

### 5.6.8 Συγκριτικά γραφήματα

Εάν επιλέξουμε τα συγκριτικά γραφήματα από το παλιό μενού επιλογών οδηγούμαστε στην σελίδα ταυτόχρονης προβολής γραφημάτων που συγκρίνουν τις τιμές των επιλεγμένων

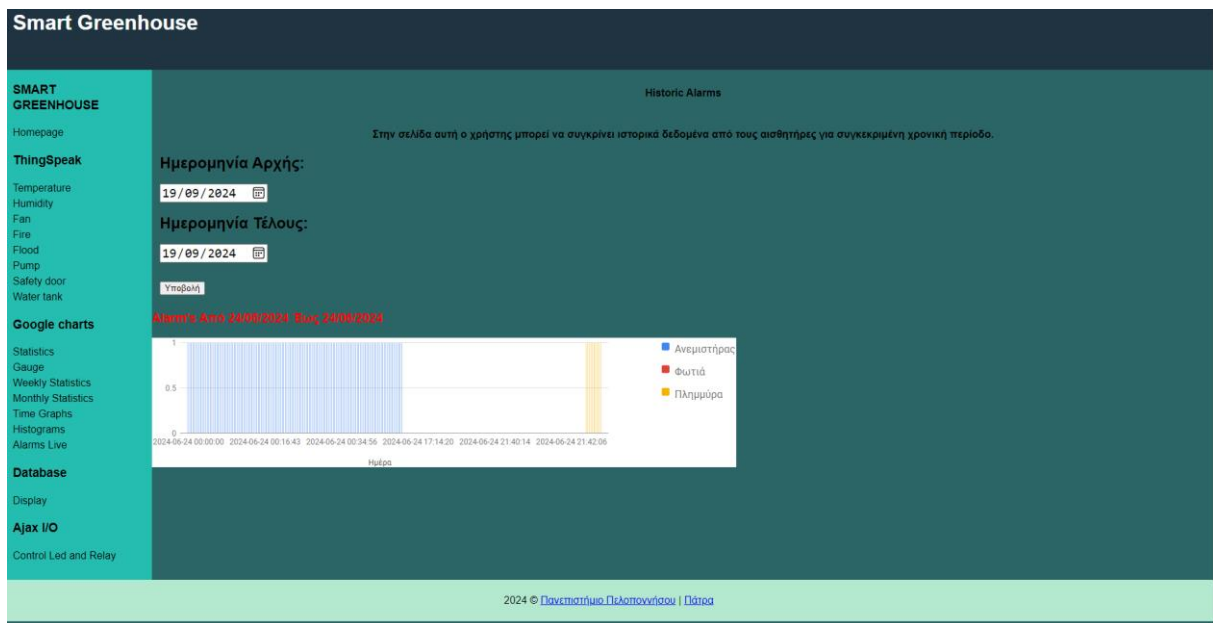
αισθητήρων (εικόνα 5.18). Η υλοποίηση γίνεται από το αρχείο «comparative 1.php» που βρίσκεται στο τμήμα A.21 Παράρτημα Α, με παρόμοιο τρόπο όπως και στο αρχείο «time\_graphs.php». Η διαφορά τώρα είναι ότι τα πεδία επιλογής στην φόρμα, είναι τύπου «checkbox», αντί «radio» και αποστέλλονται στον πίνακα επιλογών. Επίσης, κατά την δημιουργία του αντικειμένου "DataTable", με τη μέθοδο "data.addColumn()" και "data.addRows()" προστίθεται αντίστοιχα μια νέα στήλη και οι γραμμές για κάθε επιλεγμένη μέτρηση.



Εικόνα 5.18: Συγκριτικά γραφήματα

## 5.6.9 Προβολή Historic Alarms

Εάν επιλέξουμε τα Ιστορικά Alarms από το πλαίσιο μενού επιλογών οδηγούμαστε στην σελίδα προβολής γραφημάτων στήλης, για τυχόν alarms που υπήρχαν το χρονικό διάστημα επιλογής (εικόνα 5.19). Η υλοποίηση γίνεται από το αρχείο «alarms.php» που βρίσκεται στο τμήμα A21 Παράρτημα Α. Ανακτώνται από την βάση οι τιμές των αισθητήρων που έχουν υπερβεί τα όρια ασφαλείας που τους έχουν τεθεί. Τα δεδομένα, που ανακτώνται αποθηκεύονται σε έναν νέο πίνακα που ονομάζεται "\$dataTable", και βάσει τιμών αποθηκεύεται 0 ή 1 στο αντίστοιχο πεδίο. Στη συνέχεια, το \$dataArray χρησιμοποιείται για τη συμπλήρωση του πίνακα δεδομένων και την δημιουργία αντικειμένου "DataTable" για την κατασκευή γραφήματος τύπου "Bar" και προβολή μέσω του στοιχείου div HTML.



Εικόνα 5.19: Ιστορικά Alarms

## 6 Συμπεράσματα

### 6.1 Εισαγωγή

Στην παρούσα διπλωματική έγινε υλοποίηση ενός «έξυπνου» θερμοκηπίου το οποίο, στα πλαίσια του Internet of Things (IoT), το οποίο ενσωματώνει αυτοματισμούς με σκοπό την παρακολούθηση και τον έλεγχο των παραγόντων που επηρεάζουν άμεσα την παραγωγή. Έτσι, η υλοποίηση στοχεύει στη βελτιστοποίηση της παραγωγής, όπως επίσης και στην εξοικονόμηση πόρων και ενέργειας. Μέσω της πλατφόρμας Zigbee υπάρχει η δυνατότητα απομακρυσμένου ελέγχου, ενώ μέσα από τον ιστότοπο η παρακολούθηση γίνεται πιο φιλική προς το χρήστη. Ακόμη, χάρη στα συστήματα αυτοματισμών ο αγρότης εξοικονομεί τόσο χρόνο για τη συντήρηση και λειτουργία του θερμοκηπίου, όσο και χρήμα από βλάβες οι οποίες μπορούν να ανιχνευθούν εγκαίρως. Επίσης έγινε εμπλοκή πολλών τρόπων διασύνδεσης, πρωτοκόλλων επικοινωνίας, τεχνολογιών διαδικτύου και δωρεάν εργαλείων ανάπτυξης, ώστε η υλοποίηση να επιτυγχάνεται ποικιλοτρόπως (π.χ. Ethernet-WiFi διασύνδεση). Μελετήθηκαν τεχνολογίες διαδικτύου, όπως οι θεμέλιοι λίθοι του παγκόσμιου Ιστού (HTTP, URL, HTML) και η αρχιτεκτονική των διαδικτυακών εφαρμογών. Εξετάστηκαν οι δυνατότητες που προσφέρουν οι πλατφόρμες IoT και αποκτήθηκαν γνώσεις πάνω σε γλώσσες όπως η PHP και η JavaScript, χάρη στις οποίες προσδόθηκε δυναμικός και διαδραστικός χαρακτήρας στις ιστοσελίδες. Αναλύθηκε η πλατφόρμα Arduino, τόσο στο υλικό κομμάτι, όπως πλακέτες, shields, περιφερειακά, κλπ., όσο και ο τρόπος προγραμματισμού και επικοινωνίας, για την λήψη, αποθήκευση και αξιοποίηση μετρήσεων. Τέλος, ερευνήθηκαν και αναλύθηκαν οι δυνατότητες ενσύρματης (Ethernet) διασύνδεσης της πλατφόρμας Arduino στο διαδίκτυο, αλλά και η διασύνδεσή της με άλλες συσκευές μέσω πρωτοκόλλων όπως τα ZigBee, SPI, I2C. Καταλήγοντας, από την ενασχόληση με την εν λόγω διπλωματική εργασία, αποκτήθηκε η απαιτούμενη τεχνογνωσία και εμπειρία, για τον σχεδιασμό, ανάπτυξη και υλοποίηση ενός ολοκληρωμένου, αξιόπιστου και εύκολου στην εγκατάσταση συστήματος, που μπορεί να προσδιορίσει ένα θερμοκήπιο ως «έξυπνο».

### 6.2 Προβλήματα και αντιμετώπιση αυτών

Ένα από τα μεγαλύτερα προβλήματα που προέκυψαν, ήταν η έλλειψη από την αγορά της συσκευής Xbee S2. Αυτό οδήγησε στην αγορά δύο μεταχειρισμένων συσκευών με αναμονή τεσσάρων μηνών για την παράδοσή τους. Επίσης, λόγω των μεγάλων απαιτήσεων του κώδικα που φορτώθηκε στο Arduino Uno, παρουσιάστηκε πρόβλημα μεταγλώττισης, λόγω

εξάντλησης του χώρου στη μνήμη flash της συσκευής. Για τον λόγο αυτό, τα σενάρια χωρίστηκαν σε δύο Arduino Uno και ένα Arduino Mega, με αποτέλεσμα το κόστος της εργασίας να είναι ιδιαίτερα υψηλό. Η αγορά των κομματιών plexiglass ήταν ιδιαίτερα υψηλή και το πρόβλημα που παρουσιάστηκε ήταν ο τρόπος με τον οποίο θα γινόταν η κόλληση των κομματιών για την κατασκευή του θερμοκηπίου. Τελικά χρησιμοποιήθηκε κόλλα σιλικόνης ένα φθινό και αξιόπιστο υλικό. Επίσης οι αισθητήρες στάθμης νερού και καθαρότητας νερού λόγω της οξείδωσης από τα άλατα μετά την χρήση τους, παρουσίασαν μεταβολές και αποκλίσεις στις τιμές τους. Για τον λόγο αυτό χρειάζεται καθαρισμός των δύο αισθητήρων μετά την χρήση τους. Τέλος, πολλά από τα καλώδια που χρησιμοποιήθηκαν για την σύνδεση των αισθητήρων με την πλακέτα Arduino, ήταν ελαττωματικά και χρειάστηκε η αντικατάστασή τους με νέα καλώδια υψηλής αντοχής.

### **6.3 Συμπεράσματα και Μελλοντικές επεκτάσεις**

Παρά τα οφέλη που παρέχει η εγκατάσταση ενός τέτοιου συστήματος, διαθέτει πολλά περιθώρια βελτίωσης τα οποία μπορούν να εξελίσσουν περαιτέρω την λειτουργικότητά του. Για παράδειγμα μπορούν να συνδεθούν περισσότερα Zigbee ώστε να είναι δυνατή η ασύρματη μετάδοση περισσότερων πληροφοριών στο Arduino που δρα ως κεντρικός κόμβος. Επίσης, θα μπορούσε να χρησιμοποιηθεί ένα ESP8266 ώστε να υπάρχει και ασύρματη σύνδεση στο διαδίκτυο μέσω WiFi, πέρα από την ενσύρματη που υπάρχει τώρα με τη χρήση του Ethernet shield. Αναφορικά με αναβαθμίσεις των αυτοματισμών, θα μπορούσε να προστεθεί ένας αισθητήρας ο οποίος να κάνει έλεγχο του pH του εδάφους καθώς επίσης και του οξυγόνου που υπάρχει σε αυτό ώστε να ωφελήσει και αυτή η πληροφορία την παραγωγή των φυτών. Τέλος, σε επίπεδο κατανάλωσης ενέργειας μια χρήσιμη προσθήκη θα ήταν η ενσωμάτωση ενός συστήματος μέτρησης της ενέργειας που καταναλώνουν όλα τα ηλεκτρονικά στοιχεία από τα οποία απαρτίζεται το σύστημα αυτοματισμών, ιδίως λόγω της ύπαρξης συσκευών υψηλής κατανάλωσης ισχύος όπως είναι οι αντλίες. Αυτό θα μπορούσε να συνδυαστεί και με τη χρήση ανανεώσιμων πηγών ενέργειας όπως είναι οι μπαταρίες λιθίου ή τα φωτοβολταϊκά πάνελ ώστε το θερμοκήπιο να διαθέτει πλήρη ενεργειακή αυτονομία.

Σε αρκετές περιπτώσεις, η αξιοπιστία και η ποιότητα του εξοπλισμού πρέπει να προηγούνται του κόστους. Μια βλάβη μπορεί να οδηγήσει σε πολύ περισσότερο χρόνο και χρήμα για την επισκευή της. Επιπλέον, πρέπει να λάβουμε υπόψη ότι οι δωρεάν πλατφόρμες ανάπτυξης IoT συστημάτων, αν και χρήσιμες, συνήθως έχουν περιορισμούς (π.χ., περιορισμένος αριθμός πεδίων ή συχνότητα ενημέρωσης δεδομένων στο ThingSpeak), που μπορούν να επηρεάσουν τη λειτουργικότητα του συνολικού συστήματος.



## Βιβλιογραφία

- Arduino. (2024). *Arduino*. Ανάκτηση από «What is Arduino: <https://www.arduino.cc/en/Guide/Introduction>
- arduinogetstarted. (2024). *Arduino Shield*. Ανάκτηση από arduinogetstarted: <https://arduinogetstarted.com/tutorials/arduino-shield>
- ArduinoModules. (χ.χ.). *KY-017 MERCURY TILT SWITCH MODULE*. Ανάκτηση από ArduinoModules: <https://arduinomodules.info/ky-017-mercury-switch-module/>
- circuitdigest. (2022, March 21). *Everything you need to know about the Arduino Hardware*. Ανάκτηση από circuitdigest: <https://circuitdigest.com/article/everything-you-need-to-know-about-arduino-uno-board-hardware>
- Dejan. (χ.χ.). *Arduino and DS3231 Real Time Clock Tutorial*. Ανάκτηση από howtomechatronics: <https://howtomechatronics.com/tutorials/arduino/arduinods3231-real-time-clock-tutorial/>
- developers.google. (2019, September 23). «Using Google Charts. Ανάκτηση από developers.google: <https://developers.google.com/chart/interactive/docs>
- DFRobot. (2022). *Gravity: Digital Capacitive Touch Sensor for Arduino*. Ανάκτηση από DFRobot: <https://www.dfrobot.com/product-78.html>
- DFRobot. (χ.χ.). *Turbidity\_sensor\_SKU\_SEN0189*. Ανάκτηση από DFRobot.
- docs, m. w. (2024). *What is JavaScript?* Ανάκτηση από mdn web docs: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)
- docs.arduino. (2024). *Arduino Integrated Development Environment (IDE) v1*. Ανάκτηση από <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics/#sketch>
- docs.arduino.cc. (2022, August 31). *docs.arduino.cc*. Ανάκτηση από docs.arduino.cc: <https://docs.arduino.cc/retired/shields/arduino-ethernet-shield-without-poe-module>
- electrothinks. (χ.χ.). *KY-026 Flame Sensor Module*. Ανάκτηση από electrothinks: <https://www.electrothinks.com/2021/01/KY-026-flame-sensor-module.html>
- elprocus. (χ.χ.). *DHT11 Sensor and Its Working*. Ανάκτηση από elprocus: <https://www.elprocus.com/a-brief-on-dht11-sensor/>
- en.wikipedia.org. (χ.χ.). *Fritzing*. Ανάκτηση από en.wikipedia.org: <https://en.wikipedia.org/wiki/Fritzing>
- veloper.mozilla.org. (2024). *An overview of HTTP*. Ανάκτηση από eveloper.mozilla.org: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
- grobotronics. (χ.χ.). *Photoresistor Module KY-018*. Ανάκτηση από grobotronics: <https://grobotronics.com/photoresistor-module-ky-018.html?sl=en>
- Gupta, P. (2024). *Difference Between Arduino Due vs Mega*. Ανάκτηση από educba: <https://www.educba.com/arduino-due-vs-mega/>
- Gupta, S. (2021, October 6). *What Is SQL & How Does It Work? A Guide to Structured Query Language*. Ανάκτηση από springboard: <https://www.springboard.com/blog/data-analytics/what-is-sql/>
- Hopkins, J. (2021, December 1). *totalphase*. Ανάκτηση από «Understanding I2C and SPI Communication Protocols»: <https://www.totalphase.com/blog/2021/12/i2c-vs-spi-vs-uart-introduction-andcomparison-similarities-differences/>
- IBM. (2021). *What is AJAX?* Ανάκτηση από IBM: <https://www.ibm.com/docs/en/rational-software-arch/9.6.1?topic=page-asynchronous-javascript-xml-ajax-overview>
- K, A. (2021, May 11). *What is XAMPP? and How to Install XAMPP?* Ανάκτηση από devopsschool: <https://www.devopsschool.com/blog/what-is-xampp-and-how-to-install-xampp/>



- kionetworks. (2021). *Network Communication Protocols*. Ανάκτηση από kionetworks: <https://www.kionetworks.com/en-us/blog/data-center/networkcommunication-protocol>
- Kodali, R. K. (2016). IoT based smart greenhouse. *2016 IEEE region 10 humanitarian technology conference (R10-HTC)*, 1-6.
- Kumar, N. a. (2022). Development of an IoT-based smart greenhouse using Arduino. *Advances in Communication, Devices and Networking: Proceedings of ICCDN 2020*, 455-465.
- lastminuteengineers. (2024). *Interface Two Channel Relay Module with Arduino*. Ανάκτηση από lastminuteengineers: <https://lastminuteengineers.com/two-channel-relay-module-arduino-tutorial/>
- lastminuteengineers. (χ.χ.). *How Soil Moisture Sensor Works and Interface it with Arduino*. Ανάκτηση από lastminuteengineers: <https://lastminuteengineers.com/soil-moisture-sensor-arduino-tutorial/>
- Net, S. (χ.χ.). «*Ανεμιστήρας 12V DC 60x60x25mm EB60251S1 sleeve Sunon*». Ανάκτηση από Stathis Net: <https://www.stathisnet.gr/anemistiras-12v-dc-60x60x25mm-eb60251s1-sleeve-sunon>
- notepad-plus-plus.org. (2020). *What is Notepad++*. Ανάκτηση από notepad-plus-plus.org: <https://notepad-plus-plus.org/>
- Ntousakis, E. (2021). *Design and implementation of integrated control system – sensors for “smart” greenhouses, based on Internet of Things (IoT) technologies*. Ανάκτηση από <http://purl.tuc.gr/dl/dias/B7ED7D86-AABD-481B-9386-820FFBE0092E>
- php.net. (2020). *php.net*. Ανάκτηση από php.net: <https://www.php.net/manual/en/intro-what-is.php>
- randomnerdtutorials. (2021). *Guide for DS18B20 Temperature Sensor with Arduino*. Ανάκτηση από randomnerdtutorials: <https://randomnerdtutorials.com/guide-for-ds18b20-temperature-sensor-with-arduino/>
- randomnerdtutorials. (χ.χ.). *Arduino with PIR Motion Sensor*. Ανάκτηση από randomnerdtutorials: <https://randomnerdtutorials.com/arduino-with-pir-motion-sensor/>
- Research, K. (χ.χ.). *DC Mini Immersible Water Pump (6V~18V)*. Ανάκτηση από Knowledge Research: <https://www.why.gr/en/shop/open-hardware/dfrobot-en/dc-mini-immersible-water-pump-6v18v/>
- Robodo. (χ.χ.). *Water Level Sensor Depth of Detection Water Sensor for Arduino*. Ανάκτηση από Robodo: <https://robodo.in/products/water-level-sensor-depth-of-detection-water-sensor-for-arduino>
- Shirsath, D. a. (2017). IoT based smart greenhouse automation using Arduino. *International Journal of Innovative Research in Computer Science & Technology*, 234-238.
- Soheli, S. J. (2022). Smart greenhouse monitoring system using internet of things and artificial intelligence. *Wireless Personal Communications*, 3603-3634.
- thegeekpub. (χ.χ.). *Sensor Wiki: KY-012 Active Piezo-Buzzer Module*. Ανάκτηση από thegeekpub: <https://www.thegeekpub.com/wiki/sensor-wiki-ky-012-active-piezo-buzzer-module/>
- thingspeak. (χ.χ.). *Learn More About ThingSpeak*. Ανάκτηση από thingspeak: [https://thingspeak.com/pages/learn\\_more](https://thingspeak.com/pages/learn_more)
- topelectronics. (χ.χ.). *Infrared IR Receiver Module - Wireless Remote Control Kit For Arduino*. Ανάκτηση από topelectronics: <https://topelectronics.gr/electronics/sensors/ir/infrared-ir-receiver-module-wireless-remote-control-kit-for-arduino/>
- Tripathy, P. K. (2021). MyGreen: An IoT-enabled smart greenhouse for sustainable agriculture. *IEEE Consumer Electronics Magazine*, 57-62.
- w3schools. (2024). *CSS Introduction*. Ανάκτηση από w3schools: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)

- w3schools. (2024). *HTML Introduction*. Ανάκτηση από w3schools:  
[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)
- waveshare. (χ.χ.). *MQ-7 Gas Sensor*. Ανάκτηση από waveshare:  
[https://www.waveshare.com/wiki/MQ-7\\_Gas\\_Sensor](https://www.waveshare.com/wiki/MQ-7_Gas_Sensor)
- Σ. Δριμής, Δ. Κ. (2029). *ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΑΥΤΟΜΑΤΙΣΜΩΝ ΓΙΑ ΕΞΥΠΝΟ ΣΠΙΤΙ ΚΑΙ ΒΡΑΧΙΟΛΑΚΙ ΕΚΤΑΚΤΗΣ ΑΝΑΓΚΗΣ ΓΙΑ ΗΛΙΚΙΩΜΕΝΟΥΣ, ΒΑΣΙΣΜΕΝΑ ΣΤΗΝ ΠΛΑΤΦΟΡΜΑ ARDUINO, ΜΕ ΤΟΠΙΚΟ Η ΑΠΟΜΑΚΡΥΣΜΕΝΟ ΕΛΕΓΧΟ*.
- cytron.io. (2011). *Is XBee Zigbee*. Ανάκτηση από tutorial.cytron.io:  
<https://tutorial.cytron.io/2011/03/06/is-xbee-zigbee/>.

# Παράρτημα Α: Κώδικας Software Υλοποίηση Ιστότοπου, Ιστοσελίδων, Γραφικών Ιστογραμμάτων

## A.1 Κώδικας αρχείου “connection.php” – Διαχείριση σύνδεσης με την βάση δεδομένων

```
<?php
$username = "root";
$password = "";
$host = "localhost";
$db_name = "greenhouse";
$con = mysqli_connect ($host, $username, $password);
$db = mysqli_select_db ( $con, $db_name );
if(!$con){
    die("Connection failed: " . mysqli_connect_error());
}
echo " ";
?>
```

## A.2 data.php

```
<?php
include ('connection.php');
$sql_insert = "INSERT INTO data (temperature, humidity, fan, fire,
flood, pump, shades, safety_door, water_tank) VALUES
('".$_GET["temperature"]."', '".$_GET["humidity"]."',
 '".$_GET["fan"]."', '".$_GET["fire"]."', '".$_GET["flood"]."',
 '".$_GET["pump"]."', '".$_GET["shades"]."',
 '".$_GET["safety_door"]."', '".$_GET["water_tank"]."')";
if(mysqli_query($con,$sql_insert))
{
    echo "Done";
    mysqli_close($con);
}
else
{
    echo "error is " .mysqli_error($con);
}
?>
```

## A.3 display.php

```
<?php
$url=$_SERVER['REQUEST_URI'];
header("Refresh: 5; URL=$url"); // Refresh the webpage every
seconds
?>
<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Smart Greenhouse</title>
        <style type="text/css">

            body {
                margin:0;
```

```

padding:0;
font-family: Sans-Serif;
line-height: 1.5em;
background: #2C6566;
}

#header {
background: #1F3440;
color: white;
height: 100px;
}

#header h1 {

margin: 0;
padding-top: 15px;

}

main {
padding-bottom: 10010px;
margin-bottom: -10000px;
float: left;
width: 100%;
}

#nav {
padding-bottom: 10010px;
margin-bottom: -10000px;
float: left;
width: 230px;
margin-left: -100%;
background: #25bdb0;
}

#footer {
clear: left;
width: 100%;
background:#B4E8CF ;
text-align: center;
padding: 4px 0;
}

#wrapper {
overflow: hidden;
}

#content {
margin-left: 230px; /* Same as 'nav' width */
}

.innertube {
margin: 15px; /* Padding for content */
margin-top: 0;
}

p {
color: #000000;
}

nav ul {

```

```

        list-style-type: none;
        margin: 0;
        padding: 0;
    }

    nav ul a {
        color: black;
        text-decoration: none;
    }

</style>

<title>Arduino Smart Greenhouse</title>
<style type="text/css">
.table_titles {
padding-right: 20px;
padding-left: 20px;
color: #999;
}

.table_titles {
color: #000000;
background-color:#25bdb0;
}

table {
border: 2px solid #000;
}
body { font-family: "Trebuchet MS", Courier;
background-color:#2C6566;
}
</style>
</head>

<body>

    <header id="header">
        <div class="innertube">
            <h1>Smart Greenhouse</h1>
        </div>
    </header>

    <div id="wrapper">
        <main>

            <div id="content">

                <div class="content">

<h1>SMART GREENHOUSE </h1>
<table border="0" cellspacing="0" cellpadding="4">
<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">Fan</td>
<td class="table_titles">Fire</td>

```

```

<td class="table_titles">Flood</td>
<td class="table_titles">Pump</td>
<td class="table_titles">Safety door</td>
<td class="table_titles">Water tank</td>
<!--<td class="table_titles">Heat_index</td>-->
</tr>

<?php
include('connection.php');
$result = mysqli_query($con, 'SELECT * FROM data ORDER BY id DESC');
// Process every record
$odddrow = true;
while($row = mysqli_fetch_array($result))
{
if ($odddrow)
{
$css_class=' class="table_cells_odd"';
}
else
{
$css_class=' class="table_cells_even"';
}
$odddrow = !$odddrow;
echo "<tr>";
echo "<td '." . $css_class . "'>" . $row['id'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['event'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['temperature'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['humidity'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['fan'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['fire'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['flood'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['pump'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['safety_door'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['water_tank'] . "</td>";
echo "</tr>";
}

// Close the connection
mysqli_close($con);
?>
</table>
</div>
</div>

</div>
</main>

<nav id="nav">
<div class="innertube">
<h3>SMART GREENHOUSE</h3>
<ul>
<li><a href="index.html">Homepage</a></li>

<h3>ThingSpeak</h3>
<ul>
<li><a
href="temperature.php">Temperature</a></li>
<li><a
href="humidity.php">Humidity</a></li>
<li><a href="fan.php">Fan</a></li>
<li><a href="fire.php">Fire</a></li>

```

```

        <li><a href="flood.php">Flood </a></li>
        <li><a href="pump.php">Pump</a></li>
        <li><a href="safety_door.php">Safety
door</a></li>
        <li><a href="water_tank.php">Water
tank</a></li>
    </ul>

    <h3>Google charts</h3>
    <ul>
        <li><a
href="statistics1.php">Statistics</a></li>
        <li><a href="gauge.php">Gauge</a></li>
        <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
        <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
    </ul>
    <h3>Database</h3>
    <ul>
        <li><a href="display.php">Display</a></li>
    </ul>
    <h3>Ajax I/O</h3>
    <ul>
        <li><a href="index1.htm">Control Led and
Relay</a></li>
    </ul>
</div>
</nav>

</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
        </p>
    </div>
</footer>

</body>
</html>

```

## A.4 pump.php

```

<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Smart Greenhouse</title>
        <style type="text/css">

            body {
                margin:0;
                padding:0;
                font-family: Sans-Serif;
                line-height: 1.5em;

```

```

        background: #2C6566;
    }

    #header {
        background: #1F3440;
        color: white;
        height: 100px;
    }

    #header h1 {

        margin: 0;
        padding-top: 15px;

    }

    main {
        padding-bottom: 10010px;
        margin-bottom: -10000px;
        float: left;
        width: 100%;
    }

    #nav {
        padding-bottom: 10010px;
        margin-bottom: -10000px;
        float: left;
        width: 230px;
        margin-left: -100%;
        background: #25bdb0;
    }

    #footer {
        clear: left;
        width: 100%;
        background: #B4E8CF ;
        text-align: center;
        padding: 4px 0;
    }

    #wrapper {
        overflow: hidden;
    }

    #content {
        margin-left: 230px; /* Same as 'nav' width */
    }

    .innertube {
        margin: 15px; /* Padding for content */
        margin-top: 0;
    }

    p {
        color: #000000;
    }

    nav ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
    }

```



```

    }

    nav ul a {
        color: black;
        text-decoration: none;
    }

</style>

</head>

<body>

    <header id="header">
        <div class="innertube">
            <h1>Smart Greenhouse</h1>
        </div>
    </header>

    <div id="wrapper">

        <main>
            <div id="content">

                <div class="content">
                    <br>
                        <center> <strong>Διάγραμμα εκκίνησης της ανλίας
συναρτήσει χρόνου δυναμικής ανανέωσης ανα 20 δευτερόλεπτα μέσω του
Thingspeak. </strong><br></center>
                    <p>

<center> <br><br> Στο παρακάτω διάγραμμα απεικονίζονται οι τιμές
εκκίνησης της αντλίας στο "έξυπνο" θερμοκήπιο σε συνάρτηση με το
χρόνο.</p></center>
                    <center><iframe width="450" height="260" style="border: 1px solid
#cccccc;"
src="https://thingspeak.com/channels/2581223/charts/6?bgcolor=%23ff
ffff&color=%23d62020&dynamic=true&results=60&type=line&yaxis=pump&y
axismax=1&yaxismin=0"></iframe></center>
                    </div>

                </div>
            </main>

            <nav id="nav">
                <div class="innertube">
                    <h3>SMART GREENHOUSE</h3>
                    <ul>
                        <li><a href="index.html">Homepage</a></li>

                            <h3>ThingSpeak</h3>
                            <ul>
                                <li><a
href="temperature.php">Temperature</a></li>
                                <li><a
href="humidity.php">Humidity</a></li>
                                <li><a href="fan.php">Fan</a></li>
                                <li><a href="fire.php">Fire</a></li>
                                <li><a href="flood.php">Flood </a></li>

```

```

                <li><a href="pump.php">Pump</a></li>
                <li><a href="safety_door.php">Safety
door</a></li>
                <li><a href="water_tank.php">Water
tank</a></li>
            </ul>

            <h3>Google charts</h3>
            <ul>
                <li><a
href="statistics1.php">Statistics</a></li>
                <li><a href="gauge.php">Gauge</a></li>
                <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
                <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
            </ul>
            <h3>Database</h3>
            <ul>
                <li><a href="display.php">Display</a></li>
            </ul>
            <h3>Ajax I/O</h3>
            <ul>
                <li><a href="index1.htm">Control Led and
Relay</a></li>
            </ul>
        </div>
    </nav>

</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
        </p>
    </div>
</footer>

</body>
</html>

```

## A.5 fan.php

```

<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Smart Greenhouse</title>
        <style type="text/css">

            body {
                margin:0;
                padding:0;
                font-family: Sans-Serif;
                line-height: 1.5em;
                background: #2C6566;

```

```

}

#header {
    background: #1F3440;
    color: white;
    height: 100px;
}

#header h1 {

    margin: 0;
    padding-top: 15px;

}

main {
    padding-bottom: 10010px;
    margin-bottom: -10000px;
    float: left;
    width: 100%;
}

#nav {
    padding-bottom: 10010px;
    margin-bottom: -10000px;
    float: left;
    width: 230px;
    margin-left: -100%;
    background: #25bdb0;
}

#footer {
    clear: left;
    width: 100%;
    background:#B4E8CF ;
    text-align: center;
    padding: 4px 0;
}

#wrapper {
    overflow: hidden;
}

#content {
    margin-left: 230px; /* Same as 'nav' width */
}

.innertube {
    margin: 15px; /* Padding for content */
    margin-top: 0;
}

p {
    color: #000000;
}

nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}

```

```

        nav ul a {
            color: black;
            text-decoration: none;
        }
    </style>

</head>

<body>

    <header id="header">
        <div class="innertube">
            <h1>Smart Greenhouse</h1>
        </div>
    </header>

    <div id="wrapper">

        <main>
            <div id="content">

                <div class="content">
                    <br>
                        <center> <strong>Διάγραμμα κουμπιού έκτακτης
ανάγκης συναρτήσει χρόνου δυναμικής ανανέωσης ανα 20
δευτερόλεπτα μέσω του Thingspeak. </strong><br></center>
                    <p>

<center> <br><br> Στο παρακάτω διάγραμμα απεικονίζονται οι τιμές
του ανεμιστήρα εξαερισμού στο "έξυπνο" θερμοκήπιο σε συνάρτηση
με το χρόνο.</p></center>
                    <center><iframe width="450" height="260" style="border: 1px
solid #cccccc;"
src="https://thingspeak.com/channels/2581223/charts/3?bgcolor=%2
3ffffff&color=%23d62020&dynamic=true&results=60&type=line&update
=15"></iframe></center>
                    </div>

                </div>
            </main>

            <nav id="nav">
                <div class="innertube">
                    <h3>SMART GREENHOUSE</h3>
                    <ul>
                        <li><a href="index.html">Homepage</a></li>

                            <h3>ThingSpeak</h3>
                            <ul>
                                <li><a
href="temperature.php">Temperature</a></li>
                                <li><a
href="humidity.php">Humidity</a></li>
                                <li><a href="fan.php">Fan</a></li>
                                <li><a href="fire.php">Fire</a></li>
                                <li><a href="flood.php">Flood </a></li>
                                <li><a href="pump.php">Pump</a></li>

```

```

        <li><a href="safety_door.php">Safety
door</a></li>
        <li><a href="water_tank.php">Water
tank</a></li>
    </ul>

    <h3>Google charts</h3>
    <ul>
        <li><a
href="statistics1.php">Statistics</a></li>
        <li><a href="gauge.php">Gauge</a></li>
        <li><a
href="weekly_statistics1.php">Weekly Statistics</a></li>
        <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
    </ul>
    <h3>Database</h3>
    <ul>
        <li><a
href="display.php">Display</a></li>
    </ul>
    <h3>Ajax I/O</h3>
    <ul>
        <li><a href="index1.htm">Control Led and
Relay</a></li>
    </ul>
</div>
</nav>

</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#"
>Πανεπιστήμιο Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
        </p>
    </div>
</footer>

</body>
</html>

```

## A.6 fire.php

```

<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Smart Greenhouse</title>
        <style type="text/css">

            body {
                margin:0;
                padding:0;
                font-family: Sans-Serif;
                line-height: 1.5em;
                background: #2C6566;

```

```

}

#header {
    background: #1F3440;
    color: white;
    height: 100px;
}

#header h1 {

    margin: 0;
    padding-top: 15px;

}

main {
    padding-bottom: 10010px;
    margin-bottom: -10000px;
    float: left;
    width: 100%;
}

#nav {
    padding-bottom: 10010px;
    margin-bottom: -10000px;
    float: left;
    width: 230px;
    margin-left: -100%;
    background: #25bdb0;
}

#footer {
    clear: left;
    width: 100%;
    background: #B4E8CF ;
    text-align: center;
    padding: 4px 0;
}

#wrapper {
    overflow: hidden;
}

#content {
    margin-left: 230px; /* Same as 'nav' width */
}

.innertube {
    margin: 15px; /* Padding for content */
    margin-top: 0;
}

p {
    color: #000000;
}

nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}

```

```

        nav ul a {
            color: black;
            text-decoration: none;
        }
    </style>

</head>

<body>

    <header id="header">
        <div class="innertube">
            <h1>Smart Greenhouse</h1>
        </div>
    </header>

    <div id="wrapper">

        <main>
            <div id="content">

                <div class="content">
                    <br>
                        <center> <strong>Διάγραμμα ανίχνευσης φωτίας
                            συναρτήσει χρόνου δυναμικής ανανέωσης ανα 20 δευτερόλεπτα μέσω
                            του Thingspeak. </strong><br></center>
                    <p>
                        <center> <br><br> Στο παρακάτω διάγραμμα απεικονίζονται οι τιμές
                            του ανιχνευτή φωτιάς στο "έξυπνο" θερμοκήπιο σε συνάρτηση με το
                            χρόνο.</p></center>
                    <center><iframe width="450" height="260" style="border: 1px
                        solid #cccccc;"
                        src="https://thingspeak.com/channels/2581223/charts/4?bgcolor=%2
                        3ffffff&color=%23d62020&dynamic=true&results=60&type=line&update
                        =15"></iframe></center>
                    </div>

                </div>
            </main>

            <nav id="nav">
                <div class="innertube">
                    <h3>SMART GREENHOUSE</h3>
                    <ul>
                        <li><a href="index.html">Homepage</a></li>

                            <h3>ThingSpeak</h3>
                            <ul>
                                <li><a
                                    href="temperature.php">Temperature</a></li>
                                <li><a
                                    href="humidity.php">Humidity</a></li>
                                <li><a href="fan.php">Fan</a></li>
                                <li><a href="fire.php">Fire</a></li>
                                <li><a href="flood.php">Flood </a></li>
                                <li><a href="pump.php">Pump</a></li>
                            </ul>
                    </ul>
                </div>
            </nav>
        </div>
    </body>

```



```

        <li><a href="safety_door.php">Safety
door</a></li>
        <li><a href="water_tank.php">Water
tank</a></li>
    </ul>

    <h3>Google charts</h3>
    <ul>
        <li><a
href="statistics1.php">Statistics</a></li>
        <li><a href="gauge.php">Gauge</a></li>
        <li><a
href="weekly_statistics1.php">Weekly Statistics</a></li>
        <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
    </ul>
    <h3>Database</h3>
    <ul>
        <li><a
href="display.php">Display</a></li>
    </ul>
    <h3>Ajax I/O</h3>
    <ul>
        <li><a href="index1.htm">Control Led and
Relay</a></li>
    </ul>
</div>
</nav>
</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#"
>Πανεπιστήμιο Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
        </p>
    </div>
</footer>

</body>
</html>

```

## A.7 flood.php

```

<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Smart Greenhouse</title>
        <style type="text/css">

            body {
                margin:0;
                padding:0;
                font-family: Sans-Serif;
                line-height: 1.5em;
                background: #2C6566;

```

```

}

#header {
    background: #1F3440;
    color: white;
    height: 100px;
}

#header h1 {

    margin: 0;
    padding-top: 15px;

}

main {
    padding-bottom: 10010px;
    margin-bottom: -10000px;
    float: left;
    width: 100%;
}

#nav {
    padding-bottom: 10010px;
    margin-bottom: -10000px;
    float: left;
    width: 230px;
    margin-left: -100%;
    background: #25bdb0;
}

#footer {
    clear: left;
    width: 100%;
    background:#B4E8CF ;
    text-align: center;
    padding: 4px 0;
}

#wrapper {
    overflow: hidden;
}

#content {
    margin-left: 230px; /* Same as 'nav' width */
}

.innertube {
    margin: 15px; /* Padding for content */
    margin-top: 0;
}

p {
    color: #000000;
}

nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}

```

```

        nav ul a {
            color: black;
            text-decoration: none;
        }
    </style>

</head>

<body>

    <header id="header">
        <div class="innertube">
            <h1>Smart Greenhouse</h1>
        </div>
    </header>

    <div id="wrapper">

        <main>
            <div id="content">

                <div class="content">
                    <br>
                    <center> <strong>Διάγραμμα ανίχνευσης πλημμύρας
                    συναρτήσει χρόνου δυναμικής ανανέωσης ανα 20 δευτερόλεπτα μέσω του
                    Thingspeak. </strong><br></center>
                <p>

                <center> <br><br> Στο παρακάτω διάγραμμα απεικονίζονται οι τιμές
                του ανιχνευτή πλημμύρας στο "έξυπνο" θερμοκήπιο σε συνάρτηση με το
                χρόνο.</p></center>
                <center><iframe width="450" height="260" style="border: 1px solid
                #cccccc;"
                src="https://thingspeak.com/channels/2581223/charts/5?bgcolor=%23ff
                ffff&color=%23d62020&dynamic=true&results=60&type=line&update=15">
                /iframe></center>
            </div>

        </div>
    </main>

    <nav id="nav">
        <div class="innertube">
            <h3>SMART GREENHOUSE</h3>
            <ul>
                <li><a href="index.html">Homepage</a></li>

                <h3>ThingSpeak</h3>
                <ul>
                    <li><a
                    href="temperature.php">Temperature</a></li>
                    <li><a
                    href="humidity.php">Humidity</a></li>
                    <li><a href="fan.php">Fan</a></li>
                    <li><a href="fire.php">Fire</a></li>
                    <li><a href="flood.php">Flood </a></li>
                    <li><a href="pump.php">Pump</a></li>

```

```

                <li><a href="safety_door.php">Safety
door</a></li>
                <li><a href="water_tank.php">Water
tank</a></li>
            </ul>

            <h3>Google charts</h3>
            <ul>
                <li><a
href="statistics1.php">Statistics</a></li>
                <li><a href="gauge.php">Gauge</a></li>
                <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
                <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
            </ul>
            <h3>Database</h3>
            <ul>
                <li><a href="display.php">Display</a></li>
            </ul>
            <h3>Ajax I/O</h3>
            <ul>
                <li><a href="index1.htm">Control Led and
Relay</a></li>
            </ul>
        </div>
    </nav>

</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
        </p>
    </div>
</footer>

</body>
</html>

```

## A.8 gauge.php

```

<?php
// Prepare variables for database connection

$username = "root"; // enter database username, I used
"arduino" in step 2.2
$password = ""; // enter database password, I used
"arduinotest" in step 2.2
$server = "localhost";

// Connect to your database
$mysqli = new mysqli($server, $username, $password,
"greenhouse");
if($mysqli->connect_error)
    die('Connect Error (' . mysqli_connect_errno() . ') ' .
mysqli_connect_error());

```

```

// Execute SQL statement

$result = $mysqli->query("SELECT temperature, humidity, fan,
fire, flood, pump, shades, safety_door, water_tank FROM data ORDER
BY id DESC LIMIT 1"); // tin teleytaia metrisi pernw apo vasi me to
limit 1

if ($result->num_rows > 0) {

    while($row = $result->fetch_assoc()) {
        $TEValue = $row["temperature"];
        $HValue = $row["humidity"];
        $FAValue = $row["fan"];
        $FIValue = $row["fire"];
        $FLValue = $row["flood"];
        $puValue = $row["pump"];
        $shValue = $row["shades"];
        $pValue = $row["safety_door"];
        $wtValue = $row["water_tank"];
    }
}
else {
    $TEValue = 0;
    $HValue = 0;
    $FAValue = 0;
    $FIValue = 0;
    $FLValue = 0;
    $puValue = 0;
    $shValue= 0;
    $pValue = 0;
    $wtValue = 0;
}

?>
<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Smart Greenhouse</title>
        <style type="text/css">

            body {
                margin:0;
                padding:0;
                font-family: Sans-Serif;
                line-height: 1.5em;
                background: #2C6566;
            }

            #header {
                background: #1F3440;
                color: white;
                height: 100px;
            }

            #header h1 {

```

```
        margin: 0;
        padding-top: 15px;
    }

    main {
        padding-bottom: 10010px;
        margin-bottom: -10000px;
        float: left;
        width: 100%;
    }

    #nav {
        padding-bottom: 10010px;
        margin-bottom: -10000px;
        float: left;
        width: 230px;
        margin-left: -100%;
        background: #25bdb0;
    }

    #footer {
        clear: left;
        width: 100%;
        background:#B4E8CF ;
        text-align: center;
        padding: 4px 0;
    }

    #wrapper {
        overflow: hidden;
    }

    #content {
        margin-left: 230px; /* Same as 'nav' width */
    }

    .innertube {
        margin: 15px; /* Padding for content */
        margin-top: 0;
    }

    p {
        color: #000000;
    }

    nav ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
    }

    nav ul a {
        color: black;
        text-decoration: none;
    }

</style>
```

```

</head>

<body>

  <header id="header">
    <div class="innertube">
      <h1>Smart Greenhouse</h1>
    </div>
  </header>
  <div id="wrapper">

    <main>
      <div id="content">

        <div class="content">

          <script src="gauge.js"></script>
          <meta http-equiv="refresh" content="180" />
          <br>
          <center><strong>Γραφική Απεικόνιση με Google Gauge
</strong></center><br>
          <p>

```

<br><br>Στα παρακάτω γραφήματα απεικονίζονται οι τελευταίες τιμές των αισθητήρων με μορφή Gauge.

Στα δύο πρώτα γραφήματα απεικονίζονται οι τιμές της θερμοκρασίας και της υγρασίας, ενώ στα τελευταία επτά οι τιμές του ανεμιστήρα, της φωτιάς, της πλημμύρας, της αντλίας, του σκιάστρου, της πόρτας και της δεξαμενής που λειτουργούν ως αυτοματισμοί. Την στιγμή που θα ενεργοποιηθεί η συνθήκη του ανεμιστήρα, της φωτιάς, της πλημμύρας, του σκιάστρου, της πόρτας ή της αντλίας η τιμή θα δείξει '1' και θα λειτουργήσουν τα αντίστοιχα στοιχεία. </p>

```

<script>
if (!Array.prototype.forEach) {
  Array.prototype.forEach = function(cb) {
    var i = 0, s = this.length;
    for (; i < s; i++) {
      cb && cb(this[i], i, this);
    }
  }
}

document.fonts && document.fonts.forEach(function(font) {
  font.loaded.then(function() {
    if (font.family.match(/Led/)) {
      document.gauges.forEach(function(gauge) {
        gauge.update();
        gauge.options.renderTo.style.visibility =
'visible';
      });
    }
  });
});

var timers = [];

function animateGauges() {
  document.gauges.forEach(function(gauge) {
    timers.push(setInterval(function() {

```



```

        gauge.value = Math.random() *
            (gauge.options.maxValue - gauge.options.minValue) +
            gauge.options.minValue;
    }, gauge.animation.duration + 50));
    });
}

function stopGaugesAnimation() {
    timers.forEach(function(timer) {
        clearInterval(timer);
    });
}

function resize() {
    var size = parseFloat(document.getElementById('gauge-
size').value) || 400;

    document.gauges.forEach(function(gauge) {
        gauge.update({ width: size, height: size });
    });
}

function setText() {
    var text = document.getElementById('gauge-text').value;

    document.gauges.forEach(function(gauge) {
        gauge.update({ valueText: text });
    });
}
</script>

</head>
<body>

<canvas data-type="radial-gauge"
    data-width="250"
    data-height="250"
    data-units="&deg;C"
    data-title="Temperature"
    data-value=<?php echo "\".$.TEValue."\""\n"?>
    data-animate-on-init="false"
    data-animated-value="false"
    data-min-value="0"
    data-max-value="100"
    data-major-ticks="0,10,20,30,40,50,60,70,80,90,100"
    data-minor-ticks="0.1"
    data-stroke-ticks="false"
    data-highlights=' [
        { "from": 0, "to": 30, "color": "rgba(0,255,0)" },
        { "from": 30, "to": 65, "color": "rgba(255, 255, 0)" },
        { "from": 65, "to": 100, "color": "rgba(255, 0, 0)" }
    ] '
    data-color-plate="#001433"
    data-color-major-ticks="#f5f5f5"
    data-color-minor-ticks="#ddd"
    data-color-title="#fff"

```

```

data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
<</canvas>

<canvas data-type="radial-gauge"
data-width="250"
data-height="250"
data-units="%"
data-title="Humidity"
data-value=<?php echo "\"".$SHValue."\""\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="100"
data-major-ticks="0,10,20,30,40,50,60,70,80,90,100"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights=' [
    { "from": 0, "to": 30, "color": "rgba(0, 153, 0)" },
    { "from": 30, "to": 65, "color": "rgba(255, 128, 0)" },
    { "from": 65, "to": 100, "color": "rgba(204, 0, 0)" }
] '
data-color-plate="#001433"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
<</canvas>

<canvas data-type="radial-gauge"
data-width="250"
data-height="250"
data-units=""
data-title="Fan"
data-value=<?php echo "\"".$FAValue."\""\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="1"
data-major-ticks="0,1"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights=' [

```

```

        { "from": 0, "to": 0.5, "color": "rgba(255, 255, 255)"
    },
        { "from": 0.5, "to": 1, "color": "rgba(0, 128, 255)" }
    ]'
    data-color-plate="#001433"
    data-color-major-ticks="#f5f5f5"
    data-color-minor-ticks="#ddd"
    data-color-title="#fff"
    data-color-units="#ccc"
    data-color-numbers="#eee"
    data-color-needle-start="rgba(240, 128, 128, 1)"
    data-color-needle-end="rgba(255, 160, 122, .9)"
    data-value-box="true"
    data-animation-rule="bounce"
    data-animation-duration="500"
    data-border-outer-width="3"
    data-border-middle-width="3"
    data-border-inner-width="3"
<</canvas>
<canvas data-type="radial-gauge"
    data-width="250"
    data-height="250"
    data-units=""
    data-title="Fire"
    data-value=<?php echo "\"".$FIValue."\""\n"?>
    data-animate-on-init="false"
    data-animated-value="false"
    data-min-value="0"
    data-max-value="1"
    data-major-ticks="0,1"
    data-minor-ticks="0.1"
    data-stroke-ticks="false"
    data-highlights=' [
        { "from": 0, "to": 0.5, "color": "rgba(255, 255, 255)"
    },
        { "from": 0.5, "to": 1, "color": "rgba(255, 0, 0)" }
    ]'
    data-color-plate="#001433"
    data-color-major-ticks="#f5f5f5"
    data-color-minor-ticks="#ddd"
    data-color-title="#fff"
    data-color-units="#ccc"
    data-color-numbers="#eee"
    data-color-needle-start="rgba(240, 128, 128, 1)"
    data-color-needle-end="rgba(255, 160, 122, .9)"
    data-value-box="true"
    data-animation-rule="bounce"
    data-animation-duration="500"
    data-border-outer-width="3"
    data-border-middle-width="3"
    data-border-inner-width="3"
<</canvas>
<canvas data-type="radial-gauge"
    data-width="250"
    data-height="250"
    data-units=""
    data-title="Flood"
    data-value=<?php echo "\"".$FLValue."\""\n"?>

```

```

data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="1"
data-major-ticks="0,1"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights=' [
    { "from": 0, "to": 0.5, "color": "rgba(255, 255, 255)"
},
    { "from": 0.5, "to": 1, "color": "rgba(0, 0, 255)" }
]'
data-color-plate="#001433"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
</canvas>
<canvas data-type="radial-gauge"
data-width="250"
data-height="250"
data-units=""
data-title="Pump"
data-value=<?php echo "\".$puValue."\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="1"
data-major-ticks="0,1"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights=' [
    { "from": 0, "to": 0.5, "color": "rgba(255, 255, 255)"
},
    { "from": 0.5, "to": 1, "color": "rgba(128, 128, 128)"
}

]'
data-color-plate="#001433"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"

```

```

<</canvas>
    <canvas data-type="radial-gauge"
data-width="250"
data-height="250"
data-units=""
data-title="Shades"
data-value=<?php echo "\"".$shValue."\""\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="1"
data-major-ticks="0,1"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights=' [
    { "from": 0, "to": 0.5, "color": "rgba(255, 255, 255)"
},
    { "from": 0.5, "to": 1, "color": "rgba(0, 204, 0)" }
] '
data-color-plate="#001433"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
<</canvas>
<canvas data-type="radial-gauge"
data-width="250"
data-height="250"
data-units=""
data-title="Safety_door"
data-value=<?php echo "\"".$pValue."\""\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="1"
data-major-ticks="0,1"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights=' [
    { "from": 0, "to": 0.5, "color": "rgba(255, 255, 255)"
},
    { "from": 0.5, "to": 1, "color": "rgba(168, 100, 50)" }
] '
data-color-plate="#001433"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"

```

```

        data-value-box="true"
        data-animation-rule="bounce"
        data-animation-duration="500"
        data-border-outer-width="3"
        data-border-middle-width="3"
        data-border-inner-width="3"
    </canvas>
    <canvas data-type="radial-gauge"
        data-width="250"
        data-height="250"
        data-units=""
        data-title="Water_tank"
        data-value=<?php echo "\"" . $wtValue . "\"\n"?>
        data-animate-on-init="false"
        data-animated-value="false"
        data-min-value="0"
        data-max-value="1"
        data-major-ticks="0,1"
        data-minor-ticks="0.1"
        data-stroke-ticks="false"
        data-highlights='[
            { "from": 0, "to": 0.5, "color": "rgba(255, 255, 255)"
        },
            { "from": 0.5, "to": 1, "color": "rgba(50, 200, 168)" }
        ]'
        data-color-plate="#001433"
        data-color-major-ticks="#f5f5f5"
        data-color-minor-ticks="#ddd"
        data-color-title="#fff"
        data-color-units="#ccc"
        data-color-numbers="#eee"
        data-color-needle-start="rgba(240, 128, 128, 1)"
        data-color-needle-end="rgba(255, 160, 122, .9)"
        data-value-box="true"
        data-animation-rule="bounce"
        data-animation-duration="500"
        data-border-outer-width="3"
        data-border-middle-width="3"
        data-border-inner-width="3"
    </canvas>

    <p></p>
<p></p>
<p></p>

</div>
    </div>
</main>

<nav id="nav">
    <div class="innertube">
        <h3>SMART GREENHOUSE</h3>
        <ul>
            <li><a href="index.html">Homepage</a></li>

                <h3>ThingSpeak</h3>
                <ul>
                    <li><a
href="temperature.php">Temperature</a></li>

```

```

        <li><a
href="humidity.php">Humidity</a></li>
        <li><a href="fan.php">Fan</a></li>
        <li><a href="fire.php">Fire</a></li>
        <li><a href="flood.php">Flood </a></li>
        <li><a href="pump.php">Pump</a></li>
        <li><a href="safety_door.php">Safety
door</a></li>
        <li><a href="water_tank.php">Water
tank</a></li>
    </ul>

    <h3>Google charts</h3>
    <ul>
        <li><a
href="statistics1.php">Statistics</a></li>
        <li><a href="gauge.php">Gauge</a></li>
        <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
        <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
    </ul>
    <h3>Database</h3>
    <ul>
        <li><a href="display.php">Display</a></li>
    </ul>
    <h3>Ajax I/O</h3>
    <ul>
        <li><a href="index1.htm">Control Led and
Relay</a></li>
    </ul>
</div>
</nav>

</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
        </p>
    </div>
</footer>

</body>
</html>

```

## A.9 humidity.php

```

<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Smart Greenhouse</title>
    <style type="text/css">

        body {
            margin:0;

```



```

padding:0;
font-family: Sans-Serif;
line-height: 1.5em;
background: #2C6566;
}

#header {
background: #1F3440;
color: white;
height: 100px;
}

#header h1 {

margin: 0;
padding-top: 15px;

}

main {
padding-bottom: 10010px;
margin-bottom: -10000px;
float: left;
width: 100%;
}

#nav {
padding-bottom: 10010px;
margin-bottom: -10000px;
float: left;
width: 230px;
margin-left: -100%;
background: #25bdb0;
}

#footer {
clear: left;
width: 100%;
background:#B4E8CF ;
text-align: center;
padding: 4px 0;
}

#wrapper {
overflow: hidden;
}

#content {
margin-left: 230px; /* Same as 'nav' width */
}

.innertube {
margin: 15px; /* Padding for content */
margin-top: 0;
}

p {
color: #000000;
}

nav ul {

```

```

        list-style-type: none;
        margin: 0;
        padding: 0;
    }

    nav ul a {
        color: black;
        text-decoration: none;
    }

</style>

</head>

<body>

    <header id="header">
        <div class="innertube">
            <h1>Smart Greenhouse</h1>
        </div>
    </header>

    <div id="wrapper">

        <main>
            <div id="content">

                <div class="content">
                    <br>
                    <center> <strong>Διάγραμμα υγρασίας συναρτήσει χρόνου
δυναμικής ανανέωσης ανα 20 δευτερόλεπτα μέσω του Thingspeak.
</strong><br></center>
                    <p>

<center> <br><br> Στο παρακάτω διάγραμμα απεικονίζονται οι τιμές
της υγρασίας στο "έξυπνο" θερμοκήπιο σε συνάρτηση με το
χρόνο.</p></center>
                    <center><iframe width="450" height="260" style="border: 1px solid
#cccccc;"
src="https://thingspeak.com/channels/2581223/charts/2?bgcolor=%23ff
ffff&color=%23d62020&dynamic=true&results=60&type=line&update=15">
</iframe></center>
                    </div>
                </div>
            </main>

            <nav id="nav">
                <div class="innertube">
                    <h3>SMART GREENHOUSE</h3>
                    <ul>
                        <li><a href="index.html">Homepage</a></li>

                        <h3>ThingSpeak</h3>
                        <ul>
                            <li><a
href="temperature.php">Temperature</a></li>
                            <li><a
href="humidity.php">Humidity</a></li>

```

```

        <li><a href="fan.php">Fan</a></li>
        <li><a href="fire.php">Fire</a></li>
        <li><a href="flood.php">Flood </a></li>
        <li><a href="pump.php">Pump</a></li>
        <li><a href="safety_door.php">Safety
door</a></li>
        <li><a href="water_tank.php">Water
tank</a></li>
    </ul>

    <h3>Google charts</h3>
    <ul>
        <li><a
href="statistics1.php">Statistics</a></li>
        <li><a href="gauge.php">Gauge</a></li>
        <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
        <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
    </ul>
    <h3>Database</h3>
    <ul>
        <li><a href="display.php">Display</a></li>
    </ul>
    <h3>Ajax I/O</h3>
    <ul>
        <li><a href="index1.htm">Control Led and
Relay</a></li>
    </ul>
</div>
</nav>

</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
        </p>
    </div>
</footer>

</body>
</html>

```

## A.10 monthly\_statistics.php

```

<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Smart Greenhouse</title>
        <style type="text/css">

```

```

body {
    margin:0;
    padding:0;
    font-family: Sans-Serif;
    line-height: 1.5em;
    background: #2C6566;
}

#header {
    background: #1F3440;
    color: white;
    height: 100px;
}

#header h1 {

    margin: 0;
    padding-top: 15px;

}

main {
    padding-bottom: 10010px;
    margin-bottom: -10000px;
    float: left;
    width: 100%;
}

#nav {
    padding-bottom: 10010px;
    margin-bottom: -10000px;
    float: left;
    width: 230px;
    margin-left: -100%;
    background: #25bdb0;
}

#footer {
    clear: left;
    width: 100%;
    background:#B4E8CF ;
    text-align: center;
    padding: 4px 0;
}

#wrapper {
    overflow: hidden;
}

#content {
    margin-left: 230px; /* Same as 'nav' width */
}

.innertube {
    margin: 15px; /* Padding for content */
    margin-top: 0;
}

p {
    color: #000000;
}

```

```

        nav ul {
            list-style-type: none;
            margin: 0;
            padding: 0;
        }

        nav ul a {
            color: black;
            text-decoration: none;
        }

</style>

</head>

<body>

    <header id="header">
        <div class="innertube">
            <h1>Smart Greenhouse</h1>
        </div>
    </header>

    <div id="wrapper">

        <main>
            <div id="content">
                <div class="content">
                    <meta http-equiv="refresh" content="86400" />
                    <br>
                    <center><strong>Μηνιαία
Στατιστικά</strong></center><br>
                    <p>
                        <br><br> Στα παρακάτω διαγράμματα απεικονίζονται η ελάχιστη,
μέγιστη και μέση τιμή των αισθητήρων
                        (θερμοκρασίας, υγρασίας, εξαερισμού, φωτιάς, πλημμύρας,
ανιλίας, σκιάστρου, πόρτας,δεξαμενή νερού) για τον τελευταίο
μήνα.</p>
                    <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
                    <script type="text/javascript">
                        google.charts.load('current', {'packages':['corechart']});
                        google.charts.setOnLoadCallback(drawChart);

                    function drawChart () {

                        var data = new google.visualization.DataTable();
                        data.addColumn('datetime', 'event');
                        data.addColumn('number', 'Average temperature');
                        data.addColumn('number', 'Max temperature');
                        data.addColumn('number', 'Min temperature');
                        data.addColumn('number', 'Average humidity');
                        data.addColumn('number', 'Max humidity');
                        data.addColumn('number', 'Min humidity');
                        data.addColumn('number', 'Average fan');
                        data.addColumn('number', 'Max fan');
                        data.addColumn('number', 'Min fan');
                        data.addColumn('number', 'Average fire');
                        data.addColumn('number', 'Max fire');
                    }
                </div>
            </div>
        </main>
    </div>

```

```

        data.addColumn('number', 'Min fire');
        data.addColumn('number', 'Average flood');
        data.addColumn('number', 'Max flood');
        data.addColumn('number', 'Min flood');
        data.addColumn('number', 'Average pump');
        data.addColumn('number', 'Max pump');
        data.addColumn('number', 'Min pump');
        data.addColumn('number', 'Average shades');
        data.addColumn('number', 'Max shades');
        data.addColumn('number', 'Min shades');
        data.addColumn('number', 'Average safety_door');
        data.addColumn('number', 'Max safety_door');
        data.addColumn('number', 'Min safety_door');
        data.addColumn('number', 'Average water_tank ');
        data.addColumn('number', 'Max water_tank');
        data.addColumn('number', 'Min water_tank');

<?php
    // Prepare variables for database connection

    $dbusername = "root"; // enter database username, I used
"arduino" in step 2.2
    $dbpassword = ""; // enter database password, I used
"arduinotest" in step 2.2
    $server = "localhost";

    // Connect to your database
    $mysqli = new mysqli($server, $dbusername, $dbpassword,
"greenhouse");
    if($mysqli->connect_error)
        die('Connect Error (' . mysqli_connect_errno() . ') ' .
mysqli_connect_error());

    // Execute SQL statement ta grouparw ana mina kai krataw
teleytaia 12 +min max +average
    $result = $mysqli->query("SELECT YEAR(event) AS year,
MONTH(event) AS month, DAY(event) AS day, AVG(temperature),
MAX(temperature), MIN(temperature), AVG(humidity), MAX(humidity),
MIN(humidity), AVG(fan), MAX(fan), MIN(fan), AVG(fire), MAX(fire),
MIN(fire), AVG(flood), MAX(flood), MIN(flood), AVG(pump),
MAX(pump), MIN(pump), AVG(shades), MAX(shades),
MIN(shades),AVG(safety_door), MAX(safety_door),
MIN(safety_door),AVG(water_tank), MAX(water_tank), MIN(water_tank)
from data group by MONTH(event) ORDER BY event DESC limit 12");

    if ($result->num_rows > 0) {

        // output data of each row
        while($row = $result->fetch_assoc()) {
            echo "data.addRow([new Date(".$row["year"].",
".$row["month"]."-1, ".$row["day"]."),
".$row["AVG(temperature)"].", ".$row["MAX(temperature)"].",
".$row["MIN(temperature)"].", ".$row["AVG(humidity)"].",
".$row["MAX(humidity)"].", ".$row["MIN(humidity)"].",
".$row["AVG(fan)"].", ".$row["MAX(fan)"].", ".$row["MIN(fan)"].",
".$row["AVG(fire)"].", ".$row["MAX(fire)"].",
".$row["MIN(fire)"].", ".$row["AVG(flood)"].",
".$row["MAX(flood)"].", ".$row["MIN(flood)"].",
".$row["AVG(pump)"].", ".$row["MAX(pump)"].",
".$row["MIN(pump)"].", ".$row["AVG(shades)"].",

```

```

".$row["MAX(shades)"].", ".$row["MIN(shades)"].",
".$row["AVG(safety_door)"].", ".$row["MAX(safety_door)"].",
".$row["MIN(safety_door)"].", ".$row["AVG(water_tank)"].",
".$row["MAX(water_tank)"].", ".$row["MIN(water_tank)"].");\n";
    }
}
else {
}
?>
var temperature_options = {
    title: 'temperature (C) Last Month',
    subtitle: 'measured in C',
    hAxis: { format: ' '},
    seriesType: 'bars',
    width: 600,
    height: 300
};

var humidity_options = {
    title: 'humidity (%) Last Month',
    subtitle: 'measured in %',
    hAxis: { format: ' '},
    seriesType: 'bars',
    width: 600,
    height: 300
};

var fan_options = {
    title: 'Fan Last Month',
    subtitle: '',
    hAxis: { format: ' '},
    seriesType: 'bars',
    width: 600,
    height: 300
};

var fire_options = {
    title: 'Fire Last Month ',
    subtitle: '',
    hAxis: { format: ' '},
    seriesType: 'bars',
    width: 600,
    height: 300
};

var flood_options = {
    title: 'Flood Last Month ',
    subtitle: '',
    hAxis: { format: ' '},
    seriesType: 'bars',
    width: 600,
    height: 300
};

var pump_options = {
    title: 'Pump Last Month ',
    subtitle: '',
    hAxis: { format: ' '},
    seriesType: 'bars',
    width: 600,
    height: 300
};

```

```

};

var shades_options = {
  title: 'Shades Last Month ',
  subtitle: ' ',
  hAxis: { format: ' ' },
  seriesType: 'bars',
  width: 600,
  height: 300
};

var safety_door_options = {
  title: 'Safety_door Last Month ',
  subtitle: ' ',
  hAxis: { format: ' ' },
  seriesType: 'bars',
  width: 600,
  height: 300
};

var water_tank_options = {
  title: 'Water_tank Last Month ',
  subtitle: ' ',
  hAxis: { format: ' ' },
  seriesType: 'bars',
  width: 600,
  height: 300
};

var temperature_chart = new
google.visualization.ComboChart (document.getElementById('temperatur
e_div'));

var temperature_view = new
google.visualization.DataView(data);
temperature_view.setColumns ([0,1,2,3]);
temperature_chart.draw(temperature_view,
temperature_options);

//-----
var humidity_chart = new
google.visualization.ComboChart (document.getElementById('humidity_d
iv'));

var humidity_view = new google.visualization.DataView(data);
humidity_view.setColumns ([0,4,5,6]);
humidity_chart.draw(humidity_view, humidity_options);

//-----
var fan_chart = new
google.visualization.ComboChart (document.getElementById('fan_div'))
;

var fan_view = new google.visualization.DataView(data);
fan_view.setColumns ([0,7,8,9]);
fan_chart.draw(fan_view, fan_options);

//-----

```



```

    var fire_chart = new
google.visualization.ComboChart (document.getElementById('fire_div')
);

    var fire_view = new google.visualization.DataView(data);
fire_view.setColumns ([0,10,11,12]);
fire_chart.draw(fire_view, fire_options);
//-----
-----

    var flood_chart = new
google.visualization.ComboChart (document.getElementById('flood_div'
));

    var flood_view = new google.visualization.DataView(data);
flood_view.setColumns ([0,13,14,15]);
flood_chart.draw(flood_view, flood_options);
//-----
-----

    var pump_chart = new
google.visualization.ComboChart (document.getElementById('pump_div'
));

    var pump_view = new google.visualization.DataView(data);
pump_view.setColumns ([0,16,17,18]);
pump_chart.draw(pump_view, pump_options);
//-----
-----

    var shades_chart = new
google.visualization.ComboChart (document.getElementById('shades_div
'));

    var shades_view = new google.visualization.DataView(data);
shades_view.setColumns ([0,19,20,21]);
shades_chart.draw(shades_view,shades_options);
//-----
-----

    var safety_door_chart = new
google.visualization.ComboChart (document.getElementById('safety_doo
r_div'));

    var safety_door_view = new
google.visualization.DataView(data);
safety_door_view.setColumns ([0,22,23,24]);
safety_door_chart.draw(safety_door_view,safety_door_options);
//-----
-----

    var water_tank_chart = new
google.visualization.ComboChart (document.getElementById('water_tank
_div'));

    var water_tank_view = new
google.visualization.DataView(data);
water_tank_view.setColumns ([0,25,26,27]);
water_tank_chart.draw(water_tank_view,water_tank_options);
//-----
-----

}
</script>
</head>
<body>

```

```

<div style="width:100%">

</div>

<table style="border-spacing: 15px">
  <tr>
    <td style="border: 1px solid green">
      <div id="temperature_div" style="width: 600px;
height: 300px"></div>
    </td>
  <tr>
    <td style="border: 1px solid green">
      <div id="humidity_div" style="width: 600px;
height: 300px"></div>
    </td>
  </tr>
  <tr>
    <td style="border: 1px solid green">
      <div id="fan_div" style="width: 600px; height:
300px"></div>
    </td>
  <tr>
    <td style="border: 1px solid green">
      <div id="fire_div" style="width: 600px; height:
300px"></div>
    </td>
  <tr>
    <td style="border: 1px solid green">
      <div id="flood_div" style="width: 600px;
height: 300px"></div>
    </td>
  <tr>
    <td style="border: 1px solid green">
      <div id="pump_div" style="width: 600px; height:
300px"></div>
    </td>
  <tr>
    <td style="border: 1px solid green">
      <div id="shades_div" style="width: 600px;
height: 300px"></div>
    </td>
  <tr>
    <td style="border: 1px solid green">
      <div id="safety_door_div" style="width: 600px;
height: 300px"></div>
    </td>
  <tr>
    <td style="border: 1px solid green">
      <div id="water_tank_div" style="width: 600px;
height: 300px"></div>
    </td>
  <tr>
  </table>

```

```

</div>
    </div>
</main>

<nav id="nav">
    <div class="innertube">
        <h3>SMART GREENHOUSE</h3>
        <ul>
            <li><a href="index.html">Homepage</a></li>

                <h3>ThingSpeak</h3>
                <ul>
                    <li><a
href="temperature.php">Temperature</a></li>
                    <li><a
href="humidity.php">Humidity</a></li>
                    <li><a href="fan.php">Fan</a></li>
                    <li><a href="fire.php">Fire</a></li>
                    <li><a href="flood.php">Flood </a></li>
                    <li><a href="pump.php">Pump</a></li>
                    <li><a href="safety_door.php">Safety
door</a></li>
                    <li><a href="water_tank.php">Water
tank</a></li>
                </ul>

                <h3>Google charts</h3>
                <ul>
                    <li><a
href="statistics1.php">Statistics</a></li>
                    <li><a href="gauge.php">Gauge</a></li>
                    <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
                    <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
                </ul>
                <h3>Database</h3>
                <ul>
                    <li><a href="display.php">Display</a></li>
                </ul>
                <h3>Ajax I/O</h3>
                <ul>
                    <li><a href="index1.htm">Control Led and
Relay</a></li>
                </ul>
            </div>
        </nav>

    </div>

    <footer id="footer">
        <div class="innertube">
            <p>
                <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
                Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
            </p>
        </div>
    </footer>

```

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

## A.11 safety\_door.php

```
<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Smart Greenhouse</title>
    <style type="text/css">

      body {
        margin:0;
        padding:0;
        font-family: Sans-Serif;
        line-height: 1.5em;
        background: #2C6566;
      }

      #header {
        background: #1F3440;
        color: white;
        height: 100px;
      }

      #header h1 {

        margin: 0;
        padding-top: 15px;
      }

      main {
        padding-bottom: 10010px;
        margin-bottom: -10000px;
        float: left;
        width: 100%;
      }

      #nav {
        padding-bottom: 10010px;
        margin-bottom: -10000px;
        float: left;
        width: 230px;
        margin-left: -100%;
        background: #25bdb0;
      }

      #footer {
        clear: left;
        width: 100%;
        background:#B4E8CF ;
        text-align: center;
        padding: 4px 0;
      }
    </style>
  </head>
  <body>
  </body>
</html>
```

```

#wrapper {
    overflow: hidden;
}

#content {
    margin-left: 230px; /* Same as 'nav' width */
}

.innertube {
    margin: 15px; /* Padding for content */
    margin-top: 0;
}

p {
    color: #000000;
}

nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}

nav ul a {
    color: black;
    text-decoration: none;
}

</style>

</head>

<body>

<header id="header">
    <div class="innertube">
        <h1>Smart Greenhouse</h1>
    </div>
</header>

<div id="wrapper">

    <main>
        <div id="content">

            <div class="content">
                <br>
                <center> <strong>Διάγραμμα ξεκλειδώματος πόρτας
                συναρτήσει χρόνου δυναμικής ανανέωσης ανα 20 δευτερόλεπτα μέσω του
                Thingspeak. </strong><br></center>
                <p>
                <center> <br><br> Στο παρακάτω διάγραμμα απεικονίζονται οι τιμές
                του ξεκλειδώματος της πόρτας στο "έξυπνο" θερμοκήπιο σε συνάρτηση
                με το χρόνο.</p></center>
                <center><iframe width="450" height="260" style="border: 1px solid
                #cccccc;"
                src="https://thingspeak.com/channels/2581223/charts/?bgcolor=%23ff

```

```

ffff&color=%23d62020&dynamic=true&results=60&type=line&yaxis=safety
_door"></iframe></center>
</div>
    </div>
</main>

<nav id="nav">
    <div class="innertube">
        <h3>SMART GREENHOUSE</h3>
        <ul>
            <li><a href="index.html">Homepage</a></li>

                <h3>ThingSpeak</h3>
                <ul>
                    <li><a
href="temperature.php">Temperature</a></li>
                    <li><a
href="humidity.php">Humidity</a></li>
                    <li><a href="fan.php">Fan</a></li>
                    <li><a href="fire.php">Fire</a></li>
                    <li><a href="flood.php">Flood </a></li>
                    <li><a href="pump.php">Pump</a></li>
                    <li><a href="safety_door.php">Safety
door</a></li>
                    <li><a href="water_tank.php">Water
tank</a></li>
                </ul>

                <h3>Google charts</h3>
                <ul>
                    <li><a
href="statistics1.php">Statistics</a></li>
                    <li><a href="gauge.php">Gauge</a></li>
                    <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
                    <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
                </ul>
                <h3>Database</h3>
                <ul>
                    <li><a href="display.php">Display</a></li>
                </ul>
                <h3>Ajax I/O</h3>
                <ul>
                    <li><a href="index1.htm">Control Led and
Relay</a></li>
                </ul>
            </div>
        </nav>

    </div>

    <footer id="footer">
        <div class="innertube">
            <p>
                <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
                Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
            </p>
        </div>
    </footer>

```

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

## A.12 statistics.php

```
<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Smart Greenhouse</title>
    <style type="text/css">

      body {
        margin:0;
        padding:0;
        font-family: Sans-Serif;
        line-height: 1.5em;
        background: #2C6566;
      }

      #header {
        background: #1F3440;
        color: white;
        height: 100px;
      }

      #header h1 {

        margin: 0;
        padding-top: 15px;

      }

      main {
        padding-bottom: 10010px;
        margin-bottom: -10000px;
        float: left;
        width: 100%;
      }

      #nav {
        padding-bottom: 10010px;
        margin-bottom: -10000px;
        float: left;
        width: 230px;
        margin-left: -100%;
        background: #25bdb0;
      }

      #footer {
        clear: left;
        width: 100%;
        background:#B4E8CF ;
        text-align: center;
        padding: 4px 0;
      }

      #wrapper {
        overflow: hidden;
```

```

    }

    #content {
        margin-left: 230px; /* Same as 'nav' width */
    }

    .innertube {
        margin: 15px; /* Padding for content */
        margin-top: 0;
    }

    p {
        color: #000000;
    }

    nav ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
    }

    nav ul a {
        color: black;
        text-decoration: none;
    }

</style>

</head>

<body>

    <header id="header">
        <div class="innertube">
            <h1>Smart Greenhouse</h1>
        </div>
    </header>

    <div id="wrapper">

        <main>
            <div id="content">
                <div class="content">
                    <meta http-equiv="refresh" content="60" /> <!--refresh
selidas-->
                    <br> <center><strong>Διαγράμματα συναρτήσει χρόνου
δυναμικής ανανέωσης </strong></center><br>
                    <p>

                        <center><br><br> Στα παρακάτω διαγράμματα απεικονίζονται οι τελευταίες
τιμές των αισθητήρων του Smart Greenhouse σε συνάρτηση με το χρόνο. Το
τελευταίο διάγραμμα περιέχει όλες τις τιμές των αισθητήρων συναρτήσει
του χρόνου.
                        </center></p>
                        <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
                        <table align="center">

```



```

<script type="text/javascript">

    google.charts.load('current', {'packages':['line']});
    google.charts.setOnLoadCallback(drawChart);

    function drawChart() {

        var data = new google.visualization.DataTable();// ftiaxnw
pinaka gia na valw mesa ta data moy ta opoia ta pernw apo ti vasi
        data.addColumn('datetime', 'event');
        data.addColumn('number', 'temperature');
        data.addColumn('number', 'humidity');
        data.addColumn('number', 'fan');
        data.addColumn('number', 'fire');
        data.addColumn('number', 'flood');
        data.addColumn('number', 'pump');
        data.addColumn('number', 'shades');
        data.addColumn('number', 'safety_door');
        data.addColumn('number', 'water_tank');

<?php
    // Prepare variables for database connection

    $dbusername = "root"; // enter database username, I used
"arduino" in step 2.2
    $dbpassword = ""; // enter database password, I used
"arduinotest" in step 2.2
    $server = "localhost";

    // Connect to your database
    $mysqli = new mysqli($server, $dbusername, $dbpassword,
"greenhouse");
    if($mysqli->connect_error)
        die('Connect Error (' . mysqli_connect_errno() . ') ' .
mysqli_connect_error());

    // Execute SQL statement travaw apo ti vasi to time stamp kai to
spaw se year month day hour minute kai second gia na valw sto pinaka
tin timestamp opws to thelei to javascript+ oles tis metriseis
(teleytaies 40)
    $result = $mysqli->query("SELECT YEAR(event) AS year, MONTH(event)
AS month, DAY(event) AS day, HOUR(event) AS hour, MINUTE(event) AS
minute, SECOND(event) AS sec, temperature, humidity, fan, fire, flood,
pump, shades, safety_door, water_tank FROM data ORDER BY event DESC
LIMIT 5");

    if ($result->num_rows > 0) {

        // output data of each row
        while($row = $result->fetch_assoc()) {
            echo "data.addRow([new
Date(".$row["year"].",".$row["month"]."-
1,".$row["day"].",".$row["hour"].",".$row["minute"].",".$row["sec"].")
',
".$row["temperature"].",".$row["humidity"].",".$row["fan"].",".$row["f
ire"].",".$row["flood"].",".$row["pump"].",".$row["shades"].",".$row["
safety_door"].",".$row["water_tank"]."]);
\n";
        }
    }

```

```

    }
    else {
    }
?>
    var temperature_options = {
      chart: {
        title: 'Temperature vs Time',
        subtitle: 'Measured in degrees Celsius'
      },
      hAxis: { format: 'HH:mm:ss a' },
      width: 600,
      height: 300
    };

    var humidity_options = {
      chart: {
        title: 'Humidity vs Time',
        subtitle: 'Measured in %'
      },
      hAxis: { format: 'HH:mm:ss a' },
      width: 600,
      height: 300
    };

    var fan_options = {
      chart: {
        title: 'Fan vs Time'
      },
      hAxis: { format: 'HH:mm:ss a' },
      width: 600,
      height: 300
    };

    var fire_options = {
      chart: {
        title: 'Fire vs Time'
      },
      hAxis: { format: 'HH:mm:ss a' },
      width: 600,
      height: 300
    };

    var flood_options = {
      chart: {
        title: 'Flood vs Time'
      },
      hAxis: { format: 'HH:mm:ss a' },
      width: 600,
      height: 300
    };

    var pump_options = {
      chart: {
        title: 'Pump vs Time'
      },
      hAxis: { format: 'HH:mm:ss a' },
      width: 600,
      height: 300
    };

    var shades_options = {

```

```

    chart: {
        title: 'Shades vs Time'
    },
    hAxis: { format: 'HH:mm:ss a' },
    width: 600,
    height: 300
};

    var safety_door_options = {
    chart: {
        title: 'Safety_door vs Time'
    },
    hAxis: { format: 'HH:mm:ss a' },
    width: 600,
    height: 300
};
    var water_tank_options = {
    chart: {
        title: 'Water_tank vs Time'
    },
    hAxis: { format: 'HH:mm:ss a' },
    width: 600,
    height: 300
};
    var all_options = {
    chart: {
        title: 'All measurements vs Time',
        subtitle: ' '
    },
    hAxis: { format: 'HH:mm:ss a' },
    width: 600,
    height: 400
};
    var temperature_chart = new
google.charts.Line(document.getElementById('temperature_div'));

    var temperature_view = new google.visualization.DataView(data);
    temperature_view.setColumns([0,1]);
    temperature_chart.draw(temperature_view,
google.charts.Line.convertOptions(temperature_options));
//-----
-----
    var humidity_chart = new
google.charts.Line(document.getElementById('humidity_div'));

    var humidity_view = new google.visualization.DataView(data);
    humidity_view.setColumns([0,2]);
    humidity_chart.draw(humidity_view,
google.charts.Line.convertOptions(humidity_options));
//-----
-----
    var fan_chart = new
google.charts.Line(document.getElementById('fan_div'));

    var fan_view = new google.visualization.DataView(data);
    fan_view.setColumns([0,3]);
    fan_chart.draw(fan_view,
google.charts.Line.convertOptions(fan_options));
//-----
-----

```

```

    var fire_chart = new
google.charts.Line(document.getElementById('fire_div'));

    var fire_view = new google.visualization.DataView(data);
    fire_view.setColumns([0,4]);
    fire_chart.draw(fire_view,
google.charts.Line.convertOptions(fire_options));
    //-----
-----

    var flood_chart = new
google.charts.Line(document.getElementById('flood_div'));

    var flood_view = new google.visualization.DataView(data);
    flood_view.setColumns([0,5]);
    flood_chart.draw(flood_view,
google.charts.Line.convertOptions(flood_options));
    //-----
-----

    var pump_chart = new
google.charts.Line(document.getElementById('pump_div'));

    var pump_view = new google.visualization.DataView(data);
    pump_view.setColumns([0,6]);
    pump_chart.draw(pump_view,
google.charts.Line.convertOptions(pump_options));

    //-----
-----

    var shades_chart = new
google.charts.Line(document.getElementById('shades_div'));

    var shades_view = new google.visualization.DataView(data);
    shades_view.setColumns([0,7]);
    shades_chart.draw(shades_view,
google.charts.Line.convertOptions(shades_options));
    //-----
-----

    var safety_door_chart = new
google.charts.Line(document.getElementById('safety_door_div'));

    var safety_door_view = new google.visualization.DataView(data);
    safety_door_view.setColumns([0,8]);
    safety_door_chart.draw(safety_door_view,
google.charts.Line.convertOptions(safety_door_options));
    //-----
-----

    var water_tank_chart = new
google.charts.Line(document.getElementById('water_tank_div'));

    var water_tank_view = new google.visualization.DataView(data);
    water_tank_view.setColumns([0,9]);
    water_tank_chart.draw(water_tank_view,
google.charts.Line.convertOptions(water_tank_options));
    //-----
-----

    var all_chart = new
google.charts.Line(document.getElementById('all_div'));
    all_chart.draw(data,
google.charts.Line.convertOptions(all_options));
}
</script>

```

```

</head>
<body>

    <div style="width:100%">

</div>

<table style="border-spacing: 15px">
  <tr>
    <td style="border: 1px solid green">
      <div id="temperature_div" style="width: 600px;
height: 300px"></div>
    </td>
  </tr>

  <tr>
    <td style="border: 1px solid green">
      <div id="humidity_div" style="width: 600px;
height: 300px"></div>
    </td>
  </tr>

  <tr>
    <td style="border: 1px solid green">
      <div id="fan_div" style="width: 600px; height:
300px"></div>
    </td>
  </tr>

  <tr>
    <td style="border: 1px solid green">
      <div id="fire_div" style="width: 600px; height:
300px"></div>
    </td>
  </tr>

  <tr>
    <td style="border: 1px solid green">
      <div id="flood_div" style="width: 600px; height:
300px"></div>
    </td>
  </tr>

  <tr>
    <td style="border: 1px solid green">
      <div id="pump_div" style="width: 600px; height:
300px"></div>
    </td>
  </tr>

  <tr>
    <td style="border: 1px solid green">
      <div id="shades_div" style="width: 600px; height:
300px"></div>
    </td>
  </tr>

  <tr>
    <td style="border: 1px solid green">
      <div id="safety_door_div" style="width: 600px;
height: 300px"></div>

```

```

        </td>
    </tr>
    <tr>
        <td style="border: 1px solid green">
            <div id="water_tank_div" style="width: 600px;
height: 300px"></div>
        </td>
    </tr>
    <tr>
        <td style="border: 1px solid green">
            <div id="all_div" style="width: 600px; height:
300px"></div>
        </td>
    </tr>
</tr>
</table>

</div>
</main>

<nav id="nav">
    <div class="innertube">
        <h3>SMART GREENHOUSE</h3>
        <ul>
            <li><a href="index.html">Homepage</a></li>

                <h3>ThingSpeak</h3>
                <ul>
                    <li><a
href="temperature.php">Temperature</a></li>
                    <li><a href="humidity.php">Humidity</a></li>
                    <li><a href="fan.php">Fan</a></li>
                    <li><a href="fire.php">Fire</a></li>
                    <li><a href="flood.php">Flood </a></li>
                    <li><a href="pump.php">Pump</a></li>
                    <li><a href="safety_door.php">Safety
door</a></li>
                    <li><a href="water_tank.php">Water
tank</a></li>
                </ul>

                <h3>Google charts</h3>
                <ul>
                    <li><a
href="statistics1.php">Statistics</a></li>
                    <li><a href="gauge.php">Gauge</a></li>
                    <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
                    <li><a href="monthly_statistics1.php">Monthly
Statistics</a></li>
                </ul>
                <h3>Database</h3>
                <ul>
                    <li><a href="display.php">Display</a></li>
                </ul>
                <h3>Ajax I/O</h3>
                <ul>
                    <li><a href="192.168.1.13">Control Led and
Relay</a></li>
                </ul>

```

```

        </div>
    </nav>

</div>

<footer id="footer">
    <div class="innertube">
    </p>
        <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
    <p>
    </div>
</footer>

</body>
</html>

```

## A.13 temperature.php

```

<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Smart Greenhouse</title>
        <style type="text/css">

            body {
                margin:0;
                padding:0;
                font-family: Sans-Serif;
                line-height: 1.5em;
                background: #2C6566;
            }

            #header {
                background: #1F3440;
                color: white;
                height: 100px;
            }

            #header h1 {

                margin: 0;
                padding-top: 15px;
            }

            main {
                padding-bottom: 10010px;
                margin-bottom: -10000px;
                float: left;
                width: 100%;
            }

            #nav {
                padding-bottom: 10010px;
                margin-bottom: -10000px;
                float: left;

```

```

        width: 230px;
        margin-left: -100%;
        background: #25bdb0;
    }

    #footer {
        clear: left;
        width: 100%;
        background:#B4E8CF ;
        text-align: center;
        padding: 4px 0;
    }

    #wrapper {
        overflow: hidden;
    }

    #content {
        margin-left: 230px; /* Same as 'nav' width */
    }

    .innertube {
        margin: 15px; /* Padding for content */
        margin-top: 0;
    }

    p {
        color: #000000;
    }

    nav ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
    }

    nav ul a {
        color: black;
        text-decoration: none;
    }
}

</style>

</head>

<body>

    <header id="header">
        <div class="innertube">
            <h1>Smart Greenhouse</h1>
        </div>
    </header>

    <div id="wrapper">

        <main>
            <div id="content">

                <div class="content">

```



```

        <br>
        <center> <strong>Διάγραμμα θερμοκρασίας συναρτήσει χρόνου
δυναμικής ανανέωσης ανά 20 δευτερόλεπτα μέσω του Thingspeak.
</strong><br></center>
    <p>

<center> <br><br> Στο παρακάτω διάγραμμα απεικονίζονται οι τιμές της
θερμοκρασίας στο "έξυπνο" θερμοκήπιο σε συνάρτηση με το
χρόνο.</p></center>

    <br> <center><iframe width="450" height="260" style="border: 1px
solid #cccccc;"
src="https://thingspeak.com/channels/2581223/charts/1?bgcolor=%23ffffff
f&color=%23d62020&dynamic=true&results=60&type=line&update=15"></ifram
e></center>

</div>
    </div>
</main>

<nav id="nav">
    <div class="innertube">
    <h3>SMART GREENHOUSE</h3>
    <ul>
    <li><a href="index.html">Homepage</a></li>

        <h3>ThingSpeak</h3>
        <ul>
            <li><a
href="temperature.php">Temperature</a></li>
            <li><a href="humidity.php">Humidity</a></li>
            <li><a href="fan.php">Fan</a></li>
            <li><a href="fire.php">Fire</a></li>
            <li><a href="flood.php">Flood </a></li>
            <li><a href="pump.php">Pump</a></li>
            <li><a href="safety_door.php">Safety
door</a></li>
            <li><a href="water_tank.php">Water
tank</a></li>
        </ul>

        <h3>Google charts</h3>
        <ul>
            <li><a
href="statistics1.php">Statistics</a></li>
            <li><a href="gauge.php">Gauge</a></li>
            <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
            <li><a href="monthly_statistics1.php">Monthly
Statistics</a></li>
        </ul>
        <h3>Database</h3>
        <ul>
            <li><a href="display.php">Display</a></li>
        </ul>
        <h3>Ajax I/O</h3>
        <ul>
            <li><a href="index1.htm">Control Led and
Relay</a></li>

```

```

        </ul>
    </div>
</nav>

</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
        </p>
    </div>
</footer>

</body>
</html>

```

## A.14 water\_tank.php

```

<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Smart Greenhouse</title>
        <style type="text/css">

            body {
                margin:0;
                padding:0;
                font-family: Sans-Serif;
                line-height: 1.5em;
                background: #2C6566;
            }

            #header {
                background: #1F3440;
                color: white;
                height: 100px;
            }

            #header h1 {

                margin: 0;
                padding-top: 15px;
            }

            main {
                padding-bottom: 10010px;
                margin-bottom: -10000px;
                float: left;
                width: 100%;
            }

            #nav {
                padding-bottom: 10010px;
                margin-bottom: -10000px;
            }

```

```

        float: left;
        width: 230px;
        margin-left: -100%;
        background: #25bdb0;
    }

    #footer {
        clear: left;
        width: 100%;
        background:#B4E8CF ;
        text-align: center;
        padding: 4px 0;
    }

    #wrapper {
        overflow: hidden;
    }

    #content {
        margin-left: 230px; /* Same as 'nav' width */
    }

    .innertube {
        margin: 15px; /* Padding for content */
        margin-top: 0;
    }

    p {
        color: #000000;
    }

    nav ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
    }

    nav ul a {
        color: black;
        text-decoration: none;
    }
</style>

```

```
</head>
```

```
<body>
```

```

<header id="header">
  <div class="innertube">
    <h1>Smart Greenhouse</h1>
  </div>
</header>

<div id="wrapper">

  <main>
    <div id="content">

```

```

<div class="content">
  <br>
  <center> <strong>Διάγραμμα ελέγχου κατάστασης νερού
δεξαμενής συναρτήσει χρόνου δυναμικής ανανέωσης ανα 20 δευτερόλεπτα
μέσω του Thingspeak. </strong><br></center>
  <p>
<center> <br><br> Στο παρακάτω διάγραμμα απεικονίζονται οι τιμές
της ποιότητας του νερού της δεξαμενής στο "έξυπνο" θερμοκήπιο σε
συνάρτηση με το χρόνο.</p></center>
<center><iframe width="450" height="260" style="border: 1px solid
#cccccc;"
src="https://thingspeak.com/channels/2581223/charts/8?bgcolor=%23ff
ffff&color=%23d62020&dynamic=true&results=60&type=line&update=15">
/iframe></center>
</div>

  </div>
</main>

  <nav id="nav">
    <div class="innertube">
      <h3>SMART GREENHOUSE</h3>
      <ul>
        <li><a href="index.html">Homepage</a></li>

          <h3>ThingSpeak</h3>
          <ul>
            <li><a
href="temperature.php">Temperature</a></li>
            <li><a
href="humidity.php">Humidity</a></li>
            <li><a href="fan.php">Fan</a></li>
            <li><a href="fire.php">Fire</a></li>
            <li><a href="flood.php">Flood </a></li>
            <li><a href="pump.php">Pump</a></li>
            <li><a href="safety_door.php">Safety
door</a></li>
            <li><a href="water_tank.php">Water
tank</a></li>
          </ul>

          <h3>Google charts</h3>
          <ul>
            <li><a
href="statistics1.php">Statistics</a></li>
            <li><a href="gauge.php">Gauge</a></li>
            <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
            <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
          </ul>
          <h3>Database</h3>
          <ul>
            <li><a href="display.php">Display</a></li>
          </ul>
          <h3>Ajax I/O</h3>
          <ul>
            <li><a href="index1.htm">Control Led and
Relay</a></li>
          </ul>

```

```

        </div>
    </nav>

</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
        </p>
    </div>
</footer>

</body>
</html>

```

## A.15 weekly\_statistics.php

```

<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Smart Greenhouse</title>
        <style type="text/css">

            body {
                margin:0;
                padding:0;
                font-family: Sans-Serif;
                line-height: 1.5em;
                background: #2C6566;
            }

            #header {
                background: #1F3440;
                color: white;
                height: 100px;
            }

            #header h1 {

                margin: 0;
                padding-top: 15px;
            }

            main {
                padding-bottom: 10010px;
                margin-bottom: -10000px;
                float: left;
                width: 100%;
            }

            #nav {
                padding-bottom: 10010px;
                margin-bottom: -10000px;
                float: left;

```

```

        width: 230px;
        margin-left: -100%;
        background: #25bdb0;
    }

    #footer {
        clear: left;
        width: 100%;
        background:#B4E8CF ;
        text-align: center;
        padding: 4px 0;
    }

    #wrapper {
        overflow: hidden;
    }

    #content {
        margin-left: 230px; /* Same as 'nav' width */
    }

    .innertube {
        margin: 15px; /* Padding for content */
        margin-top: 0;
    }

    p {
        color: #000000;
    }

    nav ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
    }

    nav ul a {
        color: black;
        text-decoration: none;
    }
</style>

```

```
</head>
```

```
<body>
```

```

<header id="header">
  <div class="innertube">
    <h1>Smart Greenhouse</h1>
  </div>
</header>

<div id="wrapper">

  <main>
    <div id="content">
    <div class="content">
    <meta http-equiv="refresh" content="86400" />

```

```
<br>
<center><strong>Εβδομαδιαία
Στατιστικά</strong></center><br>
<p>
```

<br><br>Στα παρακάτω διαγράμματα απεικονίζονται η ελάχιστη, μέγιστη και μέση τιμή των δεδομένων (θερμοκρασίας, υγρασίας, εξαερισμού, φωτιάς, πλημμύρας, αντλίας, πόρτας, δεξαμενής ) για την κάθε μέρα της τελευταίας εβδομάδας.

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

```
function drawChart () {

var data = new google.visualization.DataTable();
data.addColumn('datetime', 'event');
data.addColumn('number', 'Average temperature');
data.addColumn('number', 'Max temperature');
data.addColumn('number', 'Min temperature');
data.addColumn('number', 'Average humidity');
data.addColumn('number', 'Max humidity');
data.addColumn('number', 'Min humidity');
data.addColumn('number', 'Average fan');
data.addColumn('number', 'Max fan');
data.addColumn('number', 'Min fan');
data.addColumn('number', 'Average fire');
data.addColumn('number', 'Max fire');
data.addColumn('number', 'Min fire');
data.addColumn('number', 'Average flood');
data.addColumn('number', 'Max flood');
data.addColumn('number', 'Min flood');
data.addColumn('number', 'Average pump');
data.addColumn('number', 'Max pump');
data.addColumn('number', 'Min pump');
data.addColumn('number', 'Average shades');
data.addColumn('number', 'Max shades');
data.addColumn('number', 'Min shades');
data.addColumn('number', 'Average safety_door');
data.addColumn('number', 'Max safety_door');
data.addColumn('number', 'Min safety_door');
data.addColumn('number', 'Average water_tank ');
data.addColumn('number', 'Max water_tank');
data.addColumn('number', 'Min water_tank');
```

```
<?php
// Prepare variables for database connection

$username = "root"; // enter database username, I used
"arduino" in step 2.2
$password = ""; // enter database password, I used
"arduinotest" in step 2.2
$server = "localhost";

// Connect to your database
mysqli = new mysqli($server, $username, $password,
"greenhouse");
```

```

    if($mysqli->connect_error)
        die('Connect Error (' . mysqli_connect_errno() . ') ' .
mysqli_connect_error());

    // Execute SQL statement ta grouparw ana mina kai krataw
teleytaia 12 +min max +average
    $result = $mysqli->query("SELECT YEAR(event) AS year,
MONTH(event) AS month, DAY(event) AS day, AVG(temperature),
MAX(temperature), MIN(temperature), AVG(humidity), MAX(humidity),
MIN(humidity), AVG(fan), MAX(fan), MIN(fan), AVG(fire), MAX(fire),
MIN(fire), AVG(flood), MAX(flood), MIN(flood), AVG(pump),
MAX(pump), MIN(pump), AVG(shades), MAX(shades),
MIN(shades),AVG(safety_door), MAX(safety_door),
MIN(safety_door),AVG(water_tank), MAX(water_tank), MIN(water_tank)
from data group by DAY(event) ORDER BY event DESC limit 12");

    if ($result->num_rows > 0) {

        // output data of each row
        while($row = $result->fetch_assoc()) {
            echo "data.addRow([new Date(".$row["year"].",
".$row["month"]."-1, ".$row["day"]."),
".$row["AVG(temperature)"].", ".$row["MAX(temperature)"].",
".$row["MIN(temperature)"].", ".$row["AVG(humidity)"].",
".$row["MAX(humidity)"].", ".$row["MIN(humidity)"].",
".$row["AVG(fan)"].", ".$row["MAX(fan)"].", ".$row["MIN(fan)"].",
".$row["AVG(fire)"].", ".$row["MAX(fire)"].",
".$row["MIN(fire)"].", ".$row["AVG(flood)"].",
".$row["MAX(flood)"].", ".$row["MIN(flood)"].",
".$row["AVG(pump)"].", ".$row["MAX(pump)"].",
".$row["MIN(pump)"].", ".$row["AVG(shades)"].",
".$row["MAX(shades)"].", ".$row["MIN(shades)"].",
".$row["AVG(safety_door)"].", ".$row["MAX(safety_door)"].",
".$row["MIN(safety_door)"].", ".$row["AVG(water_tank)"].",
".$row["MAX(water_tank)"].", ".$row["MIN(water_tank)"].");\n";
        }
    }
    else {
?>

    var temperature_options = {
        title: 'temperature (C) Last Week',
        subtitle: 'measured in C',
        hAxis: { format: 'MMM dd, yyyy' },
        seriesType: 'bars',
        width: 600,
        height: 300
    };

    var humidity_options = {
        title: 'humidity (%) Last Week',
        subtitle: 'measured in %',
        hAxis: { format: 'MMM dd, yyyy' },
        seriesType: 'bars',
        width: 600,
        height: 300
    };

    var fan_options = {

```



```

    title: 'Fan Last Week',
    subtitle: '',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
  };

  var fire_options = {
    title: 'Fire Last Week',
    subtitle: '',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
  };

  var flood_options = {
    title: 'Flood Last Week',
    subtitle: '',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
  };

  var pump_options = {
    title: 'pump Last Week',
    subtitle: '',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
  };

  var shades_options = {
    title: 'Shades Last Week',
    subtitle: '',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
  };

  var safety_door_options = {
    title: 'Safety_door Last Week ',
    subtitle: ' ',
    hAxis: { format: ' ' },
    seriesType: 'bars',
    width: 600,
    height: 300
  };

  var water_tank_options = {
    title: 'Water_tank Last Week ',
    subtitle: ' ',
    hAxis: { format: ' ' },
    seriesType: 'bars',
    width: 600,
    height: 300
  };
};

```

```

var temperature_chart = new
google.visualization.ComboChart (document.getElementById('temperatur
e_div'));

    var temperature_view = new
google.visualization.DataView (data);
    temperature_view.setColumns ([0,1,2,3]);
    temperature_chart.draw (temperature_view,
temperature_options);

//-----
var humidity_chart = new
google.visualization.ComboChart (document.getElementById('humidity_d
iv'));

    var humidity_view = new google.visualization.DataView (data);
    humidity_view.setColumns ([0,4,5,6]);
    humidity_chart.draw (humidity_view, humidity_options);

//-----
var fan_chart = new
google.visualization.ComboChart (document.getElementById('fan_div'))
;

    var fan_view = new google.visualization.DataView (data);
    fan_view.setColumns ([0,7,8,9]);
    fan_chart.draw (fan_view, fan_options);

//-----
var fire_chart = new
google.visualization.ComboChart (document.getElementById('fire_div')
);

    var fire_view = new google.visualization.DataView (data);
    fire_view.setColumns ([0,10,11,12]);
    fire_chart.draw (fire_view, fire_options);

//-----
var flood_chart = new
google.visualization.ComboChart (document.getElementById('flood_div'
));

    var flood_view = new google.visualization.DataView (data);
    flood_view.setColumns ([0,13,14,15]);
    flood_chart.draw (flood_view, flood_options);

//-----
var pump_chart = new
google.visualization.ComboChart (document.getElementById('pump_div'
));

    var pump_view = new google.visualization.DataView (data);
    pump_view.setColumns ([0,16,17,18]);
    pump_chart.draw (pump_view, pump_options);

//-----

```

```

    var shades_chart = new
google.visualization.ComboChart(document.getElementById('shades_div
'));

    var shades_view = new google.visualization.DataView(data);
shades_view.setColumns([0,19,20,21]);
shades_chart.draw(shades_view,shades_options);
//-----
-----

    var safety_door_chart = new
google.visualization.ComboChart(document.getElementById('safety_doo
r_div'));

    var safety_door_view = new
google.visualization.DataView(data);
safety_door_view.setColumns([0,22,23,24]);
safety_door_chart.draw(safety_door_view,safety_door_options);
//-----
-----

    var water_tank_chart = new
google.visualization.ComboChart(document.getElementById('water_tank
_div'));

    var water_tank_view = new
google.visualization.DataView(data);
water_tank_view.setColumns([0,25,26,27]);
water_tank_chart.draw(water_tank_view,water_tank_options);

//-----
-----
}
</script>
</head>
<body>
    <div style="width:100%">

    </div>

    <table style="border-spacing: 15px">
    <tr>
        <td style="border: 1px solid green">
            <div id="temperature_div" style="width: 600px;
height: 300px"></div>
        </td>
        <tr>
            <td style="border: 1px solid green">
                <div id="humidity_div" style="width: 600px;
height: 300px"></div>
            </td>

        </tr>
        <tr>
            <td style="border: 1px solid green">
                <div id="fan_div" style="width: 600px; height:
300px"></div>
            </td>
            <tr>
                <td style="border: 1px solid green">

```

```

        <div id="fire_div" style="width: 600px; height:
300px"></div>
        </td>
    </tr>
    <tr>
        <td style="border: 1px solid green">
            <div id="flood_div" style="width: 600px;
height: 300px"></div>
        </td>
    </tr>
    <tr>
        <td style="border: 1px solid green">
            <div id="pump_div" style="width: 600px; height:
300px"></div>
        </td>
    </tr>
    <tr>
        <td style="border: 1px solid green">
            <div id="shades_div" style="width: 600px;
height: 300px"></div>
        </td>
    </tr>
    <tr>
        <td style="border: 1px solid green">
            <div id="safety_door_div" style="width: 600px;
height: 300px"></div>
        </td>
    </tr>
    <tr>
        <td style="border: 1px solid green">
            <div id="water_tank_div" style="width: 600px;
height: 300px"></div>
        </td>
    </tr>
</table>
</div>
</div>
</main>

<nav id="nav">
    <div class="innertube">
        <h3>SMART GREENHOUSE</h3>
        <ul>
            <li><a href="index.html">Homepage</a></li>

            <h3>ThingSpeak</h3>
            <ul>
                <li><a
href="temperature.php">Temperature</a></li>
                <li><a
href="humidity.php">Humidity</a></li>
                <li><a href="fan.php">Fan</a></li>
                <li><a href="fire.php">Fire</a></li>
                <li><a href="flood.php">Flood </a></li>
                <li><a href="pump.php">Pump</a></li>
                <li><a href="safety_door.php">Safety
door</a></li>

                <li><a href="water_tank.php">Water
tank</a></li>
            </ul>
        </ul>
    </div>
</nav>

```

```

        <h3>Google charts</h3>
        <ul>
            <li><a
href="statistics1.php">Statistics</a></li>
            <li><a href="gauge.php">Gauge</a></li>
            <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
            <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
        </ul>
        <h3>Database</h3>
        <ul>
            <li><a href="display.php">Display</a></li>
        </ul>
        <h3>Ajax I/O</h3>
        <ul>
            <li><a href="index1.htm">Control Led and
Relay</a></li>
        </ul>
    </div>
</nav>

</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
        </p>
    </div>
</footer>

</body>
</html>

```

## A.16 index.html

```

<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Smart Greenhouse</title>
        <style type="text/css">

            body {
                margin:0;
                padding:0;
                font-family: Sans-Serif;
                line-height: 1.5em;
                background: #2C6566;
            }

            #header {
                background: #1F3440;
                color: white;
                height: 100px;
            }

```

```

#header h1 {
    margin: 0;
    padding-top: 15px;
}

main {
    padding-bottom: 10010px;
    margin-bottom: -10000px;
    float: left;
    width: 100%;
}

#nav {
    padding-bottom: 10010px;
    margin-bottom: -10000px;
    float: left;
    width: 230px;
    margin-left: -100%;
    background: #25bdb0;
}

#footer {
    clear: left;
    width: 100%;
    background:#B4E8CF ;
    text-align: center;
    padding: 4px 0;
}

#wrapper {
    overflow: hidden;
}

#content {
    margin-left: 230px; /* Same as 'nav' width */
}

.innertube {
    margin: 15px; /* Padding for content */
    margin-top: 0;
}

p {
    color: #000000;
}

nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}

nav ul a {
    color: black;
    text-decoration: none;
}

```

</style>

```

</head>

<body>

    <header id="header">
        <div class="innertube">
            <h1>Smart Greenhouse</h1>
        </div>
    </header>

    <div id="wrapper">

        <main>
            <div id="content">

                <div class="content">
                    <center><p><br> <br></p></center>

                    <p>

                        <br><br><br> Στη συγκεκριμένη ιστοσελίδα παρουσιάζεται μία
προσέγγιση ενός "έξυπνου" θερμοκηπίου. Αυτό επιτυγχάνεται με την
ενσωμάτωση
                            διάφορων αισθητήρων και συσκευών δράσης ούτως ώστε να
αυτοματοποιηθεί ένα μεγάλο πλήθος εργασιών που απαιτούνται για τη
                            συντήρηση και εύρυθμη λειτουργία του θερμοκηπίου. Στο
μενού της ιστοσελίδας υπάρχουν επιλογές για την παρακολούθηση
                            ορισμένων
                                απαραίτητων παραμέτρων τόσο σε μορφή γραφήματος, όσο
και σε μορφή πίνακα, ενώ παρέχονται επίσης κάποια στατιστικά.
                            Τέλος,
                                δίνεται η δυνατότητα ελέγχου των συσκευών δράσης μέσω
της ιστοσελίδας.
                                <br>
                                <center><p><br>Η υλοποίηση της μακέτας και η σύνδεση των
αισθητήρων έγινε στο πλαίσιο της διπλωματικής εργασίας του Σωτηρίου
Δριμή με επιβλέποντες καθηγητές τον Δρ.Τοπάλη Ευάγγελο και τον Δρ.
Λουκά Χαδέλλη.<br></p></center>

                                <center><p><br> <br></p></center>
                                <center><p><br>Η μακέτα προσομοίωσης του έξυπνου
θερμοκηπίου<br></p></center>

                            <p><br>
                                </p>

                            </div>
                        </div>
                    </main>

                    <nav id="nav">

```

```

<div class="innertube">
<h3>SMART GREENHOUSE</h3>
<ul>
<li><a href="index.html">Homepage</a></li>

    <h3>ThingSpeak</h3>
    <ul>
        <li><a
href="temperature.php">Temperature</a></li>
        <li><a
href="humidity.php">Humidity</a></li>
        <li><a href="fan.php">Fan</a></li>
        <li><a href="fire.php">Fire</a></li>
        <li><a href="flood.php">Flood </a></li>
        <li><a href="pump.php">Pump</a></li>
        <li><a href="safety_door.php">Safety
door</a></li>
        <li><a href="water_tank.php">Water
tank</a></li>
    </ul>

    <h3>Google charts</h3>
    <ul>
        <li><a
href="statistics1.php">Statistics</a></li>
        <li><a href="gauge.php">Gauge</a></li>
        <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
        <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
    </ul>
    <h3>Database</h3>
    <ul>
        <li><a href="display.php">Display</a></li>
    </ul>
    <h3>Ajax I/O</h3>
    <ul>
        <li><a href="index1.html">Control Led and
Relay</a></li>
    </ul>
</div>
</nav>

</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
        </p>
    </div>
</footer>

</body>
</html>

```

## A.17 index1.htm



```

<!DOCTYPE html>
<html>
  <head>
    <title>Smart Greenhouse&mdash;Footer</title>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <style type="text/css">
body { font-family: "Trebuchet MS", Courier;
background-color:#2C6566;
}
</style>
    <script>
strLED1 = "";
strLED2 = "";
strLED3 = "";
strLED4 = "";
var LED3_state = 0;
var LED4_state = 0;
function GetArduinoIO()
{
  nocache = "&nocache=" + Math.random() * 1000000;
var request = new XMLHttpRequest();
request.onreadystatechange = function()
{
  if (this.readyState == 4) {
    if (this.status == 200) {
      if (this.responseXML != null) {
        // XML file received - contains analog
values, switch values and LED states
var count;
// get analog inputs
var num_an =
this.responseXML.getElementsByTagName('analog').length;
for (count = 0; count < num_an;
count++) {
document.getElementsByClassName("analog")[count].innerHTML =
this.responseXML.getElementsByTagName('analog')[count].childNodes[0]
].nodeValue;
}
// get switch inputs
var num_an =
this.responseXML.getElementsByTagName('switch').length;
for (count = 0; count < num_an;
count++) {
document.getElementsByClassName("switches")[count].innerHTML =
this.responseXML.getElementsByTagName('switch')[count].childNodes[0]
].nodeValue;
}
// LED 1
if
(this.responseXML.getElementsByTagName('LED')[0].childNodes[0].node
Value === "checked") {
document.LED_form.LED1.checked =
true;
}
else {

```

```

document.LED_form.LED1.checked =
false;
    }
    // LED 2
    if
(this.responseXML.getElementsByTagName('LED')[1].childNodes[0].node
Value === "checked") {
        document.LED_form.LED2.checked =
true;
    }
    else {
        document.LED_form.LED2.checked =
false;
    }
    // LED 3
    if
(this.responseXML.getElementsByTagName('LED')[2].childNodes[0].node
Value === "on") {
document.getElementById("LED3").innerHTML = "lights ON (D8)";
        LED3_state = 1;
    }
    else {
document.getElementById("LED3").innerHTML = "lights OFF (D8)";
        LED3_state = 0;
    }
    // LED 4
    if
(this.responseXML.getElementsByTagName('LED')[3].childNodes[0].node
Value === "on") {
document.getElementById("LED4").innerHTML = "boiler ON (D9)";
        LED4_state = 1;
    }
    else {
document.getElementById("LED4").innerHTML = "boiler OFF (D9)";
        LED4_state = 0;
    }
    }
    }
}
// send HTTP GET request with LEDs to switch on/off if
any
request.open("GET", "ajax_inputs" + strLED1 + strLED2 +
strLED3 + strLED4 + nocache, true);
request.send(null);
setTimeout('GetArduinoIO()', 1000);
strLED1 = "";
strLED2 = "";
strLED3 = "";
strLED4 = "";
}
// service LEDs when checkbox checked/unchecked
function GetCheck()
{
    if (LED_form.LED1.checked) {
        strLED1 = "&LED1=1";
    }
}

```

```

        else {
            strLED1 = "&LED1=0";
        }
        if (LED_form.LED2.checked) {
            strLED2 = "&LED2=1";
        }
        else {
            strLED2 = "&LED2=0";
        }
    }
    function GetButton1 ()
    {
        if (LED3_state === 1) {
            LED3_state = 0;
            strLED3 = "&LED3=0";
        }
        else {
            LED3_state = 1;
            strLED3 = "&LED3=1";
        }
    }
    function GetButton2 ()
    {
        if (LED4_state === 1) {
            LED4_state = 0;
            strLED4 = "&LED4=0";
        }
        else {
            LED4_state = 1;
            strLED4 = "&LED4=1";
        }
    }
}
</script>
<style>
body
  .IO_box {
    float: left;
    margin: 0 20px 20px 0;
    border: 1px solid rgb(0, 0, 0);
    padding: 0 5px 0 5px;
    width: 120px;
  }
  h1 {
    font-size: 120%;
    color: rgb(255, 255, 255);
    margin: 0 0 10px 0;
  }
  h2 {
    font-size: 85%;
    color: #000000;
    margin: 5px 0 5px 0;
  }
  p, form, button {
    font-size: 80%;
    color: #000000;
  }
  .small_text {
    font-size: 70%;
    color: #ffffff;
  }
  {

```

```

        margin:0;
        padding:0;
        font-family: Sans-Serif;
        line-height: 1.5em;
        background: #2C6566;
    }

    #header {
        background: #1F3440;
        color: white;
        height: 100px;
    }

    #header h1 {

        margin: 0;
        padding-top: 15px;
    }

    main {
        padding-bottom: 10010px;
        margin-bottom: -10000px;
        float: left;
        width: 100%;
    }

    #nav {
        padding-bottom: 10010px;
        margin-bottom: -10000px;
        float: left;
        width: 230px;
        margin-left: -100%;
        background: #25bdb0;
    }

    #footer {
        clear: left;
        width: 100%;
        background:#B4E8CF ;
        text-align: center;
        padding: 4px 0;
    }

    #wrapper {
        overflow: hidden;
    }

    #content {
        margin-left: 230px; /* Same as 'nav' width */
    }

    .innertube {
        margin: 15px; /* Padding for content */
        margin-top: 0;
    }

    p {
        color: #000000;
    }

```

```

        nav ul {
            list-style-type: none;
            margin: 0;
            padding: 0;
        }

        nav ul a {
            color: black;
            text-decoration: none;
        }
    </style>
</head>
<header id="header">
    <div class="innertube">
        <h1>Smart Greenhouse</h1>
    </div>
</header>

<div id="wrapper">

    <main>
        <div id="content">

            <div class="content">
<body onload="GetArduinoIO()" >
                <h1>Sensors Inputs/ Control Devices Remotely</h1>
                <div class="IO_box">
                    <h2>Analog Inputs</h2>
                    <p>gas sensor: <span class="analog">...</span></p>
                    <p>temperature sensor: <span
class="analog">...</span></p>
                    <p>blinds photoresistor: <span
class="analog">...</span></p>
                    <p>soil moisture sensor: <span
class="analog">...</span></p>
                    <p>watering photoresistor: <span
class="analog">...</span></p>

                </div>
                <div class="IO_box">
                    <h2>Switch Inputs</h2>
                    <p>AC button (D2): <span
class="switches">...</span></p>
                    <p>IR receiver (D3): <span
class="switches">...</span></p>
                    <p>FAN (D5): <span class="switches">...</span></p>
                </div>
                <div class="IO_box">
                    <h2>Control Devices Using Checkboxes</h2>
                    <form id="check_LEDs" name="LED_form">
                        <input type="checkbox" name="LED1" value="0"
onclick="GetCheck()" />lights1<br /><br />
                        <input type="checkbox" name="LED2" value="0"
onclick="GetCheck()" />lights2<br /><br />
                    </form>
                </div>
                <div class="IO_box">
                    <h2>Control Devices Using Buttons</h2>
                    <button type="button" id="LED3"
onclick="GetButton1()">lights1 OFF (D8)</button><br /><br />
            </div>
        </div>
    </main>
</div>

```

```

        <button type="button" id="LED4"
onclick="GetButton2()">lights2 OFF (D9)</button><br /><br />
        <p class="small_text">D10 to D13 used by Ethernet
shield</p>
    </div>
</div>

</div>
</main>

<nav id="nav">
    <div class="innertube">
        <h3>SMART GREENHOUSE</h3>
        <ul>
            <li><a href="index.html">Homepage</a></li>

            <h3>ThingSpeak</h3>
            <ul>
                <li><a
href="temperature.php">Temperature</a></li>
                <li><a
href="humidity.php">Humidity</a></li>
                <li><a href="fan.php">Fan</a></li>
                <li><a href="fire.php">Fire</a></li>
                <li><a href="flood.php">Flood </a></li>
                <li><a href="antlia.php">Pump</a></li>
                <li><a href="porta.php">Safety
door</a></li>
                <li><a href="water_tank.php">Water
tank</a></li>
            </ul>

            <h3>Google charts</h3>
            <ul>
                <li><a
href="statistics1.php">Statistics</a></li>
                <li><a href="gauge.php">Gauge</a></li>
                <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
                <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
            </ul>
            <h3>Database</h3>
            <ul>
                <li><a href="display.php">Display</a></li>
            </ul>
            <h3>Ajax I/O</h3>
            <ul>
                <li><a href="index1.htm">Control Led and
Relay</a></li>
            </ul>
        </div>
    </nav>

</div>

<footer id="footer">
    <div class="innertube">
        <p>

```

```

                <span>2024 &copy; </span><a href="#" >Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
            </p>
        </div>
    </footer>

</body>
</html>

```

## A.18 histogram.php

```

<?php
include "connection.php";
function debug_to_console($data) {
    $output = $data;
    if (is_array($output))
        $output = implode(',', $output);

    echo "<script>console.log('Debug Objects: " . $output . "'
);</script>";
}

if (isset($_POST['selection'])) {
    $selection = mysqli_real_escape_string($con, $_POST['selection']);

    $start = mysqli_real_escape_string($con, $_POST['start']);

    $end = mysqli_real_escape_string($con, $_POST['end']);
} else {
    $selection = "temperature";
    $start = date("Y-m-d");
    $end = date("Y-m-d");
}
$text = "Θερμοκρασία";
$measurement = "Θερμοκρασία σε °C";
$start = "2024-06-24";
$end = "2024-07-24";

if ($selection == "temperature") {
    $text = "Θερμοκρασία";
    $measurement = "Θερμοκρασία σε °C";
} else if ($selection == "humidity") {
    $text = "Υγρασία";
    $measurement = "Ποσοστό Υγρασίας %";
} else if ($selection == "fan") {
    $text = "Ανεμιστήρας";
    $measurement = "Κατάσταση Ανεμιστήρα";
} else if ($selection == "fire") {
    $text = "Ανιχνευτής φωτιάς";
    $measurement = "Κατάσταση ανιχνευτή φωτιάς";
} else if ($selection == "pump") {
    $text = "Αντλία";
    $measurement = "Κατάσταση Αντλίας";
} else if ($selection == "shades") {
    $text = "Σκίαστρα";
    $measurement = "Κατάσταση σκίαστρων";
} else if ($selection == "water_tank") {
    $text = "Δεξαμενή νερού";
    $measurement = "Κατάσταση Δεξαμενής";
}

```

```

$result      =      mysqli_query($con,      "SELECT      MAX($selection),
MIN($selection),AVG($selection), event
FROM data
WHERE DATE(event)>='$start' AND DATE(event)<='$end'
group by DATE(event)
order by DATE(event) ASC");

?>
<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
  <title>Smart Greenhouse</title>
  <style type="text/css">
    body {
      margin: 0;
      padding: 0;
      font-family: Sans-Serif;
      line-height: 1.5em;
      background: #2C6566;
    }

    #header {
      background: #1F3440;
      color: white;
      height: 100px;
    }

    #header h1 {

      margin: 0;
      padding-top: 15px;

    }

    main {
      padding-bottom: 10010px;
      margin-bottom: -10000px;
      float: left;
      width: 100%;
    }

    #nav {
      padding-bottom: 10010px;
      margin-bottom: -10000px;
      float: left;
      width: 230px;
      margin-left: -100%;
      background: #25bdb0;
    }

    #footer {
      clear: left;
      width: 100%;
      background: #B4E8CF;
      text-align: center;
      padding: 4px 0;
    }
  </style>

```



```

#wrapper {
  overflow: hidden;
}

#content {
  margin-left: 230px;
  /* Same as 'nav' width */
}

.innertube {
  margin: 15px;
  /* Padding for content */
  margin-top: 0;
}

p {
  color: #000000;
}

nav ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

nav ul a {
  color: black;
  text-decoration: none;
}
</style>
</head>

<body>
<header id="header">
  <div class="innertube">
    <h1>Smart Greenhouse</h1>
  </div>
</header>
<div id="wrapper">
  <main>
    <div id="content">
      <div class="content">
        <meta http-equiv="refresh" content="86400" />
        <br>
        <center><strong>Histograms</strong></center><br>
        <p>
          <br><br> Στην σελίδα αυτή ο χρήστης μπορεί να ανακτήσει
          δεδομένα από τους αισθητήρες για συγκεκριμένη
          χρονική περίοδο και να τα παρουσιάσει σε μορφή
          ιστογράμματος. Τα δεδομένα που φαίνονται είναι η Ελάχιστη, Μέγιστη
          και Μέση τιμή.
        </p>
        <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([

```

```

        ['Ημέρα', 'Ελάχιστη', 'Μέγιστη', 'Μέσος όρος'],
        <?php
            while ($row = mysqli_fetch_array($result)) {
                echo " [ '" . $row["event"] . "' , " .
                $row["MIN($selection)"] . " , " . $row["MAX($selection)"] . " , " .
                $row["AVG($selection)"] . " ] , " ;
            }
        ?>
    ]);

    var options = {
        bars: 'vertical'
    };

    var chart = new
google.charts.Bar(document.getElementById('Bar_Stats'));
    console.log(data);
    chart.draw(data, options);
}
</script>
</body>

<body>
    <form name="bar_chart" method="post">
        <h2>Ημερομηνία Αρχής:<h2>
        <input style="font-size: 20px;" type="date"
id="start" name="start"
        value="<?php echo date("Y-m-d"); ?>"
        <h2>Ημερομηνία Τέλους:<h2>
        <input style="font-size: 20px;" type="date"
id="end" name="end"
        value="<?php echo date("Y-m-d"); ?>"
        <h1>Επιλογή Μέτρησης</h1>
        <input type="radio" name="selection"
value="temperature">
        <label for="temperature">Θερμοκρασία</label><br>
        <input type="radio" name="selection"
value="humidity">
        <label for="humidity">Υγρασία</label><br>
        <input type="radio" name="selection"
value="fan">
        <label for="fan">Κατάσταση
Ανεμιστήρα</label><br>
        <input type="radio" name="selection"
value="fire">
        <label for="fire">Κατάσταση ανιχνευτή
φωτιάς</label><br>
        <input type="radio" name="selection"
value="pump">
        <label for="pump">Κατάσταση Αντλίας</label><br>
        <input type="radio" name="selection"
value="shades">
        <label for="shades">Κατάσταση
σκιάστρων</label><br>
        <input type="radio" name="selection"
value="water_tank">
        <label for="water_tank">Κατάσταση
Δεξαμενής</label><br>
        <p><button class="button" type="submit"
value="Submit">Υποβολή </button> </p>
    </form>
<article style="width:50%;">

```

```

        <div><?php echo "<h3 style='color:red; '>" . $text . "
από "
        . date('d/m/Y', strtotime($start)) . " έως " .
        date('d/m/Y', strtotime($end)) . "</h3>"; ?></div>
        <div id="Bar_Stats" style="width: 100%; height:
80%;"></div>
    </article>
</div>
</div>
</div>
<nav id="nav">
    <div class="innertube">
        <h3>SMART GREENHOUSE</h3>
        <ul>
            <li><a href="index.html">Homepage</a></li>

            <h3>ThingSpeak</h3>
            <ul>
                <li><a href="temperature.php">Temperature</a></li>
                <li><a href="humidity.php">Humidity</a></li>
                <li><a href="fan.php">Fan</a></li>
                <li><a href="fire.php">Fire</a></li>
                <li><a href="flood.php">Flood </a></li>
                <li><a href="pump.php">Pump</a></li>
                <li><a href="safety_door.php">Safety door</a></li>
                <li><a href="water_tank.php">Water tank</a></li>
            </ul>
            <h3>Google charts</h3>
            <ul>
                <li><a href="statistics1.php">Statistics</a></li>
                <li><a href="gauge.php">Gauge</a></li>
                <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
                <li><a href="monthly_statistics1.php">Monthly
Statistics</a></li>
                <li><a href="time_graphs.php">Time Graphs</a></li>
                <li><a href="histogramms.php">Histograms</a></li>
                <li><a href="alarms_live.php">Alarms Live</a></li>
            </ul>
            <h3>Database</h3>
            <ul>
                <li><a href="display.php">Display</a></li>
            </ul>
            <h3>Ajax I/O</h3>
            <ul>
                <li><a href="index1.htm">Control Led and Relay</a></li>
            </ul>
        </div>
    </nav>

</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#">Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#">Πάτρα</a>
        </p>
    </div>
</footer>

```

```
</body>
```

```
</html>
```

## A.19 alarms\_live.htm

```
<!DOCTYPE html>
<html>

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Smart Greenhouse</title>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.0/jquery.min.js"></script>
  <script src="https://www.gstatic.com/charts/loader.js"></script>
  <style type="text/css">
    body {
      margin: 0;
      padding: 0;
      font-family: Sans-Serif;
      line-height: 1.5em;
      background: #2C6566;
    }

    #header {
      background: #1F3440;
      color: white;
      height: 100px;
    }

    #header h1 {
      margin: 0;
      padding-top: 15px;
    }

    main {
      padding-bottom: 10010px;
      margin-bottom: -10000px;
      float: left;
      width: 100%;
    }

    #nav {
      padding-bottom: 10010px;
      margin-bottom: -10000px;
      float: left;
      width: 230px;
      margin-left: -100%;
      background: #25bdb0;
    }

    #footer {
      clear: left;
      width: 100%;
      background: #B4E8CF;
      text-align: center;
      padding: 4px 0;
```

```

}

#wrapper {
  overflow: hidden;
}

#content {
  margin-left: 230px;
  /* Same as 'nav' width */
}

.innertube {
  margin: 15px;
  /* Padding for content */
  margin-top: 0;
}

p {
  color: #000000;
}

nav ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

nav ul a {
  color: black;
  text-decoration: none;
}
</style>
<script>
google.charts.load('current', {
  packages: ['gauge']
}).then(function () {
  var options_alarm = {
    width: 250, height: 250,
    redFrom: 0.5, redTo: 1,
    majorTicks: [0, 0],
    minorTicks: [],
    max: 1,
    min: 0
  };

  var data_chart_fire_alarm = new
google.visualization.Gauge(document.getElementById('chart_div1'));
  var data_chart_flood_alarm = new
google.visualization.Gauge(document.getElementById('chart_div2'));
  var data_chart_door_alarm = new
google.visualization.Gauge(document.getElementById('chart_div3'));
  drawChart();

  function drawChart() {
    var chart_fire_alarm =
google.visualization.arrayToDataTable([
      ['Label', 'Value'],

      ['', 1]
    ]);
    var chart_flood_alarm =
google.visualization.arrayToDataTable([

```

```

        ['Label', 'Value'],
        ['', 0]
    ]);
    var chart_door_alarm =
google.visualization.arrayToDataTable([
    ['Label', 'Value'],
    ['', 0]
]);
    data_chart_fire_alarm.draw(chart_fire_alarm, options_alarm);
    data_chart_flood_alarm.draw(chart_flood_alarm,
options_alarm);
    data_chart_door_alarm.draw(chart_door_alarm, options_alarm);
    $.ajax({
        url: 'Gauge_data.php',
        dataType: 'json',
        error: function (jqXHR, textStatus, errorThrown) {
            console.log(jqXHR);
            console.log(textStatus);
            console.log(errorThrown);
        }
    }).done(function (jsonData) {
        var data_chart_fire_alarm =
google.visualization.arrayToDataTable(1);
        chart_fire_alarm.draw(data_chart_fire_alarm,
options_alarm);
        var data_chart_flood_alarm =
google.visualization.arrayToDataTable(0);
        chart_flood_alarm.draw(data_chart_flood_alarm,
options_alarm);
        var data_chart_door_alarm =
google.visualization.arrayToDataTable(1);
        chart_door_alarm.draw(data_chart_door_alarm,
options_alarm);
        window.setTimeout(drawChart, 5000);
    });
}
});
</script>
</head>

<body>
<header id="header">
<div class="innertube">
<h1>Smart Greenhouse</h1>
</div>
</header>
<div id="wrapper">
<main>
<div id="content">
<div class="content">
<meta http-equiv="refresh" content="86400" />
<br>
<center><strong>Live Alarms</strong></center><br>
<p>
<br><br> Στην σελίδα αυτή ο χρήστης έχει πρόσβαση στα
alarms που διαθέτει το θερμοκήπιο. Μπορεί να ελέγξει την κατάσταση
κρίσιμων μετρήσεων από τους αισθητήρες.
</p>
<div style=" width:75%; text-align: center;">

```

```

        <p></p>
        <table style=" width:100%; display: flex; align-items:
center; justify-content: center;">
        <tr style="font-size: 25px; color:red;">
        <td>Ανίχνευση Φωτιάς</td>
        <td>Ανίχνευση Πλημμύρας</td>
        <td>Παραβίαση Θύρας</td>
        </tr>
        <tr>
        <td>
        <div id="chart_div1"></div>
        </td>
        <td>
        <div id="chart_div2"></div>
        </td>
        <td>
        <div id="chart_div3"></div>
        </td>
        <td>
        <div id="chart_div4"></div>
        </td>
        <td>
        <div id="chart_div5"></div>
        </td>
        <td>
        <div id="chart_div6"></div>
        </td>
        </tr>
        </table>
        </div>
        </div>
</main>

<nav id="nav">
<div class="innertube">
<h3>SMART GREENHOUSE</h3>
<ul>
<li><a href="index.html">Homepage</a></li>

<h3>ThingSpeak</h3>
<ul>
<li><a href="temperature.php">Temperature</a></li>
<li><a href="humidity.php">Humidity</a></li>
<li><a href="fan.php">Fan</a></li>
<li><a href="fire.php">Fire</a></li>
<li><a href="flood.php">Flood </a></li>
<li><a href="pump.php">Pump</a></li>
<li><a href="safety_door.php">Safety door</a></li>
<li><a href="water_tank.php">Water tank</a></li>
</ul>
<h3>Google charts</h3>
<ul>
<li><a href="statistics1.php">Statistics</a></li>
<li><a href="gauge.php">Gauge</a></li>
<li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
<li><a href="monthly_statistics1.php">Monthly
Statistics</a></li>
<li><a href="time_graphs.php">Time Graphs</a></li>
<li><a href="histogramms.php">Histograms</a></li>
<li><a href="alarms_live.php">Alarms Live</a></li>

```

```

        </ul>
        <h3>Database</h3>
        <ul>
            <li><a href="display.php">Display</a></li>
        </ul>
        <h3>Ajax I/O</h3>
        <ul>
            <li><a href="index1.htm">Control Led and Relay</a></li>
        </ul>
    </div>
</div>

<footer id="footer">
    <div class="innertube">
        <p>
            <span>2024 &copy; </span><a href="#">Πανεπιστήμιο
            Πελοποννήσου</a> | <a href="#">Πάτρα</a>
        </p>
    </div>
</footer>

</body>
</html>

```

## A.20 time\_graphs.php

```

<?php
include "connection.php";
if (isset($_POST['selection'])) {
    $selection = mysqli_real_escape_string($con,
    $_POST['selection']);

    $start = mysqli_real_escape_string($con, $_POST['start']);

    $end = mysqli_real_escape_string($con, $_POST['end']);
} else {
    $selection = "temperature_room";
    $start = date("Y-m-d");
    $end = date("Y-m-d");
}
$text = "Θερμοκρασία";
$measurement = "Θερμοκρασία σε °C";
$start = "2024-06-24";
$end = "2024-06-24";

if ($selection == "temperature") {
    $text = "Θερμοκρασία";
    $measurement = "Θερμοκρασία σε °C";
} else if ($selection == "humidity") {
    $text = "Υγρασία";
    $measurement = "Ποσοστό Υγρασίας %";
} else if ($selection == "fan") {
    $text = "Ανεμιστήρας";
    $measurement = "Κατάσταση Ανεμιστήρα";
} else if ($selection == "fire") {
    $text = "Ανιχνευτής φωτιάς";
    $measurement = "Κατάσταση ανιχνευτή φωτιάς";
} else if ($selection == "pump") {
    $text = "Αντλία";

```



```

    $measurement = "Κατάσταση Αντλίας";
} else if ($selection == "shades") {
    $text = "Σκίαστρα";
    $measurement = "Κατάσταση σκίαστρων";
} else if ($selection == "water_tank") {
    $text = "Δεξαμενή νερού";
    $measurement = "Κατάσταση Δεξαμενής";
}

$result = mysqli_query($con, "SELECT * FROM data WHERE
(DATE(event))>='$start' AND DATE(event)<='$end'");

?>
<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Smart Greenhouse</title>
    <style type="text/css">
        body {
            margin: 0;
            padding: 0;
            font-family: Sans-Serif;
            line-height: 1.5em;
            background: #2C6566;
        }

        #header {
            background: #1F3440;
            color: white;
            height: 100px;
        }

        #header h1 {

            margin: 0;
            padding-top: 15px;

        }

        main {
            padding-bottom: 10010px;
            margin-bottom: -10000px;
            float: left;
            width: 100%;
        }

        #nav {
            padding-bottom: 10010px;
            margin-bottom: -10000px;
            float: left;
            width: 230px;
            margin-left: -100%;
            background: #25bdb0;
        }

        #footer {

```

```

clear: left;
width: 100%;
background: #B4E8CF;
text-align: center;
padding: 4px 0;
}

#wrapper {
overflow: hidden;
}

#content {
margin-left: 230px;
/* Same as 'nav' width */
}

.innertube {
margin: 15px;
/* Padding for content */
margin-top: 0;
}

p {
color: #000000;
}

nav ul {
list-style-type: none;
margin: 0;
padding: 0;
}

nav ul a {
color: black;
text-decoration: none;
}
</style>
</head>

<body>
<header id="header">
<div class="innertube">
<h1>Smart Greenhouse</h1>
</div>
</header>
<div id="wrapper">
<main>
<div id="content">
<div class="content">
<meta http-equiv="refresh" content="86400" />
<br>
<center><strong>Time Graphs</strong></center><br>
<p>
<br><br> Στην σελίδα αυτή ο χρήστης μπορεί να ανακτήσει
δεδομένα από τους αισθητήρες για συγκεκριμένη χρονική περίοδο.
</p>
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
google.charts.load('current', { 'packages': ['corechart']
});

```

```

google.charts.setOnLoadCallback(drawChart);

function drawChart() {
    var data = google.visualization.arrayToDataTable([
        ['Χρονική Στιγμή', '<?php echo $text; ?>'],
        <?php
        while ($row = mysqli_fetch_assoc($result)) {
            echo
                "[" . $row['event'] . "," .
$row["$selection"] . "],";
        }
        ?>
    ]);
    var options = {
        title: '',
        hAxis: { title: '' },
        vAxis: { title: '<?php echo $measurement; ?>' },
    };
    var chart = new
        google.visualization.LineChart(document.getElementById('div_chart'));
    chart.draw(data, options);
}
</script>
</head>

<body>
<form name="bar_chart" method="post">
    <h2>Ημερομηνία Αρχής:<h2>
        <input style="font-size: 20px;" type="date"
id="start" name="start"
            value="<?php echo date("Y-m-d"); ?>">
    <h2>Ημερομηνία Τέλους:<h2>
        <input style="font-size: 20px;"
type="date" id="end" name="end"
            value="<?php echo date("Y-m-d");
?>">

        <h1>Επιλογή Μέτρησης</h1>
        <input type="radio" name="selection"
value="temperature">
            <label
for="temperature">Θερμοκρασία</label><br>
        <input type="radio" name="selection"
value="humidity">
            <label
for="humidity">Υγρασία</label><br>
        <input type="radio" name="selection"
value="fan">
            <label for="fan">Κατάσταση
Ανεμιστήρα</label><br>
        <input type="radio" name="selection"
value="fire">
            <label for="fire">Κατάσταση ανιχνευτή
φωτιάς</label><br>
        <input type="radio" name="selection"
value="pump">
            <label for="pump">Κατάσταση
Αντλίας</label><br>
        <input type="radio" name="selection"
value="shades">

```

```

<label for="shades">Κατάσταση
σκιάστρων</label><br>
<input type="radio" name="selection"
value="water_tank">
<label for="water_tank">Κατάσταση
Δεξαμενής</label><br>
<p><button class="button" type="submit"
value="Submit">Υποβολή </button> </p>
</div>
<article style="width:50%;">
<div><?php echo "<h3 style='color:red; '>" . $text . "
από "
. date('d/m/Y', strtotime($start)) . " έως " .
date('d/m/Y', strtotime($end)) . "</h3>"; ?></div>
<div id="div_chart" style="width: 100%; height:
80%;"></div>
</article>
</div>
</div>
</main>
<nav id="nav">
<div class="innertube">
<h3>SMART GREENHOUSE</h3>
<ul>
<li><a href="index.html">Homepage</a></li>
<h3>ThingSpeak</h3>
<ul>
<li><a href="temperature.php">Temperature</a></li>
<li><a href="humidity.php">Humidity</a></li>
<li><a href="fan.php">Fan</a></li>
<li><a href="fire.php">Fire</a></li>
<li><a href="flood.php">Flood </a></li>
<li><a href="pump.php">Pump</a></li>
<li><a href="safety_door.php">Safety door</a></li>
<li><a href="water_tank.php">Water tank</a></li>
</ul>
<h3>Google charts</h3>
<ul>
<li><a href="statistics1.php">Statistics</a></li>
<li><a href="gauge.php">Gauge</a></li>
<li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
<li><a href="monthly_statistics1.php">Monthly
Statistics</a></li>
<li><a href="time_graphs.php">Time Graphs</a></li>
<li><a href="histogramms.php">Histograms</a></li>
<li><a href="alarms_live.php">Alarms Live</a></li>
</ul>
<h3>Database</h3>
<ul>
<li><a href="display.php">Display</a></li>
</ul>
<h3>Ajax I/O</h3>
<ul>
<li><a href="index1.htm">Control Led and Relay</a></li>
</ul>
</div>
</nav>

```

```

</div>

<footer id="footer">
  <div class="innertube">
    <p>
      <span>2024 &copy; </span><a href="#">Πανεπιστήμιο
      Πελοποννήσου</a> | <a href="#">Πάτρα</a>
    </p>
  </div>
</footer>

</body>

</html>

```

## A.21 comparative1.php

```

<?php
include "connection.php"; // Προϋπόθεση η σύνδεση με την βάση
if (isset($_POST['selection'])) //Αν γίνει αίτημα
{
  $selected_measurements = $_POST['selection']; //Αποθήκευση πίνακα
  επιλογής μετρήσεων που θα προβληθούν
  $start = mysqli_real_escape_string($con, $_POST['start']);
  //Αποθήκευση ημερομηνίας αρχής
  $end = mysqli_real_escape_string($con, $_POST['end']);
  //Αποθήκευση ημερομηνίας τέλους
} else { //Αν δεν υπάρξει αίτημα τότε αρχικοποίηση στις παρακάτω
τιμές
  $selected_measurements = ["temperature"];
  $start = date("Y-m-d");
  $end = date("Y-m-d");
} // Ανάκτηση δεδομένων από την βάση, του διαστήματος και των
πεδίων που έχουν επιλεγεί
$sql_query = "SELECT event, ";
foreach ($selected_measurements as $measurement) {
  $sql_query .= "$measurement, ";
}
$sql_query = rtrim($sql_query, ", ");
$sql_query .= " FROM data WHERE (DATE(event)>='$start' AND
DATE(event)<='$end') ORDER BY event ASC";
$result = mysqli_query($con, $sql_query);
?>
<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
  <title>Smart Greenhouse</title>
  <style type="text/css">
    body {
      margin: 0;
      padding: 0;
      font-family: Sans-Serif;
      line-height: 1.5em;

```

```

    background: #2C6566;
}

#header {
    background: #1F3440;
    color: white;
    height: 100px;
}

#header h1 {

    margin: 0;
    padding-top: 15px;

}

main {
    padding-bottom: 10010px;
    margin-bottom: -10000px;
    float: left;
    width: 100%;
}

#nav {
    padding-bottom: 10010px;
    margin-bottom: -10000px;
    float: left;
    width: 230px;
    margin-left: -100%;
    background: #25bdb0;
}

#footer {
    clear: left;
    width: 100%;
    background: #B4E8CF;
    text-align: center;
    padding: 4px 0;
}

#wrapper {
    overflow: hidden;
}

#content {
    margin-left: 230px;
    /* Same as 'nav' width */
}

.innertube {
    margin: 15px;
    /* Padding for content */
    margin-top: 0;
}

p {
    color: #000000;
}

nav ul {
    list-style-type: none;

```

```

margin: 0;
padding: 0;
}

nav ul a {
color: black;
text-decoration: none;
}
</style>
</head>

<body>
<header id="header">
<div class="innertube">
<h1>Smart Greenhouse</h1>
</div>
</header>
<div id="wrapper">
<main>
<div id="content">
<div class="content">
<meta http-equiv="refresh" content="86400" />
<br>
<center><strong>Comparative Graphs</strong></center><br>
<p>
<center> <strong>Στην σελίδα αυτή ο χρήστης μπορεί να
συγκρίνει δεδομένα από τους αισθητήρες για συγκεκριμένα χρονικά
περίοδο. </strong><br></center>
</p>
<article style="width:20%; padding-left: 12px;"><!--Διαίρεση
ενότητας
σε δύο περιοχές-->
<!--Επιλογή διαστήματος και πεδίων μέτρηση-->
<form name="line_chart" method="post">
<h2>Ημερομηνία Αρχής:</h2>
<input style="font-size: 20px;" type="date" id="start"
name="start"
value="<?php echo date("Y-m-d"); ?>">

<h2>Ημερομηνία Τέλους:</h2>
<input style="font-size: 20px;" type="date" id="end" name="end"
value="<?php echo date("Y-m-d"); ?>">
<br><br>
<h2>Επιλογή Σύγκρισης</h2>
<input type="checkbox" id="temperature" name="selection[]"
value="temperature" checked>
<label for="temperature">Θερμοκρασία</label><br>
<input type="checkbox" id="humidity" name="selection[]"
value="humidity">
<label for="humidity">Υγρασία</label><br>
<input type="checkbox" id="fan" name="selection[]"
value="fan">
<label for="fan">Εξαερισμός</label><br>
<input type="checkbox" id="fire" name="selection[]"
value="fire">
<label for="fire">Φωτιά</label><br>
<p><button class="button" type="submit" value="Submit"> Υποβολή
</button> </p>
</form>
</article>
<!--Δημιουργία χρονοδιαγράμματος-->

```

```

<!--Προσθήκη βιβλιοθηκών -->
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
<!--Φόρτωση του API οπτικοποίησης και του πακέτου διαγράμματος και
ρύθμισης λειτουργίας επανάκλησης-->
google.charts.load('current', {
'packages': ['corechart'],
'mapsApiKey': ''
});
google.charts.setOnLoadCallback(comparativeChart);

function comparativeChart() { //Η συνάρτηση κατασκευής γραφήματος
var data = new google.visualization.DataTable();
data.addColumn('string', 'Event');
<?php // Στήλες του πίνακα δεδομένων του γραφήματος βάσει επιλογών
foreach ($selected_measurements as $measurement) {
$input_label = "";
switch($measurement) {
case "temperature":
$input_label = "Θερμοκρασία ";
break;
case "humidity":
$input_label = "Υγρασία";
break;
case "fan":
$input_label = "Εξαερισμός";
break;
case "fire":
$input_label = "Φωτιά";
break;
}
echo "data.addColumn('number', '$input_label');\n";
}
?>
data.addRow([ // Γραμμές του πίνακα δεδομένων του γραφήματος
<?php while($row = mysqli_fetch_array($result)) {
echo "[".$row['event'].", ";
foreach ($selected_measurements as $measurement) {
echo "{$row[$measurement]}, ";
}
echo "],\n";
} ?>
]);
var options = { //Μορφοποίηση εμφάνισης γραφήματος
title: '',
hAxis: {title: ''},
vAxis: {title: ''},
};
//Εμφάνιση του γραφήματος μέσα στο στοιχείο <div>
var chart = new
google.visualization.LineChart(document.getElementById('div_chart'
));
chart.draw(data, options);
}
</script>
<article style="width:55%;">
<!--Μορφοποίηση περιοχής προβολής ιστογράμματος-->
<div style='color:red; text-align: center;'><?php echo "<h3 >

```



```

Συγκριτικό γράφημα από " .date ('d/m/Y', strtotime($start)) . " έως "
.
.date ('d/m/Y', strtotime($end)) . "</h3>"; ?></div>
<div id="div_chart" style="width: 90%; height: 300px;"></div>
</article>
</main>

<nav id="nav">
  <div class="innertube">
    <h3>SMART GREENHOUSE</h3>
    <ul>
      <li><a href="index.html">Homepage</a></li>

      <h3>ThingSpeak</h3>
      <ul>
        <li><a href="temperature.php">Temperature</a></li>
        <li><a href="humidity.php">Humidity</a></li>
        <li><a href="fan.php">Fan</a></li>
        <li><a href="fire.php">Fire</a></li>
        <li><a href="flood.php">Flood </a></li>
        <li><a href="pump.php">Pump</a></li>
        <li><a href="safety_door.php">Safety door</a></li>
        <li><a href="water_tank.php">Water tank</a></li>
      </ul>
      <h3>Google charts</h3>
      <ul>
        <li><a href="statistics1.php">Statistics</a></li>
        <li><a href="gauge.php">Gauge</a></li>
        <li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
        <li><a href="monthly_statistics1.php">Monthly
Statistics</a></li>
        <li><a href="time_graphs.php">Time Graphs</a></li>
        <li><a href="histogramms.php">Histograms</a></li>

        <li><a href="alarms_live.php">Alarms Live</a></li>
      </ul>
      <h3>Database</h3>
      <ul>
        <li><a href="display.php">Display</a></li>
      </ul>
      <h3>Ajax I/O</h3>
      <ul>
        <li><a href="index1.htm">Control Ied and Relay</a></li>
      </ul>
    </div>
  </nav>

</div>

<footer id="footer">
  <div class="innertube">
    <p>
      <span>2024 &copy; </span><a href="#">Πανεπιστήμιο
Πελοποννήσου</a> | <a href="#">Πάτρα</a>
    </p>
  </div>
</footer>

</body>

```

```
</html>
```

## A.22 alarms.php

```
<?php
include "connection.php"; // Προϋπόθεση η σύνδεση με την βάση
$start = date("Y-m-d"); // Αρχικοποίηση μεταβλητών
$end = date("Y-m-d");
if (isset($_POST['start'])) { // Λήψη ημερομηνίας αρχής διαστήματος
    $start = $_POST['start'];
}
if (isset($_POST['end'])) { // Λήψη ημερομηνίας αρχής διαστήματος
    $end = $_POST['end'];
} //Ερώτημα ανάκτησης δεδομένων βάσης, του πίνακα arduino2_data
πάνω από τα όρια
$result = mysqli_query($con, "(SELECT event, fan, fire,
flood FROM data
WHERE (DATE(event)>='$start' AND DATE(event)<='$end') AND
(fan>0 OR fire>0 OR flood>0) ORDER BY event
ASC)");

$dataTable = array();

while ($row = mysqli_fetch_assoc($result)) {
    $fanSensorValue = ($row["fan"] > 0) ? 1 : 0;
    $fireSensorValue = ($row["fire"] < 0) ? 1 : 0;
    $floodSensorValue = ($row["flood"] > 0) ? 1 : 0;
    $event = $row["event"];
    $dataTable[$event] = array(
        "fan" => $fanSensorValue,
        "fire" => $fireSensorValue,
        "flood" => $floodSensorValue
    );
}

//Στη συνέχεια, το $dataArray χρησιμοποιείται για τη συμπλήρωση του
πίνακα δεδομένων του γραφήματος
$array = array();
$array[] = array("Ημέρα", "Ανεμιστήρας", "Φωτιά", "Πλημμύρα");
foreach ($dataTable as $event => $data) {
    $fanSensorValue = isset($data["fan"]) ? $data["fan"] :
0;
    $fireSensorValue = isset($data["fire"]) ?
$data["fire"] : 0;
    $floodSensorValue = isset($data["flood"]) ?
$data["flood"] : 0;

    $dataArray[] = array($event, $fanSensorValue, $fireSensorValue,
$floodSensorValue);
}
?>
<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
    <title>Smart Greenhouse</title>
```

```
<style type="text/css">
body {
  margin: 0;
  padding: 0;
  font-family: Sans-Serif;
  line-height: 1.5em;
  background: #2C6566;
}

#header {
  background: #1F3440;
  color: white;
  height: 100px;
}

#header h1 {

  margin: 0;
  padding-top: 15px;

}

main {
  padding-bottom: 10010px;
  margin-bottom: -10000px;
  float: left;
  width: 100%;
}

#nav {
  padding-bottom: 10010px;
  margin-bottom: -10000px;
  float: left;
  width: 230px;
  margin-left: -100%;
  background: #25bdb0;
}

#footer {
  clear: left;
  width: 100%;
  background: #B4E8CF;
  text-align: center;
  padding: 4px 0;
}

#wrapper {
  overflow: hidden;
}

#content {
  margin-left: 230px;
  /* Same as 'nav' width */
}

.innertube {
  margin: 15px;
  /* Padding for content */
  margin-top: 0;
}
```

```

p {
  color: #000000;
}

nav ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

nav ul a {
  color: black;
  text-decoration: none;
}
</style>
</head>

<body>
<header id="header">
  <div class="innertube">
    <h1>Smart Greenhouse</h1>
  </div>
</header>
<div id="wrapper">
  <main>
    <div id="content">
      <div class="content">
        <meta http-equiv="refresh" content="86400" />
        <br>
        <center><strong>Historic Alarms</strong></center><br>
        <p>
          <center> <strong> Στην σελίδα αυτή ο χρήστης μπορεί να
          συγκρίνει ιστορικά δεδομένα από τους αισθητήρες για συγκεκριμένη
          χρονική περίοδο.</strong><br></center>
        </p>

<article style="width:20%; padding-left: 12px;"><!--Διαίρεση
ενότητας σε δύο περιοχές-->
<!--Φόρμα υποβολής ημερομηνιών διαστήματος-->
<form name="bar_chart" method="post" >
  <h2>Ημερομηνία Αρχής:<h2>
  <input style="font-size: 20px;" type="date" id="start"
name="start" value="<?php echo date("Y-m-d"); ?>" >
  <h2>Ημερομηνία Τέλους:<h2>
  <input style="font-size: 20px;" type="date" id="end" name="end"
value="<?php echo date("Y-m-d"); ?>" >
<p><button class="button" type="submit" value="Submit"> Υποβολή
</button> </p>
</form>
</article>
<article style="width:55%;">

<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(<?php echo

```

```

json_encode($dataArray) ?>);
var options = {
bars: 'vertical',
series: {
0: {
bar: {
groupWidth: '20'
}
}
}
};

var chart = new
google.charts.Bar(document.getElementById('Bar_Stats'));
chart.draw(data, options);
}
</script>
<?php //Δημιουργία τίτλου γραφήματος
echo "<h3 style='color:red; '>" . "Alarm's Από "
.date('d/m/Y', strtotime($start)) . " Έως " .
.date('d/m/Y', strtotime($end)) . "</h3>";
?>
<div id="Bar_Stats" style="width: 100%; height: 50%;"></div>
</article>
</main>

<nav id="nav">
<div class="innertube">
<h3>SMART GREENHOUSE</h3>
<ul>
<li><a href="index.html">Homepage</a></li>

<h3>ThingSpeak</h3>
<ul>
<li><a href="temperature.php">Temperature</a></li>
<li><a href="humidity.php">Humidity</a></li>
<li><a href="fan.php">Fan</a></li>
<li><a href="fire.php">Fire</a></li>
<li><a href="flood.php">Flood </a></li>
<li><a href="pump.php">Pump</a></li>
<li><a href="safety_door.php">Safety door</a></li>
<li><a href="water_tank.php">Water tank</a></li>
</ul>
<h3>Google charts</h3>
<ul>
<li><a href="statistics1.php">Statistics</a></li>
<li><a href="gauge.php">Gauge</a></li>
<li><a href="weekly_statistics1.php">Weekly
Statistics</a></li>
<li><a href="monthly_statistics1.php">Monthly
Statistics</a></li>
<li><a href="time_graphs.php">Time Graphs</a></li>
<li><a href="histogramms.php">Histograms</a></li>
<li><a href="alarms_live.php">Alarms Live</a></li>
</ul>
<h3>Database</h3>
<ul>
<li><a href="display.php">Display</a></li>
</ul>
<h3>Ajax I/O</h3>

```

```
        <ul>
          <li><a href="index1.htm">Control Led and Relay</a></li>
        </ul>
      </div>
    </nav>

  </div>

  <footer id="footer">
    <div class="innertube">
      <p>
        <span>2024 &copy; </span><a href="#">Πανεπιστήμιο
        Πελοποννήσου</a> | <a href="#">Πάτρα</a>
      </p>
    </div>
  </footer>

</body>

</html>
```

# Παράρτημα Β: Κώδικας Software Υλοποίηση Arduino

## B.1 Arduino Mega

```
* B.1.1 Variables initilisation, Libraries

* B.1.1.2 Libraries
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
#include <DS1302.h>
#include <dht.h>
#include "ThingSpeak.h"

* B.1.1.3 Relays initialisation
#define REQ_BUF_SZ 60
#define RELAY1 47
#define dht_dpin A1

/**
 * B.1.1.4 automatic watering initialisation
 * Σενάριο αυτόματου ποτίσματος
 */
int pump = 33;
int sensorPin2 = A4;
int sensorPin1 = A3;
int water_value;
int sun_Value1 = 0;
//B.1.1.5 SD web server initialization
File webFile;
char HTTP_req[REQ_BUF_SZ] = {0};
char req_index = 0;
EthernetServer server1(80);
boolean LED_state[4] = {0};
//B.1.1.6 Network variables initialization
int Wait_between_dumps = 0;

byte mac[] = { 0x90, 0xA2, 0xDA, 0x10, 0x40, 0x4F };

EthernetClient client;

IPAddress ip(192, 168, 1, 23);
IPAddress serv(192, 168, 1, 22);
unsigned long myChannelNumber = 2581223;
const char * myWriteAPIKey = "DMX2800VAOGYY6YD";
int period = 20000;
unsigned long time_now = 0;
dht dht;

/**
//B.1.1.7 shades initialization
 * Σενάριο ελέγχου σκιάστρου
 */
int tiltSwitch = 32;
int shades;
int buzzer = 6;
int greenled1 = 49;
```

```

int redled1 = 42;

/**
//B.1.1.8 fire and flood initialization
 * Σενάριο ανίχνευσης φωτιάς και πλημμύρας
 */
int Grove_Water_Sensor = 28;
int FLAME = 22;
int ALARM = 27;
int flood = 48;
int fire = 46;

/**
//B.1.1.9 shutter and interior lighting initialization
 * Σενάριο ελέγχου παντζουριών και εσωτερικού φωτισμού
 */
int led = 45;
int sensorPin = A2;
int sun_Value = 0;
int led1 = 44;
DS1302 rtc(24,25,26);
Time t;
const int OnHour = 20;
const int OnMin = 30;
const int OffHour = 00;
const int OffMin = 06;

/**
//B.1.1.10 ventilation and air quality initialization
 * Σενάριο ελέγχου εξαερισμού και ποιότητας αέρα
 */
int sensorValue;
int fan1 = 0;
int pinButton = 2;
int stateLED = HIGH;
int stateButton;
int previous = LOW;
long time = 0;
long debounce = 200;
int fan = 5;
int AC = 29;

//B.1.2 initialize void setup
void setup()
//B.1.2.1 Ethernet shield connection

{
  pinMode(10, OUTPUT);
  digitalWrite(10, HIGH);
  Serial.begin(9600);
  SD.begin(4);
  Ethernet.init(10);

  Serial.println("Initialize Ethernet with DHCP:");
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
  }
}

```



```

        if (Ethernet.hardwareStatus() == EthernetNoHardware) {
            Serial.println("Ethernet shield was not found. Sorry,
can't run without hardware. :(");
            while (true) delay(1);
        }
        if (Ethernet.linkStatus() == LinkOFF) {
            Serial.println("Ethernet cable is not connected.");
        }

        Ethernet.begin(mac, ip, serv );
    }
    else {
        Serial.print(" DHCP assigned IP ");
        Serial.println(Ethernet.localIP());
    }
    Ethernet.begin(mac, ip);
    server1.begin();

    delay(1000);
    ThingSpeak.begin(client);

//B.1.2.2 sensors
pinMode(30, OUTPUT);
pinMode(RELAY1, OUTPUT);
pinMode(pump, OUTPUT);

/**
 * Σενάριο ελέγχου σκιάστρου
 */
pinMode(buzzer, OUTPUT);
pinMode(greenled1, OUTPUT);
pinMode(redled1, OUTPUT);
pinMode (tiltSwitch, INPUT);

/**
 * Σενάριο ανίχνευσης φωτιάς και πλημμύρας
 */
Serial.println("Fire and Flood test");
pinMode (FLAME, INPUT);
pinMode (ALARM, OUTPUT);
pinMode (flood, OUTPUT);
pinMode (Grove_Water_Sensor, INPUT);
pinMode (fire, OUTPUT );

/**
 * Σενάριο ελέγχου εξαερισμού και ποιότητας αέρα
 */
pinMode (pinButton, INPUT);
pinMode (fan, OUTPUT);
Serial.println("Air Pollution Level\n\n");
pinMode (AC, OUTPUT);
Serial.println("Humidity and temperature\n\n");

/**
 * Σενάριο ελέγχου παντζουριών και εσωτερικού φωτισμού
 */
rtc.halt(false);

```

```

    rtc.writeProtect(false);
    pinMode (led,OUTPUT);
    pinMode (led1,OUTPUT);
}
//B.1.3 Arduino loop

void loop() {
    db();

    /**
 * B.1.3.1 Automatic watering calculation function
 * Σενάριο αυτόματου ποτίσματος
 */
    int pump = digitalRead(pump);
    water_value = analogRead(sensorPin1);
    sun_Value1 = analogRead(sensorPin2);
    water_value = map(water_value,350,1023,120,0);
    sun_Value1 = map(sun_Value1,15,800,100,0);

    if (sun_Value1 < 60 && (water_value < 40 || water_value < 80 ))
    {
        digitalWrite(pump,HIGH);
    }
    else {
        digitalWrite(pump,LOW);
    }

    Serial.println(water_value);
    Serial.println(sun_Value1);

    /**
 * B.1.3.2 Shades calculation function
 * Σενάριο ελέγχου σκιάστρου
 */
    shades = digitalRead (tiltSwitch);
    if (shades == LOW) {
        digitalWrite(buzzer, HIGH);
        delay(100);
        digitalWrite(buzzer, LOW);
        delay(100);
        digitalWrite(redled1, HIGH);
        digitalWrite(greenled1, LOW);
        Serial.println("*shades open");
    }
    else {
        digitalWrite(greenled1, HIGH);
        digitalWrite(redled1, LOW);
        digitalWrite(buzzer, LOW);
        Serial.println("shades closed");
    }

    /**
 * B.1.3.3 ventilation and air quality calculation function      *
 Σενάριο ελέγχου εξαερισμού και ποιότητας αέρα
 */
    stateButton = digitalRead(pinButton);
    sensorValue = analogRead(0);

```

```

int fan = digitalRead(fan);

if (sensorValue > 100) {
    digitalWrite(fan, HIGH);
}
else {
    digitalWrite(fan, LOW );
}
Serial.print("Air Pollution = ");
Serial.println(sensorValue, DEC);

if ((dht.temperature > 28 || dht.temperature > 24) ||
(dht.temperature < 15 || dht.temperature < 22)) {
    digitalWrite(AC, HIGH);
}
else {
    digitalWrite(AC, LOW );
}

if (dht.temperature >= 21 && dht.temperature<=24) {
    if(stateButton == HIGH && previous == LOW && millis() -
time > debounce) {
        if(stateLED == HIGH) {
            stateLED = LOW;
        }
        else {
            stateLED = HIGH;
        }
        time = millis();
    }
}

digitalWrite(AC, stateLED);
previous = stateButton;}

dht.read11(dht_dpin);

Serial.print("Current humidity = ");
Serial.print(dht.humidity);
Serial.print("% ");
Serial.print("Temperature = ");
Serial.print(dht.temperature);
Serial.println("C ");

/**
 * B.1.3.4 shutter and interior lighting calculation function
 * Σενάριο ελέγχου παντζουριών και εσωτερικού φωτισμού
 */
t = rtc.getTime();
Serial.print(t.hour);
Serial.print(" hour(s), ");
Serial.print(t.min);
Serial.print(" minute(s)");
Serial.println(" ");

sun_Value = analogRead(sensorPin);
sun_Value = map(sun_Value,100,800,100,0);
if (sun_Value < 50){

```

```

digitalWrite(led, HIGH);
Serial.println("LIGHT ON");
if (t.hour == OnHour && t.min == OnMin) {
    digitalWrite(led1, HIGH);
    Serial.println("pantzouri kleisto");
}
}
else if (sun_Value > 40) {
    digitalWrite(led, LOW);
    Serial.println("LIGHT OFF");
    if (t.hour == OffHour && t.min == OffMin) {
        digitalWrite(led1, LOW);
        Serial.println("pantzouri anoixto");
    }
}

/**
* B.1.3.5 fire and flood calculation function
* Σενάριο ανίχνευσης φωτιάς και πλημμύρας
*/
int flood = digitalRead(Grove_Water_Sensor);
int fire = digitalRead(FLAME);
if (flood == HIGH && fire == LOW) {
    digitalWrite(ALARM, HIGH);
    digitalWrite(fire, HIGH);
    digitalWrite(flood, HIGH);
    Serial.println("Flood fire !");
}
else if (digitalRead(Grove_Water_Sensor) == HIGH) {
    digitalWrite(ALARM, HIGH);
    delay(200);
    digitalWrite(ALARM, LOW);
    delay(200);
    Serial.println("Flood!");
    digitalWrite(flood, HIGH);
}
else if (fire == LOW) {
    Serial.println("Fire! Fire!");
    digitalWrite(ALARM, HIGH);
    delay(50);
    digitalWrite(ALARM, LOW);
    delay(50);
    digitalWrite(fire, HIGH);
}
else {
    digitalWrite(ALARM, LOW);
    digitalWrite(flood, LOW);
    Serial.println("Peace");
    Serial.println("Not Flood");
    digitalWrite(fire, LOW);
}
* B.1.3.6 thingspeak calculation function
if(millis() > (time_now + period)){
    dht.read11(dht_dpin);
    float temp = dht.temperature;
    float hum = dht.humidity;

```

```

    ThingSpeak.setField(1, temp);
    ThingSpeak.setField(2, hum);
    ThingSpeak.setField(3, fan );
    ThingSpeak.setField(4, !fire );
    ThingSpeak.setField(5, flood );
    ThingSpeak.setField(6, pump );

int x =ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " +
String(x));
  }
time_now = millis();
}

* B.1.3.7 Arduino communication with website

EthernetClient client = server1.available(); // try to get client
if (client) { // got client?
  boolean currentLineIsBlank = true;
  while (client.connected()) {
    if (client.available()) { // client data available to
read
        char c = client.read(); // read 1 byte (character)
from client
        // limit the size of the stored received HTTP
request
        // buffer first part of HTTP request in HTTP_req
array (string)
        // leave last element in array as 0 to null
terminate string (REQ_BUF_SZ - 1)
        if (req_index < (REQ_BUF_SZ - 1)) {
            HTTP_req[req_index] = c; // save HTTP
request character
            req_index++;
        }
        // last line of client request is blank and ends
with \n
        // respond to client only after last line received
        if (c == '\n' && currentLineIsBlank) {
            // send a standard http response header
            client.println("HTTP/1.1 200 OK");
            // remainder of header follows below, depending
on if
            // web page or XML page is requested
            // Ajax request - send XML file
            if (StrContains(HTTP_req, "ajax_inputs")) {
                // send rest of HTTP header
                client.println("Content-Type: text/xml");
                client.println("Connection: keep-alive");
                client.println();
                SetLEDs();
            }
        }
    }
}
}

```

```

        // send XML file containing input states
        XML_response(client);
    }
    else { // web page request
        // send rest of HTTP header
        client.println("Content-Type: text/html");
        client.println("Connection: keep-alive");
        client.println();
        // send web page
        webFile = SD.open("index1.htm"); //
open web page file
        if (webFile) {
            while(webFile.available()) {
                client.write(webFile.read()); //
send web page to client
            }
            webFile.close();
        }
        // display received HTTP request on serial port
        Serial.print(HTTP_req);
        // reset buffer index and all buffer elements
to 0
        req_index = 0;
        StrClear(HTTP_req, REQ_BUF_SZ);
        break;
    }
    // every line of text received from the client ends
with \r\n
    if (c == '\n') {
        // last character on line of received text
        // starting new line with next character read
        currentLineIsBlank = true;
    }
    else if (c != '\r') {
        // a text character was received from client
        currentLineIsBlank = false;
    }
    } // end if (client.available())
} // end while (client.connected())
delay(1); // give the web browser time to receive the
data
    client.stop(); // close the connection
} // end if (client)
}

* B.1.3.8 Arduino communication with database

void db() {
    float temp = dht.temperature;
    float hum = dht.humidity;
    int fan =digitalRead(fan);
    int flood=digitalRead(Grove_Water_Sensor);
    int fire = digitalRead(FLAME); // read flame sensor

```

```

int shades = digitalRead (tiltSwitch) ;
int pump =digitalRead(pump);

Wait_between_dumps++;
if (Wait_between_dumps >=10)
{
  if (client.connect(serv, 80)) { //Connecting at the IP address and
port we saved before
Serial.println("connected");
client.print("GET /greenhouse/data.php?"); //Connecting and Sending
values to database

client.print("temperature=");
client.print(temp);
client.print("&&humidity=");
client.print(hum);
client.print("&&fan=");
client.print( !fan );
client.print("&&fire=");
client.print( !fire );
client.print("&&flood=");
client.print( flood );
client.print("&&pump=");
client.print(pump);
client.print("&&shades=");
client.print( !shades );
Wait_between_dumps = 0;

client.println( " HTTP/1.1");
client.print( "Host: " );
client.println(serv);
client.println( "Connection: close" );
client.println();
client.println();

client.stop(); //Closing the connection
}
else {
// if you didn't get a connection to the server:
Serial.println("connection failed");
}

}

}

* B.1.3.9 Arduino send the XML file with analog values, switch
status
// and LED status
void XML_response(EthernetClient cl)
{
  int analog_val; // stores value read from analog
inputs
  int count; // used by 'for' loops
  int sw_arr[] = {2, 3, 5}; // pins interfaced to switches

  cl.print("<?xml version = \"1.0\" ?>");

```

```

cl.print("<inputs>");
// read analog inputs
for (count = 0; count <= 4; count++) { // A0 to A4
    analog_val = analogRead(count);
    cl.print("<analog>");
    cl.print(analog_val);
    cl.println("</analog>");
}
// read switches
for (count = 0; count < 3; count++) {
    cl.print("<switch>");
    if (digitalRead(sw_arr[count])) {
        cl.print("ON");
    }
    else {
        cl.print("OFF");
    }
    cl.println("</switch>");
}
// checkbox LED states
// LED1
cl.print("<LED>");
if (LED_state[0]) {
    cl.print("checked");
}
else {
    cl.print("unchecked");
}
cl.println("</LED>");
// LED2
cl.print("<LED>");
if (LED_state[1]) {
    cl.print("checked");
}
else {
    cl.print("unchecked");
}
cl.println("</LED>");
// button LED states
// LED3
cl.print("<LED>");
if (LED_state[2]) {
    cl.print("on");
}
else {
    cl.print("off");
}
cl.println("</LED>");
// LED4
cl.print("<LED>");
if (LED_state[3]) {
    cl.print("on");
}
else {
    cl.print("off");
}
cl.println("</LED>");

```



```

    cl.print("</inputs>");
}

// sets every element of str to 0 (clears array)
void StrClear(char *str, char length)
{
    for (int i = 0; i < length; i++) {
        str[i] = 0;
    }
}

// searches for the string sfind in the string str
// returns 1 if string found
// returns 0 if string not found
char StrContains(char *str, char *sfind)
{
    char found = 0;
    char index = 0;
    char len;

    len = strlen(str);

    if (strlen(sfind) > len) {
        return 0;
    }
    while (index < len) {
        if (str[index] == sfind[found]) {
            found++;
            if (strlen(sfind) == found) {
                return 1;
            }
        }
        else {
            found = 0;
        }
        index++;
    }

    return 0;
}

// checks if received HTTP request is
switching on/off LEDs // also saves the state of the LEDs
void SetLEDs(void)
{
    // LED 1 (pin 6)
    if (StrContains(HTTP_req, "LED1=1")) {
        LED_state[0] = 1; // save LED state
        digitalWrite(30, HIGH);
    }
    else if (StrContains(HTTP_req, "LED1=0")) {
        LED_state[0] = 0; // save LED state
        digitalWrite(30, LOW);
    }
    // LED 2 (pin 7)

```

```

    if (StrContains(HTTP_req, "LED2=1")) {
        LED_state[1] = 1; // save LED state
        digitalWrite(47, HIGH);
    }
    else if (StrContains(HTTP_req, "LED2=0")) {
        LED_state[1] = 0; // save LED state
        digitalWrite(47, LOW);
    }
    // LED 3 (pin 8)
    if (StrContains(HTTP_req, "LED3=1")) {
        LED_state[2] = 1; // save LED state
        digitalWrite(30, HIGH);
    }
    else if (StrContains(HTTP_req, "LED3=0")) {
        LED_state[2] = 0; // save LED state
        digitalWrite(30, LOW);
    }
    // LED 4 (pin 9)
    if (StrContains(HTTP_req, "LED4=1")) {
        LED_state[3] = 1; // save LED state
        digitalWrite(47, HIGH);
    }
    else if (StrContains(HTTP_req, "LED4=0")) {
        LED_state[3] = 0; // save LED state
        digitalWrite(47, LOW);
    }
}

```

## B.2 Arduino Uno -Σενάριο Ασφάλειας Εξωτερικής Πύρτας

```

* B.2.1.1 Libraries
#include <SPI.h>
#include <Ethernet.h>
#include "ThingSpeak.h"
#include <IRremote.h>

#define ctsPin 9
//B.2.1.2 Network variables initialization
byte mac[] = { 0xB0, 0xAC, 0xD2, 0x2A, 0xD7, 0x91 };
byte ip[] = {192, 168, 1, 24 };
byte serv[] = {192, 168, 1, 22};
EthernetClient client;
int Wait_between_dumps = 0;

unsigned long myChannelNumber = 2581223;
const char * myWriteAPIKey = "DMX2800VAOGYY6YD";
//byte mac[] = { 0x90, 0xA2, 0xDA, 0x10, 0x40, 0x4F };
int period1 = 15000;
unsigned long time_now1 = 0;
//B.2.1.3 variables initialization
int led1 = 6;
int sensor = 2;
int state = LOW;
int val = 0;
int flag1 = 0;

```

```

int flag = 0;
int counter = 0;
int led2 = 7;
int led3 = 8;
int period = 10000;
unsigned long time_now = 0;
int led = 5;
int a = 0;
int b = 0;
int c = 0;
const int buttonPin = 4;
int buttonState = 0;
int RECV_PIN = 3;

IRrecv irrecv(RECV_PIN);
decode_results results;
//B.2.1.4 void setup initialization
void setup() {
    Ethernet.init(10);

    Serial.println("Initialize Ethernet with DHCP:");
    if (Ethernet.begin(mac) == 0) {
        Serial.println("Failed to configure Ethernet using DHCP");

        if (Ethernet.hardwareStatus() == EthernetNoHardware) {
            Serial.println("Ethernet shield was not found. Sorry,
can't run without hardware. :(");
            while (true) delay(1);
        }
        if (Ethernet.linkStatus() == LinkOFF) {
            Serial.println("Ethernet cable is not connected.");
        }

        Ethernet.begin(mac, ip, serv );
    }
    else {
        Serial.print(" DHCP assigned IP ");
        Serial.println(Ethernet.localIP());
    }

    delay(1000);

    ThingSpeak.begin(client);

    Ethernet.begin(mac, ip);

//B.2.1.5 ports initialization
    pinMode(led1, OUTPUT);
    pinMode(sensor, INPUT);
    pinMode(led2, OUTPUT);
    pinMode(led3, OUTPUT);
    pinMode(ctsPin, INPUT);
    pinMode(led, OUTPUT);

    Serial.begin(9600);
    irrecv.enableIRIn();

```

```

}
//B.2.1.6 arduino loop initialization
void loop() {
  db();
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    translateIR();
    for (int z = 0; z < 2; z++) {
      irrecv.resume(); // receive the next value
    }
  }
//B.2.1.7 safety door calculation function
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    if (millis() > (time_now + period)) {
      if (flag + flag1 == 2)
        digitalWrite(led3, HIGH);
      else digitalWrite(led3, LOW);
      counter = 0;

      Serial.println("-----");
      time_now = millis();
      a = 0; b = 0; c = 0;
      flag1 = 0;
      digitalWrite(led, LOW);
    }
    else {
      if (digitalRead(ctsPin) == HIGH) counter = counter+1;

      if (counter < 5 && counter > 1){
        digitalWrite(led2, HIGH);
        flag = 1;
      }
      else {
        flag = 0;
        digitalWrite(led2, LOW);
      }
      if (a + b + c == 3) {
        digitalWrite(led, HIGH);
        flag1 = 1;
      }
    }
  }

  Serial.println(counter);

  val = digitalRead(sensor);
  if (val == HIGH) {
    digitalWrite(led1, HIGH);
    delay(100);

    if (state == LOW) {
      Serial.println("Motion detected!");
      state = HIGH;
    }
  }
  else {

```

```

digitalWrite(led1, LOW);
delay(200);

if (state == HIGH) {
    Serial.println("Motion stopped!");
    state = LOW;
}
}

if(millis() > (time_now1 + period1)) {
    ThingSpeak.setField(7, digitalRead(buttonPin));
    int x = ThingSpeak.writeFields(myChannelNumber,
myWriteAPIKey);

    if(x == 200) {
        Serial.println("Channel update successful.");
    }
    else {
        Serial.println("Problem updating channel. HTTP error
code " + String(x));
    }
    time_now1 = millis();
}
}

void db() {
    Wait_between_dumps++;

    if (Wait_between_dumps >= 10) {
        if (client.connect(serv, 80)) {
            Serial.println("connected");
            client.print("GET /greenhouse/data.php?");
            client.print("safety_door=");
            client.println(digitalRead(buttonPin));
            client.stop();
        }
        else {
            Serial.println("connection failed");
        }
    }
}

//B.2.1.8 IR translate calculation function

void translateIR() {
    switch(results.value) {
        case 0xFFA25D:
            a=1;
            break;
        case 0xFF629D:
            b=1;
            break;
        case 0xFFE21D:
            c=1;
            break;
        default:
            Serial.println("OTHER");
    }
}

```

```
    delay(200);  
}
```

### B.3 Arduino Uno-Water tank

```
* B.3.1.1 Libraries  
  
#include <SPI.h> //ethernet  
#include <Ethernet.h>  
//B.3.1.2 Network variables initialization  
byte mac[] = { 0x90, 0xA2, 0xDA, 0x10, 0x40, 0x4F };  
byte ip[] = {192, 168, 1, 15 }; //Enter the IP of ethernet shield  
byte serv[] = {192, 168, 1, 16} ; //Enter the IPv4 address  
EthernetClient client;  
int Wait_between_dumps = 0;  
  
#include "ThingSpeak.h" //thingspeak  
unsigned long myChannelNumber = 2581223;  
const char * myWriteAPIKey = "DMX2800VAOGYY6YD";  
//byte mac[] = { 0x90, 0xA2, 0xDA, 0x10, 0x40, 0x4F };  
int period1 = 15000;  
unsigned long time_now1 = 0;  
  
#include <OneWire.h>  
#include <DallasTemperature.h>  
// Libraries for the LCD Display and I2C  
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>  
// Data wire is conntec to the Arduino digital pin 4  
#define ONE_WIRE_BUS 4  
  
//B.3.1.3 variables initialization  
  
#define INCREASE_BUTTON_PIN 5  
#define DECREASE_BUTTON_PIN 6  
// Setup a oneWire instance to communicate with any OneWire devices  
OneWire oneWire(ONE_WIRE_BUS);  
  
// Pass our oneWire reference to Dallas Temperature sensor  
DallasTemperature sensors(&oneWire);  
// Setup LCD display  
LiquidCrystal_I2C lcd(0x27, 20, 4);  
  
int sensorPin = A0;  
float volt;  
float ntu;  
  
const int RelayPin = 8;  
  
int water_tank = 0; // value of water_tank status to send to the  
website via ThingSpeak  
  
// Selection of vegetable, using one button to increase the choice  
variable and another to decrease  
int selection = 0;  
// User must change this to begin watering process  
int watering_command = 0;
```

```

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

//B.3.1.4 void setup initialization

void setup()
{
    Ethernet.init(10); // Most Arduino Ethernet hardware
//thingspeak
    //Serial.begin(115200); //Initialize serial

    // start the Ethernet connection:
    Serial.println("Initialize Ethernet with DHCP:");
    if (Ethernet.begin(mac) == 0) {
        Serial.println("Failed to configure Ethernet using DHCP");
        // Check for Ethernet hardware present
        if (Ethernet.hardwareStatus() == EthernetNoHardware) {
            Serial.println("Ethernet shield was not found. Sorry, can't
run without hardware. :(");
            while (true) {
                delay(1); // do nothing, no point running without Ethernet
hardware
            }
        }
        if (Ethernet.linkStatus() == LinkOFF) {
            Serial.println("Ethernet cable is not connected.");
        }
        // try to congifure using IP address instead of DHCP:
        Ethernet.begin(mac, ip, serv );
    } else {
        Serial.print(" DHCP assigned IP ");
        Serial.println(Ethernet.localIP());
    }
    // give the Ethernet shield a second to initialize:

    delay(1000);

    ThingSpeak.begin(client); // Initialize ThingSpeak

Ethernet.begin(mac, ip); //ethernet

    // Start serial communication for debugging purposes
    Serial.begin(9600);
    // Start up the library
    sensors.begin();
//B.3.1.5 lcd initializstion
    lcd.init();
    lcd.clear();
    lcd.backlight();
//B.3.1.6 ports initialization
    pinMode(RelayPin, OUTPUT);

    pinMode(INCREASE_BUTTON_PIN, INPUT);
    pinMode(DECREASE_BUTTON_PIN, INPUT);
}

```

```

//B.3.1.7 arduino loop initialization

void loop()
{
  //db();
  Read_RemoteXbee_Loop();
  // Call sensors.requestTemperatures() to issue a global
  temperature and Requests to all devices on the bus
  sensors.getTempCByIndex(0)
  sensors.requestTemperatures();

  Serial.print("Celsius temperature: ");
  // Why "byIndex"? You can have more than one IC on the same bus.
  0 refers to the first IC on the wire
  Serial.print(sensors.getTempCByIndex(0));
  Serial.print(" - Fahrenheit temperature: ");
  Serial.println(sensors.getTempCByIndex(0));
  delay(1000);
  Serial.print("NTU: ");
  Serial.println(ntu);
  Serial.print("light value: ");
  Serial.println(LDR);

  // ----- Water turbidity -----
  // We take 800 readings from the sensor and take the average
  value to counter the sensor's noise.
  //B.3.1.8 tank water quality calculation function

  volt = 0;
  for(int i=0; i<800; i++)
  {
    volt += ((float)analogRead(sensorPin)/1023)*5;
  }
  volt = volt/800;
  volt = round_to_dp(volt,2);
  if(volt < 2.5){
    ntu = 3000;
  }else{
    ntu = -1120.4*square(volt)+5742.3*volt-4353.8;
  }

  // Water pump
  enum vegetable {
    tomato = 1,
    cucumber = 2,
    lettuce = 3
  };

  int increase_button_state = digitalRead(INCREASE_BUTTON_PIN);
  if (increase_button_state == 1 && selection <= 2) // selection <=
  length(vegetable)
    selection += 1;

  int decrease_button_state = digitalRead(DECREASE_BUTTON_PIN);
  if (decrease_button_state == 1 && selection > 0)
    selection -= 1;

  int watering_command=1;
  if (watering_command == 1) {

```



```

    if (selection == 1) { // tomato
        if (LDR > 60 || ntu > 2500 || sensors.getTempCByIndex(0) < 21
|| sensors.getTempCByIndex(0) > 27) {
            digitalWrite(RelayPin, LOW); // The water pump stops
            Serial.println("Watering tomatoes aborted due to out of
limits temperature or turbidity. Please check the water inside the
tank.");
        }
        else {
            digitalWrite(RelayPin, HIGH); // The water pump starts
watering the tomatoes
        }
    }
    else if (selection == 2) { // cucumber
        if (LDR > 60 || ntu > 2500 || sensors.getTempCByIndex(0) < 15
|| sensors.getTempCByIndex(0) > 29) {
            digitalWrite(RelayPin, LOW); // The water pump stops
            Serial.println("Watering of cucumbers aborted due to out
of limits temperature or turbidity. Please check the water inside
the tank.");
        }
        else {
            digitalWrite(RelayPin, HIGH); // The water pump starts
watering the cucumbers
        }
    }
    else if (selection == 3) { // lettuce
        if (LDR > 60 || ntu > 2500 || sensors.getTempCByIndex(0) < 10
|| sensors.getTempCByIndex(0) > 20) {
            digitalWrite(RelayPin, LOW); // The water pump stops
            Serial.println("Watering of lettuce aborted due to out of
limits temperature or turbidity. Please check the water inside the
tank.");
        }
        else {
            digitalWrite(RelayPin, HIGH); // The water pump starts
watering the lettuce
        }
    }
    else {
        Serial.println("Watering process is inactive");
        digitalWrite(RelayPin, LOW);
    }
}
//B.3.1.9 lcd calculation function

//value indication on the display
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("selected vegetable:");
lcd.print(selection);

    lcd.setCursor(0, 1);
    lcd.print("degC:");
    lcd.print(sensors.getTempCByIndex(0));

    lcd.setCursor(0, 2);
    lcd.print("NTU:");
    lcd.print(ntu);
    delay(10);

```

```

lcd.setCursor(0, 3);
lcd.print("light value:");
lcd.print(LDR);

//thingspeak
if(millis() > (time_now1 + period1)){

    ThingSpeak.setField(8,water_tank );

int x =ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
if(x == 200){
    Serial.println("Channel update successful.");
}
else{
    Serial.println("Problem updating channel. HTTP error code " +
String(x));
}
time_now1 = millis();
}

}

void db(){
    int water_tank = digitalRead(water_tank);
    Wait_between_dumps++;
if (Wait_between_dumps >= 10){
//ethernet
    if (client.connect(serv, 80)) { //Connecting at the IP address and
port we saved before
Serial.println("connected");
client.print("GET /greenhouse/data.php?"); //Connecting and Sending
values to database

client.print("&&water_tank=");
client.print( water_tank );

client.stop(); //Closing the connection
}
else {
// if you didn't get a connection to the server:
Serial.println("connection failed");
}

}
}

//B.3.1.10 xbee loop initialization
void Read_RemoteXbee_Loop() {
// check if the frame has at least 22 bits
if (Serial.available() >= 21) {
// check if it has start byte 7E
if (Serial.read() == 0x7E) {
// Rejection of the first 19 bytes
for (int i = 1; i < 19; i++) {
byte discardbyte = Serial.read();

```

```

}
//the First byte is the value of the LDR
// 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
//mathematical operation to create 2 bits of LSB and MSB in one
byte

LDR = analogLSB1 + (analogMSB1 * 256);
LDR = map(LDR, 0, 1030, 0, 100);
}
}

float round_to_dp( float in_value, int decimal_place )
{
    float multiplier = powf( 10.0f, decimal_place );
    in_value = roundf( in_value * multiplier ) / multiplier;
    return in_value;
}

```