

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΦΑΡΜΟΣΜΕΝΑ ΗΛΕΚΤΡΟΝΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Αυτόματο σύστημα θερμοκηπίου και ο έλεγχός του
από απόσταση (έξυπνο θερμοκήπιο)»



Του φοιτητή
Νταμοτσίδα Βασίλειου
Αρ. Μητρώου: 52017Μ

Επιβλέπων
Όνοματεπώνυμο: Γιακουμής Άγγελος
Βαθμίδα: Επίκουρος Καθηγητής

Ημερομηνία 26/02/2024

Τίτλος Δ.Ε. Αυτόματο σύστημα θερμοκηπίου και ο έλεγχός του από απόσταση (έξυπνο θερμοκήπιο)

Κωδικός Δ.Ε. 21394

Ονοματεπώνυμο φοιτητή Νταμοτσίδης Βασίλειος

Ονοματεπώνυμο εισηγητή Γιακουμής Άγγελος

Ημερομηνία ανάληψης Δ.Ε. 27/10/21

Ημερομηνία περάτωσης Δ.Ε. 26/02/2024

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Μεταπτυχιακό Πρόγραμμα Σπουδών «Εφαρμοσμένα Ηλεκτρονικά Συστήματα» στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Νταμοτσίδα Βασίλειου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Στους φίλους και στην οικογένειά μου»

Πρόλογος

Στόχος της παρούσας Διπλωματικής Εργασίας είναι η δημιουργία ενός αποτελεσματικού συστήματος διαχείρισης θερμοκηπίου από απόσταση. Τα οφέλη της ενασχόλησης με αυτή την εργασία είναι τρία. Πρώτον, η κατανόηση της λειτουργίας των μικροελεγκτών της σειράς STM32 και ESP32, η εξοικείωση με αυτά, καθώς και η κατασκευή πλακέτας. Δεύτερον, η ενασχόληση με IoT πρωτόκολλα επικοινωνίας οδήγησε στην καλύτερη κατανόηση για το πως λειτουργεί το διαδίκτυο. Και τρίτον, η δημιουργία ιστοσελίδας από το μηδέν (front-end, back-end και βάση δεδομένων), το οποίο από μόνο του αποτελεί μεγάλη πρόκληση, αφού χρειάστηκε η ενασχόληση με γλώσσες προγραμματισμού πέραν του κλάδου της ηλεκτρονικής (PHP, Python, Javascript, SQL).

Περίληψη

Η παρούσα Διπλωματική Εργασία πραγματεύεται την υλοποίηση ενός αποτελεσματικού συστήματος απομακρυσμένης διαχείρισης ενός θερμοκηπίου. Στην αρχή της βιβλιογραφίας γίνεται μια μικρή ιστορική αναδρομή σε συστήματα θερμοκηπίων και πώς αυτά έχουν εξελιχθεί με την πάροδο των ετών. Επιπλέον, αναφέρονται κάποιες βασικές λειτουργίες που πρέπει να εκτελεί ένα θερμοκήπιο. Στη συνέχεια, γίνεται αναφορά για τις μέχρι σήμερα αντίστοιχες εργασίες και δημοσιεύσεις, έτσι ώστε να φανεί ότι το σύστημα που αναπτύχθηκε προσφέρει περισσότερες δυνατότητες και καινοτομίες. Έπειτα, γίνεται μια αναλυτική παρουσίαση του πρωτόκολλου MQTT, καθώς και η σύγκρισή του με άλλα πρωτόκολλα επικοινωνίας. Παρακάτω γίνεται μια εκτενής ανάλυση-επεξήγηση των αισθητήρων, των επιμέρους εξαρτημάτων, όπως επίσης και των αναπτυξιακών πλακετών που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος. Ύστερα ακολουθεί ο σχεδιασμός και ο προγραμματισμός του ηλεκτρονικού κυκλώματος το οποίο αποτελεί και το σημαντικότερο κεφάλαιο της εργασίας. Επίσης γίνεται η παρουσίαση του λειτουργικού συστήματος (front-end και back-end) το οποίο αποτελεί εξίσου σημαντικό κεφάλαιο. Πιο συγκεκριμένα, παρουσιάζεται η ιστοσελίδα του συστήματος όπου από εκεί γίνεται η διαχείριση του θερμοκηπίου (παρακολούθηση μετρήσεων και ενεργοποίηση/απενεργοποίηση λειτουργιών). Τέλος, παρουσιάζονται τα συμπεράσματα καθώς και οι μελλοντικές επεκτάσεις-βελτιώσεις του αναπτυχθέντος συστήματος.

«Automatic greenhouse system and its remote control (smart greenhouse)»

«Ntamotsidis Vasileios»

Abstract

This Diploma Thesis deals with the implementation of an effective remote management system of a greenhouse. At the beginning of the bibliography there is a short historical review of greenhouse systems and how they have evolved over the years. In addition, some basic functions that a greenhouse should perform are listed. Then, reference is made to the corresponding papers and publications to date, so that it can be seen that the system developed offers more possibilities and innovations. Then, a detailed presentation of the MQTT protocol is given, as well as its comparison with other communication protocols. Below is an extensive analysis and explanation of the sensors, the individual components, as well as the development boards that were used to implement the system. Then follows the design and programming of the electronic circuit which is the most important chapter of this thesis. The operating system (front-end and back-end) is also presented, which is an equally important chapter. More specifically, the website of the system is presented, where the greenhouse is managed from there (monitoring of measurements and activation/deactivation of functions). Finally, the conclusions as well as the future extensions-improvements of the developed system are presented.

Ευχαριστίες

Πρώτο από όλους θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή της Διπλωματικής μου Εργασίας κ. Άγγελο Γιακουμή για την βοήθεια που μου προσέφερε σε τεχνικό επίπεδο αλλά και για την καθοδήγηση στην περάτωσή της. Επίσης, ιδιαίτερες ευχαριστίες θα ήθελα να δώσω στους γονείς μου και στον αδερφό μου που με στηρίζουν και μου συμπαραστέκονται στον αγώνα μου.

Περιεχόμενα

| | |
|----------------------------------------------|----|
| Πρόλογος..... | 5 |
| Περίληψη..... | 6 |
| Abstract | 7 |
| Ευχαριστίες | 8 |
| Περιεχόμενα | 9 |
| Κατάλογος Εικόνων | 12 |
| Κατάλογος Πινάκων..... | 14 |
| Συνομογραφίες..... | 15 |
| Κεφάλαιο 1ο: Το θερμοκήπιο..... | 16 |
| 1.1 Εισαγωγή..... | 16 |
| 1.2 Ιστορική αναδρομή..... | 16 |
| 1.3 Βασικές λειτουργίες θερμοκηπίων | 18 |
| 1.4 Αντίστοιχες εργασίες-έρευνες | 18 |
| 1.5 Επίλογος..... | 19 |
| Κεφάλαιο 2ο: Πρωτόκολλο MQTT..... | 20 |
| 2.1 Εισαγωγή..... | 20 |
| 2.2 Ιστορική αναδρομή..... | 20 |
| 2.3 Το μοτίβο Publish / Subscribe..... | 21 |
| 2.4 Client, Broker και Συνδέσεις..... | 22 |
| 2.5 Publish, Subscribe και Unsubscribe | 24 |
| 2.6 Χαρακτηριστικά πρωτοκόλλου MQTT..... | 26 |
| 2.6.1 Topics | 26 |
| 2.6.2 Quality of Service (QoS)..... | 27 |
| 2.6.3 Persistent Session | 29 |
| 2.6.4 Retained Messages | 30 |
| 2.6.5 Last Will and Testament..... | 30 |
| 2.6.6 Keep Alive..... | 31 |
| 2.7 Σύγκριση MQTT με άλλα πρωτόκολλα | 31 |
| 2.7.1 Πρωτόκολλο HTTP | 31 |
| 2.7.2 Πρωτόκολλο AMQP | 33 |
| 2.7.3 Πρωτόκολλο CoAP | 34 |
| 2.7.4 Πρωτόκολλο ZeroMQ..... | 36 |

| | | |
|--------------------------------------------------------------|----------------------------------------------|----|
| 2.8 | Επίλογος..... | 37 |
| Κεφάλαιο 3ο: Αισθητήρες και λοιπά υλικά..... | | 38 |
| 3.1 | Εισαγωγή..... | 38 |
| 3.2 | Αισθητήρες..... | 38 |
| 3.2.1 | Αισθητήρας DHT22 | 38 |
| 3.2.2 | Αισθητήρας MG-811 | 41 |
| 3.2.3 | Αισθητήρας υγρασίας εδάφους | 45 |
| 3.2.4 | Αισθητήρας βροχής..... | 45 |
| 3.2.5 | Μηχανικός διακόπτης (Tamper)..... | 46 |
| 3.3 | Λοιπά εξαρτήματα..... | 46 |
| 3.3.1 | Stepper motor | 46 |
| 3.3.2 | MIC29302WT | 47 |
| 3.3.3 | LF33CV..... | 48 |
| 3.3.4 | Οθόνη ST7735..... | 48 |
| 3.3.5 | Τροφοδοσία | 49 |
| 3.3.6 | Ρελέ και τρανζίστορ | 49 |
| 3.3.7 | SIM800L | 50 |
| 3.3.8 | STM32F103C8T6 | 51 |
| 3.3.9 | ESP32..... | 52 |
| 3.4 | Επίλογος..... | 52 |
| Κεφάλαιο 4ο: Σχεδιασμός και προγραμματισμός κυκλώματος | | 53 |
| 4.1 | Εισαγωγή..... | 53 |
| 4.2 | Σχεδιασμός κυκλώματος | 53 |
| 4.3 | Υλοποίηση πλακέτας και θερμοκηπίου..... | 57 |
| 4.4 | Προγραμματισμός μικροελεγκτών | 61 |
| 4.4.1 | Ρυθμίσεις STM32F103C8T6..... | 61 |
| 4.4.2 | Διαγράμματα ροής..... | 65 |
| 4.5 | Μπλοκ-διάγραμμα ηλεκτρονικού κυκλώματος..... | 68 |
| 4.6 | Επίλογος..... | 68 |
| Κεφάλαιο 5ο: Το λειτουργικό σύστημα | | 69 |
| 5.1 | Εισαγωγή..... | 69 |
| 5.2 | MQTT Broker..... | 70 |
| 5.3 | Βάση δεδομένων MySQL | 71 |
| 5.4 | Ιστοσελίδα του συστήματος | 72 |
| 5.5 | Μηνύματα SMS..... | 76 |

| | | |
|--------------------------------------------------------|--------------------------|-----|
| 5.6 | Python..... | 77 |
| 5.7 | Επίλογος..... | 78 |
| Κεφάλαιο 6ο: Συμπεράσματα και προτάσεις βελτίωσης..... | | 79 |
| 6.1 | Συμπεράσματα..... | 79 |
| 6.2 | Προτάσεις βελτίωσης..... | 79 |
| ΒΙΒΛΙΟΓΡΑΦΙΑ..... | | 80 |
| ΠΑΡΑΡΤΗΜΑ Α: ΚΩΔΙΚΑΣ STM32..... | | 86 |
| ΠΑΡΑΡΤΗΜΑ Β: ΚΩΔΙΚΑΣ ESP32..... | | 111 |
| ΠΑΡΑΡΤΗΜΑ Γ: ΚΩΔΙΚΑΣ ΡΥΘΜΟΝ..... | | 124 |
| ΠΑΡΑΡΤΗΜΑ Δ: ΚΩΔΙΚΑΣ SQL..... | | 131 |
| ΠΑΡΑΡΤΗΜΑ Ε: ΚΩΔΙΚΕΣ ΙΣΤΟΣΕΛΙΔΑΣ..... | | 135 |

Κατάλογος Εικόνων

| | |
|-----------------------------------------------------------------|----|
| Εικόνα 1. Το πρωτόκολλο MQTT..... | 21 |
| Εικόνα 2. Το MQTT βασίζεται στο TCP/IP..... | 22 |
| Εικόνα 3. Σύνδεση του Client στον Broker..... | 23 |
| Εικόνα 4. Το μήνυμα PUBLISH..... | 24 |
| Εικόνα 5. Η διαδικασία του PUBLISH/SUBSCRIBE..... | 25 |
| Εικόνα 6. Η διαδικασία του UNSUBSCRIBE..... | 25 |
| Εικόνα 7. Quality of Service 0..... | 27 |
| Εικόνα 8. Quality of Service 1..... | 28 |
| Εικόνα 9. Quality of Service 2..... | 28 |
| Εικόνα 10. Τρόπος ανάκτησης ιστοσελίδων με το HTTP..... | 32 |
| Εικόνα 11. Αποστολή πληροφορίας στο HTTP με τη μέθοδο POST..... | 32 |
| Εικόνα 12. Αρχιτεκτονική πρωτόκολλου AMQP..... | 34 |
| Εικόνα 13. Αλληλεπίδραση μεταξύ Client και Server..... | 35 |
| Εικόνα 14. Τρόπος λειτουργίας του πρωτόκολλου CoAp..... | 35 |
| Εικόνα 15. Τρόπος λειτουργίας του πρωτόκολλου ZeroMQ..... | 36 |
| Εικόνα 16. Ο Broker του ZeroMQ..... | 37 |
| Εικόνα 17. Αισθητήρας DHT22..... | 38 |
| Εικόνα 18. Η σειρά των 40bit..... | 39 |
| Εικόνα 19. Το start signal και το response signal..... | 40 |
| Εικόνα 20. Διαχωρισμός 0 και 1 bit..... | 40 |
| Εικόνα 21. Διάγραμμα ροής της λειτουργίας του DHT22..... | 41 |
| Εικόνα 22. Η πλακέτα SEN0159..... | 41 |
| Εικόνα 23. Ο αισθητήρας MG-811..... | 42 |
| Εικόνα 24. Η λογαριθμική κλίμακα του MG-811..... | 43 |
| Εικόνα 25. Κύκλωμα τροφοδοσίας του MG-811..... | 43 |
| Εικόνα 26. Κύκλωμα ενίσχυσης της εξόδου του MG-811..... | 44 |
| Εικόνα 27. Αισθητήρας υγρασίας εδάφους..... | 45 |
| Εικόνα 28. Αισθητήρας βροχής..... | 45 |
| Εικόνα 29. Μηχανικός διακόπτης (tamper)..... | 46 |
| Εικόνα 30. Βηματικός κινητήρας..... | 46 |
| Εικόνα 31. Το εσωτερικό του βηματικού κινητήρα..... | 47 |
| Εικόνα 32. Το ολοκληρωμένο ULN2003AN..... | 47 |
| Εικόνα 33. Ο σταθεροποιητής MIC29302WT..... | 47 |
| Εικόνα 34. Ο σταθεροποιητής LF33CV..... | 48 |
| Εικόνα 35. Η οθόνη ST7335..... | 48 |
| Εικόνα 36. Το τροφοδοτικό MeanWell..... | 49 |
| Εικόνα 37. Το ρελέ PE014005 και το τρανζίστορ BC337..... | 50 |
| Εικόνα 38. Το SIM800L..... | 50 |
| Εικόνα 39. Αναπτυξιακή πλακέτα STM32F103C8T6..... | 51 |
| Εικόνα 40. Αναπτυξιακή πλακέτα ESP32..... | 52 |
| Εικόνα 41. Το ηλεκτρονικό κύκλωμα του συστήματος..... | 56 |
| Εικόνα 42. Υλοποίηση κυκλώματος στο ράστερ..... | 57 |
| Εικόνα 43. Διαδικασία κατασκευής θερμοκηπίου..... | 57 |
| Εικόνα 44. Το σχέδιο της πλακέτας από το Eagle..... | 57 |

| | |
|----------------------------------------------------------------|----|
| Εικόνα 45. Αποχάλκωση πλακέτας..... | 58 |
| Εικόνα 46. Τοποθέτηση υλικών στην πλακέτα | 58 |
| Εικόνα 47. Εφαρμογή της πλακέτας στο θερμοκήπιο..... | 58 |
| Εικόνα 48. Αυτοκόλλητο πρόσοψης της συσκευής | 59 |
| Εικόνα 49. Μπροστινή όψη θερμοκηπίου..... | 59 |
| Εικόνα 50. Πλαϊνή όψη θερμοκηπίου | 60 |
| Εικόνα 51. Το εσωτερικό του θερμοκηπίου..... | 60 |
| Εικόνα 52. Ρύθμιση των pins του STM32 | 62 |
| Εικόνα 53. Ρύθμιση USART 1 | 62 |
| Εικόνα 54. Ρύθμιση USART 2..... | 63 |
| Εικόνα 55. Ρύθμιση ADC | 63 |
| Εικόνα 56. Ρύθμιση SPI..... | 64 |
| Εικόνα 57. Ρύθμιση του Timer (A)..... | 64 |
| Εικόνα 58. Ρύθμιση του Timer (B) | 65 |
| Εικόνα 59. Διάγραμμα ροής STM32..... | 66 |
| Εικόνα 60. Διάγραμμα ροής ESP32..... | 67 |
| Εικόνα 61. Μπλοκ-διάγραμμα ηλεκτρονικού κυκλώματος | 68 |
| Εικόνα 62. Μπλοκ-διάγραμμα του συστήματος | 69 |
| Εικόνα 63. Mosquitto Broker | 70 |
| Εικόνα 64. MQTT Explorer | 70 |
| Εικόνα 65. Περιβάλλον του XAMPP..... | 71 |
| Εικόνα 66. Πίνακας "personal data" της βάσης δεδομένων..... | 71 |
| Εικόνα 67. Πίνακας "greenhouses" της βάσης δεδομένων | 71 |
| Εικόνα 68. Πίνακας "measurements" της βάσης δεδομένων | 72 |
| Εικόνα 69. Σελίδα εγγραφής και login του χρήστη..... | 72 |
| Εικόνα 70. Σελίδα επαναφοράς κωδικού πρόσβασης | 73 |
| Εικόνα 71. Η αρχική σελίδα της ιστοσελίδας | 73 |
| Εικόνα 72. Λειτουργία αλλαγής κωδικού πρόσβασης | 73 |
| Εικόνα 73. Λειτουργία διαγραφής λογαριασμού χρήστη..... | 74 |
| Εικόνα 74. Λειτουργία προσθήκης και διαγραφής θερμοκηπίου..... | 74 |
| Εικόνα 75. Διαγράμματα θερμοκρασίας και υγρασίας | 74 |
| Εικόνα 76. Διαγράμματα υγρασίας εδάφους..... | 75 |
| Εικόνα 77. Διαγράμματα CO2 και βροχής..... | 75 |
| Εικόνα 78. Εντολές θερμοκηπίου..... | 75 |
| Εικόνα 79. Διάγραμμα ροής κεντρικής σελίδας..... | 76 |
| Εικόνα 80. Μηνύματα SMS από θερμοκήπιο | 76 |
| Εικόνα 81. Διάγραμμα ροής Python..... | 77 |
| Εικόνα 82. Διαχείριση μηνυμάτων από την Python..... | 78 |

Κατάλογος Πινάκων

| | |
|---------------------------------------------------------------------|----|
| Πίνακας 1. Σύγκριση όγκου μηνύματος μεταξύ MQTT και HTTP | 33 |
| Πίνακας 2. Σύγκριση χρόνου αποστολής μηνύματος..... | 33 |
| Πίνακας 3. Ομοιότητες και διαφορές μεταξύ MQTT και HTTP | 33 |
| Πίνακας 4. Σύγκριση MQTT και AMQP | 34 |
| Πίνακας 5. Σύγκριση MQTT και CoAP..... | 35 |
| Πίνακας 6. Σύγκριση MQTT και ZeroMQ..... | 37 |
| Πίνακας 7. Χαρακτηριστικά του DHT22 | 38 |
| Πίνακας 8. Χαρακτηριστικά του MG-811 | 43 |
| Πίνακας 9. Χαρακτηριστικά του MIC29302WT | 47 |
| Πίνακας 10. Χαρακτηριστικά του LF33CV | 48 |
| Πίνακας 11. Ηλεκτρικά χαρακτηριστικά του ST7335 | 49 |
| Πίνακας 12. Ηλεκτρικά χαρακτηριστικά του τροφοδοτικού MeanWell..... | 49 |
| Πίνακας 13. Ηλεκτρικά χαρακτηριστικά του SIM800L | 50 |
| Πίνακας 14. Pinout του SIM800L..... | 51 |
| Πίνακας 15. Μοτίβο για την περιστροφή του βηματικού κινητήρα..... | 53 |
| Πίνακας 16. Υλικά της πλακέτας | 56 |
| Πίνακας 17. Περιγραφή λειτουργίας των pin..... | 61 |

Συντομογραφίες

| | |
|----------|----------------------------------------------------------------------|
| IoT | Internet of Things |
| MQTT | Message Queuing Telemetry Transport |
| QoS | Quality of Service |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| PHP | Personal Home Page / Hypertext Preprocessor |
| SQL | Structured Query Language |
| CONNACK | Connection Acknowledgement |
| SUBACK | Subscribe Acknowledgement |
| UNSUBACK | Unsubscribe Acknowledgement |
| PUBACK | Publish Acknowledgement |
| PUBREC | Publish Received |
| PUBREL | Publish Release |
| PUBCOMP | Publish Complete |
| PINGREQ | Ping Request |
| PINGRESP | Ping Response |
| LWT | Last Will and Testament |
| JSON | JavaScript Object Notation |
| DUP | Duplicate |
| PLC | Programmable Logic Controller |
| OASIS | Organization for the Advancement of Structured Information Standards |
| IBM | International Business Machines |
| TCP/IP | Transmission Control Protocol / Internet Protocol |
| HTTP | Hypertext Transfer Protocol |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| AMQP | Advanced Message Queuing Protocol |
| CoAP | Constrained Application Protocol |
| UDP | User Datagram Protocol |
| API | Application Programming Interface |
| ADC | Analog to Digital Converter |
| SPI | Serial Peripheral Interface |
| SCK | Serial Clock |
| SDA | Serial Data |
| PWM | Pulse Width Modulation |
| CAN | Controller Area Network |
| BLE | Bluetooth Low Energy |

Κεφάλαιο 1ο: Το θερμοκήπιο

1.1 Εισαγωγή

Το θερμοκήπιο είναι ένα προστατευμένο περιβάλλον στο οποίο καλλιεργούνται και αναπτύσσονται φυτά. Είναι μια κατασκευή ειδικά σχεδιασμένη για να μπορεί να κρατάει σταθερές τις συνθήκες ή και να ρυθμίζει τη θερμοκρασία και την υγρασία στο εσωτερικό του. Υπάρχουν διάφοροι τύποι θερμοκηπίων, όμως όλα έχουν κατασκευαστεί για τον ίδιο σκοπό. Έχουν μεγάλες επιφάνειες από διαφανή υλικά, όπου αιχμαλωτίζουν το ηλιακό φως και συνεπώς τη θερμότητα. Στις μέρες μας, τα θερμοκήπια κατασκευάζονται με διάφορα υλικά. Αυτά είναι το ξύλο, το σίδερο και το πολυαιθυλένιο (το πιο κοινό πλαστικό). Τα θερμοκήπια ποικίλουν, μπορεί να είναι μικρά υπόστεγα έως και κτίρια βιομηχανικού μεγέθους. Τα μεγάλα θερμοκήπια είναι εξοπλισμένα με διάφορες ελεγχόμενες λειτουργίες, όπως ψύξη, θέρμανση, φωτισμός, εξαερισμός, κ.α.. Συνήθως ο έλεγχος τους γίνεται αυτοματοποιημένα από υπολογιστές για την αποδοτικότερη λειτουργία τους και συνεπώς να υπάρχουν οι βέλτιστες συνθήκες για καλλιέργεια.

1.2 Ιστορική αναδρομή

Ρωμαϊκή αυτοκρατορία

Πριν την ανάπτυξη των θερμοκηπίων οι καλλιέργειες περιορίζονταν από τις καιρικές συνθήκες. Οι άνθρωποι περιορίζονταν στα είδη και στην εποχή του έτους κατά την οποία μπορούσαν να καλλιεργήσουν φυτά. Γύρω στο 30 μ.Χ., η Ρωμαϊκή Αυτοκρατορία έκανε την πρώτη καταγεγραμμένη προσπάθεια να χτίσει ένα τεχνητό περιβάλλον. Λόγω της εξασθενημένης υγείας του αυτοκράτορα Τιβέριου, οι γιατροί συνέστησαν στον αυτοκράτορα να τρώει ένα αγγούρι την ημέρα. Όμως, επειδή τα αγγούρια δύσκολα αναπτύσσονται ολόκληρο τον χρόνο, οι Ρωμαίοι σχεδίασαν ένα περιβάλλον, όπως ένα θερμοκήπιο, για να έχουν διαθέσιμα αγγούρια ολόκληρο τον χρόνο για τον αυτοκράτορα. Τα φύτευαν σε καρότσια, όπου την ημέρα τα βγάζανε στον ήλιο και τη νύχτα τα πήγαιναν μέσα για να παραμείνουν ζεστά.

15^{ος} αιώνας Κορέα

Η Κορέα τον 15^ο αιώνα, κατά τη διάρκεια της δυναστείας του Joseon, ήταν αυτή που εφηύρε νέο σχεδιασμό για το θερμοκήπιο. Τη δεκαετία του 1450, ο Soon ui Jeon περιέγραψε στο χειρόγραφο του (με ονομασία Sangayogok) το πρώτο θερμοκήπιο το οποίο θερμαινόταν τεχνητά. Ο Soon Ui Jeon ήταν γιατρός της βασιλικής οικογένειας και το Sangayogok προοριζόταν να παρέχει σημαντικές γνώσεις σχετικά με τη γεωργία και τη καθαριότητα. Ο Soon ui Jeon είχε γράψει πως να κατασκευάσει ένα θερμοκήπιο με το οποίο θα μπορούσε να καλλιεργεί λαχανικά και άλλου είδους φυτά το χειμώνα. Η κορεατική σχεδίαση προσθέτει στη δομή του θερμοκηπίου ένα σύστημα θέρμανσης, το οποίο χρησιμοποιούταν σε οικιακούς χώρους. Αυτό το σύστημα είναι ένας σωλήνας όπου προχωράει κάτω από το δάπεδο που ξεκινάει από μια πηγή θερμότητας. Εκτός από αυτό το σύστημα υπήρχε και ένα καζάνι με νερό, το οποίο το θέρμαιναν με στόχο να αυξήσουν τη θερμοκρασία αλλά και την υγρασία στο θερμοκήπιο. Ήταν από τα πρώτα θερμοκήπια όπου μπορούσαν και έλεγχαν τη θερμοκρασία. Συνδύαζαν και άλλες μεθόδους, όπως ημιδιαφανή παράθυρα και στάχνα στους τοίχους προκειμένου να συγκρατηθεί η θερμότητα από τον ήλιο.

17^{ος} αιώνας

Τον 17^ο αιώνα η έννοια του θερμοκηπίου εμφανίστηκε στην Ολλανδία και ύστερα στην Αγγλία. Κάποιες από τις πρώτες προσπάθειες απαιτούσαν αρκετή εργασία για να κλείσουν την νύχτα. Υπήρχε σοβαρό πρόβλημα σχετικά με την θερμότητα. Το πρώτο θερμαινόμενο θερμοκήπιο στο Ηνωμένο Βασίλειο ολοκληρώθηκε στο Chelsea Physic Garden το 1681. Η Ολλανδία, ακόμα και σήμερα διαθέτει τα μεγαλύτερα θερμοκήπια, τα οποία μπορούν να παράγουν τεράστιες ποσότητες λαχανικών σε ετήσια βάση. Με την εξέλιξη της τεχνολογίας (καλύτερο γυαλί, βελτιωμένες τεχνικές κατασκευής), οι

πειραματισμοί στον σχεδιασμό θερμοκηπίων συνεχίστηκαν στην Ευρώπη. Το θερμοκήπιο στο Παλάτι των Βερσαλλιών ήταν ένα παράδειγμα (μήκος 150μ, πλάτος 13μ, ύψος 14μ).

19^{ος} αιώνας

Κατά τη διάρκεια του 1800, ο Γάλλος βοτανολόγος Charles Lucien Bonaparte ήταν υπεύθυνος του πρώτου πρακτικού σύγχρονου θερμοκηπίου στο Leiden της Ολλανδίας, για την καλλιέργεια φαρμακευτικών τροπικών φυτών. Χάρη την ανάπτυξη της επιστήμης της βοτανικής διαδόθηκε η έννοια του θερμοκηπίου στα πανεπιστήμια. Οι Γάλλοι, είχαν ονομάσει τα πρώτα τους θερμοκήπια ως πορτοκαλιές, επειδή τα χρησιμοποιούσαν για την προστασία των πορτοκαλιών από το κρύο. Έπειτα, αφού οι ανανάδες έγιναν δημοφιλείς, χτίστηκαν θερμοκήπια για ανανάδες.

19^{ος} αιώνας Αγγλία

Στην Αγγλία, κατά την Βικτοριανή εποχή, είχαν κατασκευαστεί τα μεγαλύτερα θερμοκήπια που έχουν σχεδιαστεί μέχρι σήμερα. Ως άμεσο αποτέλεσμα της αποικιακής επέκτασης, ο σκοπός των θερμοκηπίων άλλαξε από τη γεωργία στην κηπουρική. Η γρήγορη μεταφορά της γνώσης μεταξύ των αποικιών σχετικά με τα φυτά και τα κηπευτικά, συνέβαλε στη Βικτοριανή γοητεία. Τα θερμοκήπια είχαν γίνει θεάματα για να διασκεδάσουν τον κόσμο. Πιστεύεται ότι η συλλογή των φυτών ήταν οι αληθινές αντανάκλασεις του κόσμου, αλλά στην πραγματικότητα ήταν στερεότυπες ρυθμίσεις εξωτικών φυτών όπου συμβόλιζαν ακριβώς που βρίσκονταν οι βρετανικές αποικίες και πόσο μακριά έφτανε η εξουσία τους. Για να υποστηριχθεί η βρετανική εξουσία, τα θερμοκήπια είχαν γίνει επιχειρήματα αποικιακής εξουσίας. Μια ξεχωριστή σχεδίαση θερμοκηπίων τον 19^ο αιώνα ήταν αυτά με μεγάλο ύψος, ώστε να χωράνε και δέντρα. Συνήθως βρίσκονταν σε δημόσιους χώρους (κήπους και πάρκα) και αποτελούσαν παράδειγμα της ανάπτυξης της αρχιτεκτονικής από γυαλί και σίδηρο. Η τεχνολογία αυτή χρησιμοποιήθηκε σε σιδηροδρομικούς σταθμούς, αγορές, εκθεσιακούς χώρους και γενικότερα σε κτίρια όπου χρειαζόταν μεγάλος και ανοιχτός χώρος. Άλλα μεγάλα θερμοκήπια που κατασκευάστηκαν τον 19ο αιώνα περιελάμβαναν το Κρυστάλλινο Παλάτι της Νέας Υόρκης, το Glaspalast του Μονάχου και τα Βασιλικά Θερμοκήπια του Λάκεν (1874–1895) για τον βασιλιά Λεοπόλδο Β' του Βελγίου. Το 1880 κατασκευάστηκε στην Ιαπωνία από τον Samuel Cocking (Βρετανός έμπορος βοτάνων) το πρώτο θερμοκήπιο.

20^{ος} αιώνας

Τον 20^ο αιώνα, προστέθηκε σε πολλά θερμοκήπια ο γεωδαιτικός θόλος. Τέτοια παραδείγματα είναι, το Eden Project στην Κορνουάλη, το Ινστιτούτο Rodale στην Πενσυλβάνια, το Climatron στον Βοτανικό Κήπο του Μισούρι στο Σεντ Λούις και η Toyota Motor Manufacturing Kentucky. Ένα ακόμα δημοφιλές σχήμα για μεγάλα και ψηλά θερμοκήπια είναι και η πυραμίδα. Τη δεκαετία του 1960 όπου έγιναν ευρέως διαθέσιμα φαρδιά φύλλα πολυαιθυλενίου, προσαρμόστηκαν και οι δομές των θερμοκηπίων. Τα θερμοκήπια με σκελετό από κρίκους (σαν τούνελ) κατασκευάζονταν από διάφορες εταιρείες, αλλά και συχνά από τους ίδιους τους καλλιεργητές. Η εξέλιξη της κατασκευής των θερμοκηπίων μείωσε το κόστος σημαντικά, με αποτέλεσμα να κατασκευάζονταν όλο και περισσότερα. Τη δεκαετία του 1970, η προσθήκη πιο αποτελεσματικών αναστολών υπεριώδους ακτινοβολίας, έκανε πιο ανθεκτικά τα φιλμ πολυαιθυλενίου. Μετά το 1980 διαδόθηκαν και τα θερμοκήπια που έχουν δυνατότητα σύνδεσης με υδρορροές. Αυτά τα θερμοκήπια καλύπτονται με δομημένα πολυκαρβονικά υλικά ή με διπλό στρώμα μεμβράνης πολυαιθυλενίου όπου ανάμεσά τους υπάρχει αέρας και λειτουργεί ως μονωτικό για την αποτελεσματική θέρμανσή τους.

1.3 Βασικές λειτουργίες θερμοκηπίων

Παρακάτω αναφέρονται κάποιες σημαντικές λειτουργίες που πρέπει να διαθέτει ένα θερμοκήπιο.

Εξαερισμός

Ο εξαερισμός είναι μία από τις σημαντικότερες λειτουργίες που πρέπει να έχει ένα θερμοκήπιο. Αν δεν υπάρχει σωστός εξαερισμός, τότε τα φυτά εντός του θερμοκηπίου μπορεί να αποκτήσουν προβλήματα. Ο σκοπός του εξαερισμού είναι η ρύθμιση της θερμοκρασίας και της υγρασίας στο βέλτιστο επίπεδο για να αποφευχθεί η ανάπτυξη φυτικών παθογόνων. Επίσης, ο εξαερισμός διασφαλίζει την παροχή του φρέσκου αέρα, ο οποίος χρειάζεται για τη φωτοσύνθεση και την αναπνοή των φυτών.

Ψύξη

Η ψύξη είναι ακόμα μία σημαντική λειτουργία που πρέπει να διαθέτει ένα θερμοκήπιο. Τις ημέρες που υπάρχει υψηλή θερμοκρασία θα πρέπει το εσωτερικό του θερμοκηπίου να μένει δροσερό. Μπορεί να γίνει είτε χειροκίνητα είτε αυτοματοποιημένα. Συνήθως αυτή η λειτουργία γίνεται αυτοματοποιημένα όταν πρόκειται για μεγάλης έκτασης θερμοκήπια και όταν έχουν περίπλοκα συστήματα.

Θέρμανση

Η θέρμανση είναι ένα από τα μεγαλύτερα κόστη ενός θερμοκηπίου. Το πρόβλημα που προκύπτει με τη θερμοκρασία είναι ότι ενώ τα καλύμματα πρέπει να είναι λεπτά για να περνάει το ηλιακό φως, το οποίο είναι απαραίτητο για την φωτοσύνθεση και για την αύξηση της θερμοκρασίας στο εσωτερικό τους, δεν μπορούν να κάνουν καλή μόνωση με αποτέλεσμα να χάνεται σημαντική ποσότητα θερμότητας. Στα περισσότερα θερμοκήπια, όταν απαιτείται επιπλέον θερμότητα, επιτυγχάνεται είτε με χρήση φυσικού αερίου είτε με ηλεκτρικές αντιστάσεις.

Φωτισμός

Όπως είναι γνωστό, τα φυτά για να αναπτυχθούν πρέπει να κάνουν φωτοσύνθεση. Την ημέρα αυτό επιτυγχάνεται με το φως του ήλιου. Υπάρχουν όμως θερμοκήπια που είναι εξοπλισμένα με φώτα, τα οποία ανάβουν την νύχτα με αποτέλεσμα να αυξάνεται η απόδοση σε ορισμένες καλλιέργειες.

Διοξειδίου του άνθρακα

Τα οφέλη του διοξειδίου του άνθρακα (CO₂) σε ένα θερμοκήπιο σχετικά με την ενίσχυση της ανάπτυξης των φυτών είναι γνωστά περίπου εκατό χρόνια. Με τα σωστά επίπεδα του CO₂ μπορεί ακόμα και να μειωθεί η χρήση του νερού, αφού ένα μεγάλο ποσοστό χάνεται κατά την εξάτμιση. Όμως, όταν δεν είναι εφικτή η συγκέντρωση της απαραίτητης ποσότητας του CO₂, τότε πρέπει να γίνει ο ελεγχόμενος εμπλουτισμός του. [19]

1.4 Αντίστοιχες εργασίες-έρευνες

Αντίστοιχες εργασίες-έρευνες για συστήματα θερμοκηπίων υπάρχουν αρκετές στο διαδίκτυο. Παρακάτω γίνεται μια μικρή αναφορά μερικών τέτοιων εργασιών που έχουν υλοποιηθεί στο παρελθόν.

Ένα παρόμοιο σύστημα σαν της παρούσας εργασίας έχει υλοποιήσει και η Vijitha Ananthi, το οποίο έχει δημοσιευθεί το 2019 με τίτλο «Automation using IoT in greenhouse environment». Είναι ένα αυτοματοποιημένο σύστημα το οποίο μετράει θερμοκρασία και υγρασία με τον αισθητήρα DHT11, την υγρασία εδάφους με το αισθητήριο YL69, και την φωτεινότητα με τον αισθητήρα GL5528. Οι μετρήσεις αυτές επεξεργάζονται με το NETDUINO το οποίο ενεργοποιεί/απενεργοποιεί λειτουργίες στο θερμοκήπιο για να κρατάει σταθερές τις συνθήκες. Επίσης, οι μετρήσεις αυτές αποστέλλονται σε ένα cloud, όπου και αυτό με τη σειρά του στέλνει μηνύματα SMS στον διαχειριστή του θερμοκηπίου. [69]

Παρόμοιο σύστημα για θερμοκήπια παρουσιάζεται και στη δημοσίευση των Dia Mehdi Faris και Mahmood Basil Mahmood με τίτλο «Data acquisition of greenhouse using Arduino». Το σύστημά τους συλλέγει μετρήσεις όπως θερμοκρασία και υγρασία (μέσω του αισθητήρα DHT11), υγρασία εδάφους (μέσω του αισθητήρα YL69), μετράει τη φωτεινότητα καθώς και τη συγκέντρωση του Διοξειδίου του άνθρακα (CO₂) με τον αισθητήρα MG-811. Οι μετρήσεις συλλέγονται και αποστέλλονται σε υπολογιστή με το Arduino mega. Τέλος, η απεικόνιση των παραπάνω μετρήσεων γίνονται μέσω του προγράμματος LabVIEW. [70]

Οι μηχανικοί Shatadru Bipasha Biswas και M. Tariq Iqbal στη δημοσίευση «Solar water pumping system control using a low cost ESP32 microcontroller» παρουσιάζουν ένα αυτοματοποιημένο σύστημα άντλησης νερού το οποίο λειτουργεί με ηλιακά πάνελ. Το σύστημα αυτό συλλέγει με τον μικροελεγκτή ESP32 μετρήσεις όπως θερμοκρασία και υγρασία (μέσω του αισθητήρα DHT11) και επίσης την υγρασία εδάφους. Στη μονάδα ESP32 συνδέεται και ένας αισθητήρας όπου παρακολουθεί τη στάθμη του νερού σε μια δεξαμενή. Η μονάδα του ESP32 με βάση κάποιες προκαθορισμένες τιμές αποφασίζει πότε να αντλήσει νερό για να γεμίσει την δεξαμενή. Τέλος, οι μετρήσεις αυτές απεικονίζονται σε έναν server ο οποίος είναι εγκατεστημένος στο ESP32 και ο χρήστης μπορεί από εκεί να επέμβει στη λειτουργία της αντλίας.[66]

Ένα ακόμα παρόμοιο σύστημα που βασίζεται στην ίδια λογική όπως η παρούσα εργασία είναι αυτό των Kunja Bihari Swain, G. Santamanyu και Amiya Ranjan Senapati με τίτλο «Smart industry pollution monitoring and controlling using LabVIEW based IoT». Το σύστημα τους προσφέρει λύση για απομακρυσμένη παρακολούθηση αισθητήρων. Μετράνε τη θερμοκρασία και την υγρασία (μέσω του αισθητήρα DHT11), τη συγκέντρωση του διοξειδίου του άνθρακα (με τον αισθητήρα MG-811) και το pH. Οι μετρήσεις γίνονται με τη χρήση Arduino όπου η διαχείριση αυτού γίνεται μέσω του προγράμματος LabVIEW. Όλες οι μετρήσεις αποστέλλονται σε μια βάση δεδομένων MySQL και απεικονίζονται σε μια ιστοσελίδα όπου από εκεί μπορεί να γίνει διαχείριση κάποιων λειτουργιών. Τέλος, σχεδίασαν και μια εφαρμογή σε Android περιβάλλον για να προσφέρει το σύστημά τους μεγαλύτερη ευελιξία.[67]

1.5 Επίλογος

Σε αυτό το κεφάλαιο έγινε αρχικά μια μικρή εισαγωγή σχετικά με την έννοια του θερμοκηπίου και ύστερα η ιστορική αναδρομή αυτών και πώς εξελίχθηκαν με την πάροδο των ετών. Έπειτα παρουσιάστηκαν κάποιες βασικές λειτουργίες που πρέπει να εκτελεί ένα θερμοκήπιο. Τέλος, αναφέρθηκαν κάποιες αντίστοιχες εργασίες και δημοσιεύσεις που έχουν γίνει μέχρι σήμερα.

Κεφάλαιο 2ο: Πρωτόκολλο MQTT

2.1 Εισαγωγή

Το MQTT είναι πλέον ευρέως το πιο συχνά χρησιμοποιούμενο πρωτόκολλο για ανταλλαγή μηνυμάτων στο Διαδίκτυο των Πραγμάτων (Internet of Things, IoT) μεταξύ Server-Client. Το MQTT σημαίνει Message Queuing Telemetry Transport και είναι ένα πρωτόκολλο ανταλλαγής μηνυμάτων για δίκτυα κυρίως με μικρό εύρος ζώνης. Είναι ένα ελαφρύ και απλό πρωτόκολλο, το οποίο έχει σχεδιαστεί με τέτοιο τρόπο έτσι ώστε να είναι εύκολο στην εφαρμογή του. Βασίζεται στο δυαδικό σύστημα το οποίο κάνει τα πακέτα μηνυμάτων πολύ μικρά, το οποίο έχει ως αποτέλεσμα να είναι ελαφριά και επομένως πολύ γρήγορα στη μεταφορά τους σε σχέση με άλλα πρωτόκολλα. Αποτελείται από ένα σύνολο κανόνων που καθορίζει τον τρόπο με τον οποίο οι συσκευές μπορούν να επικοινωνούν μεταξύ τους. Χρησιμοποιείται για ανταλλαγή μηνυμάτων μεταξύ του IoT, όπως για παράδειγμα οι ενσωματωμένες συσκευές, αισθητήρια, βιομηχανικά PLC κ.α..

Το πρωτόκολλο αυτό βασίζεται σε συμβάντα και συνδέει τις συσκευές χρησιμοποιώντας το μοτίβο Publish/Subscribe. Ο Publisher (αποστολέας) και ο Subscriber (παραλήπτης) ενός μηνύματος επικοινωνούν μέσω των Topics (θεμάτων). Αυτό σημαίνει ότι οι δύο αυτοί δεν έχουν άμεση σύνδεση μεταξύ τους. Αυτή η μεταξύ τους σύνδεση επιτυγχάνεται μέσω του MQTT Broker. Με τη σειρά του ο MQTT Broker φιλτράρει όλα τα μηνύματα που φτάνουν σε αυτόν και τα διανέμει στους κατάλληλους Subscribers.

Όσα αναφέρθηκαν έως εδώ είναι μια γενική εικόνα του πρωτόκολλου MQTT. Παρακάτω γίνεται αναφορά στην ιστορική αναδρομή για το πρωτόκολλο αυτό και στη συνέχεια αναλύεται το πως λειτουργεί.

2.2 Ιστορική αναδρομή

Το πρωτόκολλο MQTT δημιουργήθηκε το 1999 από τους Arlen Nipper και Andy Stanford-Clark. Εφαρμόστηκε αρχικά για την παρακολούθηση αγωγών πετρελαίου εντός ενός συστήματος βιομηχανικού ελέγχου SCADA. Επειδή οι συσκευές ήταν συνδεδεμένες μέσω δορυφόρου και οι δορυφορικές επικοινωνίες την τότε εποχή ήταν πολύ ακριβές, ο Arlen και ο Andy χρειάζονταν ένα πρωτόκολλο επικοινωνίας το οποίο θα είχε στόχο την χαμηλότερη δυνατή κατανάλωση της μπαταρίας, αλλά και ταυτόχρονα αποδοτικό στο εύρος ζώνης. Τα χαρακτηριστικά του πρωτόκολλου είναι τα εξής:

- Ποιότητα της υπηρεσίας κατά την ανταλλαγή μηνυμάτων (Quality of Service, QoS)
- Αποτελεσματικό και ελαφρύ
- Διαγνωστικά δεδομένων
- Απλό στη χρήση
- Συνεχής επίγνωση της συνεδρίας

Αν και οι στόχοι αυτοί εξακολουθούν να υπάρχουν στο MQTT, η αρχική του σχεδίαση έχει αλλάξει από τα ενσωματωμένα συστήματα για εφαρμογές IoT. Αυτή η αλλαγή στη σχεδίαση είχε προκαλέσει σύγχυση σχετικά με τον ορισμό του MQTT. Η απάντηση όμως ήταν ότι το MQTT είναι η ονομασία του πρωτόκολλου, και όχι ακρωνύμιο. Το MQTT σημαίνει «Message Queuing Telemetry Transport».

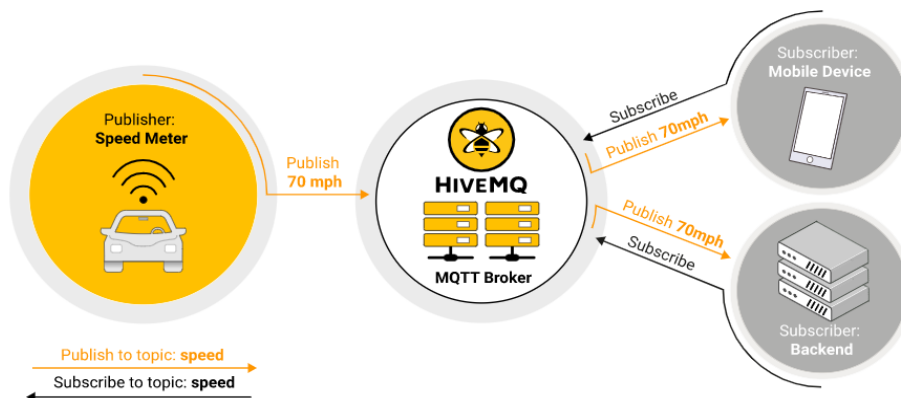
Το «MQ» παραπέμπει στη MQ Series. Είναι ένα προϊόν που αναπτύχθηκε από την IBM για να υποστηρίζει τη μεταφορά τηλεμετρίας MQ. Όταν ο Arlen και ο Andy υλοποίησαν το πρωτόκολλο, το ονόμασαν από το προϊόν της IBM. Αρκετές πηγές χαρακτηρίζουν λανθασμένα το MQTT ως

πρωτόκολλο ουράς μηνυμάτων. Γεγονός το οποίο δεν ισχύει. Την επόμενη δεκαετία, η IBM χρησιμοποίησε το πρωτόκολλο για εσωτερικό σκοπό. Το 2010 κυκλοφόρησε η έκδοση MQTT 3.1, ως έκδοση χωρίς δικαιώματα. Και από τότε η χρήση του είναι ελεύθερη για όλους.

Τρία χρόνια αργότερα, ανακοινώθηκε ότι το MQTT θα τυποποιηθεί από την OASIS. Η OASIS είναι ένας οργανισμός με στόχο την προώθηση προτύπων. Τα AMQP, SAML και DocBook είναι κάποια πρότυπα που είχαν κυκλοφορήσει από την OASIS. Η διαδικασία της τυποποίησης κράτησε περίπου έναν χρόνο. Στις 29 Οκτωβρίου 2014 το MQTT έγινε πλέον επισήμως εγκεκριμένο πρότυπο της OASIS. Η αλλαγή της έκδοσης από την 3.1 σε 3.1.1 σημαίνει ότι έχουν γίνει κάποιες βελτιώσεις. Στις 7 Μαρτίου του 2019, η OASIS επικύρωσε τη νέα έκδοση του MQTT, το MQTT 5. Σε αυτήν την βελτιωμένη έκδοση έχουν εισαχθεί νέες δυνατότητες. Οι δυνατότητες αυτές μπορούν να καλύψουν τις εφαρμογές IoT που απαιτούν περισσότερη αξιοπιστία και διαχείριση των σφαλμάτων σε κρίσιμες περιπτώσεις. [20]

2.3 Το μοντίβο Publish / Subscribe

Το μοντέλο του Publish-Subscribe στο MQTT διαφέρει από την αρχιτεκτονική του κλασικού μοντέλου Server-Client. Στο μοντέλο του Server-Client ο Client επικοινωνεί απευθείας με το τελικό σημείο. Αντίθετα, στο μοντέλο του Publish-Subscribe οι Clients δεν επικοινωνούν ποτέ απευθείας μεταξύ τους. Δηλαδή ο Publisher δεν έχει άμεση σύνδεση με τον Subscriber. Για την ακρίβεια, δεν γνωρίζουν καν ότι ο άλλος υπάρχει. Οπότε τη μεταξύ τους επικοινωνία τη διαχειρίζεται ένα τρίτο στοιχείο, ο Broker. Μια από τις σημαντικότερες δουλειές που έχει να κάνει ένας Broker είναι να φιλτράρει όλα τα μηνύματα που λαμβάνει και τη συνέχεια να τα διανέμει στους σωστούς Subscribers.



Εικόνα 1. Το πρωτόκολλο MQTT

Το σημαντικότερο σημείο αυτού του μοντέλου είναι η αποσύνδεση του Publisher από τον Subscriber. Αυτή η αποσύνδεση μπορεί να γίνει με τους εξής 3 τρόπους:

- **Χωρική αποσύνδεση:** Ο Publisher και ο Subscriber δεν επικοινωνούν άμεσα μεταξύ τους. Παρεμβάλλεται ανάμεσά τους ο Broker.
- **Χρονική αποσύνδεση:** Ο Publisher και ο Subscriber δεν είναι αναγκαίο να λειτουργούν ταυτόχρονα. Αν και τα μηνύματα παραδίδονται σχεδόν σε πραγματικό χρόνο, ο Broker έχει τη δυνατότητα να αποθηκεύσει τα μηνύματα για τους Subscribers που είναι αποσυνδεδεμένοι (offline) και να τα στείλει μόλις επανασυνδεθούν. Για να γίνει αυτό το Quality of Service (QoS) πρέπει να είναι μεγαλύτερο του μηδέν.
- **Αποσύνδεση συγχρονισμού:** Δεν χρειάζεται να διακόπτονται-μπλοκάρονται οι λειτουργίες και στα δύο μέλη κατά την αποστολή και λήψη μηνυμάτων. Αυτό γίνεται επειδή οι περισσότερες

βιβλιοθήκες των Clients βασίζονται σε επανακλήσεις (callbacks). Επομένως, η ροή των μηνυμάτων είναι συνήθως ασύγχρονη.

Φιλτράρισμα μηνυμάτων

Ο Broker είναι το στοιχείο που παίζει τον σημαντικότερο ρόλο στο πρωτόκολλο MQTT. Ο λόγος που τον καθορίζει ως το σημαντικότερο στοιχείο είναι ότι φιλτράρει τα μηνύματα που λαμβάνει, έτσι ώστε κάθε Subscriber να λάβει το σωστό μήνυμα. Ο πιο διαδεδομένος και εύκολος τρόπος φιλτραρίσματος των μηνυμάτων, είναι το φιλτράρισμα με βάση το Topic. Όταν ένας Client κάνει Subscribe σε ένα Topic στον Broker για να λαμβάνει τα μηνύματα αυτού του Topic, ο Broker είναι υπεύθυνος να διαβιβάζει στον Subscriber κάθε μήνυμα που δέχεται από τον Publisher σχετικά με αυτό το Topic. [21]

2.4 Client, Broker και Συνδέσεις

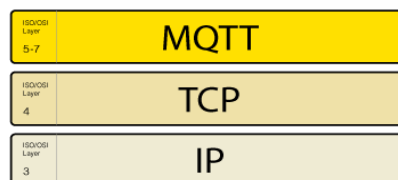
Client

Με τον όρο Client αναφερόμαστε πάντοτε σε έναν MQTT Client. Ενώ με τον όρο MQTT Client εννοούμε και τον Publisher αλλά και τον Subscriber. Ο Publisher είναι αυτός που δημοσιεύει/στέλνει κάποιο μήνυμα προς τον Broker, ενώ Subscriber είναι αυτός που λαμβάνει τα μηνύματα από τον Broker. Ο ίδιος Client μπορεί να είναι ταυτόχρονα Publisher και Subscriber. Ένας Client μπορεί να είναι μια οποιαδήποτε συσκευή (π.χ. server, μικροελεγκτής), ο οποίος χρησιμοποιεί την βιβλιοθήκη MQTT για να συνδεθεί με τον Broker μέσω του διαδικτύου. Είναι ουσιαστικά μια συσκευή η οποία επικοινωνεί με το πρωτόκολλο MQTT. Η ευκολία εφαρμογής του MQTT και η διάθεσή του σε πολλές γλώσσες προγραμματισμού (όπως για παράδειγμα C, Java, Javascript, Python, PHP, κ.α.) είναι ο σημαντικότερος λόγος όπου το πρωτόκολλο αυτό θεωρείται το ιδανικότερο για μικρές συσκευές (π.χ. IOT συσκευές).

Broker

Με τον όρο Broker αναφερόμαστε πάντοτε σε έναν MQTT Broker. Ενώ με τον όρο MQTT Broker εννοούμε το στοιχείο το οποίο είναι υπεύθυνο στο να διαχειρίζεται τα μηνύματα των Clients, να τα φιλτράρει έτσι ώστε να γνωρίζει σε ποιον να διαβιβάσει το αντίστοιχο μήνυμα, τον προσδιορισμό του ποιος Client έχει κάνει Subscribe στο αντίστοιχο Topic. Είναι ουσιαστικά ο πυρήνας του πρωτόκολλου. Ένας κεντρικός κόμβος όπου πρέπει να περάσει κάθε μήνυμα. Διατηρεί στοιχεία από τις συνδέσεις όλων των Clients. Επίσης ο Broker έχει ευθύνη στο να ελέγχει την ταυτότητα του Client κατά τη μεταξύ τους σύνδεση. Είναι σημαντικός ο έλεγχος αυτός, επειδή ο Broker είναι συνδεδεμένος και εκτεθειμένος στο διαδίκτυο.

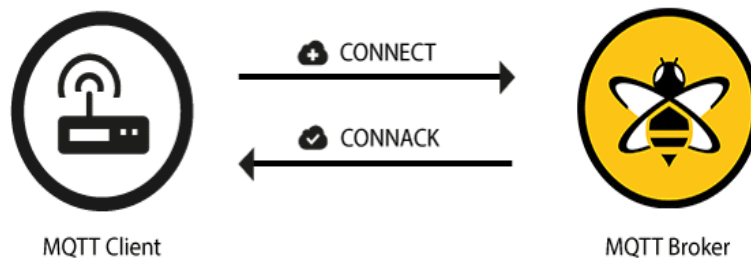
Αφού ο Broker είναι υπεύθυνος για τέτοιου είδους διεργασίες, μπορούμε να καταλάβουμε ότι η περίπτωση αύξησης του αριθμού των συνδέσεων (π.χ. εκατομμύρια συνδέσεις) αποτελεί μια πρόκληση. Για να γίνει αυτό πρέπει να χρησιμοποιηθούν περισσότεροι από ένας Broker. Τέτοιες καταστάσεις αντιμετωπίζονται ομαδοποιώντας τους Brokers και χρησιμοποιώντας εξισορροπητές φορτίου.



Εικόνα 2. Το MQTT βασίζεται στο TCP/IP

Σύνδεση στο MQTT

Το πρωτόκολλο MQTT στηρίζεται στο TCP/IP. Η σύνδεση στο MQTT γίνεται πάντα μεταξύ του Client και του Broker. Δύο Clients δεν συνδέονται ποτέ απευθείας μεταξύ τους. Μεσολαβεί ενδιάμεσα ο Broker. Για να ξεκινήσει μια σύνδεση, ο Client στέλνει ένα μήνυμα CONNECT στον Broker. Αν αυτή η εντολή δεν έχει σωστή σύνταξη ή περάσει αρκετό χρονικό διάστημα χωρίς αποστολή μηνύματος, τότε ο Broker απλά κλείνει τη σύνδεση. Αν ο Broker λάβει το μήνυμα CONNECT τότε ο Broker απαντάει με το μήνυμα CONNACK και έναν κωδικό κατάστασης. Όταν πραγματοποιηθεί η σύνδεση, τότε αυτή μένει ενεργή μέχρι ένας από τους δύο να τη διακόψει. Ο Client από την πλευρά του στέλνοντας μήνυμα αποσύνδεσης και ο Broker απλά κλείνει τη σύνδεση από την δική του πλευρά αν δεν λάβει κάποιο μήνυμα σε εύλογο χρονικό διάστημα.



Εικόνα 3. Σύνδεση του Client στον Broker

Το μήνυμα CONNECT περιλαμβάνει κάποια χαρακτηριστικά τα οποία αναλύονται σε άλλη ενότητα. Όμως παρακάτω αναφέρονται συνοπτικά, και αυτά είναι τα εξής:

- **ClientId**, το αναγνωριστικό του Client στον Broker και είναι μοναδικό.
- **Clean session flag**, σημαία για λειτουργία καθαρής περιόδου.
- **Username/Password**, για έλεγχο ταυτότητας στον Broker.
- **Will Message (LWT)**, το τελευταίο επιθυμητό μήνυμα σε περίπτωση ανορθόδοξης αποσύνδεσης.
- **Keep Alive**, μέγιστο χρονικό διάστημα που μεσολαβεί μεταξύ Client-Broker χωρίς να ανταλλάξουν μήνυμα, αλλιώς διακόπτεται η σύνδεση.
- **Session present flag**, αν υπάρχει διαθέσιμη μόνιμη περίοδο λειτουργίας.
- **Return code**, η κατάσταση της σύνδεσης (επιτυχής ή όχι).

Οι επιστρεφόμενοι κωδικοί στο πρωτόκολλο MQTT είναι οι εξής:

- **Return code 0**, επιτυχής σύνδεση.
- **Return code 1**, αποτυχημένη σύνδεση, μη αποδεκτό πρωτόκολλο.
- **Return code 2**, αποτυχημένη σύνδεση, μη αποδεκτός Client.
- **Return code 3**, αποτυχημένη σύνδεση, μη διαθέσιμος server.
- **Return code 4**, αποτυχημένη σύνδεση, λάθος όνομα και κωδικός.
- **Return code 5**, αποτυχημένη σύνδεση, μη εξουσιοδοτημένο. [22]

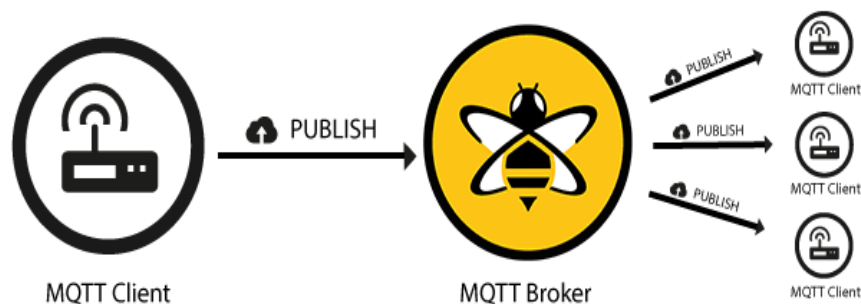
2.5 Publish, Subscribe και Unsubscribe

Publish

Μόλις επιτευχθεί η σύνδεση μεταξύ του Publisher και του Broker, τότε ο Publisher έχει τη δυνατότητα να ξεκινήσει να στέλνει μηνύματα. Τα μηνύματα αυτά φιλτράρονται με βάση τα Topics. Έτσι ο Broker μπορεί εύκολα να γνωρίζει σε ποιον θα διαβιβάσει το αντίστοιχο μήνυμα. Κάθε μήνυμα (payload) περιέχει τα δεδομένα σε μορφή byte. Η δομή του μηνύματος καθορίζεται αποκλειστικά από τον Publisher, η οποία μπορεί να έχει τη μορφή κειμένου, δυαδικού συστήματος ή ακόμα και μορφής JSON. Το μήνυμα που στέλνει ο Publisher έχει αρκετά χαρακτηριστικά. Παρακάτω παρουσιάζονται αυτά τα χαρακτηριστικά συνοπτικά, αλλά αργότερα θα αναλυθούν με περισσότερη λεπτομέρεια. Αυτά είναι τα εξής:

- **Αναγνωριστικό πακέτου (Packet Identifier):** Το αναγνωριστικό του πακέτου προσδιορίζει τη μοναδικότητα ενός μηνύματος καθώς ο Publisher και ο Broker επικοινωνούν. Αυτό γίνεται για να μην χαθούν ή μπλεχτούν τα μηνύματα.
- **Topic:** Το Topic είναι μια απλή συμβολοσειρά η οποία περιέχει κάθετους για να ξεχωρίζονται τα επίπεδα (π.χ. `tospitimou/dwmatio1/thermokrasia`).
- **Ποιότητα υπηρεσίας (Quality of Service – QoS):** Αυτό το χαρακτηριστικό παίρνει τιμές 1, 2, ή 3 και καθορίζει την ποιότητα της υπηρεσίας κατά την επικοινωνία.
- **Ωφέλιμο φορτίο (Payload):** Το Payload είναι το περιεχόμενο του μηνύματος.
- **Retain Flag:** Αυτό το χαρακτηριστικό καθορίζει αν ο Broker θα αποθηκεύσει το τελευταίο (πιο πρόσφατο) μήνυμα που έλαβε. Έτσι ώστε όταν συνδεθεί νέος Subscriber τότε να του στείλει αυτό το μήνυμα.
- **DUP flag:** Αυτό το χαρακτηριστικό καθορίζει αν ένα μήνυμα έχει σταλθεί ξανά σε περίπτωση που ο παραλήπτης δεν έλαβε το αρχικό.

Όταν ο Publisher στέλνει ένα μήνυμα, το μόνο που τον ενδιαφέρει είναι να παραδοθεί το μήνυμα στον Broker. Δεν τον ενδιαφέρει και ούτε παίρνει πληροφορίες για αν ο Broker παρέδωσε το μήνυμα ή πόσοι Subscribers υπάρχουν. Με την σειρά του ο Broker είναι υπεύθυνος να παραδώσει αυτό το μήνυμα στους κατάλληλους Subscribers.



Εικόνα 4. Το μήνυμα PUBLISH

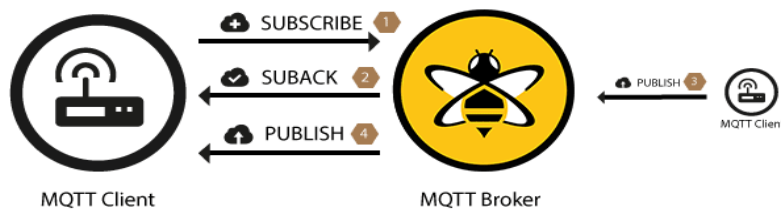
Subscribe

Ένας Subscriber για να λαμβάνει τα μηνύματα που στέλνει ο Publisher, θα πρέπει να κάνει Subscribe στο Topic που έχει κάνει Publish ο Publisher (δηλαδή ο αποστολέας). Το Topic βρίσκεται στον Broker, οπότε ο Publisher και ο Subscriber επικοινωνούν μέσω αυτού. Αν δεν υπάρχει κανένας Subscriber σε κάποιο Topic, τότε τα μηνύματα που στέλνει ο Publisher σε αυτό το Topic δεν έχουν καμία αξία. Επομένως, ένας Client για να κάνει Subscribe σε ένα Topic, πρέπει να στείλει το μήνυμα Subscribe στον Broker. Το μήνυμα αυτό περιέχει το Topic και το QoS (Quality of Service). Μία ευθύνη που έχει ο Broker σε αυτό το σημείο είναι να έχει ευθύνη του να ξεχωρίζει τα μηνύματα των Clients. Δηλαδή κάθε μήνυμα έχει το αναγνωριστικό κωδικό (αναγνωριστικό πακέτου).

Suback

Το Suback προκύπτει από το Subscribe Acknowledgement και είναι το μήνυμα που στέλνει ο Broker στον Subscriber για να του επιβεβαιώσει ότι το Subscribe έγινε με επιτυχία. Το μήνυμα αυτό αποτελείται από το αναγνωριστικό πακέτου του αρχικού μηνύματος που έκανε Subscribe ο Client, για αναγνώριση και διαχωρισμό του μηνύματος και ακόμα έναν επιστρεφόμενο κωδικό (return code). Ο επιστρεφόμενος αυτός κωδικός είναι ξεχωριστός για κάθε Subscribe που έχει γίνει, και καθορίζει το Quality of Service (QoS). Εφόσον ολοκληρωθούν με επιτυχία τα μηνύματα Subscribe και Suback, τότε ο Subscriber θα αρχίσει να λαμβάνει κάθε μήνυμα που φτάνει στον Broker στο αντίστοιχο Topic.

- **Return Code 0**, επιτυχία και το QoS είναι 0.
- **Return Code 1**, επιτυχία και το QoS είναι 1.
- **Return Code 2**, επιτυχία και το QoS είναι 2.
- **Return Code 128**, αποτυχία στο Subscribe.



Εικόνα 5. Η διαδικασία του PUBLISH/SUBSCRIBE

Unsubscribe

Το μήνυμα Unsubscribe είναι το αντίστοιχο μήνυμα του Subscribe όταν ένας Client επιθυμεί να διαγραφεί από ένα Topic που βρίσκεται στον Broker και να σταματήσει πλέον να λαμβάνει μηνύματα. Επομένως και αυτό το μήνυμα περιέχει το αναγνωριστικό πακέτου και μια λίστα από Topic από τα οποία επιθυμεί να κάνει Unsubscribe ο Client. Σε αυτήν την περίπτωση δεν αποστέλλεται το QoS, αρκεί μόνο το Topic. Έτσι ο Broker με τη σειρά του καταργεί και αυτός το Subscribe του Client στο αντίστοιχο Topic και δεν θα του διαβιβάζει πλέον τα μηνύματα.

Unsuback

Το μήνυμα Unsuback προκύπτει από το Unsubscribe Acknowledgement και είναι το μήνυμα που στέλνει ο Broker στον Client για να του επιβεβαιώσει ότι το Unsubscribe έγινε με επιτυχία. Το μήνυμα αυτό περιέχει μονάχα το αναγνωριστικό πακέτου του αρχικού μηνύματος Unsubscribe. Εφόσον ολοκληρωθούν με επιτυχία τα μηνύματα Unsubscribe και Unsuback, τότε ο Client μπορεί να υποθέσει ότι έχει διαγραφεί από το Topic στο οποίο λάμβανε μηνύματα. [23]



Εικόνα 6. Η διαδικασία του UNSUBSCRIBE

2.6 Χαρακτηριστικά πρωτόκολλου MQTT

2.6.1 Topics

Στο MQTT, η λέξη Topic αποτελεί μια συμβολοσειρά που χρησιμοποιεί ο Broker για να φιλτράρει τα μηνύματα των clients. Ένα Topic μπορεί να αποτελείται από ένα ή περισσότερα θεματικά επίπεδα. Κάθε επίπεδο του Topic διαχωρίζεται με μια κάθετο προς τα εμπρός. Πρέπει να σημειωθεί ότι κάθε Topic θα πρέπει να έχει τουλάχιστον έναν χαρακτήρα. Παρακάτω ακολουθούν παραδείγματα έγκυρων Topics:

- Topic 1: **tospitimu/dwmatio1/lampa**
- Topic 2: **tospitimu/dwmatio1/thermokrasia**
- Topic 3: **tospitimu/dwmatio1/ugrasia**
- Topic 4: **tospitimu/dwmatio2/lampa**

Τα Topics είναι case-sensitive. Δηλαδή, τα πεζά γράμματα διαφέρουν από τα κεφαλαία. Για παράδειγμα, τα παρακάτω δύο Topics που ακολουθούν είναι διαφορετικά.

- Topic 1: **tospitimu/dwmatio1/lampa**
- Topic 2: **Tospitimu/dwmatio1/lampa**

Όταν ένας Client κάνει Subscribe σε ένα Topic, μπορεί να κάνει Subscribe ακριβώς και μόνο στο Topic που τον ενδιαφέρει για κάποιο μήνυμα ή μπορεί να χρησιμοποιήσει χαρακτήρες μπαλαντέρ (wildcards) για να κάνει Subscribe σε πολλά Topics ταυτόχρονα. Διευκρινίζεται ότι οι χαρακτήρες μπαλαντέρ χρησιμοποιούνται μόνο για Subscribe και όχι για Publish. Υπάρχουν δύο είδη μπαλαντέρ: α)μονού-επιπέδου (+) και β)πολλαπλών-επιπέδων (#).

Όπως φαίνεται και από το όνομά του, ο μπαλαντέρ μονού-επιπέδου (+) αντικαθιστά μόνο ένα επίπεδο Topic. Για παράδειγμα, παρακάτω το Topic 1 αντικαθιστά τα Topic 2 και Topic 3. Όχι όμως το Topic 4. Έτσι ο Client κάνει ευκολότερα Subscribe σε περισσότερα Topics.

- Topic 1: **tospitimu/+/lampa**
- Topic 2: **tospitimu/dwmatio1/lampa**
- Topic 3: **tospitimu/dwmatio2/lampa**
- Topic 4: **tospitimu/dwmatio1/thermokrasia**

Αντίθετα, ο μπαλαντέρ πολλαπλών-επιπέδων (#) αντικαθιστά πολλά επίπεδα του Topic. Οπότε ο χαρακτήρας hash (#) τοποθετείται ως ο τελευταίος χαρακτήρας του Topic. Για παράδειγμα, το Topic 1 αντικαθιστά τα Topic 2 και Topic 3. Έτσι ο Client κάνει ευκολότερα Subscribe σε περισσότερα Topics.

- Topic 1: **tospitimu/dwmatio1/#**
- Topic 2: **tospitimu/dwmatio1/lampa**
- Topic 3: **tospitimu/dwmatio1/thermokrasia**

Τα Topics που ξεκινούν με το σύμβολο \$ έχουν διαφορετικό σκοπό. Χρησιμοποιούνται για εσωτερικά στατιστικά στοιχεία του Broker. Οι Clients δεν μπορούν να κάνουν Publish σε αυτά τα Topics. Παρακάτω ακολουθούν κάποια παραδείγματα τέτοιου είδους Topic.

- **\$\$SYS/broker/clients/connected**
- **\$\$SYS/broker/clients/disconnected**
- **\$\$SYS/broker/messages/sent [24]**

2.6.2 Quality of Service (QoS)

Επειδή το πρωτόκολλο MQTT διαχειρίζεται την αποστολή και λήψη μηνυμάτων, και εγγυάται την παράδοσή τους ακόμα και σε αναξιόπιστα δίκτυα, τότε η Ποιότητα της Υπηρεσίας (Quality of Service, QoS) είναι αυτή που κάνει την επικοινωνία αξιόπιστη. Είναι ένα από τα σημαντικότερα χαρακτηριστικά του MQTT. Δίνει στον Client τη δυνατότητα να επιλέξει το QoS που ταιριάζει ανάλογα με το πόσο αξιόπιστο είναι το δίκτυο ή και το είδος της εφαρμογής. Είναι ένα είδος συμφωνίας που κάνει ο αποστολέας και ο παραλήπτης ενός μηνύματος, η οποία ορίζει την εγγύηση της παράδοσης του μηνύματος. Υπάρχουν 3 επίπεδα QoS στο πρωτόκολλο MQTT:

- Το πολύ μία φορά (QoS 0)
- Τουλάχιστον μία φορά (QoS 1)
- Ακριβώς μία φορά (QoS 2)

Όταν γίνεται αναφορά για το QoS, τότε πρέπει να ληφθούν υπόψη και οι δύο πλευρές σχετικά με την παράδοση ενός μηνύματος:

- Αποστολή μηνύματος από τον Client στον Broker (Publisher).
- Αποστολή μηνύματος από τον Broker σε έναν Client (Subscriber).

Ο Publisher που στέλνει ένα μήνυμα στον Broker καθορίζει και το QoS αυτού του μηνύματος. Ο Broker με τη σειρά του, διανέμει αυτό το μήνυμα στους Subscribers, χρησιμοποιώντας το QoS που είχε ορίσει κάθε Subscriber. Αυτός ο καθορισμός γίνεται κατά την αποστολή του μηνύματος Subscribe προς τον Broker. Εάν ένας Subscriber ορίσει χαμηλότερο QoS από αυτό του Publisher, τότε ο Broker στέλνει το μήνυμα με το χαμηλότερο QoS. Παρακάτω αναλύεται κάθε QoS ξεχωριστά.

QoS 0 - το πολύ μία φορά

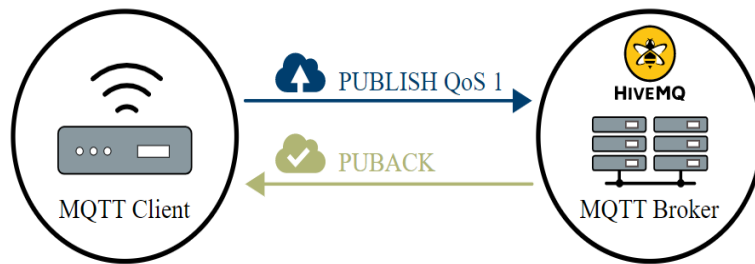
Το χαμηλότερο επίπεδο του QoS είναι το μηδέν. Σε αυτό το επίπεδο δεν υπάρχει εγγύηση παράδοσης του μηνύματος. Ο παραλήπτης δεν επιβεβαιώνει ότι παρέλαβε το μήνυμα και έτσι δεν αποθηκεύεται και δεν μεταδίδεται ξανά από τον αποστολέα. Το QoS 0 αποκαλείται συχνά ως «fire and forget».



Εικόνα 7. Quality of Service 0

QoS 1 - τουλάχιστον μία φορά

Στο QoS 1 υπάρχει εγγύηση ότι το μήνυμα παραδίδεται τουλάχιστον μία φορά στον παραλήπτη. Ο αποστολέας αποθηκεύει το μήνυμα μέχρι να λάβει το μήνυμα PUBACK (Publish Acknowledgement) από τον παραλήπτη, το οποίο επιβεβαιώνει τη λήψη του μηνύματος. Είναι πιθανό ένα μήνυμα να σταλεί αρκετές φορές.



Εικόνα 8. Quality of Service 1

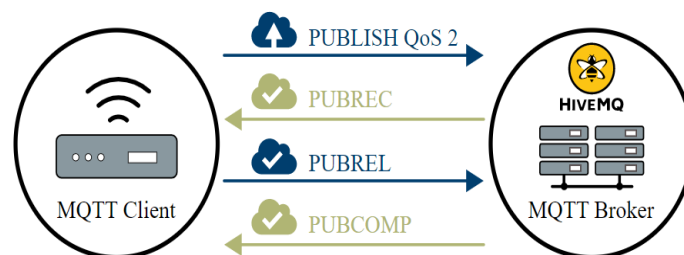
Ο αποστολέας χρησιμοποιεί το αναγνωριστικό του μηνύματος (ID) σε κάθε μήνυμα για να αντιστοιχίσει το μήνυμα PUBLISH με το αντίστοιχο μήνυμα PUBACK. Εάν ο αποστολέας δεν λάβει το μήνυμα PUBACK μέσα σε μικρό χρονικό διάστημα, τότε στέλνει ξανά το μήνυμα PUBLISH. Εάν όμως ένας παραλήπτης λάβει ένα μήνυμα με QoS 1, τότε μπορεί να το επεξεργαστεί αμέσως. Για παράδειγμα, εάν ο παραλήπτης είναι ο Broker, τότε ο Broker στέλνει το μήνυμα σε όλους τους Subscribers και στη συνέχεια απαντά με ένα πακέτο PUBACK.

Εάν ο Publisher στείλει ξανά το ίδιο μήνυμα, τότε ορίζει το flag διπλότυπου (DUP). Στο QoS 1, αυτό το flag DUP χρησιμοποιείται για εσωτερικό σκοπό μόνο και δεν υποβάλλεται σε επεξεργασία από τον Broker ή τον Client. Ανεξάρτητα από το flag DUP, ο παραλήπτης του μηνύματος στέλνει το PUBACK.

QoS 2 - ακριβώς μία φορά

Το QoS 2 είναι το υψηλότερο επίπεδο ποιότητας υπηρεσίας. Σε αυτό το επίπεδο υπάρχει η εγγύηση ότι κάθε μήνυμα λαμβάνεται ακριβώς μία φορά από τους παραλήπτες. Το QoS 2 είναι το ασφαλέστερο αλλά και το πιο αργό σε σχέση με τα δύο προηγούμενα. Είναι μια χειραγία τεσσάρων μερών μεταξύ αποστολέα-παραλήπτη. Ο αποστολέας και ο παραλήπτης χρησιμοποιούν το αναγνωριστικό του αρχικού μηνύματος PUBLISH για να συντονιστούν.

Όταν ένας παραλήπτης λαμβάνει ένα μήνυμα PUBLISH με QoS 2, επεξεργάζεται το μήνυμα και απαντά στον αποστολέα με το μήνυμα PUBREC (Publish Received). Εάν ο αποστολέας δεν λάβει το μήνυμα PUBREC από τον παραλήπτη, στέλνει ξανά το μήνυμα PUBLISH με το flag DUP μέχρι να λάβει επιβεβαίωση. Μόλις ο αποστολέας λάβει το μήνυμα PUBREC, ο αποστολέας μπορεί να απορρίψει με ασφάλεια το αρχικό μήνυμα PUBLISH. Έπειτα ο αποστολέας αποθηκεύει το μήνυμα PUBREC και απαντά στον παραλήπτη με το μήνυμα PUBREL (Publish Release).



Εικόνα 9. Quality of Service 2

Εφόσον ο παραλήπτης λάβει το μήνυμα PUBREL, τότε μπορεί να διαγράψει όλες τις αποθηκευμένες καταστάσεις και να απαντήσει στον αποστολέα με το μήνυμα PUBCOMP (Publish Complete). Το ίδιο ισχύει όταν ο αποστολέας λάβει το μήνυμα PUBCOMP. Μέχρι όμως ο παραλήπτης να ολοκληρώσει τις λειτουργίες του και να στείλει το μήνυμα PUBCOMP, αποθηκεύει μια αναφορά στο αναγνωριστικό

(ID) του αρχικού μηνύματος PUBLISH. Αυτό το σημείο είναι σημαντικό, έτσι ώστε να αποφευχθεί η δεύτερη επεξεργασία του μηνύματος. Αφού ο αποστολέας λάβει το πακέτο PUBCOMP, το αναγνωριστικό του μηνύματος γίνεται διαθέσιμο για να χρησιμοποιηθεί ξανά.

Όταν ολοκληρωθεί η ροή των μηνυμάτων στο QoS 2, και οι δύο πλευρές είναι σίγουρες ότι το μήνυμα έχει παραδοθεί. Στην περίπτωση που ένα μήνυμα χαθεί στην πορεία, ο αποστολέας είναι υπεύθυνος να στείλει ξανά το μήνυμα μέσα σε μικρό χρονικό διάστημα. Αυτό ισχύει είτε ο αποστολέας είναι Client είτε Broker. Είναι ευθύνη του παραλήπτη να απαντήσει σε κάθε μήνυμα αναλόγως. [25]

2.6.3 Persistent Session

Για να μπορεί να λαμβάνει μηνύματα από έναν Broker ένας Client, πρέπει να συνδεθεί με τον Broker κάνοντας Subscribe στα Topics. Εάν η σύνδεση μεταξύ του Client και του Broker διακοπεί κατά τη διάρκεια μιας μη μόνιμης συνεδρίας, αυτά τα Topics χάνονται και ο Client είναι αναγκασμένος να κάνει ξανά Subscribe. Κάθε φορά που διακόπτεται η σύνδεση, η επανεγγραφή επιβαρύνει τους Clients. Για να αποφευχθεί αυτό, ο Client μπορεί να ζητήσει μια μόνιμη περίοδο λειτουργίας όταν συνδέεται με τον Broker. Οι μόνιμες συνεδρίες αποθηκεύουν όλες τις πληροφορίες που έχουν σχέση με τον Client. Το Client-Id που παρέχει ο Client κατά την έναρξη μιας σύνδεσης με τον Broker προσδιορίζει τη συνεδρία.

Αν ένας Client είναι εκτός σύνδεσης και είχε ορίσει μόνιμη συνεδρία, όταν επανασυνδεθεί οι πληροφορίες είναι άμεσα διαθέσιμες. Οι πληροφορίες που αποθηκεύει ο Broker είναι οι εξής:

- Αν υπάρχει συνεδρία.
- Όλα τα Subscribe που έχει κάνει ο Client.
- Όλα τα μηνύματα σε QoS 1 ή 2 που είναι σε ροή και ο Client δεν έχει επιβεβαιώσει ακόμη.
- Όλα τα νέα μηνύματα με QoS 1 ή 2 που δεν έλαβε ο Client όσο ήταν αποσυνδεδεμένος.
- Όλα τα μηνύματα με QoS 2 που έλαβε ο Client και δεν έχει επιβεβαιώσει πλήρως.

Όταν ένας Client συνδέεται με τον Broker, έχει τη δυνατότητα να ζητήσει μια μόνιμη συνεδρία. Ο Client χρησιμοποιεί το flag Clean-Session για να δηλώσει στον Broker το είδος της συνεδρίας. Όταν το flag Clean-Session είναι αληθής, ο Client δεν θέλει μόνιμη συνεδρία. Σε αυτή την περίπτωση αν ο Client αποσυνδεθεί, τότε όλες οι πληροφορίες και τα μηνύματα χάνονται. Αντίθετα, όταν το flag Clean-Session είναι ψευδής, τότε ο Broker δημιουργεί μια μόνιμη συνεδρία για τον Client. Σε αυτή την περίπτωση όλες οι πληροφορίες και τα μηνύματα αποθηκεύονται μέχρι ο Client να ζητήσει εκ νέου καθαρή συνεδρία.

Από την έκδοση MQTT 3.1.1, το μήνυμα CONNACK περιέχει το flag «Session Present». Αυτό το flag ενημερώνει τον Client αν μια παλιότερη σύνδεση εξακολουθεί να είναι διαθέσιμη από τον Broker. Όπως ο Broker έτσι και ο Client πρέπει και αυτός να αποθηκεύει μια μόνιμη συνεδρία. Όταν ένας Client ζητήσει από τον server να κρατήσει δεδομένα σύνδεσης, ο Client είναι υπεύθυνος να αποθηκεύσει τις παρακάτω πληροφορίες:

- Όλα τα μηνύματα QoS 1 ή 2 που είναι σε ροή και δεν έχουν ακόμη επιβεβαιωθεί από τον Broker.
- Όλα τα μηνύματα QoS 2 που έλαβε από τον Broker και δεν έχουν αναγνωριστεί πλήρως.

Ο Broker αποθηκεύει μια συνεδρία μέχρι ο Client να επανασυνδεθεί και να λάβει το μήνυμα. Όμως, αν ένας Client δεν επανασυνδεθεί για αρκετό χρονικό διάστημα τότε η μνήμη του λειτουργικού συστήματος είναι αυτή που περιορίζει την αποθήκευση των μηνυμάτων. [26]

2.6.4 Retained Messages

Στο πρωτόκολλο MQTT, ο Publisher δεν γνωρίζει πραγματικά εάν ένας Subscriber λαμβάνει τα μηνύματα. Ο Publisher μπορεί να βεβαιωθεί μόνο ότι το μήνυμα παραδίδεται με ασφάλεια στον Broker. Το ίδιο ισχύει και με τον Subscriber. Ο Client που κάνει Subscribe σε κάποια Topics δεν γνωρίζει στην πραγματικότητα αν και πότε ο Publisher θα κάνει Publish. Ενδεχομένως μπορεί να περάσει αρκετό διάστημα, ίσως και ώρες, για να στείλει ο Publisher ένα νέο μήνυμα. Μέχρι να γίνει Publish το επόμενο μήνυμα, ο Subscriber όλο αυτό το χρονικό διάστημα απλά περιμένει. Σε αυτήν την περίπτωση τα διατηρούμενα μηνύματα είναι αυτά που παίρνουν αξία.

Ένα διατηρούμενο μήνυμα είναι ένα απλό μήνυμα με το Retained Flag να είναι αληθής. Ο Broker αποθηκεύει το πιο πρόσφατο μήνυμα και το QoS σχετικά με το Topic. Κάθε Client που κάνει Subscribe σε ένα Topic, τότε αμέσως λαμβάνει το διατηρούμενο μήνυμα. Ο Broker αποθηκεύει μονάχα ένα διατηρούμενο μήνυμα ανά Topic, το οποίο είναι και το πιο πρόσφατο.

Η δυνατότητα αυτή είναι πολύ χρήσιμη για τους νέους Subscribers, αφού λαμβάνουν σχετική ενημέρωση για την κατάσταση του Topic αμέσως μετά το Subscribe. Είναι σημαντικό να αναφερθεί ότι ένα μήνυμα που διατηρείται δεν έχει καμία σχέση με τα «Persistent Sessions». Για τη διαγραφή ενός Retained Message αρκεί να σταλεί στο Topic ένα μήνυμα με payload μηδέν byte. Με τη σειρά του ο Broker θα διαγράψει το διατηρούμενο μήνυμα με αποτέλεσμα οι νέοι Subscribers να μην το λάβουν. Τέλος, να σημειωθεί ότι δεν είναι απαραίτητο να διαγραφεί, επειδή κάθε νέο μήνυμα που διατηρείται αντικαθιστά το προηγούμενο. [27]

2.6.5 Last Will and Testament

Επειδή το πρωτόκολλο MQTT χρησιμοποιείται συχνά σε εφαρμογές με αναξιόπιστα δίκτυα, είναι λογικό να θεωρηθεί ότι ορισμένοι από τους MQTT Clients να αποσυνδέονται ανορθόδοξα κατά περιόδους. Μια τέτοια αποσύνδεση μπορεί να συμβεί λόγω απώλειας δικτύου, άδειας μπαταρίας κ.α. Γνωρίζοντας εάν ένας Client αποσυνδέθηκε ορθά (με το μήνυμα DISCONNECT) ή ανορθόδοξα (χωρίς μήνυμα DISCONNECT), τότε το πρωτόκολλο ανταποκρίνεται σωστά. Η δυνατότητα «Last Will and Testament» (LWT) είναι αυτή η οποία παρέχει έναν τρόπο στους Clients να ανταποκρίνονται κατάλληλα σε ανορθόδοξες αποσυνδέσεις.

Η δυνατότητα Last Will and Testament χρησιμοποιείται όταν ένας Client που έχει αποσυνδεθεί ανορθόδοξα να ειδοποιηθούν οι υπόλοιποι Clients σχετικά με αυτή την αποσύνδεση. Κάθε Client μπορεί να καθορίσει το Last Will μήνυμα κατά την έναρξη της επικοινωνίας του με τον Broker. Το Last Will μήνυμα είναι ένα κανονικό μήνυμα με Topic, retained message flag, QoS και payload. Ο Broker κρατάει το μήνυμα μέχρι να εντοπίσει ότι ο Client έχει αποσυνδεθεί ανορθόδοξα. Με την σειρά του ο Broker στέλνει το Last Will μήνυμα σε όλους τους Subscribers του Topic του Last Will μηνύματος. Εάν όμως, ο Client αποσυνδεθεί με ορθό τρόπο στέλνοντας μήνυμα DISCONNECT, τότε ο Broker απορρίπτει το αποθηκευμένο μήνυμα LWT.

Οι Clients μπορούν να καθορίσουν το μήνυμα LWT στο μήνυμα CONNECT, με το οποίο ξεκινά η σύνδεση μεταξύ του Client και του Broker. Σύμφωνα με την προδιαγραφή του MQTT 3.1.1, ο Broker πρέπει να στείλει το LWT ενός Client στις εξής περιπτώσεις:

- Ο Broker εντοπίζει σφάλμα ή αποτυχία δικτύου.
- Ο Broker κλείνει τη σύνδεση λόγω μη συμβατότητας πρωτόκολλου.
- Ο Client δεν καταφέρνει να επικοινωνήσει εντός του διαστήματος Keep Alive.
- Ο Client δεν στέλνει μήνυμα DISCONNECT πριν τερματίσει τη σύνδεση. [28]

2.6.6 Keep Alive

Το MQTT βασίζεται στο πρωτόκολλο Transmission Control Protocol (TCP). Αυτό το πρωτόκολλο επικοινωνίας διασφαλίζει ότι τα πακέτα μηνυμάτων μεταφέρονται μέσω του διαδικτύου με έναν τρόπο αξιόπιστο, ταξινομημένο και χωρίς λάθη. Ωστόσο, με την πάροδο του χρόνου, η μεταφορά των μηνυμάτων μεταξύ των δύο πλευρών που επικοινωνούν, μπορεί να είναι εκτός συγχρονισμού (π.χ. εάν κάποιο από τα μέρη έχει κολλήσει ή έχει σφάλματα στη μετάδοση). Στο πρωτόκολλο TCP αυτή η κατάσταση (δηλαδή η κατάσταση της ημιτελούς σύνδεσης) χαρακτηρίζεται ως μισάνοιχτη σύνδεση (half-open connection). Σημειώνεται εδώ ότι η μία πλευρά της σύνδεσης συνεχίζει να λειτουργεί κανονικά, χωρίς όμως να ειδοποιείται για την αποτυχία της άλλης πλευράς. Έτσι, η πλευρά που είναι ακόμα συνδεδεμένη και συνεχίζει να στέλνει μηνύματα, περιμένει μηνύματα επιβεβαίωσης από την άλλη πλευρά. Αυτό όπως καταλαβαίνουμε είναι πρόβλημα. Λύση στο πρόβλημα των μισάνοιχτων συνδέσεων δίνει η δυνατότητα «Keep Alive».

Το Keep Alive εξασφαλίζει ότι η σύνδεση μεταξύ του Client και του Broker είναι ακόμα ανοιχτή και ότι και οι δύο γνωρίζουν ότι είναι συνδεδεμένοι. Όταν ο Client δημιουργεί μια σύνδεση με τον Broker, τότε ο Client στέλνει ένα χρονικό διάστημα σε μορφή δευτερολέπτων στον Broker. Αυτός ο χρόνος καθορίζει το μέγιστο χρονικό διάστημα κατά το οποίο ο Client και ο Broker ενδέχεται να μην ανταλλάξουν μηνύματα μεταξύ τους. Δηλαδή, είναι το μέγιστο χρονικό διάστημα που επιτρέπεται να περάσει μεταξύ δύο διαδοχικών μηνυμάτων. Στην περίπτωση που δεν υπάρχουν μηνύματα προς αποστολή, τότε ο Client είναι υπεύθυνος να στείλει ένα πακέτο PINGREQ σε χρονικό διάστημα μικρότερο από αυτό του Keep Alive.

Εάν η ανταλλαγή των μηνυμάτων γίνεται συχνά και δεν ξεπερνιέται το χρονικό διάστημα Keep Alive, τότε δεν χρειάζεται να σταλεί το πακέτο PINGREQ για να διαπιστωθεί ότι η σύνδεση είναι ακόμα ενεργή. Αντίθετα, στην περίπτωση που ο Client δεν στείλει κάποιο μήνυμα εντός του διαστήματος Keep Alive, τότε πρέπει να στείλει ένα πακέτο PINGREQ στον Broker. Αυτό γίνεται για να βεβαιωθούν και οι δύο πλευρές ότι και οι δύο είναι διαθέσιμοι και μπορούν να ανταλλάξουν μηνύματα. Αν ο Broker από την πλευρά του δε λάβει πακέτο PINGREQ σε διάστημα μεγαλύτερο από αυτό του Keep Alive, τότε ο Broker πρέπει να αποσυνδέσει τον Client. Ομοίως, αν ο Client δεν λάβει πακέτο PINGRESP από τον Broker σε εύλογο χρονικό διάστημα, τότε ο Client κλείνει τη σύνδεση.

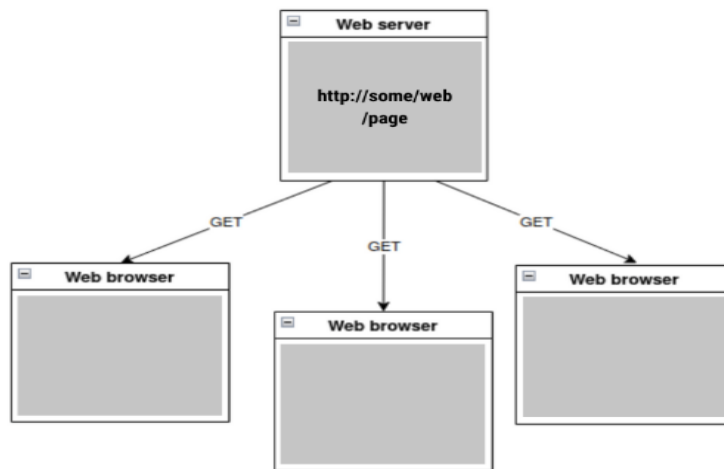
Από τα παραπάνω συμπεραίνουμε ότι υπάρχουν δύο πακέτα για την επιβεβαίωση της επικοινωνίας. Το ένα είναι το PINGREQ και το άλλο το PINGRESP. Το PINGREQ αποστέλλεται από τον Client και γνωστοποιεί στον Broker ότι ο Client είναι ακόμα διαθέσιμος. Ο Client έχει τη δυνατότητα να στείλει ένα πακέτο PINGREQ όποτε θέλει. Από την άλλη πλευρά, όταν ο Broker λάβει ένα πακέτο PINGREQ, πρέπει να απαντήσει στον Client με ένα πακέτο PINGRESP για να του δείξει ότι και αυτός συνεχίζει να είναι διαθέσιμος. Σημειώνεται ότι τα πακέτα PINGREQ και PINGRESP δεν περιλαμβάνουν το payload, δηλαδή το ωφέλιμο μήνυμα. [29]

2.7 Σύγκριση MQTT με άλλα πρωτόκολλα

2.7.1 Πρωτόκολλο HTTP

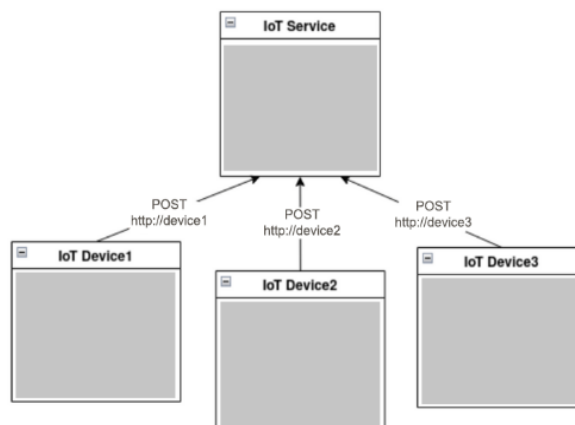
Το πρωτόκολλο HTTP (HyperText Transfer Protocol) επινοήθηκε ως στοιχείο του παγκόσμιου Ιστού (World Wide Web) για τη μεταφορά εγγράφων. Είναι το πιο γνωστό πρωτόκολλο με το οποίο λειτουργούν τα προγράμματα περιήγησης. Οι Server περιέχουν πόρους οι οποίοι προσδιορίζονται με τα URLs, δηλαδή με τις διευθύνσεις (π.χ <https://aes.iee.ihu.gr>), όπου ένας HTTP Client μπορεί να κάνει

αιτήματα GET, POST, PUT και DELETE. Το πιο απλό μοντέλο αυτού του πρωτόκολλου είναι όταν ένα πρόγραμμα περιήγησης ανακτά ιστοσελίδες από έναν Server με τη μέθοδο GET.



Εικόνα 10. Τρόπος ανάκτησης ιστοσελίδων με το HTTP

Σε εφαρμογές IoT η χρήση του HTTP πρωτόκολλου επιτρέπει στις συσκευές να στέλνουν την κατάστασή τους (π.χ. μια μέτρηση) σε έναν πόρο με την μέθοδο POST (με τη μορφή ενός link).



Εικόνα 11. Αποστολή πληροφορίας στο HTTP με τη μέθοδο POST

Μόνο το πρώτο μήνυμα στο MQTT είναι μεγάλο, και ο λόγος είναι επειδή μετά τη σύνδεση ακολουθεί το μήνυμα CONNACK. Ύστερα τα πακέτα μηνυμάτων είναι πολύ μικρότερα σε όγκο. Ενώ σε αντίθεση με το HTTP όπου σε κάθε μήνυμα παρέχεται η κεφαλίδα (header) με αποτέλεσμα να αυξάνει τον συνολικό όγκο του μηνύματος και να κάνει πιο αργή τη μετάδοση. Παρακάτω φαίνονται ο όγκος των μηνυμάτων σε κάθε πρωτόκολλο και οι χρόνοι μετάδοσης αυτών.

Πίνακας 1. Σύγκριση όγκου μηνύματος μεταξύ MQTT και HTTP

| Μηνύματα | MQTT | HTTP |
|-----------------------------------------|-------------------|--------|
| Δημιουργία σύνδεσης (bytes) | 5572 | 2261 |
| Κατά την αποσύνδεση (bytes) | 376 (προαιρετικό) | 0 |
| Για κάθε μήνυμα μετά τη σύνδεση (bytes) | 388 | 3285 |
| Για ένα μήνυμα (bytes) | 6336 | 5546 |
| Για δέκα μηνύματα (bytes) | 9829 | 55460 |
| Για εκατό μηνύματα (bytes) | 44748 | 554600 |

Πίνακας 2. Σύγκριση χρόνου αποστολής μηνύματος

| Μηνύματα | MQTT (για κάθε μήνυμα) | HTTP (για κάθε μήνυμα) |
|----------|------------------------|------------------------|
| 1 | 113ms | 289ms |
| 100 | 47ms | 289ms |
| 1000 | 43ms | 289ms |

Πίνακας 3. Ομοιότητες και διαφορές μεταξύ MQTT και HTTP

| MQTT | HTTP |
|------------------------------|---------------------------------------|
| Publish-Subscribe | Αίτημα-Ανταπόκριση (Request-Response) |
| Topics | URIs |
| TCP/IP | TCP/IP |
| Ένα προς ένα, ένα προς πολλά | Ένα προς ένα |
| QoS 0, 1, 2 | Πρέπει να υλοποιηθεί |

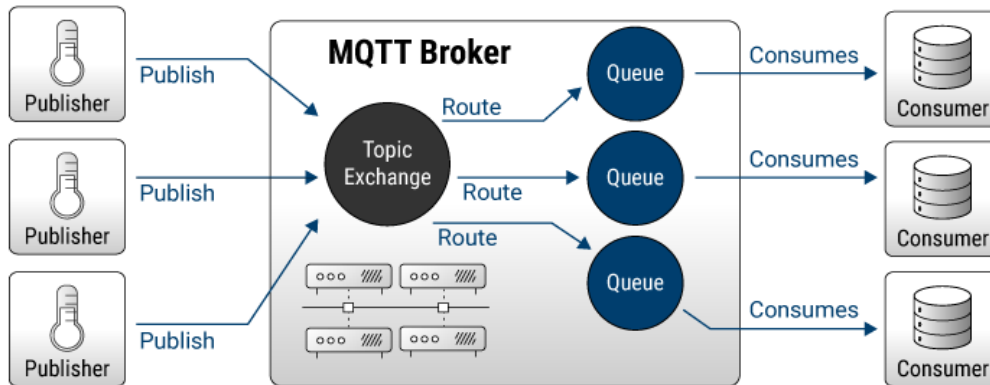
Κλείνοντας, το πλεονέκτημα με το HTTP είναι ότι το πρωτόκολλο αυτό είναι γνωστό σχεδόν σε όλους τους προγραμματιστές και χρησιμοποιείται σε πάρα πολλές εφαρμογές. Αν και τα δύο πρωτόκολλα βασίζονται στο TCP/IP, το MQTT είναι σχεδιασμένο αποκλειστικά για εφαρμογές IoT. Και ο λόγος είναι ότι το HTTP δεν διαθέτει χρήσιμες λειτουργίες όπως οι ουρές μηνυμάτων, το QoS, τα διατηρούμενα μηνύματα και φυσικά έχει ογκώδη μηνύματα. [30]

2.7.2 Πρωτόκολλο AMQP

Το πρωτόκολλο AMQP (Advanced Message Queuing Protocol) είναι ένα πρωτόκολλο ανταλλαγής μηνυμάτων και σχεδιάστηκε για εφαρμογές γενικής χρήσης, ενώ το MQTT για εφαρμογές IoT. Η έκδοση AMQP 0.9.1 αποτελεί προϊόν της εταιρείας AMQP, ενώ η έκδοση AMQP 1.0 αποτελεί πρότυπο του οργανισμού OASIS (όπως και οι τελευταίες εκδόσεις του MQTT). Το πρωτόκολλο AMQP υποστηρίζει πολλαπλές τοπολογίες δρομολόγησης, οι οποίες καθορίζονται ανάλογα την εφαρμογή. Είναι ένα ευέλικτο πρωτόκολλο αλλά ταυτόχρονα περίπλοκο, και έχει υψηλότερο κόστος στην εγκατάστασή του σε σχέση με το MQTT.

Το πρωτόκολλο αυτό χρησιμοποιεί είτε το μοντέλο Publish-Subscribe είτε την point-to-point επικοινωνία για τη μετάδοση των μηνυμάτων. Μια εφαρμογή πρέπει να δημιουργήσει μια ουρά (queue), εφόσον δεν υπάρχει, και ύστερα να συνδέσει αυτή την ουρά σε μια ανταλλαγή (exchange). Αυτή η ανταλλαγή είναι και ο στόχος της λειτουργίας της αποστολής των μηνυμάτων, όπου στη συνέχεια δρομολογεί τα μηνύματα σε ουρές. Υπάρχουν τέσσερις τύποι ανταλλαγής και είναι οι εξής:

- **Direct**, το κλειδί της δρομολόγησης ταιριάζει με το όνομα ουράς.
- **Fanout**, κάθε ουρά που συνδέεται σε αυτήν λαμβάνει κάθε μήνυμα.
- **Topic**, τα μηνύματα δρομολογούνται σε ουρές με βάση το μοτίβο που χρησιμοποιείται κατά τη σύνδεση της ουράς με την ανταλλαγή.
- **Headers**, παρόμοιο με το Topic αλλά χρησιμοποιεί κεφαλίδες στην αντιστοίχιση.



Εικόνα 12. Αρχιτεκτονική πρωτόκολλου AMQP

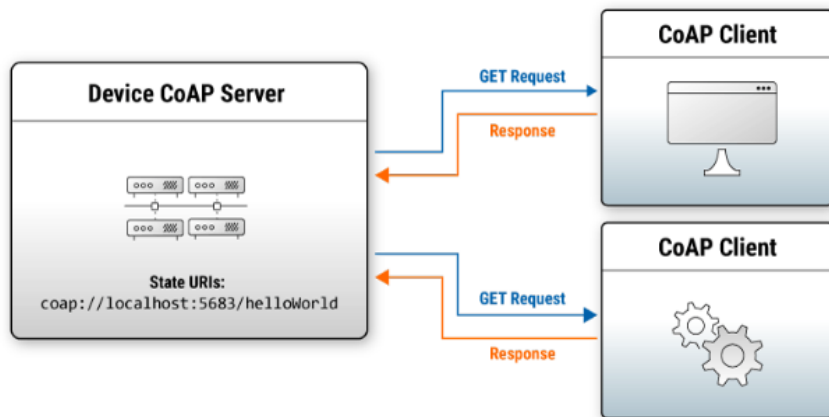
Το πλεονέκτημα που έχει το πρωτόκολλο AMQP είναι ότι έχει πολλές δυνατότητες στις ουρές μηνυμάτων. Όμως η πολυπλοκότητα, το υψηλό κόστος και η σύγχυση ανάμεσα στις εκδόσεις του δεν είναι αποτελεσματικό για εφαρμογές IoT. Παρακάτω φαίνονται κάποιες από τις ομοιότητες και διαφορές μεταξύ των πρωτόκολλων. [31]

Πίνακας 4. Σύγκριση MQTT και AMQP

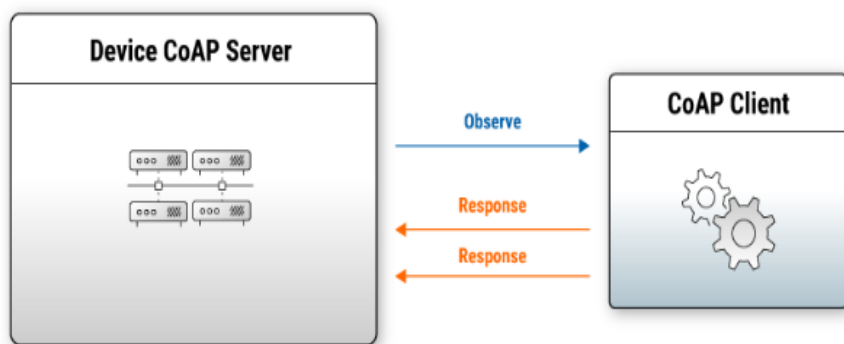
| MQTT | AMQP |
|-------------------|----------------------------------------------------------------------|
| Publish-Subscribe | Ουρές, πολλαπλή εκπομπή(fanout), Publish-Subscribe, ερώτηση-απάντηση |
| Topics | Exchanges, ουρές |
| TCP/IP | TCP/IP |
| QoS 0 ,1, 2 | QoS 0 ,1 |

2.7.3 Πρωτόκολλο CoAP

Το πρωτόκολλο CoAp (Constrained Application Protocol), είναι ένα πρωτόκολλο Server-Client το οποίο έχει σχεδιαστεί για εφαρμογές IoT. Χρησιμοποιεί τις εντολές GET, POST, PUT και DELETE όπως και το πρωτόκολλο HTTP. Είχε σχεδιαστεί για εφαρμογές machine-to-machine υποστηρίζοντας περιορισμένες συσκευές και δίκτυα μέσω του HTTP. Μια συσκευή μπορεί να είναι CoAP Client είτε Server. Η αλληλεπίδραση μεταξύ Server-Client γίνεται με το μοτίβο αιτήματος-απόκρισης (request-response). Για να πάρουμε πληροφορίες από τον Server αρκεί να στείλουμε ένα αίτημα GET στη διεύθυνση (link) που θέλουμε. Έτσι, υπάρχει η δυνατότητα ταυτόχρονης αποστολής αιτημάτων σε πολλαπλές συσκευές (με τη μορφή ενός γενικότερου link, όπως για παράδειγμα ενεργοποίηση φώτων μαζικά).



Εικόνα 13. Αλληλεπίδραση μεταξύ Client και Server



Εικόνα 14. Τρόπος λειτουργίας του πρωτόκολλου CoAp

Η επέκταση της παρατήρησης (Observe extension) του CoAP δίνει τη δυνατότητα στους Clients να κάνουν Subscribe σε ένα link του Server, έτσι ώστε κάθε φορά που αλλάζει η κατάσταση του εκάστοτε link, να ενημερώνονται αυτόματα για την αλλαγή αυτή. Όπως στο QoS=0 στο MQTT, το ίδιο ισχύει και εδώ, δεν υπάρχει καμία επιβεβαίωση μηνυμάτων. Ενώ σε επίπεδο QoS=1, αν δεν επιβεβαιωθεί η λήψη ενός μηνύματος τότε επαναλαμβάνεται η αποστολή.

Το πλεονέκτημα που έχει το πρωτόκολλο CoAP είναι ότι η δομή του είναι παρόμοια με το πρωτόκολλο HTTP. Με αποτέλεσμα να το κάνει εύκολο στη χρήση του αφού το HTTP είναι γνωστό από τους περισσότερους προγραμματιστές. Ενώ το μειονέκτημα είναι ότι αφού βασίζεται στο πρωτόκολλο UDP και όχι στο TCP, το καθιστά αναξιόπιστο. Παρακάτω φαίνονται κάποιες ομοιότητες και διαφορές μεταξύ των πρωτόκολλων CoAP και MQTT. [32]

Πίνακας 5. Σύγκριση MQTT και CoAP

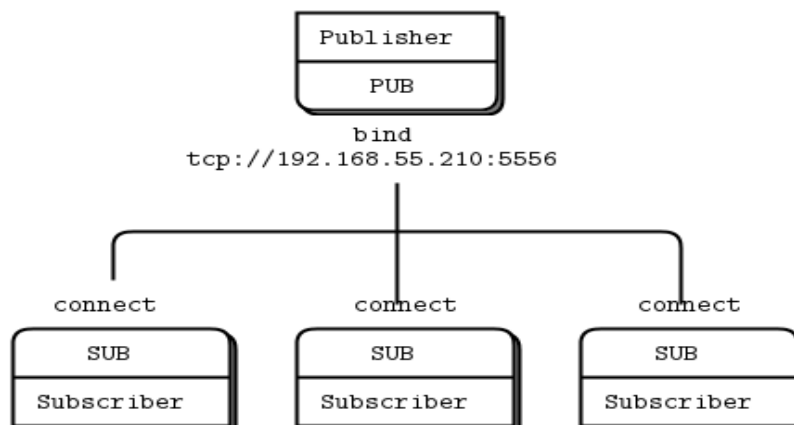
| MQTT | CoAP |
|------------------------------|------------------------------|
| Topics | URIs |
| Client-Server | Client-Server |
| TCP/IP | UDP |
| Ένα προς ένα, ένα προς πολλά | Ένα προς ένα, ένα προς πολλά |
| QoS 0, 1, 2 | Παρόμοιο με QoS 0 και 1 |

2.7.4 Πρωτόκολλο ZeroMQ

Το πρωτόκολλο ZeroMQ είναι μια βιβλιοθήκη εφαρμογών η οποία έχει σχεδιαστεί για τη γρήγορη διανομή μηνυμάτων (π.χ. για ενημερώσεις τιμών, alerts, κτλ). Η λειτουργία του βασίζεται σε μεγάλο βαθμό στο peer-to-peer δίκτυο, δηλαδή όλοι οι κόμβοι ZeroMQ έχουν τις ίδιες δυνατότητες. Σε μια εφαρμογή ZeroMQ, η μία εφαρμογή μπορεί να είναι ο Publisher και η άλλη ο Subscriber. Αυτό όμως εξαρτάται από τις εφαρμογές και από την μεταξύ τους συμφωνία. Το πρωτόκολλο αυτό παρουσιάζει ένα API (Application Programming Interface), το οποίο μοιάζει με τα TCP/IP sockets. Για να συνδεθούν δύο εφαρμογές, η μία καλεί την `zmq_bind()` για να γίνει ο Server και η άλλη συνδέεται στην πρώτη με την `zmq_connect()`. Για την δημιουργία των ZeroMQ sockets όπου λειτουργούν τα APIs γίνεται κλήση της `zmq_socket()`. Οι τύποι αυτών των sockets είναι οι εξής:

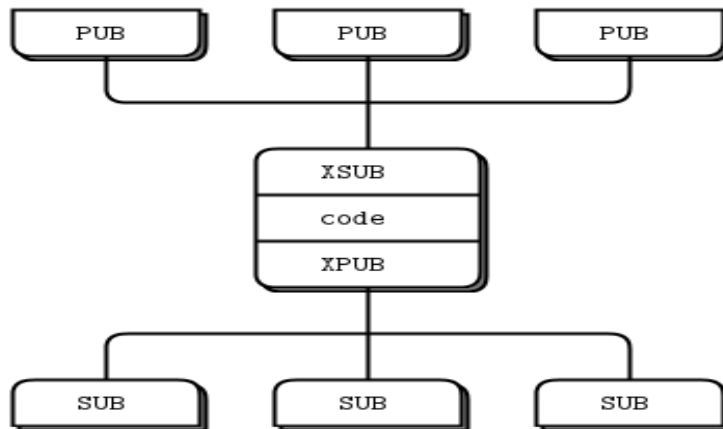
- ZMQ_CLIENT ή ZMQ_SERVER, δηλαδή Client/Server
- ZMQ_RADIO ή ZMQ_DISH, εναλλακτικά pub/sub
- ZMQ_(X)PUB ή ZMQ_(X)SUB, ζεύγος publish/subscribe – αποστολή μηνυμάτων από έναν προς πολλά
- ZMQ_PULL ή ZMQ_PUSH, ένας προς έναν
- ZMQ_PAIR, peer-to-peer επικοινωνία
- ZMQ_STREAM, σύνδεση σε μη όμοιο-ομότιμο ZeroMQ
- ZMQ_REQ ή ZMQ_REP and ZMQ_DEALER, ZMQ_ROUTER, ζεύγη αιτήματος-ανταπόκρισης

Αυτοί οι τύποι των sockets χρησιμοποιούνται μόνο με αντίστοιχα sockets. Άλλοι έχουν ενσωματωμένη ουρά μηνυμάτων, ενώ άλλοι όχι. Για παράδειγμα, ο τύπος ZMQ_PUB, δηλαδή ο Publisher, απορρίπτει τα μηνύματα που δεν μπορούν να παραδοθούν (όπως και στο QoS=0 του MQTT). Ενώ από την άλλη πλευρά, ο τύπος ZMQ_SUB, δηλαδή ο Subscriber, βάζει τα εισερχόμενα μηνύματα σε μια ουρά.



Εικόνα 15. Τρόπος λειτουργίας του πρωτόκολλου ZeroMQ

Ενώ το πρωτόκολλο ZeroMQ δεν χρησιμοποιεί Broker (όπως το MQTT), σε ορισμένες περιπτώσεις εισάγεται ένας κεντρικός κόμβος που βοηθάει στην ανίχνευση των Publishers και Subscribers. Σε αυτές τις περιπτώσεις ο κόμβος αυτός παίρνει τον ίδιο ρόλο όπως ο Broker στο MQTT.



Εικόνα 16. Ο Broker του ZeroMQ

Παρακάτω φαίνονται κάποιες διαφορές μεταξύ των δύο πρωτόκολλων

Πίνακας 6. Σύγκριση MQTT και ZeroMQ

| MQTT | ZeroMQ |
|------------------------------|--------------------------------|
| Client/Server | Peer-to-peer |
| Topics | ZeroMQ sockets |
| TCP/IP | TCP/IP, UDP |
| Ένα προς πολλά, ένα προς ένα | Διάφοροι τρόποι |
| QoS 0, 1, 2 | Ανάλογα με τον τύπο του socket |

Το πλεονέκτημα του ZeroMQ είναι ότι θεωρείται μια ευέλικτη εργαλειοθήκη που δίνει τη δυνατότητα πολλαπλών λύσεων σε ζητήματα ανταλλαγής μηνυμάτων. Όμως για να συμπεριφερθεί όπως το MQTT θα πρέπει ο προγραμματιστής να το υλοποιήσει μόνος του από την αρχή με τα εργαλεία που του διαθέτει το πρωτόκολλο ZeroMQ. Τέλος, το ZeroMQ δεν παρέχει την ασφάλεια που παρέχει το MQTT. [33]

2.8 Επίλογος

Σε αυτό το κεφάλαιο έγινε μια εκτενής ανάλυση του πρωτόκολλου επικοινωνίας MQTT. Αρχικά, εξηγήθηκε τί είναι και έπειτα παρουσιάστηκε η ιστορική του αναδρομή. Στη συνέχεια, έγινε αναφορά στην αρχιτεκτονική του πρωτόκολλου αυτού και πώς λειτουργεί, και αναλύθηκαν τα βασικά του χαρακτηριστικά. Τέλος, ακολουθεί η σύγκριση του MQTT με άλλα πρωτόκολλα επικοινωνίας και συμπεραίνεται η ευκολία και η χρηστικότητά του σε εφαρμογές IoT.

Κεφάλαιο 3ο: Αισθητήρες και λοιπά υλικά

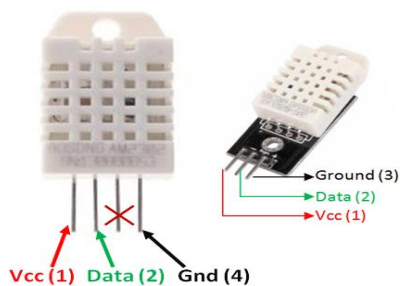
3.1 Εισαγωγή

Αφού στο προηγούμενο κεφάλαιο έγινε εκτενής ανάλυση του πρωτόκολλου MQTT αλλά και η σύγκρισή του με άλλα πρωτόκολλα επικοινωνίας, σε αυτό το κεφάλαιο αναλύονται τα επιμέρους εξαρτήματα για το σύστημα του θερμοκηπίου που υλοποιήθηκε. Πιο συγκεκριμένα, αναλύεται η λειτουργία των αισθητήρων, γίνεται αναφορά των μικροελεγκτών και των λοιπών υλικών που χρησιμοποιήθηκαν.

3.2 Αισθητήρες

3.2.1 Αισθητήρας DHT22

Ο αισθητήρας DHT22 είναι ένας αισθητήρας που μετράει θερμοκρασία και σχετική υγρασία, και διαθέτει βαθμονομημένη ψηφιακή έξοδο. Είναι ένα αισθητήριο καλής ποιότητας, μικρό σε μέγεθος, με χαμηλή κατανάλωση ενέργειας, ανταποκρίνεται γρήγορα και μπορεί να χρησιμοποιηθεί σε μεγάλες αποστάσεις (μετάδοση σήματος έως και 20 μέτρα από τον μικροελεγκτή). Έχει τρεις ακροδέκτες και η επικοινωνία με τον μικροελεγκτή γίνεται μέσω 1-Wire. Το πεδίο εφαρμογής του είναι ευρύ, όπως για παράδειγμα, αφυγραντήρες, αυτοκίνητα, data loggers, σε διάφορες οικιακές συσκευές και γενικότερα όπου χρειάζεται επιτήρηση θερμοκρασίας και υγρασίας.

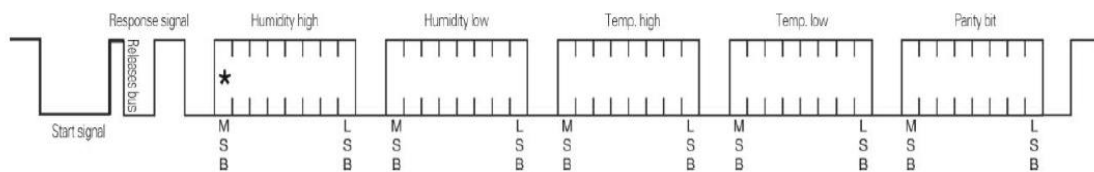


Εικόνα 17. Αισθητήρας DHT22

Πίνακας 7. Χαρακτηριστικά του DHT22

| Pin | Περιγραφή |
|----------------------------|------------------------------|
| Τάση τροφοδοσίας VDD | 3,3V – 5,5V (συνιστάται 5V) |
| SDA | Serial data, αμφίδρομη πόρτα |
| NC | No Connection |
| GND | Ground |
| Μέση κατανάλωση | 300μΑ |
| Θερμοκρασία | -40 έως 80 (°C) |
| Σχετική υγρασία | 0 έως 99,9 (%RH) |
| Ακρίβεια θερμοκρασίας | ±0,5 (°C) |
| Ακρίβεια σχετικής υγρασίας | ±2 (%RH) |
| Ανάλυση θερμοκρασίας | 0,1 (°C) |
| Ανάλυση σχετικής υγρασίας | 0,1 (%RH) |

Το αισθητήριο DHT22 χρησιμοποιεί μια απλουστευμένη τεχνολογία για την επικοινωνία με τον μικροελεγκτή, την 1-Wire. Με το 1-Wire εννοούμε ότι η επικοινωνία γίνεται μέσω μόνο μίας γραμμής. Στις περισσότερες εφαρμογές απαιτείται η χρήση pull-up αντίστασης. Οπότε όταν η επικοινωνία είναι αδρανής, τότε η κατάσταση είναι HIGH. Επειδή η επικοινωνία του αισθητηρίου βασίζεται στη δομή MASTER-SLAVE, σημαίνει ότι ο αισθητήρας ανταποκρίνεται μόνο όταν τον καλέσει ο μικροελεγκτής. Να σημειωθεί ότι ο μικροελεγκτής πρέπει να συμμορφωθεί με την ακολουθία του 1-Wire για να μπορέσει να λάβει σωστά τα δεδομένα. Σε περίπτωση μη συμμόρφωσης, ο αισθητήρας δεν ανταποκρίνεται. Ουσιαστικά ο μικροελεγκτής και ο αισθητήρας πρέπει να συγχρονιστούν. Ο αριθμός των δεδομένων που μεταδίδει ο αισθητήρας είναι 40bit. Παρακάτω φαίνονται τα δεδομένα που στέλνει ο αισθητήρας.



Εικόνα 18. Η σειρά των 40bit

- **Start signal**, ο μικροελεγκτής θέτει το pin SDA στο LOW για χρονικό διάστημα τουλάχιστον 800μs έτσι ώστε να προετοιμάσει τον αισθητήρα.
- **Response signal**, ο αισθητήρας θέτει το pin SDA στο LOW για 80μs και ύστερα στο HIGH πάλι για 80μs ως απάντηση στο Start signal.
- **Data format**, μετά τη λήψη του Start signal ο αισθητήρας στέλνει τα δεδομένα (40 bit).
- **Humidity**, η υγρασία είναι τα πρώτα 16 bit και είναι 10 φορές μεγαλύτερη από την πραγματική τιμή.
- **Temperature**, η θερμοκρασία είναι τα επόμενα 16 bit και είναι 10 φορές μεγαλύτερη από την πραγματική τιμή. Όταν το MSB(bit15) είναι '1', η θερμοκρασία είναι αρνητική. Ενώ όταν το MSB(bit15) είναι '0', η θερμοκρασία είναι θετική. Τα επόμενα bit (0 έως 14) εκφράζουν τη θερμοκρασία.
- **Parity bit**, είναι το bit ισοτιμίας, Parity bit = Humidity high + Humidity low + Temperature high + Temperature low.

Αρχικά ο μικροελεγκτής στέλνει το Start signal για να προετοιμάσει τον αισθητήρα. Μόλις ο αισθητήρας λάβει το σήμα αυτό ανταποκρίνεται με το Response signal. Έπειτα, στέλνει τα δεδομένα (40 bit). Αυτά είναι το Humidity high, Humidity low, Temperature high, Temperature low και το Parity bit. Αφού τα στείλει, μπαίνει ξανά σε Sleep Mode και περιμένει για το επόμενο Start signal. Παρακάτω φαίνεται ένα παράδειγμα υπολογισμού με τα δεδομένα που στέλνει ο αισθητήρας.

Έστω ότι έχουμε: $0000\ 0010 + 1001\ 0010 + 0000\ 0001 + 0000\ 1101 = 1010\ 0010$ (Bit ισοτιμίας)

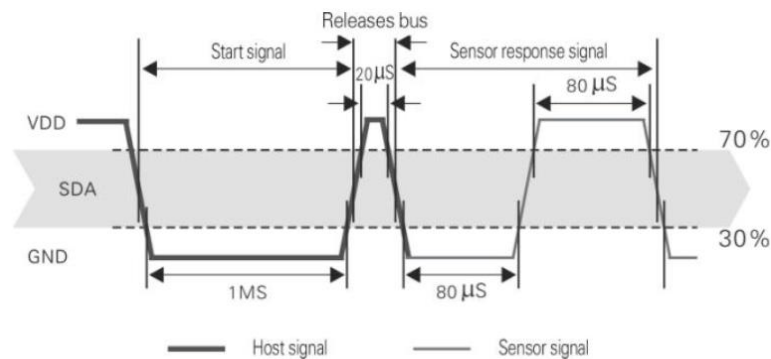
Σχετική Υγρασία = $(2 \times 256) + (9 \times 16) + 2 = 658$

Θερμοκρασία = $(1 \times 256) + (0 \times 16) + 13 = 269$

Άρα, η Σχετική Υγρασία είναι 65,8% και η Θερμοκρασία είναι 26,9°C.

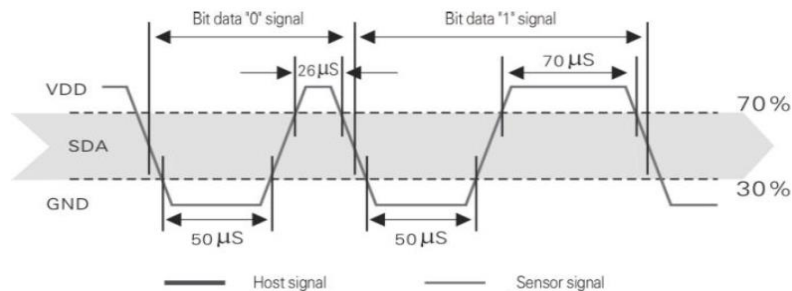
Αφού ενεργοποιηθεί ο αισθητήρας πρέπει να περιμένουμε 2 δευτερόλεπτα για να σταθεροποιηθεί. Σε αυτό το χρονικό διάστημα δεν γίνεται καμία αποστολή προς τον μικροελεγκτή. Απλά ο αισθητήρας παίρνει τις μετρήσεις από το περιβάλλον, τις καταγράφει και μπαίνει αυτόματα σε Sleep Mode. Λόγο της pull-up αντίστασης το pin SDA μεταβαίνει σε κατάσταση HIGH. Αυτή τη χρονική στιγμή ο αισθητήρας βρίσκεται σε input κατάσταση περιμένοντας κάποιο εξωτερικό σήμα.

Με τη σειρά του ο μικροελεγκτής θέτει το pin σε output και σε κατάσταση LOW για περισσότερο από 800μs (τυπικά 1ms). Και μετά πάλι σε κατάσταση input και HIGH λόγω της pull-up αντίστασης. Ύστερα ο αισθητήρας ανταποκρίνεται για χρόνο 80μs σε κατάσταση LOW και αμέσως μετά για χρόνο 80μs σε κατάσταση HIGH για να ενημερώσει τον μικροελεγκτή ότι θα του στείλει τα δεδομένα. Αυτό φαίνεται στην παρακάτω εικόνα.



Εικόνα 19. Το start signal και το response signal

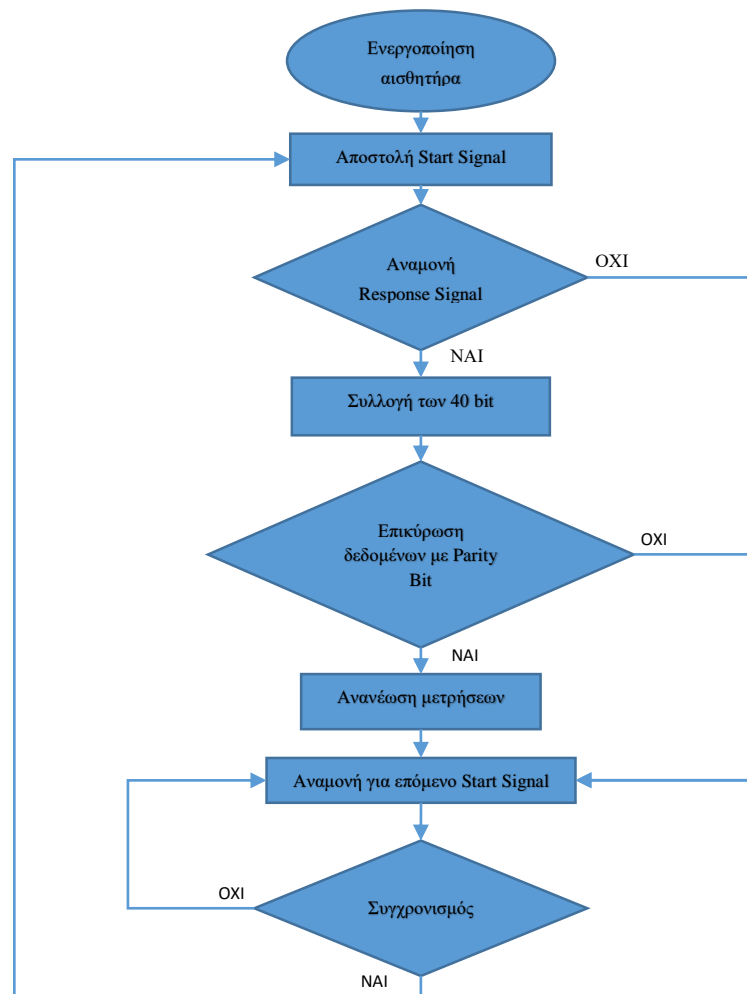
Τέλος, ο αισθητήρας στέλνει στον μικροελεγκτή την ακολουθία των 40 bit η οποία είναι η σχετική υγρασία, η θερμοκρασία και το bit ισοτιμίας. Το bit ισοτιμίας χρησιμοποιείται για το αν τα δεδομένα στάλθηκαν σωστά.



Εικόνα 20. Διαχωρισμός 0 και 1 bit

- Όταν έχουμε 50μs LOW και 26-28μs HIGH, τότε το bit είναι '0'.
- Όταν έχουμε 50μs LOW και 70μs HIGH, τότε το bit είναι '1'.

Μετά την αποστολή των 40 bit, ο αισθητήρας μένει σε κατάσταση LOW για 50μs και ύστερα πάει σε κατάσταση input και HIGH λόγω της pull-up αντίστασης. Ταυτόχρονα ο αισθητήρας παίρνει ξανά μετρήσεις και τις καταγράφει. Μόλις ολοκληρώσει τις λειτουργίες του μπαίνει σε Sleep Mode και περιμένει το επόμενο Start signal από τον μικροελεγκτή. [5][49]



Εικόνα 21. Διάγραμμα ροής της λειτουργίας του DHT22

3.2.2 Αισθητήρας MG-811

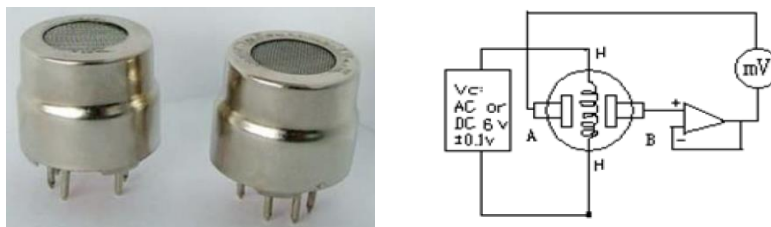
Για την μέτρηση της συγκέντρωσης του διοξειδίου του άνθρακα (CO₂) χρησιμοποιήθηκε ο αισθητήρας MG811 της εταιρείας DFRobot (μοντέλο SEN0159). Είναι ένας υψηλής ποιότητας αναλογικός αισθητήρας. Η τάση εξόδου του αισθητήρα μειώνεται όσο αυξάνεται η συγκέντρωση του CO₂. Ο αισθητήρας αυτός έχει ενσωματωμένο ένα ποτενσιόμετρο με το οποίο δίνεται η δυνατότητα της ρύθμισης του ορίου της τάσης (threshold) για σύγκριση του CO₂. Όταν η τάση γίνει μικρότερη από το όριο (δηλαδή αυξηθεί η συγκέντρωση του CO₂) τότε απελευθερώνεται ένα ψηφιακό σήμα (ON/OFF) δείχνοντας ότι το CO₂ είναι σε ψηλό επίπεδο. Στην παρούσα εργασία μας ενδιαφέρει μόνο η αναλογική έξοδος του αισθητήρα.



Εικόνα 22. Η πλακέτα SEN0159

Το μοντέλο SEN0159 έχει ενσωματωμένο τον αισθητήρα MG-811. Ο αισθητήρας αυτός είναι ιδιαίτερα ευαίσθητος στις μεταβολές του CO₂ και λιγότερο στο αλκοόλ και το μονοξείδιο του άνθρακα (CO). Επίσης, έχει χαμηλή εξάρτηση από τη θερμοκρασία και την υγρασία. Επειδή τέτοιου είδους αισθητήρια απαιτούν θέρμανση για την σωστή τους απόδοση, το μοντέλο SEN0159 έχει ενσωματωμένο κύκλωμα θέρμανσης. Η τάση λειτουργίας του είναι τα 5V, αλλά ανυψώνεται στα 6V η οποία απαιτείται για το κύκλωμα θέρμανσης. Τέλος, επειδή η έξοδος του αισθητήρα MG-811 είναι της τάξης των millivolt, υπάρχει ενσωματωμένο κύκλωμα ενίσχυσης του σήματος εξόδου.

Ο αισθητήρας MG-811 είναι ένας χημικός αισθητήρας που βασίζεται στην αρχή λειτουργίας της μπαταρίας στερεού ηλεκτρολύτη. Όταν ο αισθητήρας εκτεθεί σε περιβάλλον με υψηλό CO₂, τότε μεταξύ του αρνητικού και του θετικού πόλου δημιουργείται μια αντίδραση. Αυτό έχει ως αποτέλεσμα τη δημιουργία της ηλεκτροκινητικής δύναμης (EMF) μεταξύ του ευαίσθητου ηλεκτροδίου του αισθητήρα και του ηλεκτροδίου αναφοράς. Η τάση εξόδου του αισθητήρα μπορεί να χρησιμοποιηθεί για τον υπολογισμό του CO₂.



Εικόνα 23. Ο αισθητήρας MG-811

Επομένως, αφού ο αισθητήρας είναι ηλεκτροχημικός πρέπει να βαθμονομηθεί πριν την χρήση. Με βάση τον κατασκευαστή ο αισθητήρας πρέπει να λειτουργήσει στον καθαρό αέρα (έξω στο περιβάλλον 400ppm) για 48 ώρες συνεχόμενα. Αφού περάσουν οι 48 ώρες η τάση που βγάζει το SEN0159 είναι η τάση αναφοράς (δηλαδή τα 400ppm). Αυτή η τάση είναι ενισχυμένη 8,5 φορές. Οπότε στον πηγαίο κώδικα η παράμετρος ZERO_POINT_VOLTAGE θα ρυθμιστεί σε $V_{out}/8,5$. Η ενίσχυση αυτή καθορίζεται από το μοντέλο SEN0159. Σε περίπτωση χρήσης μόνο του αισθητήρα MG-811 η ενίσχυση μπορεί να είναι οποιαδήποτε, αρκεί να μην βλάψει τον μικροελεγκτή. Ο τύπος για τη βαθμονόμηση του αισθητήρα που δίνει ο κατασκευαστής είναι ο παρακάτω.

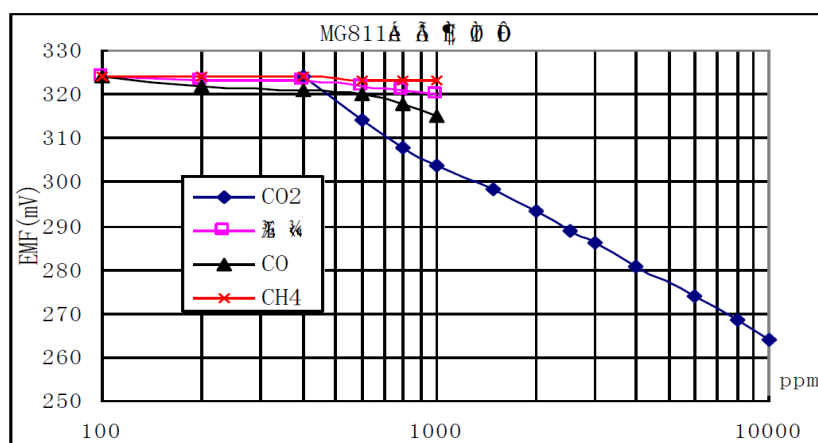
$$ppm = 10 \frac{\frac{V_{outCO_2} - ZeroPointVoltage}{GAIN} - \frac{ReactionVoltage}{2,602-3}}{2,602-3} + 2,602 \quad (3.1)$$

Όπου,

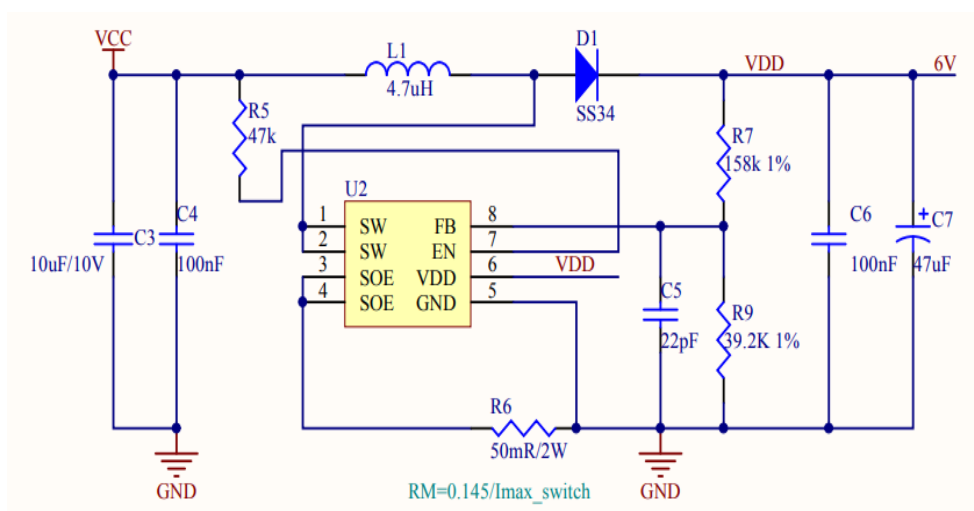
- **ppm**, parts per million, μονάδα μέτρησης CO₂
- **V_{outCO₂}**, τάση εξόδου SEN0159/8,5
- **GAIN=8,5**
- **ZeroPointVoltage**, η τάση που βγάζει ο αισθητήρας στον καθαρό αέρα (400ppm στο περιβάλλον)
- **ReactionVoltage**, μεταβολή της τάσης από τα 400ppm στα 1000ppm

Πίνακας 8. Χαρακτηριστικά του MG-811

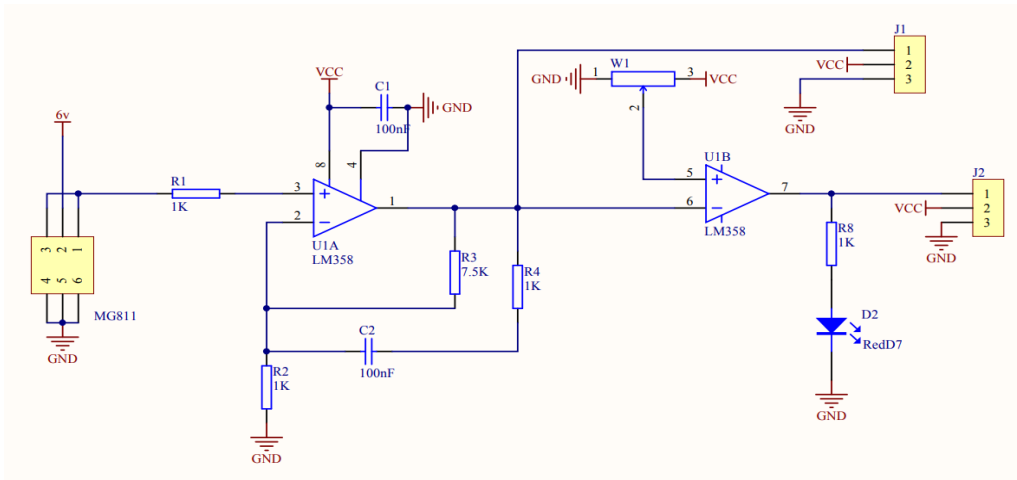
| Παράμετρος | Τιμή |
|------------------------------------|--------------|
| Τάση εξόδου MG-811 | 30-50mV |
| Εύρος μέτρησης CO ₂ | 350-10000ppm |
| Τάση για θέρμανση αισθητήρα | 6V |
| Τάση λειτουργίας SEN0159 | 5V |
| Ρεύμα λειτουργίας θέρμανσης MG-811 | 150-200mA |



Εικόνα 24. Η λογαριθμική κλίμακα του MG-811



Εικόνα 25. Κύκλωμα τροφοδοσίας του MG-811



Εικόνα 26. Κύκλωμα ενίσχυσης της εξόδου του MG-811

Στις δύο παραπάνω εικόνες φαίνονται τα ηλεκτρονικά κυκλώματα που ενσωματώνονται στο πλακετάκι SEN0159. Η πρώτη απεικονίζει το κύκλωμα της τροφοδοσίας, το οποίο παίρνει για είσοδο τα 5V και η έξοδος που ανυψώνεται στα 6V. Η ανύψωση της τάσης γίνεται με το ολοκληρωμένο GS1662. Πρόκειται για έναν χαμηλής τάσης DC/DC μετατροπέα με υψηλή απόδοση. Η τάση εξόδου (δηλαδή τα 6V) ρυθμίζεται από τις αντιστάσεις R7 και R9. Η σχέση με την οποία υπολογίζεται η τάση εξόδου είναι η εξής:

$$V_{out} = \left(1 + \frac{R7}{R9}\right) * 1,212 \quad (3.2)$$

Οπότε βάζοντας τιμές στην σχέση (3.2), έχουμε:

$$V_{out} = \left(1 + \frac{R7}{R9}\right) * 1,212 = \left(1 + \frac{158K\Omega}{39,2K\Omega}\right) * 1,212 = (1 + 4,03) * 1,212 = 6,09V$$

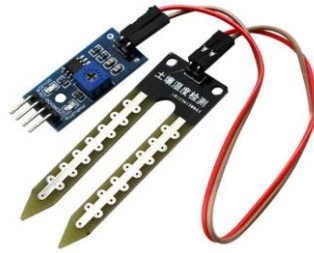
Στη δεύτερη εικόνα φαίνεται το υπόλοιπο κύκλωμα. Βλέπουμε ότι ο αισθητήρας τροφοδοτείται με τα 6V και η έξοδος του πηγαίνει μέσω της αντίστασης R1 στο ολοκληρωμένο LM358 (U1A). Εκεί γίνεται ενίσχυση της τάσης εξόδου του αισθητήρα. Είναι ένας απλός μη-αναστρέφων ενισχυτής, και ο υπολογισμός του γίνεται με την σχέση (3.3).

$$V_{out} = \left(1 + \frac{R3}{R2}\right) * V_{co2} \quad (3.3)$$

Όπου V_{co2} , η τάση εξόδου του αισθητήρα MG-811. Στη συνέχεια, η ενισχυμένη έξοδος του LM358 (U1A), οδηγείται πάλι στο LM358 (U1B) στο ποδαράκι 6. Το U1B λειτουργεί ως συγκριτής. Συγκρίνει την τάση του αισθητήρα με την τάση που έρχεται από το ποτενσιόμετρο. Με το ποτενσιόμετρο ρυθμίζεται το επιθυμητό όριο (threshold) του CO2. Έχουμε ουσιαστικά και μια ψηφιακή έξοδο του σήματος. Έτσι, στην έξοδο του U1B υπάρχει το led D2, με το οποίο μπορούμε να καταλάβουμε πότε έχει ξεπεραστεί το επιθυμητό όριο του CO2. [6][7][9][34][35]

3.2.3 Αισθητήρας υγρασίας εδάφους

Ο αισθητήρας υγρασίας εδάφους είναι ένας αισθητήρας ο οποίος δείχνει την περιεκτικότητα του νερού στο έδαφος. Αποτελείται από δύο αγώγιμες πλάκες που λειτουργούν ως ανιχνευτές. Η αρχή λειτουργίας του βασίζεται στην αλλαγή της αντίστασης (με την παρουσία νερού) μεταξύ των δύο αγώγιμων πλακών. Είναι ένας απλός αισθητήρας και πολύ εύκολος στη χρήση του. Συνήθως, κατά την αγορά συνοδεύεται μαζί με το βοηθητικό πλακετάκι, το οποίο έχει ως είσοδο τον αισθητήρα και ως έξοδο ένα αναλογικό και ένα ψηφιακό σήμα. Το ψηφιακό σήμα ελέγχεται μέσω του ολοκληρωμένου LM393 και ενός ποτενσιόμετρου που ρυθμίζει το threshold. Στην παρούσα εργασία χρησιμοποιείται το αναλογικό σήμα του αισθητήρα. Οπότε, η επεξεργασία των μετρήσεων γίνεται από τον ADC του μικροελεγκτή. [36][52]



Εικόνα 27. Αισθητήρας υγρασίας εδάφους

3.2.4 Αισθητήρας βροχής

Ίδιας λογικής του αισθητήρα υγρασίας εδάφους είναι και ο αισθητήρας ανίχνευσης βροχής. Έχει εκτεθειμένες δύο πλάκες οι οποίες είναι ειδικά διαμορφωμένες και πρέπει να γεφυρωθούν (με τη χρήση νερού) για να αρχίσουν να βγάζουν μετρήσεις. Η μία πλάκα έχει τάση ενώ η άλλη πλάκα είναι το ευαίσθητο κομμάτι του αισθητήρα. Είναι ίδιες ακριβώς και η μόνη διαφορά είναι σε ποια πλάκα θα εφαρμοστεί η τάση. Κατά την αγορά συνοδεύεται μαζί με το βοηθητικό πλακετάκι, το οποίο έχει ως είσοδο τον αισθητήρα και ως έξοδο ένα αναλογικό και ένα ψηφιακό σήμα (ακριβώς όπως και στον αισθητήρα υγρασίας εδάφους). Στην παρούσα εργασία ο αισθητήρας συνδέεται στην αναλογική είσοδο του μικροελεγκτή. [37][53]



Εικόνα 28. Αισθητήρας βροχής

3.2.5 Μηχανικός διακόπτης (Tampet)

Ο μηχανικός διακόπτης (tampet) είναι ένας κλασικός διακόπτης και η χρήση του στην παρούσα εργασία αποσκοπεί στην ανίχνευση της κατάστασης των παραθύρων του θερμοκηπίου. Δηλαδή αν τα παράθυρα είναι ανοιχτά ή κλειστά. [54]



Εικόνα 29. Μηχανικός διακόπτης (tampet)

3.3 Λοιπά εξαρτήματα

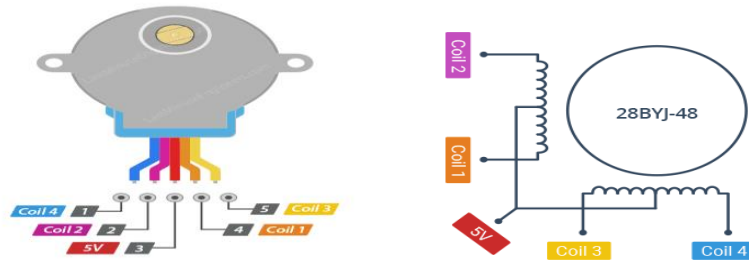
3.3.1 Stepper motor

Ο βηματικός κινητήρας στην παρούσα εργασία χρησιμοποιείται για να ανοιγοκλείνουμε τα παράθυρα του θερμοκηπίου. Το θετικό με τους βηματικούς κινητήρες είναι ότι μπορούμε να τους περιστρέψουμε και να τους τοποθετήσουμε ακριβώς στο σημείο που θέλουμε. Έχουν δηλαδή διακριτά βήματα. Στην παρακάτω εικόνα φαίνεται ο βηματικός κινητήρας (μοντέλο 28BYJ-48). Συνήθως συνοδεύονται με το ολοκληρωμένο ULN2003AN. Το ολοκληρωμένο ULN2003AN είναι ένας driver τον οποίο θα εξηγήσουμε αργότερα.



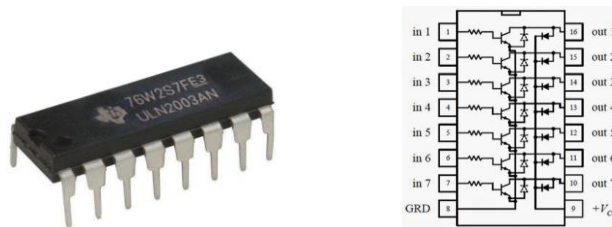
Εικόνα 30. Βηματικός κινητήρας

Είναι ένας κινητήρας που αποτελείται από 5 καλώδια. Έχει δύο πηνία και καθένα από αυτά έχουν μια μεσαία λήψη. Αυτές οι δύο μεσαίες λήψεις ενώνονται εσωτερικά μεταξύ τους και αποτελούν το 5^ο καλώδιο, το κόκκινο όπου είναι και η τροφοδοσία του (5V-240mA). Το ένα άκρο του πηνίου μαζί με το κόκκινο σχηματίζουν μια φάση. Έτσι, το 28BYJ-48 έχει 4 φάσεις. Το κόκκινο καλώδιο είναι πάντα στο HIGH, και όταν κάποιο από τα υπόλοιπα καλώδια τεθεί στο LOW δημιουργείται μια φάση. Επομένως, για να έχουμε περιστροφή του βηματικού κινητήρα πρέπει οι φάσεις να παίρνουν λογικές καταστάσεις HIGH και LOW με κάποιο μοτίβο. Το μοτίβο αυτό υπάρχει και φαίνεται στον πηγαίο κώδικα του STM32 στα Παραρτήματα. [38][55]



Εικόνα 31. Το εσωτερικό του βηματικού κινητήρα

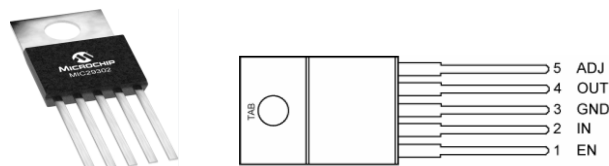
Το ολοκληρωμένο ULN2003AN όπως είπαμε και προηγουμένως είναι ένας driver για το βηματικό κινητήρα. Επειδή η κατανάλωση του βηματικού κινητήρα είναι αρκετά μεγάλη (5V/240mA) δεν μπορεί να οδηγηθεί απευθείας από τον μικροελεγκτή. Επομένως, το ULN2003AN αφού είναι ικανό να βγάλει υψηλή τάση και αρκετό ρεύμα, μπορεί να οδηγήσει το βηματικό κινητήρα πολύ εύκολα. Το ολοκληρωμένο ULN2003AN αποτελείται από 7 ζεύγη ανοιχτού συλλέκτη συνδεσμολογίας Darlington. Κάθε μια συνδεσμολογία μπορεί να οδηγήσει φορτίο μέχρι και 50V/500mA. [10][56][57]



Εικόνα 32. Το ολοκληρωμένο ULN2003AN

3.3.2 MIC29302WT

Το MIC29302WT είναι ένας ειδικός σταθεροποιητής με ρυθμιζόμενη τάση εξόδου. Χρησιμοποιείται στην παρούσα εργασία για την σταθεροποίηση της τάσης στα 4,1V όπως ορίζει ο κατασκευαστής του SIM800L module. [11][58]



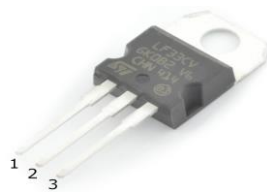
Εικόνα 33. Ο σταθεροποιητής MIC29302WT

Πίνακας 9. Χαρακτηριστικά του MIC29302WT

| Pin | Περιγραφή |
|----------------------------|----------------------------|
| Pin ενεργοποίησης Enable | Έως V_{in} |
| Τάση εισόδου (V_{in}) | Έως 26V |
| Τάση εξόδου (V_{out}) | Ρυθμιζόμενο (για εδώ 4,1V) |
| Ρεύμα εξόδου (I_{out}) | Έως 3A |

3.3.3 LF33CV

Το LF33CV είναι ένας σταθεροποιητής στα 3,3V και τροφοδοτεί τα περισσότερα εξαρτήματα του κυκλώματος. Παρακάτω φαίνονται τα ηλεκτρικά του χαρακτηριστικά. [12][59]



Εικόνα 34. Ο σταθεροποιητής LF33CV

Πίνακας 10. Χαρακτηριστικά του LF33CV

| Pin | Περιγραφή |
|---------------------------|-----------|
| Τάση εξόδου (Vout, pin 3) | 3,3V |
| Τάση εισόδου (Vin, pin 1) | 16V (max) |
| GND (pin 2) | Ground |
| Ρεύμα εξόδου (Iout) | 1A |

3.3.4 Οθόνη ST7735

Η TFT-LCD οθόνη (μοντέλο ST7735) είναι μια έγχρωμη οθόνη 1,8 ιντσών (128x160 pixels). Το ST7735 είναι ένας driver για οθόνες που υποστηρίζουν γραφικά με 262000 χρώματα. Επικοινωνεί με τον μικροελεγκτή μέσω του πρωτοκόλλου επικοινωνίας SPI (Serial Peripheral Interface), οπότε μπορεί να συνδεθεί απευθείας. Τα δεδομένα της οθόνης μπορούν να αποθηκευτούν στη μνήμη RAM που διαθέτει. Μπορεί να εκτελέσει λειτουργίες read/write χωρίς εξωτερικό clock ελαχιστοποιώντας έτσι την κατανάλωση ενέργειας. Τέλος, λόγω των ολοκληρωμένων κυκλωμάτων τροφοδοσίας που είναι απαραίτητα για την οδήγηση των υγρών κρυστάλλων, υπάρχει η δυνατότητα υλοποίησης ενός συστήματος οθόνης με λίγα εξαρτήματα. [13][39][60]



Εικόνα 35. Η οθόνη ST7735

Πίνακας 11. Ηλεκτρικά χαρακτηριστικά του ST7335

| Pin | Περιγραφή |
|-------|------------------------|
| Led | 3,3V |
| SCK | Serial clock (SCK) SPI |
| SDA | Serial data (SDA) SPI |
| DC | Data Commands |
| Reset | Reset |
| CS | Chip Select |
| GND | Ground |
| Vcc | 3,3V |

3.3.5 Τροφοδοσία

Ολόκληρο το σύστημα του θερμοκηπίου τροφοδοτείται από το τροφοδοτικό της εταιρείας Mean Well. Το μοντέλο είναι HDR-60-5. Πρόκειται για ένα οικονομικό τροφοδοτικό αρκετά καλής ποιότητας με 91% απόδοση. Στην παρακάτω εικόνα φαίνεται το τροφοδοτικό και ύστερα τα ηλεκτρικά του χαρακτηριστικά. [14][61]



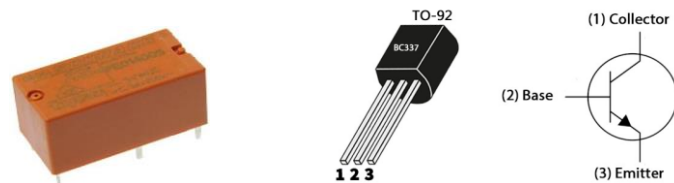
Εικόνα 36. Το τροφοδοτικό MeanWell

Πίνακας 12. Ηλεκτρικά χαρακτηριστικά του τροφοδοτικού MeanWell

| Παράμετρος | Τιμή |
|----------------------------|------------|
| Τάση εισόδου (V_{in}) | 100-240VAC |
| Ρεύμα εισόδου (I_{in}) | 1,8A |
| Συχνότητα | 50/60Hz |
| Τάση εξόδου (V_{out}) | 5VDC |
| Ρεύμα εξόδου (I_{out}) | 6,5A |

3.3.6 Ρελέ και τρανζίστορ

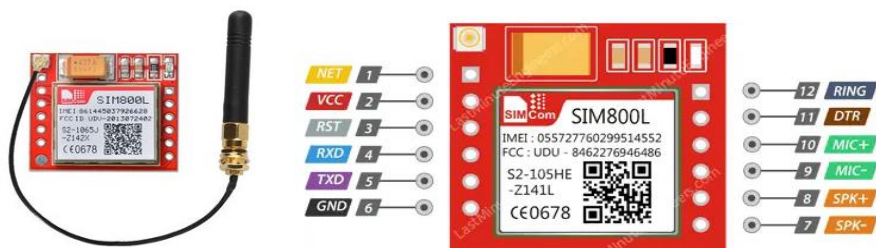
Επειδή στην πλακέτα ενσωματώνονται ρελέ για την ενεργοποίηση/απενεργοποίηση λειτουργιών του θερμοκηπίου, καλό θα ήταν να αναφερθούν ποια ακριβώς μοντέλα από το κάθε είδος χρησιμοποιήθηκαν. Το ρελέ που χρησιμοποιήθηκε είναι της εταιρείας SCHRACK το μοντέλο PE014005, και πρόκειται για ένα 5V/5A ρελέ. Ενώ το τρανζίστορ που επιλέχθηκε (για την όπλιση αυτού του ρελέ) είναι το BC337 (NPN τρανζίστορ). Παρακάτω φαίνονται αυτά τα δύο εξαρτήματα. [15][16][62][63]



Εικόνα 37. Το ρελέ PE014005 και το τρανζίστορ BC337

3.3.7 SIM800L

Στην παρούσα εργασία ένα εξάρτημα που χρησιμοποιήθηκε ακόμα, είναι το module SIM800L. Πρόκειται για ένα GSM module της εταιρείας SIMCOM. Είναι απλό στη χρήση του και επικοινωνεί με τον μικροελεγκτή μέσω της σειριακής θύρας. Οι εντολές που στέλνονται στο SIM module λέγονται «AT+Commands». Οι εντολές αυτές υπάρχουν στο διαδίκτυο, είναι πάρα πολλές και σχετίζονται με τις λειτουργίες της μονάδας (π.χ. αν υπάρχει σύνδεση με το τηλεφωνικό δίκτυο, αποστολή μηνύματος SMS, κ.α.). Η χρήση του στην παρούσα εργασία έγινε για αποστολή μηνυμάτων SMS σε καταστάσεις «ALERT» στον διαχειριστή σχετικά με το θερμοκήπιο (π.χ. πολύ χαμηλή ή πολύ υψηλή θερμοκρασία κ.α.). Μπορεί να θεωρείται απλή, αλλά είναι αρκετά σημαντική η αποστολή μηνυμάτων SMS. Είναι σημαντική επειδή μπορεί να προκύψει κάποιο σοβαρό ζήτημα και ο διαχειριστής του θερμοκηπίου να ξεχαστεί να συνδεθεί στο account του για να δει τις μετρήσεις.



Εικόνα 38. Το SIM800L

Το παραπάνω module έχει ενσωματωμένο στην πλακέτα ένα κόκκινο led το οποίο δείχνει την κατάσταση της μονάδας. Υπάρχουν τρεις καταστάσεις και αυτές είναι οι εξής:

- **Αναβοσβήνει κάθε 1s**, είναι σε λειτουργία αλλά δεν υπάρχει σύνδεση στο δίκτυο κινητής τηλεφωνίας.
- **Αναβοσβήνει κάθε 2s**, η σύνδεση δεδομένων GPRS είναι ενεργή.
- **Αναβοσβήνει κάθε 3s**, η μονάδα είναι συνδεδεμένη στο δίκτυο κινητής τηλεφωνίας και είναι έτοιμο για αποστολή/λήψη μηνυμάτων ή και κλήσεων. [17][40]

Πίνακας 13. Ηλεκτρικά χαρακτηριστικά του SIM800L

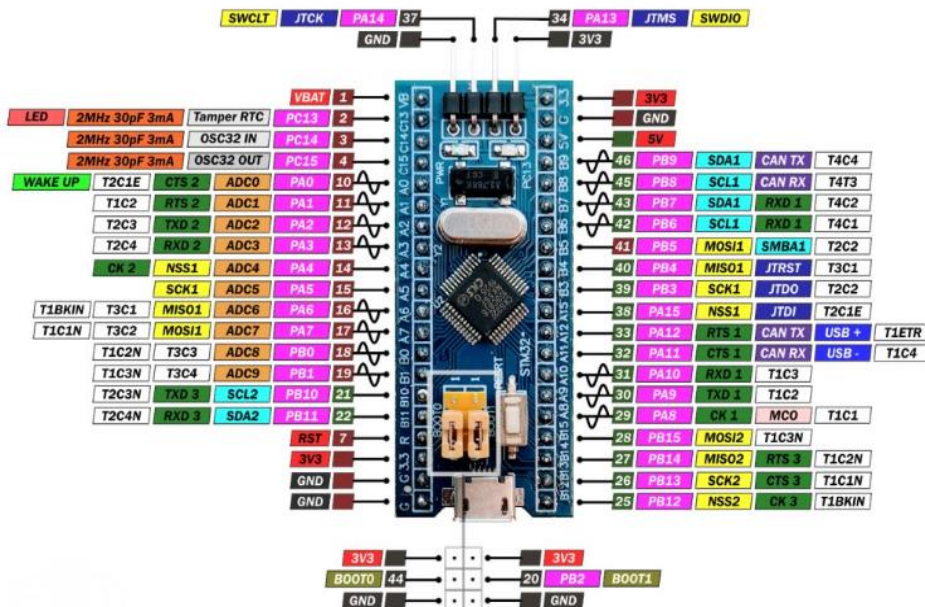
| Παράμετρος | Τιμή |
|------------------------|------------------------------|
| Τάση λειτουργίας (Vcc) | 3,4V - 4,4V (επιθυμητή 4,1V) |
| Κατανάλωση | Έως και 2A |
| Κατάσταση αδράνειας | 1mA |

Πίνακας 14. Pinout του SIM800L

| Pin | Περιγραφή |
|-------------|-----------------------------|
| NET | Σύνδεση ελικοειδούς κεραίας |
| VCC | Τροφοδοσία |
| RST | Reset |
| RXD | Receive (USART) |
| TXD | Transmit (UASRT) |
| GND | Ground |
| SPK- / SPK+ | Σύνδεση ακουστικού |
| MIC- / MIC+ | Σύνδεση μικροφώνου |
| DTR | Έλεγχος του Sleep mode |
| RING | Ενδείκτης κλήσεων |

3.3.8 STM32F103C8T6

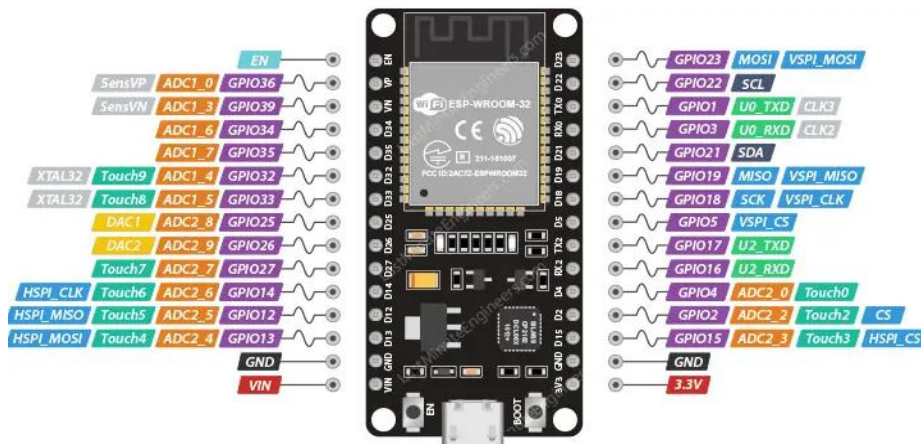
Το αναπτυξιακό πλακετάκι της εταιρείας ST αποτελεί τον πυρήνα του συστήματος του θερμοκηπίου. Εκεί γίνονται όλες οι απαραίτητες μετρήσεις και λειτουργίες. Πρόκειται για έναν μικροελεγκτή υψηλής απόδοσης με επεξεργαστή τον Cortex-M3 που λειτουργεί σε συχνότητα 72MHz, με ενσωματωμένες μνήμες υψηλής ταχύτητας (μνήμη FLASH έως και 128 KByte και SRAM έως 20 KByte) και μια εκτεταμένη γκάμα από I/Os. Έχει δύο 12-bit ADCs και το καθένα από 10 κανάλια. Ακόμα έχει τρεις 16-bit timers γενικής χρήσης και επιπλέον ένας PWM timer. Επίσης, τα πρωτόκολλα επικοινωνίας που διαθέτει είναι: δύο I2C και δύο SPI, τρία USART, ένα USB και ένα CAN. Το εύρος της τάσης τροφοδοσίας είναι από 2V έως 3,6V. Ο μικροελεγκτής έχει δυνατότητα λειτουργίας με χαμηλή κατανάλωση όπου αυτή απαιτηθεί (low power). Αυτά είναι κάποια από τα πολλά χαρακτηριστικά που διαθέτει ο μικροελεγκτής STM32F103C8T6 που τον καθιστούν κατάλληλο σε ένα ευρύ φάσμα εφαρμογών. [41][64]



Εικόνα 39. Αναπτυξιακή πλακέτα STM32F103C8T6

3.3.9 ESP32

Το αναπτυξιακό πλακετάκι ESP32-WROOM-32 είναι μια εξίσου ισχυρή μονάδα με μικροεπεξεργαστή που ενσωματώνει WiFi, Bluetooth και BLE (Bluetooth Low Energy). Η λειτουργία WiFi το καθιστά ικανό να εφαρμόζεται σε πολλούς τομείς, όπως σε δίκτυα αισθητήρων με χαμηλή κατανάλωση έως και αρκετά απαιτητικές διεργασίες (π.χ. αποκωδικοποίηση MP3, κωδικοποίηση φωνής, κ.α.). Το module ESP32 αποτελεί τον πυρήνα από αυτό το πλακετάκι. Πρόκειται για έναν διπύρηνο επεξεργαστή 32-bit με clock από 80MHz έως και 240MHz, όπου ο έλεγχος τους μπορεί να γίνει μεμονωμένα. Έχει ενσωματωμένο ένα ευρύ φάσμα περιφερειακών, όπως I2C, SPI, UART, PWM, ADC, pins για χωρητικούς αισθητήρες αφής (capacitive touch sensors), διεπαφή για κάρτα SD, και πάρα πολλά ακόμη. Το εύρος της τάσης λειτουργίας είναι από 3V έως 3,6V και μέση κατανάλωση 80mA. [18][65]



Εικόνα 40. Αναπτυξιακή πλακέτα ESP32

3.4 Επίλογος

Στο κεφάλαιο αυτό έγινε ανάλυση διάφορων κομματιών του ηλεκτρονικού κυκλώματος που σχεδιάστηκε. Πιο συγκεκριμένα, παρουσιάστηκαν οι αισθητήρες που χρησιμοποιήθηκαν, η αρχή λειτουργίας τους και ποια είναι η χρήση τους. Επιπλέον αναφέρθηκαν τα λοιπά εξαρτήματα κατόπιν και οι αναπτυξιακές πλακέτες που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος.

Κεφάλαιο 4ο: Σχεδιασμός και προγραμματισμός κυκλώματος

4.1 Εισαγωγή

Αφού στο προηγούμενο κεφάλαιο αναλύθηκαν όλα τα εξαρτήματα που χρησιμοποιήθηκαν, σε αυτό το κεφάλαιο παρατίθεται η ανάπτυξη του συστήματος. Αρχικά, αναλύεται ο σχεδιασμός του ηλεκτρονικού κυκλώματος. Στη συνέχεια, αναφέρονται οι προγραμματιστικές ρυθμίσεις που απαιτούνται για τον μικροελεγκτή STM32 στο περιβάλλον STM32CubeIDE, καθώς και τα διαγράμματα ροής των πηγαίων κωδίκων του STM32 και του ESP32.

4.2 Σχεδιασμός κυκλώματος

Ο μικροελεγκτής που χρησιμοποιήθηκε είναι ο STM32F103C8T6 της ST, ο οποίος αποτελεί και τον πυρήνα του κυκλώματος. Αν και το κύκλωμα αποτελείται από αρκετά εξαρτήματα, ο μικροελεγκτής θεωρείται πυρήνας επειδή είναι υπεύθυνος για πολλές λειτουργίες (συλλογή μετρήσεων, απεικόνιση στην οθόνη, απόφαση αποστολής SMS και αποστολή μετρήσεων στην ιστοσελίδα, κ.α.).

Στα σημεία POWER_SUPPLY-1 και POWER_SUPPLY-2 εφαρμόζεται η τροφοδοσία του κυκλώματος η οποία είναι 5V. Στα 5V συνδέεται και ο αισθητήρας CO2 μέσω των κλεμών CO2-2 και CO2-3. Ενώ το σημείο CO2-1 πηγαίνει στην αναλογική είσοδο PA0 του μικροελεγκτή. Στα pin-headers με ονομασία TFT συνδέεται η οθόνη όπου και απεικονίζει τις μετρήσεις των αισθητηρίων. Η τροφοδοσία της οθόνης είναι τα 3,3V και εφαρμόζεται στα σημεία TFT-7 και TFT-8. Τα pins TFT-2, TFT-3, TFT-4, TFT-5 και TFT-6 πηγαίνουν στα ποδαράκια PA5, PA7, PB12, PB13 και PB14 του μικροελεγκτή αντίστοιχα, για τον χειρισμό της οθόνης. Το pin TFT-1 πρέπει να συνδεθεί στην τροφοδοσία. Η τάση των 3,3V παρέχεται από τον σταθεροποιητή LF33CV, ο οποίος τροφοδοτείται από τα 5V.

Επειδή το ρεύμα του μικροελεγκτή δεν είναι αρκετό για να οδηγήσει τους δύο βηματικούς κινητήρες, τα ολοκληρωμένα IC3 και IC4 είναι οι drivers (ULN2003AN) που λύνουν αυτό το πρόβλημα. Τα pins PC15, PC14, PC13 και PB8 του μικροελεγκτή συνδέονται στις εισόδους I1, I2, I3 και I4 του IC3 αντίστοιχα. Οι αντίστοιχοι έξοδοι του driver συνδέονται στα καλώδια του βηματικού κινητήρα στα χρώματα μπλε, ροζ, κίτρινο και πορτοκαλί αντίστοιχα. Ομοίως έχουμε και για το δεύτερο βηματικό κινητήρα, ο οποίος ελέγχεται μέσω του driver IC4. Στον παρακάτω πίνακα φαίνεται το μοτίβο που πρέπει να ακολουθηθεί για να περιστραφεί ο βηματικός κινητήρας (περισσότερο κατανοητό στον κώδικα του STM32 στα Παραρτήματα).

Πίνακας 15. Μοτίβο για την περιστροφή του βηματικού κινητήρα

| Βήμα | Είσοδος driver 1η | Είσοδος driver 2η | Είσοδος driver 3η | Είσοδος driver 4η |
|------|-------------------|-------------------|-------------------|-------------------|
| 1 | 1 | 1 | 0 | 0 |
| 2 | 0 | 1 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 |

Το MIC29302WT είναι ένας ειδικός σταθεροποιητής, οποίος χρησιμοποιήθηκε για την τροφοδοσία του SIM800L. Η τροφοδοσία που χρειάζεται το SIM module είναι 4,1V. Η τάση αυτή υπολογίζεται με την σχέση (4.1).

$$R8 = R12 * \left(\frac{V_{out}}{1,24} - 1 \right) \quad (4.1)$$

Οπότε θέτοντας την αντίσταση $R12=43K\Omega$ και την επιθυμητή τάση εξόδου $V_{out}=4,1V$, έχουμε:

$$R8 = R12 * \left(\frac{V_{out}}{1,24} - 1 \right) = 43 * \left(\frac{4,1}{1,24} - 1 \right) = 43 * (3,3 - 1) = 43 * 2,3 = 99,17K\Omega$$

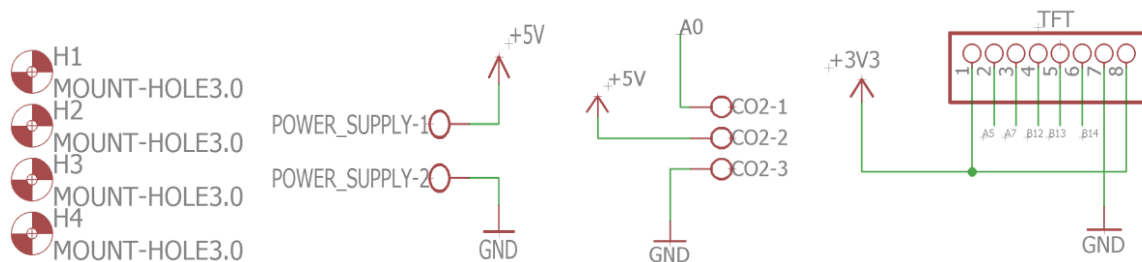
Επομένως, η εμπορική τιμή της αντίστασης R8 πρέπει να είναι 100KΩ. Το SIM module επικοινωνεί με τον μικροελεγκτή σειριακά (μέσω USART2) και οι εντολές που δέχεται είναι οι AT Commands. Το tact switch που είναι συνδεδεμένο στο SIM800L υπάρχει σε περίπτωση που χρειαστεί να το κάνουμε reset.

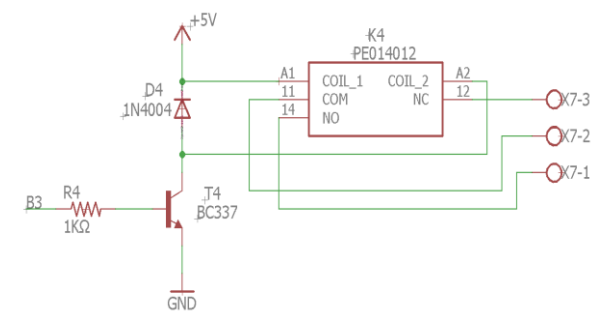
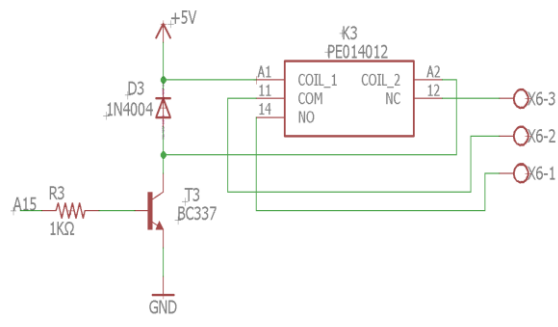
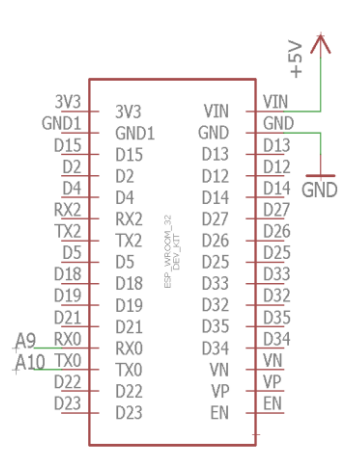
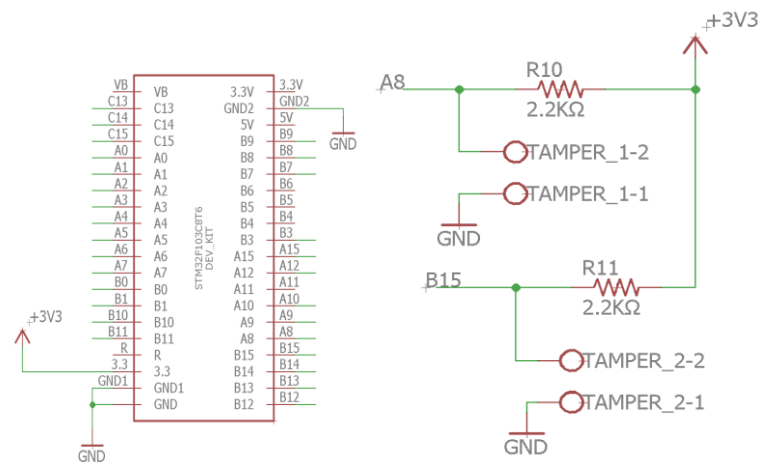
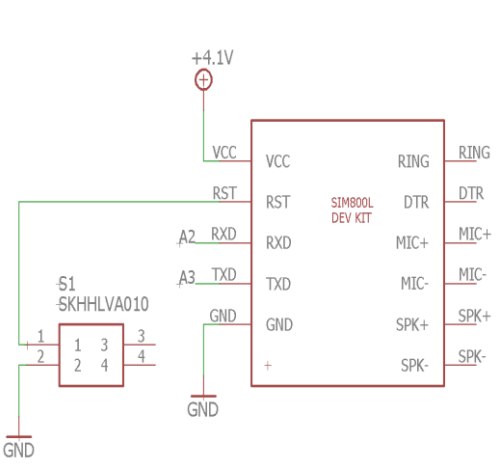
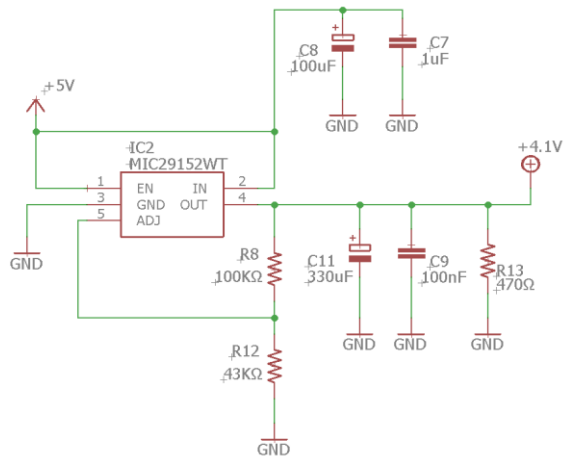
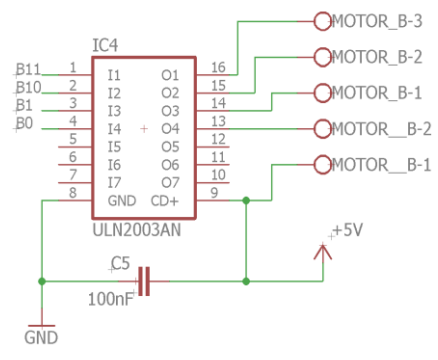
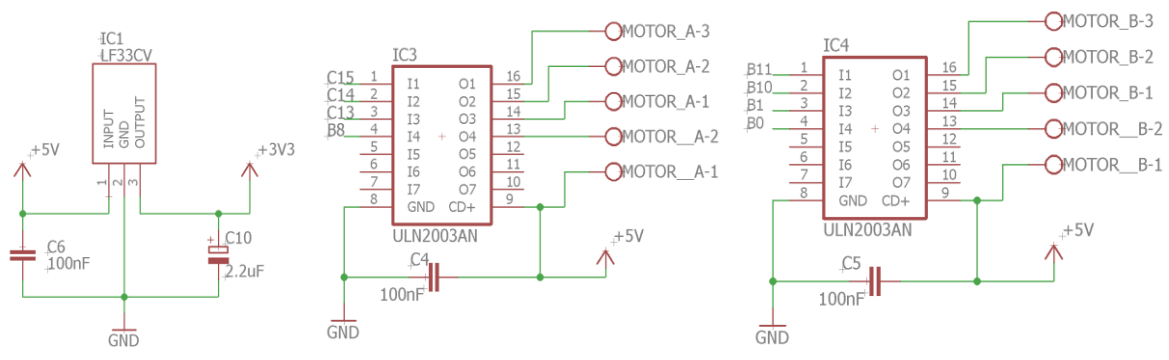
Τα δύο tamper που συνδέονται στα pins PA8 και PB15 του μικροελεγκτή, χρησιμεύουν στο να γνωρίζουμε εάν τα παράθυρα του θερμοκηπίου είναι είτε ανοιχτά είτε κλειστά. Όλες οι μετρήσεις που συλλέγει ο μικροελεγκτής αποστέλλονται στο ESP32, όπου και αυτός με τη σειρά του τα στέλνει προς την ιστοσελίδα. Η επικοινωνία του STM32 με το ESP32 γίνεται σειριακά (μέσω USART1).

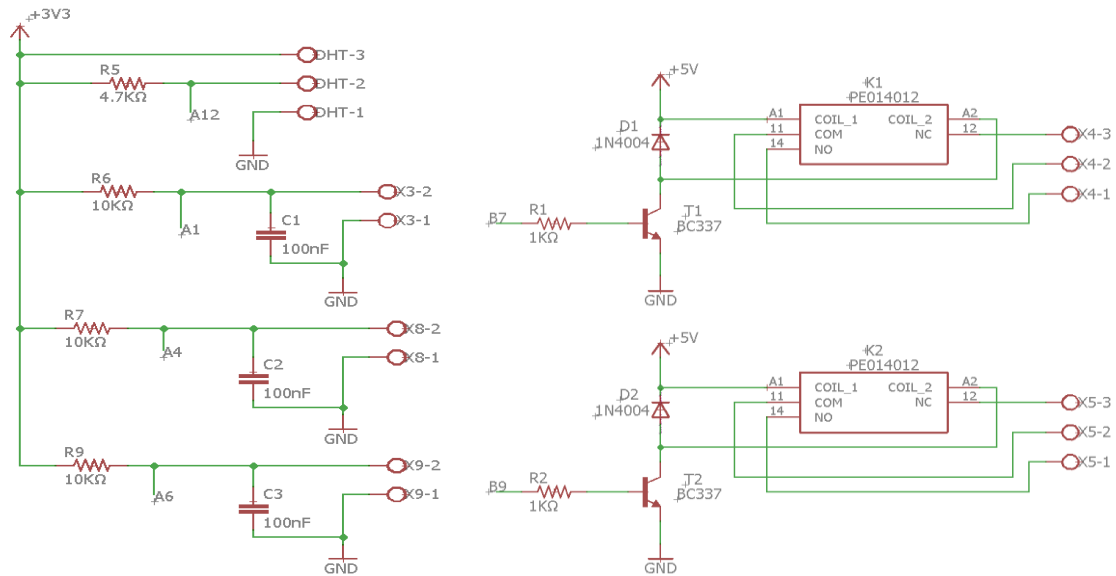
Στη συνέχεια έχουμε 4 αισθητήρες οι οποίοι τροφοδοτούνται με 3,3V. Ο ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας DHT22 συνδέεται στο pin PA12 του μικροελεγκτή και επικοινωνεί μέσω 1-Wire. Στα αναλογικά pins PA1 και PA4 του μικροελεγκτή συνδέονται οι αισθητήρες υγρασίας εδάφους και στο pin PA6 ο αισθητήρας βροχής.

Τέλος, υπάρχουν τέσσερα ρελέ όπου δίνεται η δυνατότητα ενεργοποίησης/απενεργοποίησης κάποιων λειτουργιών του θερμοκηπίου. Με το ρελέ συνδεδεμένο στο pin PA15 του μικροελεγκτή μπορούμε να αυξήσουμε τη θερμοκρασία εντός του θερμοκηπίου (ενεργοποίηση θέρμανσης), ενώ με το ρελέ στο pin PB9 μπορούμε αν θέλουμε να μειώσουμε τη θερμοκρασία (π.χ. ενεργοποίηση ανεμιστήρα). Με το pin PB3 δίνεται η δυνατότητα όπλισης του ρελέ για προσθήκη CO2 στο θερμοκήπιο. Η παρακολούθηση του CO2 είναι πολύ σημαντική σε ένα θερμοκήπιο. Και το pin PB7 του μικροελεγκτή μπορεί να οπλίσει το ρελέ για την ενεργοποίηση κάποιας ηλεκτροβάνας σε περίπτωση που θέλει ο διαχειριστής του θερμοκηπίου να ποτίσει.

Η ανάπτυξη του πηγαίου κώδικα για τον μικροελεγκτή έγινε στο περιβάλλον STM32CubeIDE και ενσωματώθηκε μέσω του ST-Link από τα pins PA13 και PA14. Παρακάτω απεικονίζεται το ηλεκτρονικό κύκλωμα που σχεδιάστηκε μέσω του προγράμματος Eagle.







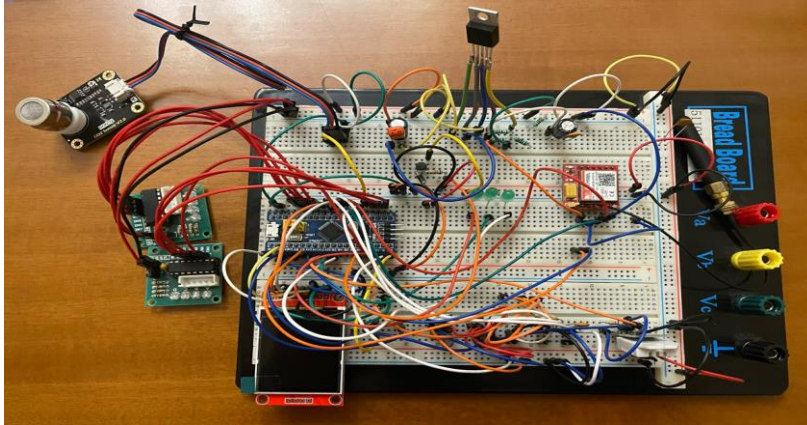
Εικόνα 41. Το ηλεκτρονικό κύκλωμα του συστήματος

Πίνακας 16. Υλικά της πλακέτας

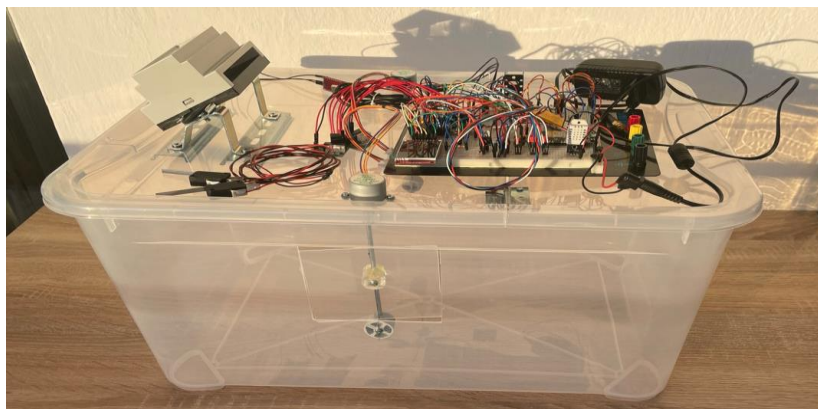
| Υλικό | Τιμή |
|----------------------------|-------------------------------|
| R1, R2, R3, R4 | 1KΩ |
| R5 | 4,7KΩ |
| R6, R7, R9 | 10KΩ |
| R8 | 100KΩ |
| R10, R11 | 2,2KΩ |
| R12 | 43KΩ |
| R13 | 470Ω |
| C1, C2, C3, C4, C5, C6, C9 | 100nF |
| C7 | 1μF |
| C8 | 100μF |
| C10 | 2,2μF |
| C11 | 330μF |
| D1, D2, D3, D4 | 1N4007 |
| T1, T2, T3, T4 | BC337 |
| K1, K2, K3, K4 | PE014005 |
| S1 | Tact Switch 6x6 5.00mm κάθετο |
| IC1 | LF33CV |
| IC2 | MIC29302WT |
| IC3, IC4 | ULN2003AN |
| IC5 | ESP32-WROOM-32 |
| IC6 | STM32F103C8T6 |
| IC7 | SIM800L |

4.3 Υλοποίηση πλακέτας και θερμοκηπίου

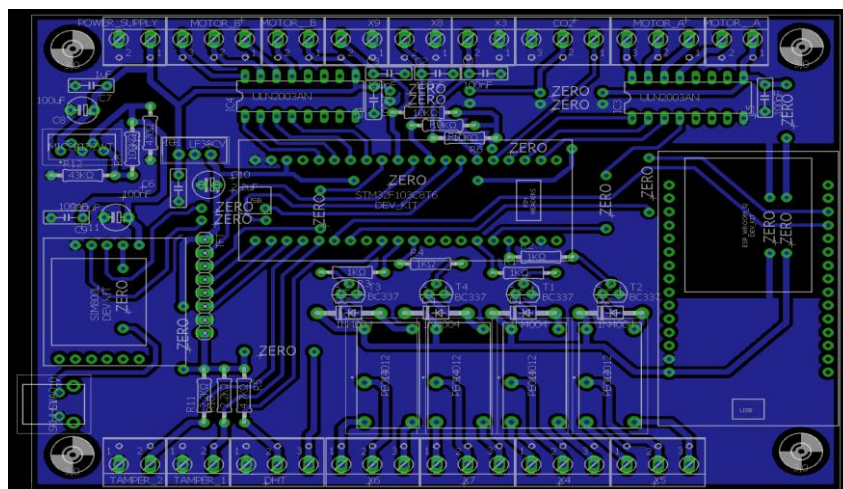
Παρακάτω ακολουθούν μια σειρά εικόνων από την υλοποίηση της πλακέτας η οποία έγινε με τον παραδοσιακό τρόπο, και της κατασκευής του θερμοκηπίου στην οποία εφαρμόζεται η πλακέτα με τους αισθητήρες και τα λοιπά εξαρτήματα.



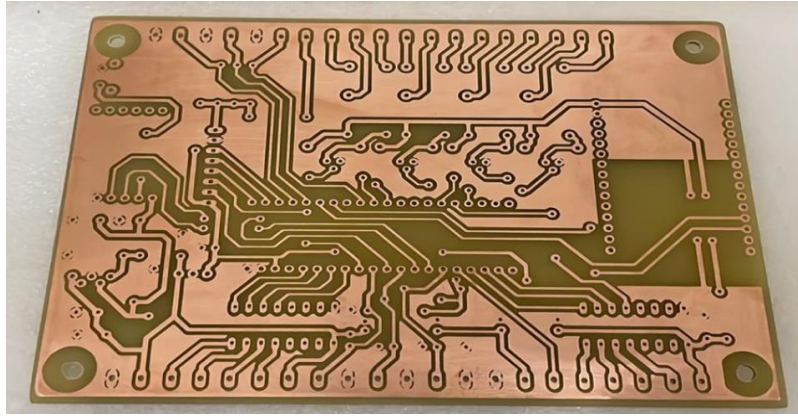
Εικόνα 42. Υλοποίηση κυκλώματος στο ράστερ



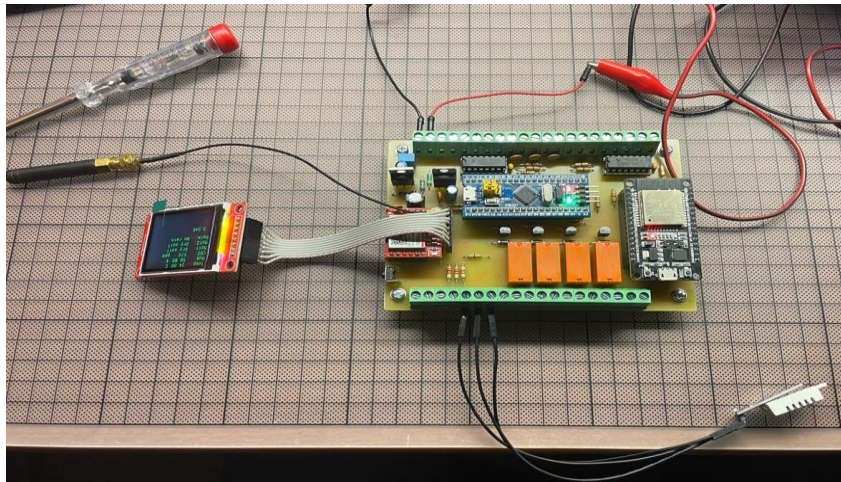
Εικόνα 43. Διαδικασία κατασκευής θερμοκηπίου



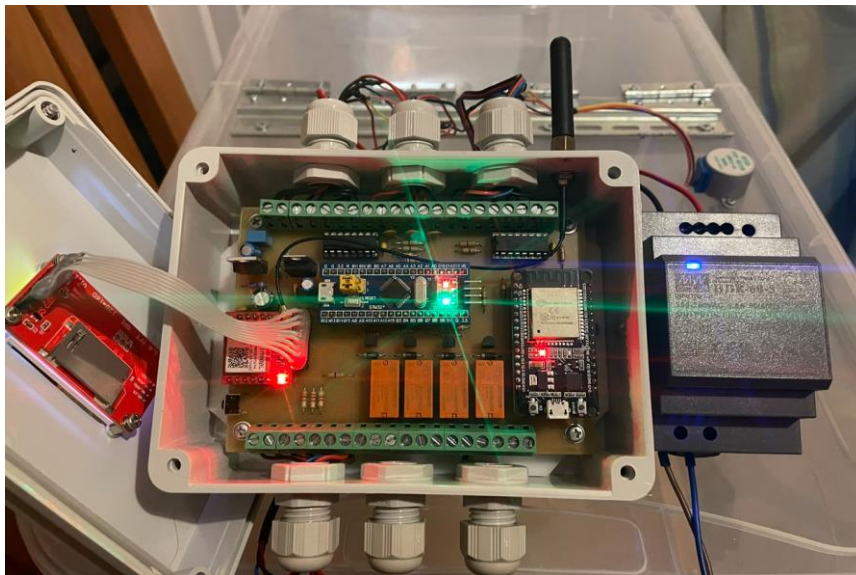
Εικόνα 44. Το σχέδιο της πλακέτας από το Eagle



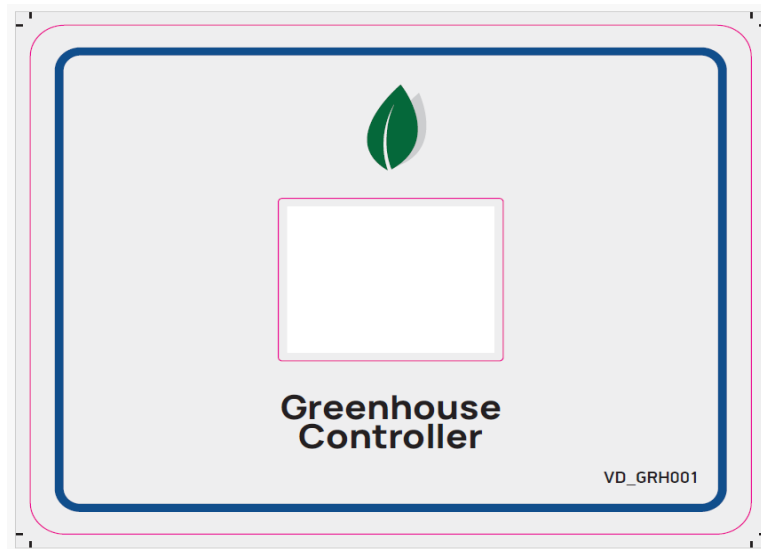
Εικόνα 45. Αποχάλκωση πλακέτας



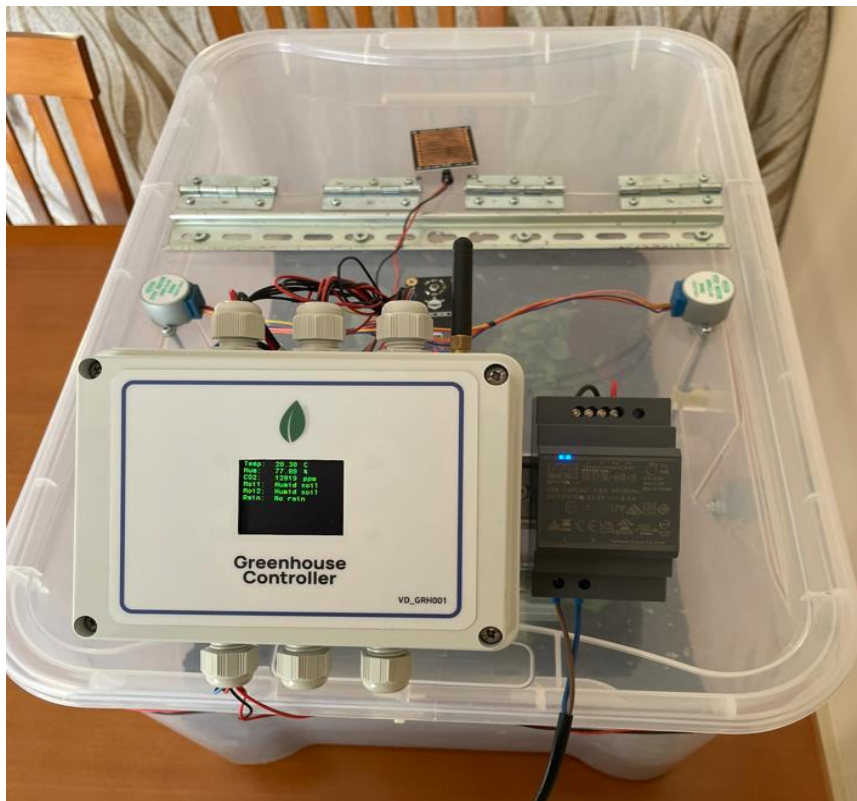
Εικόνα 46. Τοποθέτηση υλικών στην πλακέτα



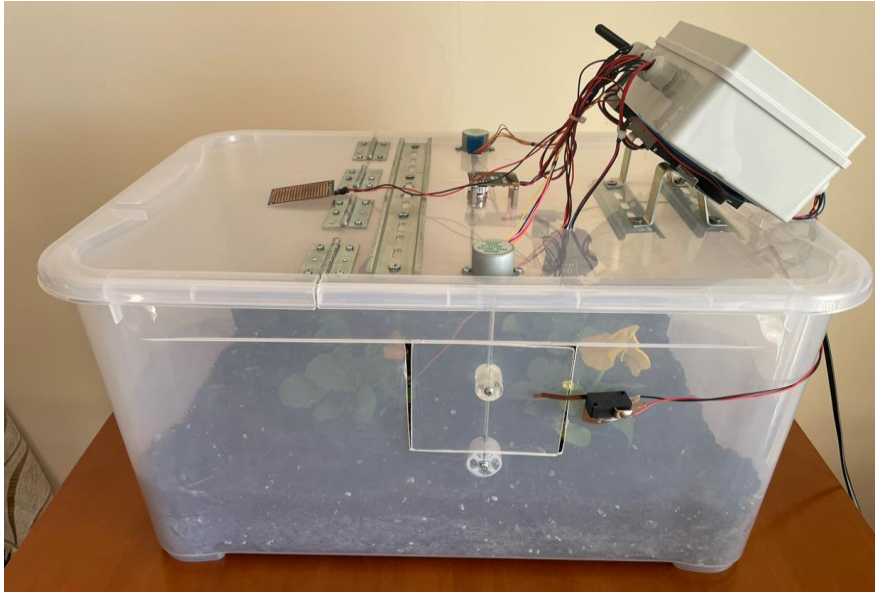
Εικόνα 47. Εφαρμογή της πλακέτας στο θερμοκήπιο



Εικόνα 48. Αυτοκόλλητο πρόσφυγής της συσκευής



Εικόνα 49. Μπροστινή όψη θερμοκηπίου



Εικόνα 50. Πλαϊνή όψη θερμοκηπίου



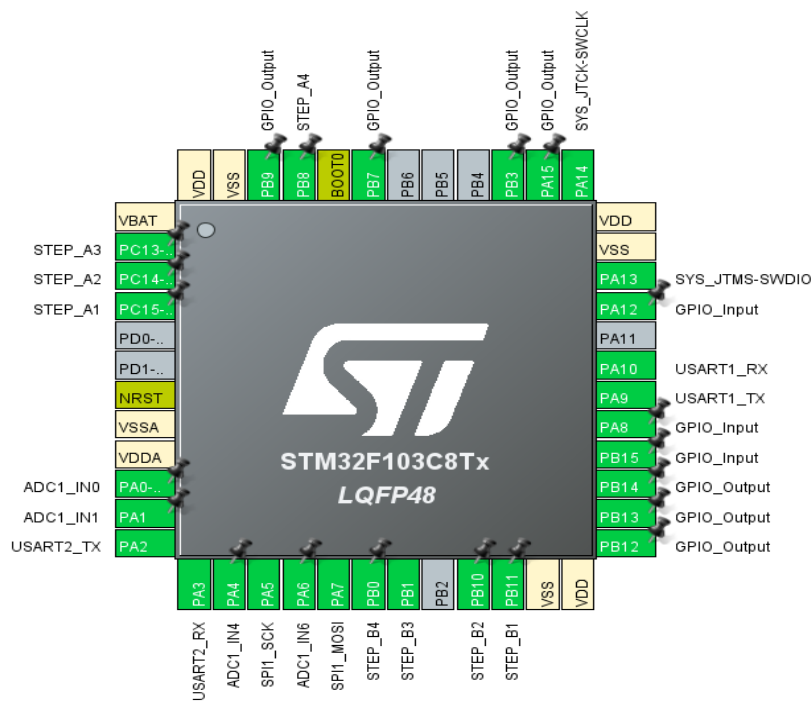
Εικόνα 51. Το εσωτερικό του θερμοκηπίου

4.4 Προγραμματισμός μικροελεγκτών

4.4.1 Ρυθμίσεις STM32F103C8T6

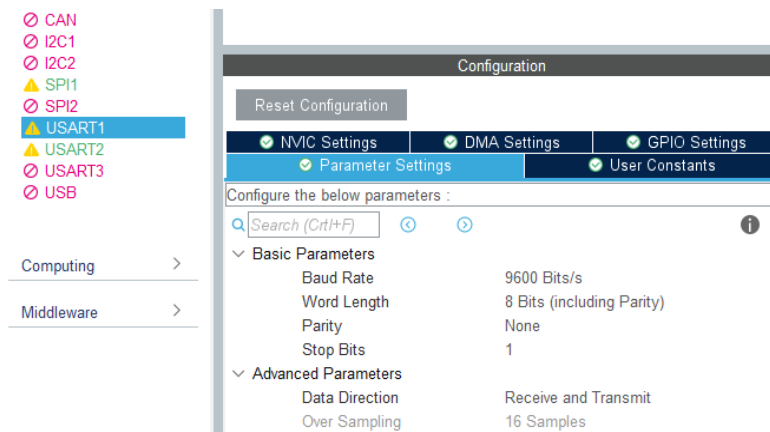
Πίνακας 17. Περιγραφή λειτουργίας των pin

| Pin | Ονομασία | Είσοδος/ Έξοδος | Λειτουργία |
|------|----------------|------------------|-----------------------|
| PA0 | ADC1_IN0 | Είσοδος | CO2 |
| PA1 | ADC1_IN1 | Είσοδος | Υγρασία εδάφους 1 |
| PA2 | USART2_Tx | Έξοδος | SIM800L |
| PA3 | USART2_Rx | Είσοδος | SIM800L |
| PA4 | ADC1_IN4 | Είσοδος | Υγρασία εδάφους 2 |
| PA5 | SPI1_SCK | Έξοδος | Οθόνη |
| PA6 | ADC1_IN6 | Είσοδος | Αισθητήρας βροχής |
| PA7 | SPI1_MOSI | Έξοδος | Οθόνη |
| PA8 | GPIO_Input | Είσοδος | Tamper 1 |
| PA9 | USART1_Tx | Έξοδος | ESP32 |
| PA10 | USART1_Rx | Είσοδος | ESP32 |
| PA12 | GPIO_Input | Είσοδος / Έξοδος | DHT22 |
| PA13 | SYS_JTMS-SWDIO | Έξοδος | ST-LINK |
| PA14 | SYS_JTCK-SWCLK | Έξοδος | ST-LINK |
| PA15 | GPIO_Output | Έξοδος | Ρελέ θέρμανσης |
| PB0 | STEP_B4 | Έξοδος | Βηματικός κινητήρας B |
| PB1 | STEP_B3 | Έξοδος | Βηματικός κινητήρας B |
| PB3 | GPIO_Output | Έξοδος | Ρελέ CO2 |
| PB7 | GPIO_Output | Έξοδος | Ρελέ ποτίσματος |
| PB8 | STEP_A4 | Έξοδος | Βηματικός κινητήρας A |
| PB9 | GPIO_Output | Έξοδος | Ρελέ ψύξης |
| PB10 | STEP_B2 | Έξοδος | Βηματικός κινητήρας B |
| PB11 | STEP_B1 | Έξοδος | Βηματικός κινητήρας B |
| PB12 | GPIO_Output | Έξοδος | Οθόνη (Data Command) |
| PB13 | GPIO_Output | Έξοδος | Οθόνη (Reset) |
| PB14 | GPIO_Output | Έξοδος | Οθόνη (Chip Select) |
| PB15 | GPIO_Input | Είσοδος | Tamper 2 |
| PC13 | STEP_A3 | Έξοδος | Βηματικός κινητήρας A |
| PC14 | STEP_A2 | Έξοδος | Βηματικός κινητήρας A |
| PC15 | STEP_A1 | Έξοδος | Βηματικός κινητήρας A |



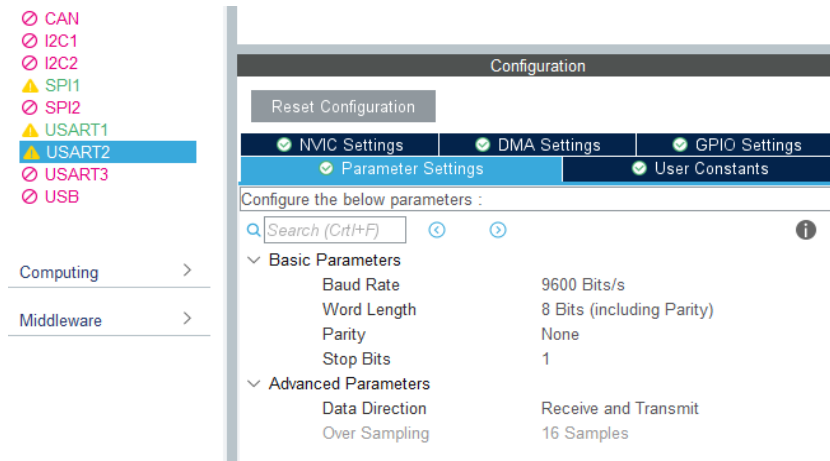
Εικόνα 52. Ρύθμιση των pins του STM32

Στην παρακάτω εικόνα φαίνονται οι ρυθμίσεις που έχουν γίνει για την σειριακή επικοινωνία του STM32 με το ESP32 μέσω του USART1. Είναι ρυθμισμένο σε ασύγχρονη λειτουργία και έχει ενεργοποιηθεί το global interrupt. Το Baud Rate είναι στα 9600Bits/s, 8 Bits, χωρίς Parity Bit και Stop Bits 1.



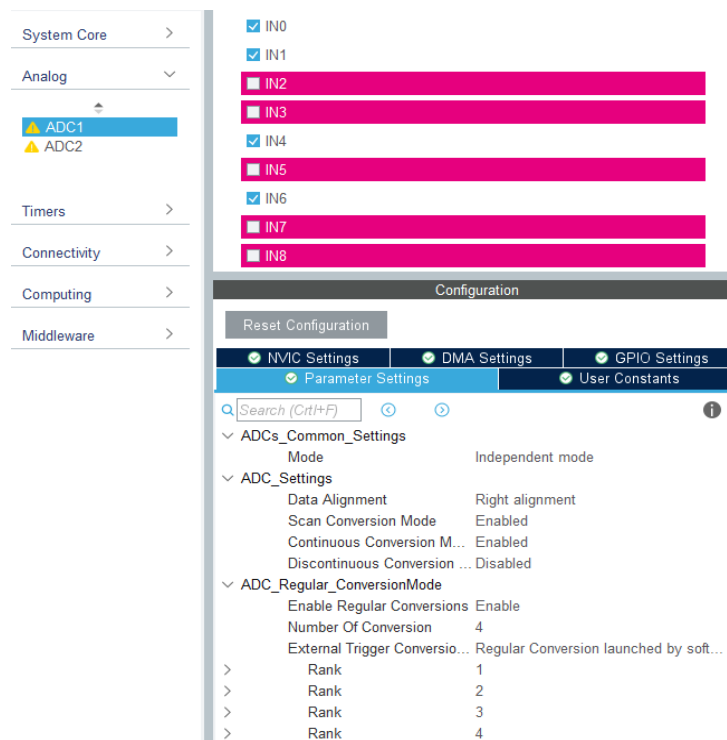
Εικόνα 53. Ρύθμιση USART 1

Στη συνέχεια απεικονίζονται οι ρυθμίσεις που έχουν γίνει για την σειριακή επικοινωνία του STM32 με το SIM800L μέσω του USART2. Ίδιες ρυθμίσεις έχουν γίνει και στο USART2. Δηλαδή ασύγχρονη λειτουργία και ενεργοποιημένο το global interrupt. Baud Rate στα 9600Bits/s, 8 Bits, χωρίς Parity Bit και 1 Stop Bit.



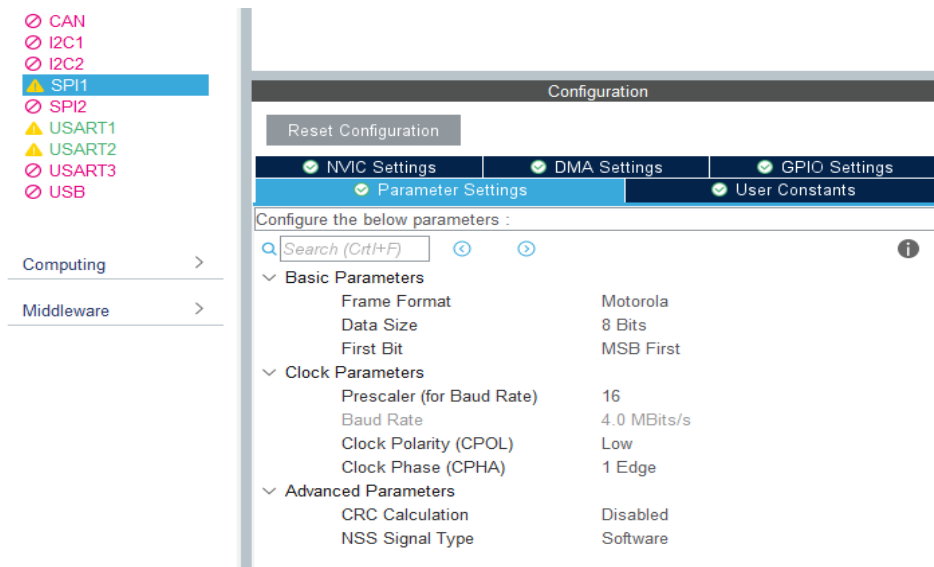
Εικόνα 54. Ρύθμιση USART 2

Για την συλλογή των μετρήσεων από τους αναλογικούς αισθητήρες χρησιμοποιήθηκαν από τον ADC1 τα κανάλια 0, 1, 4 και 6. Τα οποία αντιστοιχούν στο αισθητήριο του CO2, στην υγρασία εδάφους 1, στην υγρασία εδάφους 2, και στον αισθητήρα βροχής. Χρησιμοποιήθηκε 12-Bit ανάλυση (4096 βήματα) για την μετατροπή του αναλογικού σήματος σε ψηφιακό. Στην παρακάτω εικόνα φαίνονται οι ρυθμίσεις που έγιναν για τον ADC1. Έχουν γίνει enable τα Scan Conversion Mode και Continuous Conversion Mode. Και τέλος το Number of Conversion στο 4.



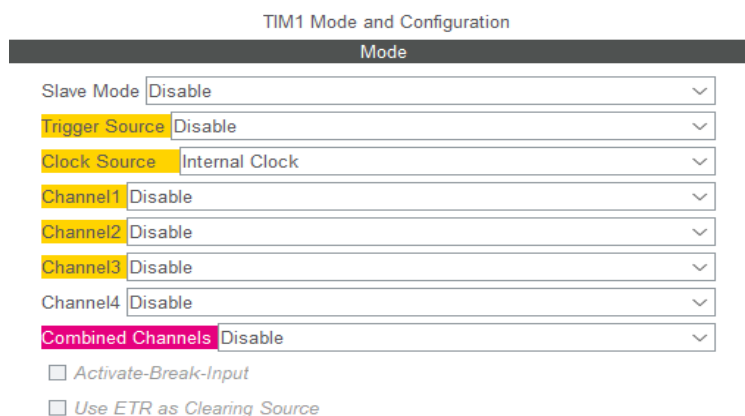
Εικόνα 55. Ρύθμιση ADC

Για την απεικόνιση των μετρήσεων στην οθόνη χρησιμοποιήθηκε η διεπαφή SPI1 όπως φαίνεται παρακάτω. Το mode του SPI έχει ρυθμιστεί ως Half-Duplex Master. Το οποίο σημαίνει ότι τόσο ο μικροελεγκτής όσο και η οθόνη μπορούν να στέλνουν δεδομένα, αλλά όχι ταυτόχρονα.

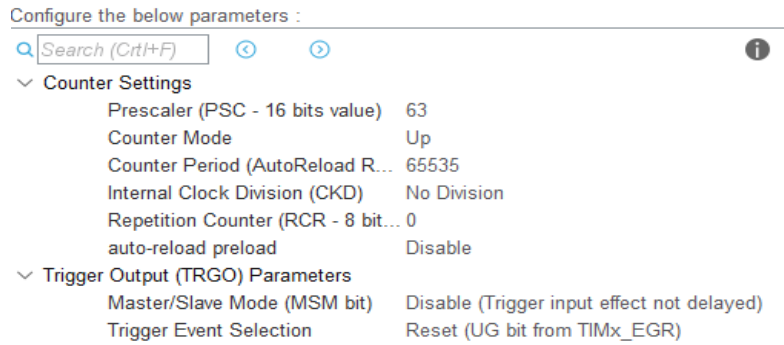


Εικόνα 56. Ρύθμιση SPI

Τέλος, για το αισθητήριο DHT22 χρησιμοποιήθηκε ο Timer 1 του STM32F103C8T6 και ενεργοποιήθηκε το εσωτερικό Clock. Όπως αναλύθηκε σε προηγούμενη ενότητα, το αισθητήριο αυτό επικοινωνεί με τον μικροελεγκτή μέσω πρωτόκολλου 1-Wire. Για να γίνει εφικτή αυτή η επικοινωνία, ο Timer πρέπει να ρυθμιστεί έτσι ώστε να κάνει υπερχείλιση κάθε 1μs. Για να το πετύχουμε αυτό χρησιμοποιήθηκε μια ξεχωριστή ρουτίνα (υπάρχει στο Παράρτημα εντός του πηγαίου κώδικα του STM32). Επειδή το Clock είναι στα 64MHz και για να έχουμε υπερχείλιση κάθε 1μs, τότε η συχνότητα του Timer πρέπει να είναι στο 1MHz ($F = \frac{1}{1\mu s}$). Ουσιαστικά ο counter θα μετρήσει από το μηδέν μέχρι αυτήν την τιμή. Για αυτό και η τιμή του Prescaler είναι 63 (64-1). Πρέπει να σημειωθεί ότι Prescaler θα προσθέσει το 1. Οπότε κάθε μέτρηση θα ισοδυναμεί με 1μs. Παρακάτω φαίνονται οι ρυθμίσεις για τον Timer 1.



Εικόνα 57. Ρύθμιση του Timer (A)



Εικόνα 58. Ρύθμιση του Timer (B)

4.4.2 Διαγράμματα ροής

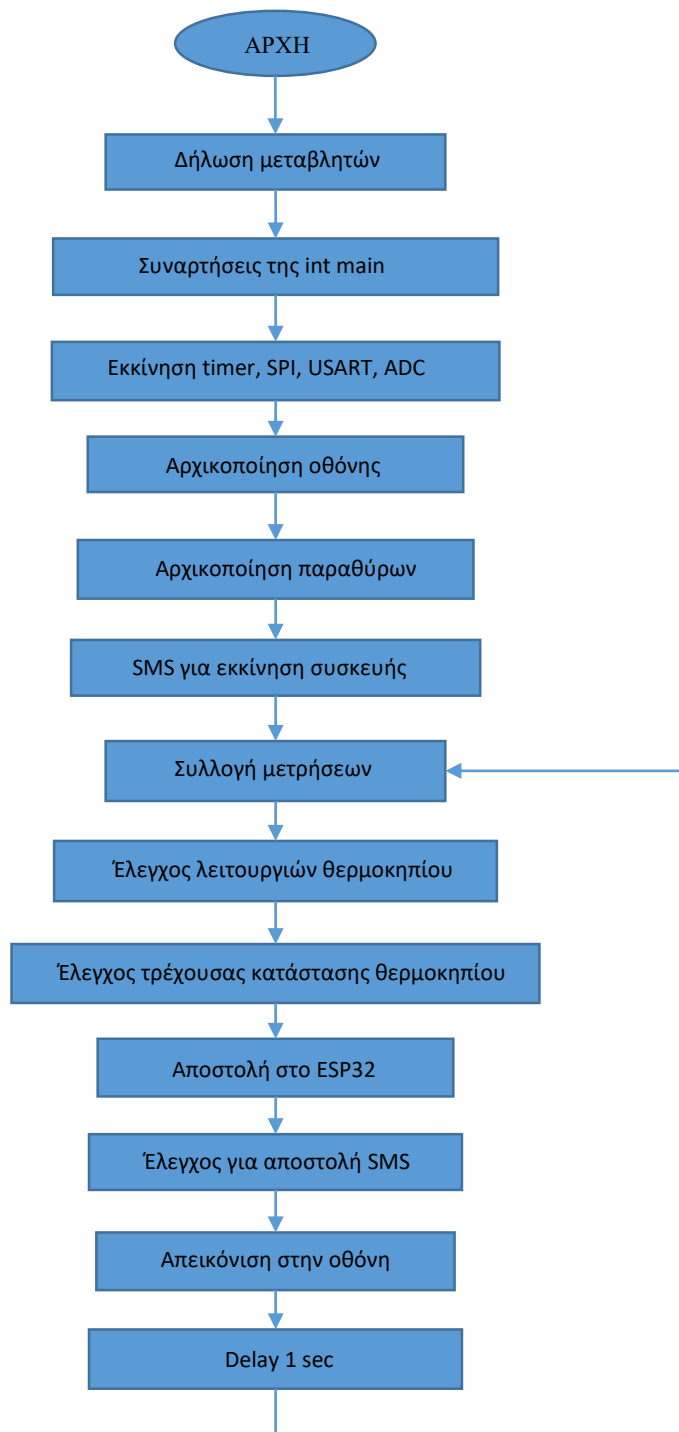
Εκτός από την υλοποίηση του ηλεκτρονικού κυκλώματος, για να μπορέσει να λειτουργήσει το κύκλωμα θα πρέπει να γραφεί και ο ανάλογος κώδικας. Το σύστημα του θερμοκηπίου αποτελείται από δύο μέρη πηγαίου κώδικα. Το ένα μέρος είναι στους μικροελεγκτές και το άλλο στον υπολογιστή που υλοποιεί την ιστοσελίδα. Σε αυτή την ενότητα θα γίνει επεξήγηση του πηγαίου κώδικα των μικροελεγκτών.

Αρχικά, μόλις τροφοδοτηθεί η πλακέτα και πριν ξεκινήσει ο ατέρμων βρόχος του κώδικα, εκτελούνται κάποιες ενέργειες. Η αρχικοποίηση όλων των περιφερειακών (SPI, TIMER, USART, ADC, κτλ.), η αρχικοποίηση της οθόνης, το κλείσιμο των παραθύρων (για να υπάρχει ο έλεγχος της θέσης των βηματικών κινητήρων) και η αποστολή SMS στον διαχειριστή του θερμοκηπίου για να λάβει γνώση ότι έγινε εκκίνηση/επανεκκίνηση της συσκευής (έτσι θα γνωρίζει εάν είχε κοπεί το ρεύμα) είναι οι πρώτες ενέργειες του κώδικα. Έπειτα ακολουθεί ο ατέρμων βρόχος.

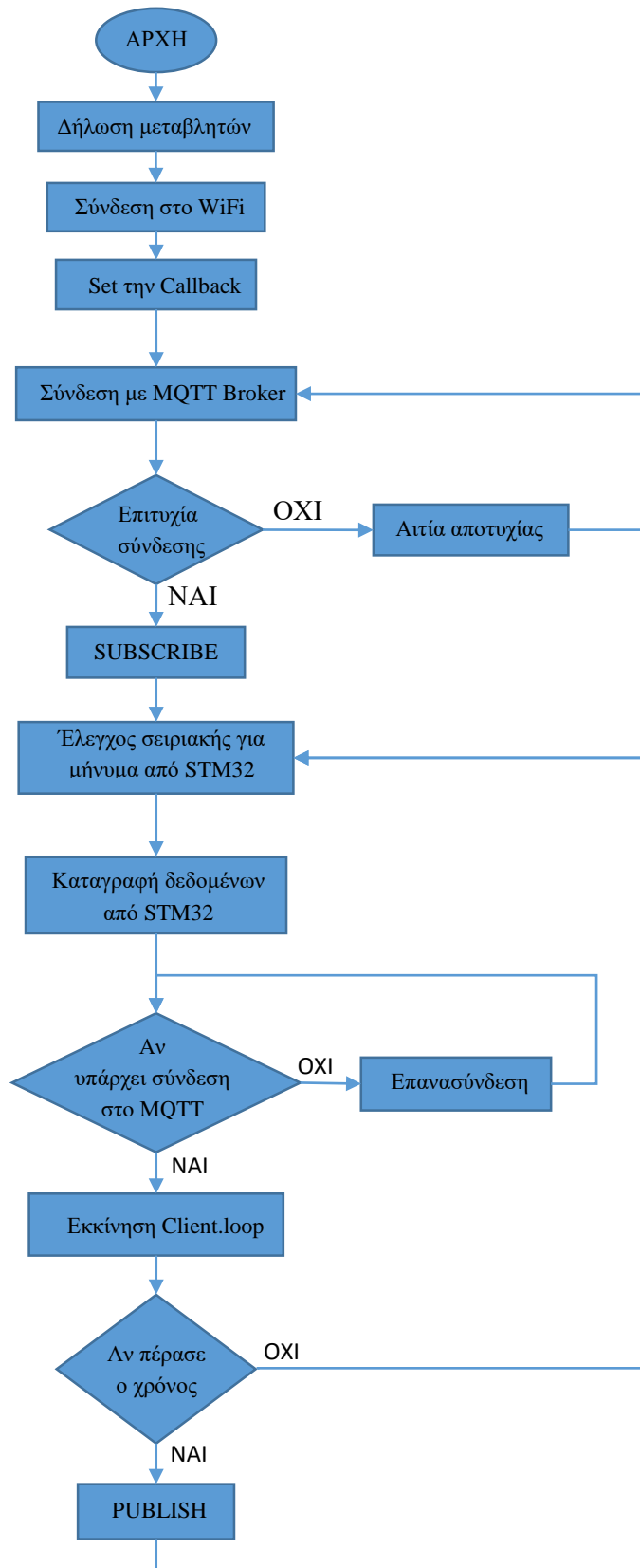
Ο ατέρμων βρόχος αποτελείται από πέντε ρουτίνες. Η πρώτη ρουτίνα είναι αυτή που ελέγχει διαδοχικά τους αισθητήρες και συλλέγει τις μετρήσεις. Η δεύτερη ρουτίνα βασίζεται στο interrupt της σειριακής θύρας. Πιο συγκεκριμένα, κάθε φορά που καλείται αυτή η ρουτίνα γίνεται αρχικά ένας έλεγχος (μέσω ενός flag) για το αν έχει γίνει διακοπή από το ESP32. Αν δεν υπάρχει διακοπή ο κώδικας προχωράει στην επόμενη ρουτίνα. Αν υπάρχει διακοπή, τότε ελέγχεται το μήνυμα που έφτασε και εκτελούνται οι ανάλογες ενέργειες (ενεργοποίηση/απενεργοποίηση ρελέ, άνοιγμα/κλείσιμο παραθύρων). Μετά από αυτήν την ρουτίνα ήταν σημαντικό να υπάρξει άλλη μία ακόμα η οποία θα ελέγχει συνέχεια την τρέχουσα κατάσταση του θερμοκηπίου, ώστε να υπάρχει ανάδραση (feedback) των ενεργειών στην ιστοσελίδα.

Έπειτα, ακολουθεί η ρουτίνα με την οποία στέλνονται όλες οι μετρήσεις και οι τρέχουσες καταστάσεις των λειτουργιών (π.χ. οπλισμένα ρελέ, ανοιχτά παράθυρα) στο USART1, δηλαδή στο ESP32. Η πέμπτη ρουτίνα του ατέρμονα βρόχου ελέγχει αν θα γίνει αποστολή μηνύματος SMS προς στον καταχωρημένο τηλεφωνικό αριθμό, ο οποίος είναι ο διαχειριστής του θερμοκηπίου. Στην παρούσα εργασία το σύστημα στέλνει επιπλέον τρία SMS εκτός από το αρχικό (επανεκκίνηση συσκευής). Το πρώτο από τα τρία SMS αφορά την υψηλή θερμοκρασία εντός του θερμοκηπίου και το δεύτερο την χαμηλή θερμοκρασία. Ενώ το τρίτο SMS αφορά την βροχή. Δηλαδή ότι βρέχει στο χώρο όπου βρίσκεται το θερμοκήπιο και τα παράθυρα είναι ανοιχτά. Τέλος, η τελευταία ρουτίνα είναι υπεύθυνη να απεικονίζει τις μετρήσεις των αισθητήρων στην οθόνη της συσκευής.

Ο πηγαίος κώδικας του ESP32 κάνει ελάχιστα πράγματα αλλά πολύ σημαντικά για το σύστημα. Συνδέεται στον MQTT Broker, στέλνει τις μετρήσεις του STM32 στην ιστοσελίδα και είναι υπεύθυνος να μεταφέρει τις εντολές ενεργοποίησης/απενεργοποίησης από την ιστοσελίδα στο θερμοκήπιο. Παρακάτω φαίνονται τα διαγράμματα ροής του STM32 και ESP32.

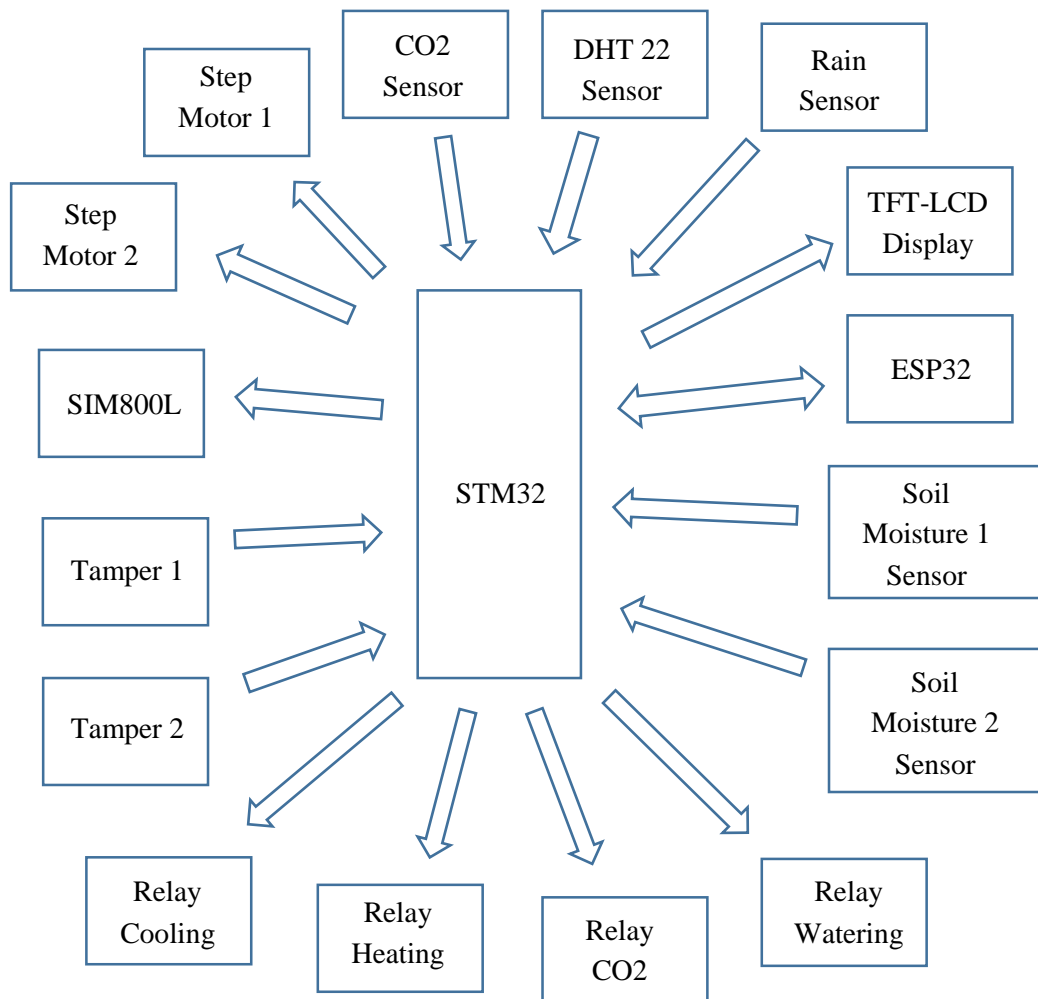


Εικόνα 59. Διάγραμμα ροής STM32



Εικόνα 60. Διάγραμμα ροής ESP32

4.5 Μπλοκ-διάγραμμα ηλεκτρονικού κυκλώματος



Εικόνα 61. Μπλοκ-διάγραμμα ηλεκτρονικού κυκλώματος

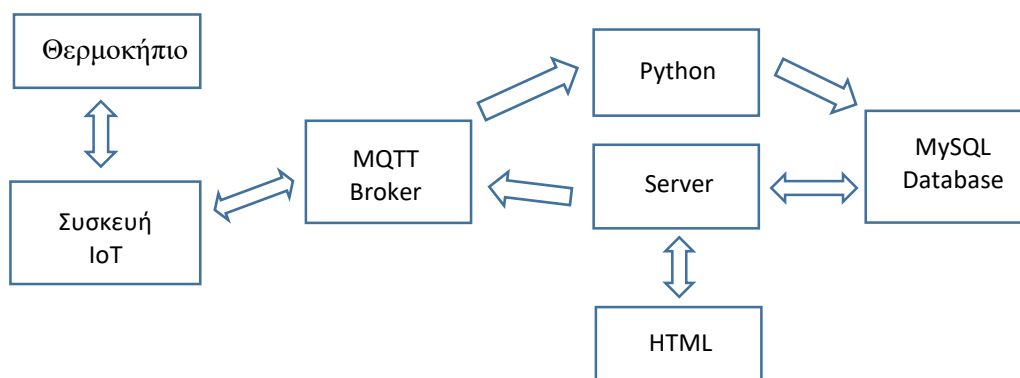
4.6 Επίλογος

Σε αυτό το κεφάλαιο το οποίο αποτελεί το ένα από τα δύο σημαντικότερα κεφάλαια της Διπλωματικής Εργασίας, παρουσιάστηκε ο σχεδιασμός και ο προγραμματισμός του ηλεκτρονικού κυκλώματος. Αρχικά, παρουσιάστηκε ο σχεδιασμός του ηλεκτρονικού κυκλώματος που έγινε στο ράστερ και έπειτα η υλοποίηση του μέσω του προγράμματος EAGLE. Στη συνέχεια, αναφέρθηκαν οι προγραμματιστικές ρυθμίσεις που έγιναν στο περιβάλλον STM32CubeIDE, καθώς και τα διαγράμματα ροής των πηγαίων κωδίκων των μικροελεγκτών STM32 και ESP32. Επίσης, επισυνάπτονται εικόνες από την κατασκευή του θερμοκηπίου.

Κεφάλαιο 5ο: Το λειτουργικό σύστημα

5.1 Εισαγωγή

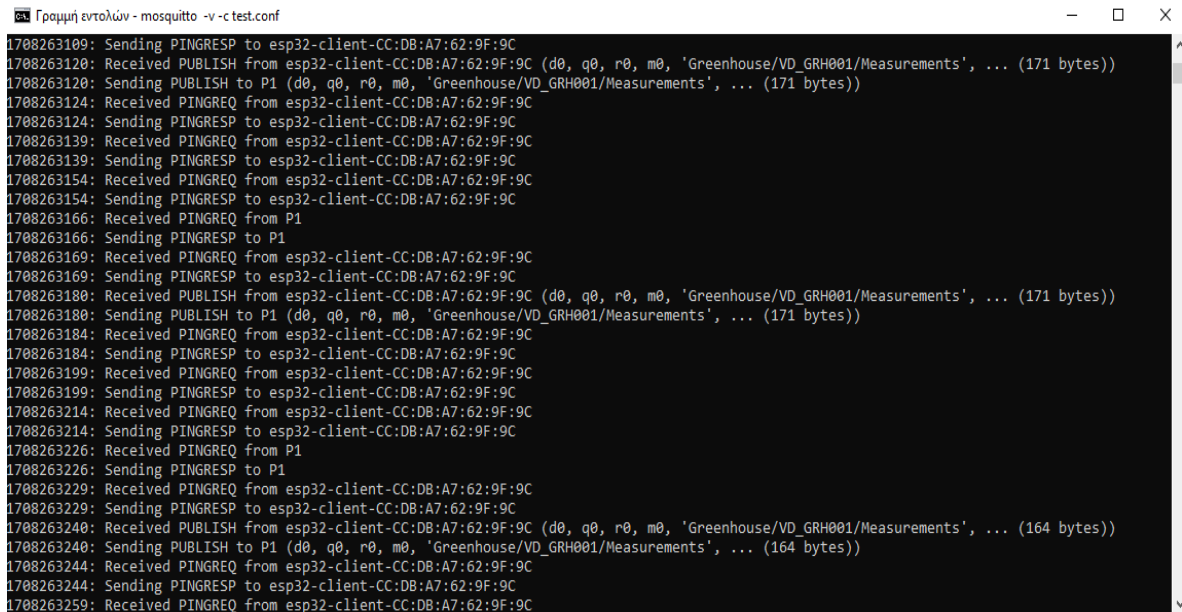
Στην παρούσα εργασία έχει γίνει κατασκευή βάσης δεδομένων και ιστοσελίδας για την αποθήκευση και την απεικόνιση των μετρήσεων, αλλά και την αποστολή εντολών για ενεργοποίηση/απενεργοποίηση λειτουργιών. Έχει γίνει χρήση της γλώσσας HTML (HyperText Markup Language), με την οποία δίνεται η δυνατότητα απεικόνισης εγγράφων και διάφορων στοιχείων της ιστοσελίδας. Μαζί με την HTML χρησιμοποιήθηκε και η CSS (Cascading Style Sheets), με την οποία ορίζεται η εμφάνιση (στυλ μορφοποίησης) και η διάταξη του κειμένου μιας ιστοσελίδας. Αυτές οι δύο γλώσσες διαχειρίζονται το front-end κομμάτι της ιστοσελίδας. Για το back-end της ιστοσελίδας χρησιμοποιήθηκε η γλώσσα PHP (αρχικά ονομάστηκε Personal Home Page, και ύστερα Hypertext Preprocessor). Είναι μια ισχυρή γλώσσα προγραμματισμού για την ανάπτυξη ιστοσελίδων και όχι μόνο. Συνήθως, η χρήση της γίνεται για την επικοινωνία με την βάση δεδομένων, όπου οι εντολές εκτελούνται δυναμικά. Η βάση δεδομένων που χρησιμοποιήθηκε είναι η MySQL. Η SQL είναι μια γλώσσα με την οποία δίνεται η δυνατότητα δημιουργίας, τροποποίησης και εξαγωγής πληροφοριών από τη βάση δεδομένων. Στο front-end κομμάτι συνήθως στις μέρες μας χρησιμοποιείται η γλώσσα Javascript, όπου και αυτή αποτελεί μια ισχυρή γλώσσα προγραμματισμού (πλέον και στο back-end). Στη παρούσα εργασία χρησιμοποιήθηκε το framework Chart.js. Πρόκειται για ένα framework της Javascript, με το οποίο μπορούμε εύκολα να σχεδιάζουμε γραφήματα. Επίσης, χρησιμοποιήθηκε και το sweetalert, το οποίο είναι και αυτό framework της Javascript. Είναι ένα framework με το οποίο μπορούμε να παρουσιάσουμε εύκολα και όμορφα μηνύματα προς τον χρήστη. Πέρα από αυτές τις γλώσσες χρησιμοποιήθηκε και η γλώσσα προγραμματισμού Python. Πρόκειται για μια γλώσσα υψηλού επιπέδου με ευρεία χρήση (π.χ. machine learning). Στη παρούσα εργασία χρησιμοποιήθηκε για τη λειτουργία του συστήματος γενικότερα. Πιο συγκεκριμένα, έχει γραφεί πρόγραμμα το οποίο τρέχει στον υπολογιστή συνεχόμενα όπου και γίνεται διαχείριση των δεδομένων που έρχονται από τον Broker. Είναι ουσιαστικά και αυτή μια back-end γλώσσα προγραμματισμού. Ο Broker που χρησιμοποιήθηκε για την διαχείριση των μηνυμάτων του πρωτόκολλου MQTT είναι ο Mosquitto Broker. Τέλος, χρησιμοποιήθηκε η εφαρμογή XAMPP, όπου από εκεί γίνεται εύκολα η δημιουργία και η διαχείριση της βάσης δεδομένων MySQL και του Apache server. [42-47]



Εικόνα 62. Μπλοκ-διάγραμμα του συστήματος

5.2 MQTT Broker

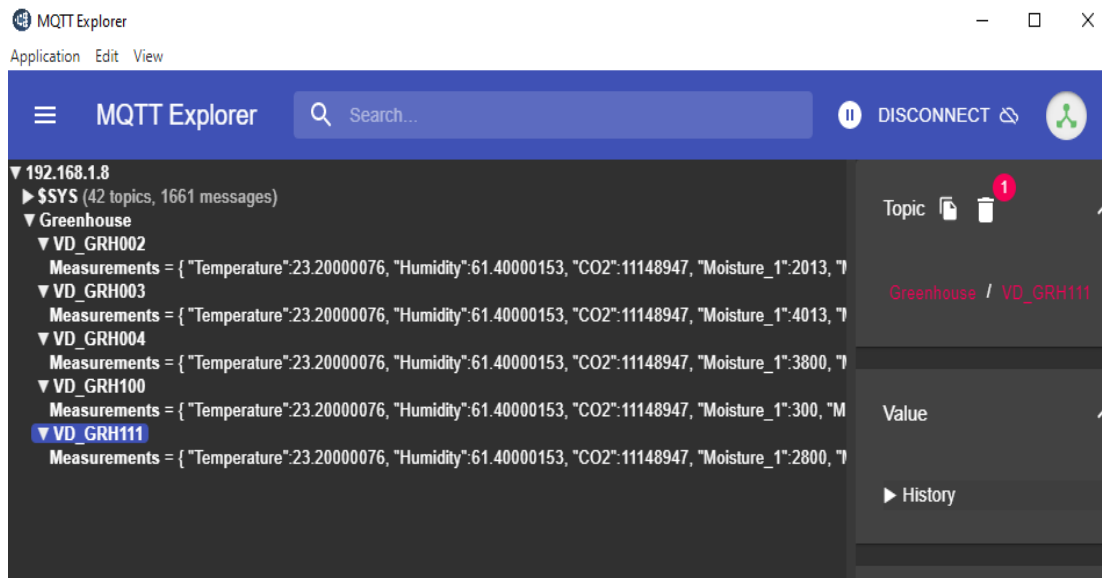
Παρακάτω φαίνονται τα μηνύματα που διαχειρίζεται ο Mosquitto Broker (MQTT Broker) από το command window του υπολογιστή.



```
Γραμμή εντολών - mosquitto -v -c test.conf
1708263109: Sending PINGRESP to esp32-client-CC:DB:A7:62:9F:9C
1708263120: Received PUBLISH from esp32-client-CC:DB:A7:62:9F:9C (d0, q0, r0, m0, 'Greenhouse/VD_GRH001/Measurements', ... (171 bytes))
1708263120: Sending PUBLISH to P1 (d0, q0, r0, m0, 'Greenhouse/VD_GRH001/Measurements', ... (171 bytes))
1708263124: Received PINGREQ from esp32-client-CC:DB:A7:62:9F:9C
1708263124: Sending PINGRESP to esp32-client-CC:DB:A7:62:9F:9C
1708263139: Received PINGREQ from esp32-client-CC:DB:A7:62:9F:9C
1708263139: Sending PINGRESP to esp32-client-CC:DB:A7:62:9F:9C
1708263154: Received PINGREQ from esp32-client-CC:DB:A7:62:9F:9C
1708263154: Sending PINGRESP to esp32-client-CC:DB:A7:62:9F:9C
1708263166: Received PINGREQ from P1
1708263166: Sending PINGRESP to P1
1708263169: Received PINGREQ from esp32-client-CC:DB:A7:62:9F:9C
1708263169: Sending PINGRESP to esp32-client-CC:DB:A7:62:9F:9C
1708263180: Received PUBLISH from esp32-client-CC:DB:A7:62:9F:9C (d0, q0, r0, m0, 'Greenhouse/VD_GRH001/Measurements', ... (171 bytes))
1708263180: Sending PUBLISH to P1 (d0, q0, r0, m0, 'Greenhouse/VD_GRH001/Measurements', ... (171 bytes))
1708263184: Received PINGREQ from esp32-client-CC:DB:A7:62:9F:9C
1708263184: Sending PINGRESP to esp32-client-CC:DB:A7:62:9F:9C
1708263199: Received PINGREQ from esp32-client-CC:DB:A7:62:9F:9C
1708263199: Sending PINGRESP to esp32-client-CC:DB:A7:62:9F:9C
1708263214: Received PINGREQ from esp32-client-CC:DB:A7:62:9F:9C
1708263214: Sending PINGRESP to esp32-client-CC:DB:A7:62:9F:9C
1708263226: Received PINGREQ from P1
1708263226: Sending PINGRESP to P1
1708263229: Received PINGREQ from esp32-client-CC:DB:A7:62:9F:9C
1708263229: Sending PINGRESP to esp32-client-CC:DB:A7:62:9F:9C
1708263240: Received PUBLISH from esp32-client-CC:DB:A7:62:9F:9C (d0, q0, r0, m0, 'Greenhouse/VD_GRH001/Measurements', ... (164 bytes))
1708263240: Sending PUBLISH to P1 (d0, q0, r0, m0, 'Greenhouse/VD_GRH001/Measurements', ... (164 bytes))
1708263244: Received PINGREQ from esp32-client-CC:DB:A7:62:9F:9C
1708263244: Sending PINGRESP to esp32-client-CC:DB:A7:62:9F:9C
1708263259: Received PINGREQ from esp32-client-CC:DB:A7:62:9F:9C
```

Εικόνα 63. Mosquitto Broker

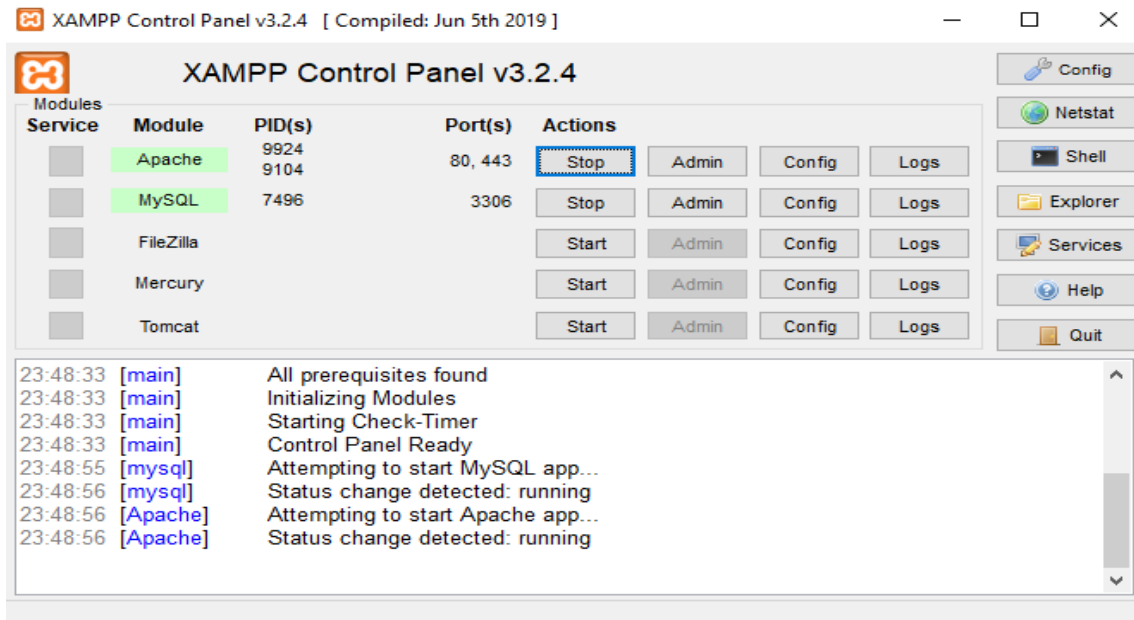
Στην παρακάτω εικόνα φαίνεται το περιβάλλον του MQTT Explorer. Με αυτήν την εφαρμογή γίνεται εύκολη η διαχείριση του MQTT Broker επειδή μπορούμε να δούμε αναλυτικά τα Topics του MQTT.



Εικόνα 64. MQTT Explorer

5.3 Βάση δεδομένων MySQL

Παρακάτω φαίνεται το περιβάλλον του XAMPP. Είναι ένα open source λογισμικό το οποίο περιέχει έναν Apache server και μια βάση δεδομένων MySQL και μπορούμε πολύ εύκολα να δημιουργήσουμε δυναμικές ιστοσελίδες σε γλώσσα προγραμματισμού PHP.



Εικόνα 65. Περιβάλλον του XAMPP

| Name | Surname | Email | Password | Must_change_password |
|---------|------------|------------------------------|----------------------------------------------------------|----------------------|
| Βασίλης | Νταμοσιδής | ntamotsidisvasilis@gmail.com | \$2y\$10\$1jF5I/NBUpn5.zbaLhJrs.9K07UojkTj6yXx0d/IJz... | |
| Γιάννης | Γιάννου | giannis123@gmail.com | \$2y\$10\$3Ad2Rsp0YnEhGRKP5hjrguJix2xZjdMnMZRQHfSECxV... | |
| Χρήστος | Χρήστου | xristos123@gmail.com | \$2y\$10\$epK1cHbGvbxTnPCv9gV4AOYm5eRUdIAzHAFc2szDQgR... | |
| Σωτήρης | Σωτηρίου | sotiris123@gmail.com | \$2y\$10\$aJrtCDMdexJ7W8CsV/46HuNZ5lkpo/5/exVUX713rz3... | |
| Νίκος | Νικολάου | nikos123@gmail.com | \$2y\$10\$RbvffTU3VPozuoTb6gAcYe.TDaKZiffEGB6OgVT8Brt... | |

Εικόνα 66. Πίνακας "personal data" της βάσης δεδομένων

| Email | Greenhouse_name | Greenhouse_serial_number | Topic | Windows | Watering | Cooling | Heating | Dioxide |
|------------------------------|-----------------|--------------------------|-------------------------------|---------|----------|---------|---------|---------|
| ntamotsidisvasilis@gmail.com | Πολίχνη | VD_GRH001 | Greenhouse/VD_GRH001/Commands | 0 | 0 | 0 | 0 | 0 |
| ntamotsidisvasilis@gmail.com | Πολίχνη | VD_GRH002 | Greenhouse/VD_GRH002/Commands | 0 | 0 | 0 | 0 | 0 |
| ntamotsidisvasilis@gmail.com | Χαλκιδική | VD_GRH003 | Greenhouse/VD_GRH003/Commands | 0 | 0 | 0 | 0 | 0 |
| ntamotsidisvasilis@gmail.com | Χαλκιδική | VD_GRH004 | Greenhouse/VD_GRH004/Commands | 0 | 0 | 0 | 0 | 0 |
| giannis123@gmail.com | Χαλκιδική | VD_GRH100 | Greenhouse/VD_GRH100/Commands | 0 | 0 | 0 | 0 | 0 |
| xristos123@gmail.com | Λάρισα | VD_GRH111 | Greenhouse/VD_GRH111/Commands | 0 | 0 | 0 | 0 | 0 |

Εικόνα 67. Πίνακας "greenhouses" της βάσης δεδομένων

| Email | Temperature | Humidity | Carbon_Dioxide | Soil_Moisture_1 | Soil_Moisture_2 | Rain | Datetime | Greenhouse_serial_number |
|-------|-------------|----------|----------------|-----------------|-----------------|------|---------------------------|--------------------------|
| | 22.30 | 61.60 | 1452 | 1 | 1 | 1 | Ωρα:21:49 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.30 | 61.10 | 1450 | 1 | 1 | 1 | Ωρα:21:50 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.30 | 61.20 | 1449 | 1 | 1 | 1 | Ωρα:21:51 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.30 | 61.20 | 1449 | 1 | 1 | 1 | Ωρα:21:52 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.30 | 61.60 | 1451 | 1 | 1 | 1 | Ωρα:21:53 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.30 | 61.40 | 1453 | 1 | 1 | 1 | Ωρα:21:54 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.30 | 61.60 | 1453 | 1 | 1 | 1 | Ωρα:21:55 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.20 | 61.70 | 1450 | 1 | 1 | 1 | Ωρα:21:56 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.30 | 61.30 | 1452 | 1 | 1 | 1 | Ωρα:21:57 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.30 | 61.80 | 1455 | 1 | 1 | 1 | Ωρα:21:58 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.20 | 61.70 | 1455 | 1 | 1 | 1 | Ωρα:21:59 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.30 | 61.90 | 1458 | 1 | 1 | 1 | Ωρα:22:00 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.20 | 61.80 | 1460 | 1 | 1 | 1 | Ωρα:22:01 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.10 | 61.90 | 1463 | 1 | 1 | 1 | Ωρα:22:02 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.10 | 61.70 | 1466 | 1 | 1 | 1 | Ωρα:22:03 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.20 | 62.00 | 1470 | 1 | 1 | 1 | Ωρα:22:04 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.20 | 62.00 | 1470 | 1 | 1 | 1 | Ωρα:22:05 Ημερ:16-01-2024 | VD_GRH001 |
| | 22.10 | 62.00 | 1475 | 1 | 1 | 1 | Ωρα:22:06 Ημερ:16-01-2024 | VD_GRH001 |

Εικόνα 68. Πίνακας "measurements" της βάσης δεδομένων

Στις εικόνες που προηγήθηκαν απεικονίζονται οι πίνακες της βάσης δεδομένων «thermokiριο» στο περιβάλλον «rhrMyAdmin». Στους πίνακες αυτούς αποθηκεύονται τα στοιχεία του χρήστη (το password του χρήστη είναι κρυπτογραφημένο), τα θερμοκήπια και σε ποιόν χρήστη ανήκουν και επίσης οι μετρήσεις από τους αισθητήρες των θερμοκηπίων.

5.4 Ιστοσελίδα του συστήματος

Παρακάτω φαίνονται οι σελίδες από το σύστημα εγγραφής του χρήστη (δημιουργία λογαριασμού) και το σύστημα του login.

Εικόνα 69. Σελίδα εγγραφής και login του χρήστη

Στη συνέχεια απεικονίζεται η σελίδα με τη δυνατότητα επαναφοράς του κωδικού πρόσβασης σε περίπτωση που ο χρήστης «κλειδωθεί» έξω από τον λογαριασμό του.

Για να επαναφέρετε τον κωδικό πρόσβασής σας, εισάγετε το email σας και πατήστε "Αποστολή κωδικού".

Αποστολή κωδικού

Εικόνα 70. Σελίδα επαναφοράς κωδικού πρόσβασης



Εικόνα 71. Η αρχική σελίδα της ιστοσελίδας

Αλλαγή κωδικού πρόσβασης

Υποβολή

Εικόνα 72. Λειτουργία αλλαγής κωδικού πρόσβασης

Παραπάνω φαίνεται η δυνατότητα της αλλαγής του κωδικού πρόσβασης του χρήστη στην εφαρμογή. Και στη συνέχεια η δυνατότητα της διαγραφής του λογαριασμού εάν το επιθυμεί ο χρήστης.

Διαγραφή λογαριασμού

Email

Κωδικός πρόσβασης

* Προσοχή η ενέργεια αυτή είναι μη αναστρέψιμη

Υποβολή

Εικόνα 73. Λειτουργία διαγραφής λογαριασμού χρήστη

Προσθήκη θερμοκηπίου

Email

Όνομα θερμοκηπίου

Serial number θερμοκηπίου

Υποβολή

Αφαίρεση θερμοκηπίου

Email

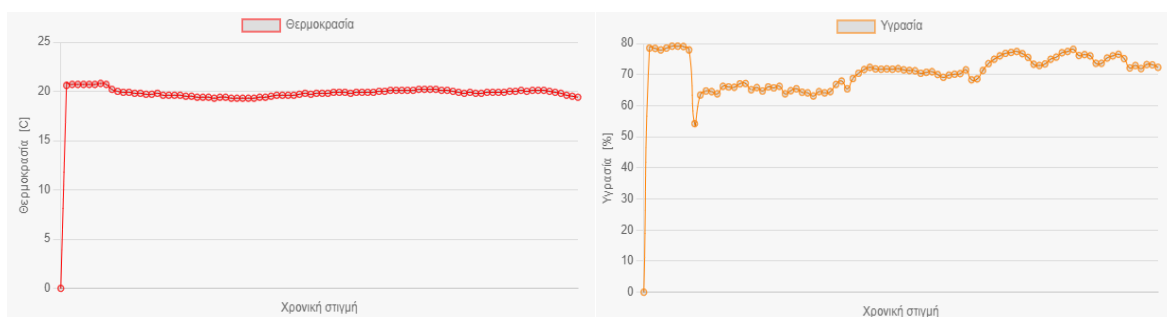
Όνομα θερμοκηπίου

Serial number θερμοκηπίου

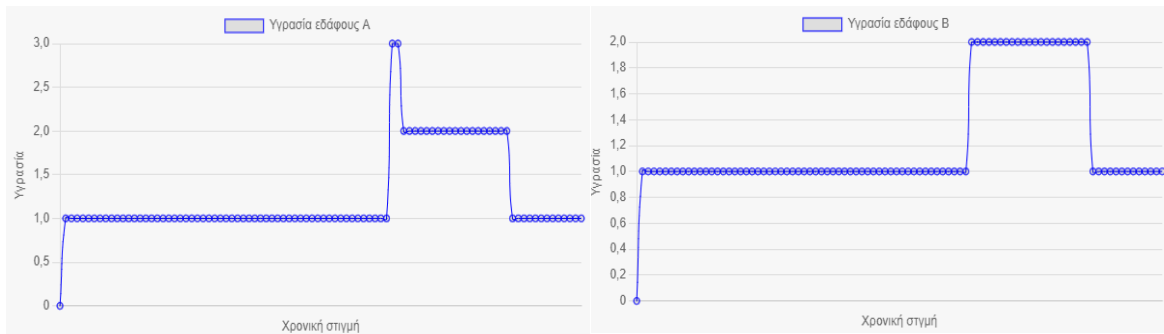
Υποβολή

Εικόνα 74. Λειτουργία προσθήκης και διαγραφής θερμοκηπίου

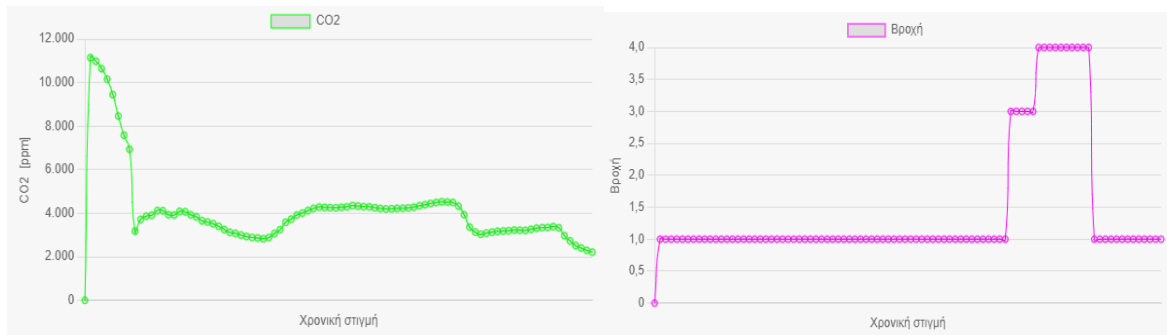
Παραπάνω απεικονίζεται η σελίδα στην οποία γίνεται η προσθήκη ή διαγραφή ενός θερμοκηπίου του χρήστη.



Εικόνα 75. Διαγράμματα θερμοκρασίας και υγρασίας

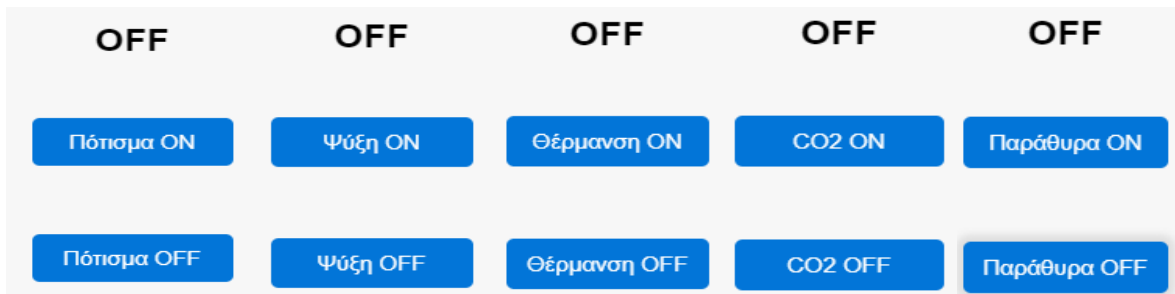


Εικόνα 76. Διαγράμματα υγρασίας εδάφους

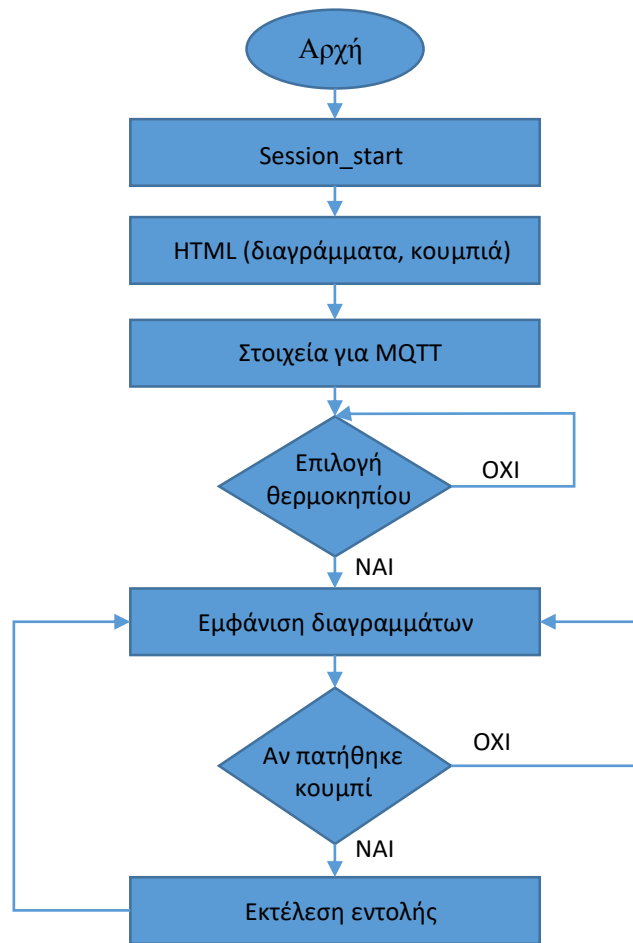


Εικόνα 77. Διαγράμματα CO2 και βροχής

Παρακάτω απεικονίζονται οι εντολές που μπορούν να σταλούν προς το θερμοκήπιο. Αρχικά γίνεται η επιλογή του επιθυμητού θερμοκηπίου από ένα drop down μενού και ύστερα η εκτέλεση της εντολής. Επίσης, δίνεται η δυνατότητα απεικόνισης της τρέχουσας κατάστασης των εντολών, δηλαδή να είναι ενεργοποιημένη ή όχι.

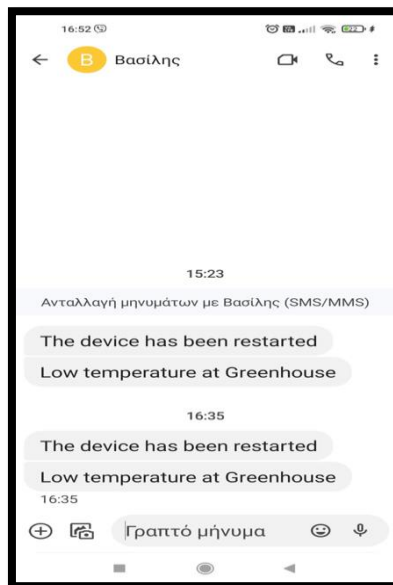


Εικόνα 78. Εντολές θερμοκηπίου



Εικόνα 79. Διάγραμμα ροής κεντρικής σελίδας

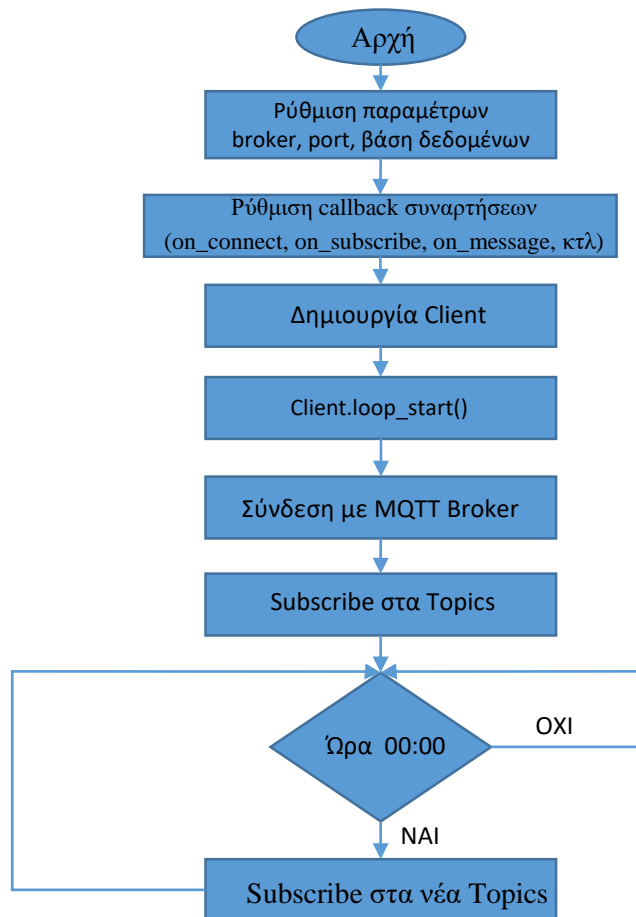
5.5 Μηνύματα SMS



Εικόνα 80. Μηνύματα SMS από θερμοκήπιο

5.6 Python

Ο πηγαίος κώδικας που εκτελείται με την Python τρέχει στον υπολογιστή (δηλαδή στον server). Αρχικά, ρυθμίζονται οι παράμετροι όπως τα στοιχεία της βάσης δεδομένων, ο Broker και η πόρτα στην οποία δέχεται τα μηνύματα ο Mosquitto Broker. Έπειτα, ρυθμίζονται οι συναρτήσεις callback (on_connect, on_subscribe, on_message, κτλ), οι οποίες ξεκινάνε να εκτελούνται όταν ξεκινήσει και η loop.start(). Στη συνέχεια, ξεκινάει η κανονική ροή του προγράμματος, όπου δημιουργείται ο Client, ξεκινάει η ρουτίνα loop.start() και γίνεται η σύνδεση με τον MQTT Broker. Αφού γίνει η σύνδεση, τότε το πρόγραμμα πάει και κάνει subscribe σε όλα τα Topic που υπάρχουν στη βάση δεδομένων. Έπειτα, το πρόγραμμα μπαίνει σε έναν ατέρμων βρόχο. Εκεί, απλά γίνεται ο έλεγχος της ώρας για το πότε θα κάνει το επόμενο Subscribe. Το πρόγραμμα είναι ρυθμισμένο να κάνει Subscribe κάθε βράδυ στις 00:00. Επομένως το σύστημα αυτό λειτουργεί αυτόνομα. Παράλληλα όμως, εκτελούνται και οι callback συναρτήσεις. Η πιο σημαντική είναι η on_message, όπου μέσα σε αυτήν γίνεται η λήψη του μηνύματος, ο διαχωρισμός των μετρήσεων και η καταχώρησή τους στη βάση δεδομένων. Παρακάτω φαίνεται το διάγραμμα ροής του πηγαίου κώδικα της Python.



Εικόνα 81. Διάγραμμα ροής Python

```

===== RESTART: C:\Users\Vasilis\Desktop\Κώδικας Διπλωματικής Python.py =====
Creating new instance
Connecting to broker
Connected OK Returned code = 0
Subscribe on topic OK
Subscribe on topic OK
Subscribe on topic OK
Subscribe on topic OK
Subscribe on topic OK
Subscribe on topic OK
Message topic = Greenhouse/VD_GRH001/Measurements
Message qos = 0
message retain flag = 0
Message received : {"Temperature":20.60000038,"Humidity":78.5,"CO2":11143,"Moisture_1":3778,"Moisture_2":3987,"Rain":4021,"Windows":0,"Potisma":0,"Psiksi":0,"Thermansi":0,"Dioxide":0}
Message topic = Greenhouse/VD_GRH001/Measurements
Message qos = 0
message retain flag = 0
Message received : {"Temperature":20.70000076,"Humidity":78.40000153,"CO2":10976,"Moisture_1":3776,"Moisture_2":3987,"Rain":4020,"Windows":0,"Potisma":0,"Psiksi":0,"Thermansi":0,"Dioxide":0}
Message topic = Greenhouse/VD_GRH001/Measurements
Message qos = 0
message retain flag = 0
Message received : {"Temperature":20.70000076,"Humidity":77.90000153,"CO2":10638,"Moisture_1":3789,"Moisture_2":3985,"Rain":4025,"Windows":0,"Potisma":0,"Psiksi":0,"Thermansi":0,"Dioxide":0}
Message topic = Greenhouse/VD_GRH001/Measurements
Message qos = 0
message retain flag = 0
Message received : {"Temperature":20.70000076,"Humidity":78.5,"CO2":10157,"Moisture_1":3784,"Moisture_2":3987,"Rain":4020,"Windows":0,"Potisma":0,"Psiksi":0,"Thermansi":0,"Dioxide":0}
Message topic = Greenhouse/VD_GRH001/Measurements
Message qos = 0
message retain flag = 0
Message received : {"Temperature":20.70000076,"Humidity":79.09999847,"CO2":9447,"Moisture_1":3779,"Moisture_2":3987,"Rain":4022,"Windows":0,"Potisma":0,"Psiksi":0,"Thermansi":0,"Dioxide":0}

```

Εικόνα 82. Διαχείριση μηνυμάτων από την Python

5.7 Επίλογος

Στο κεφάλαιο αυτό το οποίο αποτελεί το δεύτερο από τα δύο σημαντικά κεφάλαια της Διπλωματικής Εργασίας, παρουσιάστηκε το λειτουργικό σύστημα. Αρχικά, έγινε μια μικρή εισαγωγή των επιμέρους κομματιών μιας ιστοσελίδας και πώς αυτά εφαρμόστηκαν στην παρούσα εργασία. Κατόπιν, παρουσιάστηκε το μπλοκ διάγραμμα ολόκληρου του συστήματος, το οποίο κάνει κατανοητή τη μεθοδολογία που αναπτύχθηκε. Στη συνέχεια, έγινε αναφορά σχετικά με τον Mosquitto Broker, ο οποίος είναι ένας MQTT Broker. Έπειτα ακολουθεί η διασύνδεση της βάσης δεδομένων MySQL με το σύστημα, η οποία χρησιμεύει στην αποθήκευση των χρηστών αλλά και του ιστορικού των μετρήσεων από τους αισθητήρες του θερμοκηπίου. Τέλος, παρουσιάστηκαν το front-end και back-end της ιστοσελίδας που αναπτύχθηκε.

Κεφάλαιο 6ο: Συμπεράσματα και προτάσεις βελτίωσης

6.1 Συμπεράσματα

Το σύστημα που αναπτύχθηκε στην παρούσα εργασία χρησιμοποιεί αισθητήρες θερμοκρασίας, υγρασίας, υγρασίας εδάφους και διοξειδίου του άνθρακα. Η συλλογή των μετρήσεων με τον μικροελεγκτή STM32F103C8T6 και η αποστολή τους στον MQTT Broker με το ESP32, αλλά και η αποστολή μηνύματος SMS προσφέρει μεγάλη ευελιξία στις μελλοντικές εξελίξεις του συστήματος. Η σύνδεση της συσκευής με το διαδίκτυο κάνει την διαχείριση του θερμοκηπίου πολύ εύκολη. Από την άλλη πλευρά του συστήματος, η ιστοσελίδα προσφέρει και αυτή αρκετές σύγχρονες δυνατότητες. Εκτός από την παρακολούθηση των μετρήσεων ενός από τα πολλά θερμοκήπια που μπορεί να διαθέτει ένας χρήστης, υπάρχει και η δυνατότητα αποστολής εντολών με σκοπό την ενεργοποίηση/απενεργοποίηση λειτουργιών, όπως είναι το πότισμα, ψύξη, θέρμανση, προσθήκη διοξειδίου του άνθρακα και διαχείριση των παραθύρων. Πρόκειται για ένα σύστημα το οποίο προσφέρει βασικές δυνατότητες, αλλά αν εφαρμοστούν οι βελτιώσεις που αναφέρθηκαν προηγουμένως, μπορεί να γίνει ένα πολύ ανταγωνιστικό σύστημα.

6.2 Προτάσεις βελτίωσης

Η παρούσα εργασία αποτελεί ένα βασικό σύστημα παρακολούθησης θερμοκηπίου από απόσταση. Οι βελτιώσεις που θα μπορούσαν να γίνουν στο υπάρχων σύστημα, αναφέρονται παρακάτω.

Αρχικά, στο κομμάτι του hardware μπορούν να χρησιμοποιηθούν επιπλέον αισθητήρια. Για παράδειγμα, θα μπορούσε να χρησιμοποιηθεί ένας αισθητήρας για την μέτρηση της φωτεινότητας και στη συνέχεια να εγκατασταθούν φώτα πάνω στο θερμοκήπιο με τη δυνατότητα του απομακρυσμένου ελέγχου τους. Όπως αναφέρθηκε και στην εισαγωγή, είναι σημαντικό να υπάρχει επαρκής φωτεινότητα εντός του θερμοκηπίου αφού επηρεάζει σημαντικά την ανάπτυξη των φυτών. Πέραν αυτού του αισθητήρα θα μπορούσαν να ενσωματωθούν και άλλοι αισθητήρες όπως αισθητήρας ανίχνευσης καπνού και κίνησης με στόχο την ασφάλεια του θερμοκηπίου.

Στη συνέχεια, μια επιπλέον προσθήκη στην κεντρική σελίδα της εφαρμογής που θα μπορούσε να γίνει, είναι η δυνατότητα φιλτραρίσματος των αποτελεσμάτων των μετρήσεων. Δηλαδή ο χρήστης να έχει τη δυνατότητα να επιλέξει το διάστημα των μετρήσεων που θα θέλει να παρατηρήσει. Ακόμα, θα μπορούσε να δοθεί η δυνατότητα στον χρήστη να εισάγει τιμές με σκοπό το σύστημα να κρατάει σταθερές τις συνθήκες (αυτόματη λειτουργία) του θερμοκηπίου με βάση τα δεδομένα που θα δίνει ο χρήστης.

Επίσης, θα ήταν χρήσιμο να χρησιμοποιηθεί η γλώσσα Javascript, η οποία με την τεχνολογία που διαθέτει το περιβάλλον της εφαρμογής να γίνει πιο διαδραστικό και η εμπειρία του χρήστη καλύτερη (π.χ. αυτόματη ανανέωση μετρήσεων).

Τέλος, εφόσον αυξηθούν οι χρήστες της εφαρμογής θα ήταν λειτουργικά πιο αποδοτικό η βελτίωση της βάσης δεδομένων. Πιο συγκεκριμένα, στην παρούσα εργασία οι μετρήσεις όλων των θερμοκηπίων αποθηκεύονται σε έναν πίνακα στη βάση δεδομένων και ο διαχωρισμός τους γίνεται με βάση το serial number της συσκευής. Στην περίπτωση που θα υπάρξουν πολλοί χρήστες καλό θα ήταν να σχεδιαστεί στη βάση δεδομένων ένας πίνακας για κάθε χρήστη. Αυτό θα έχει ως αποτέλεσμα τη γρηγορότερη φόρτωση των δεδομένων στην ιστοσελίδα και φυσικά την αποδοτικότερη λειτουργία του server και της βάσης δεδομένων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία

- [1] Geoffrey Brown, *Discovering the STM32 Microcontroller*, 2012
- [2] Carmine Noviello, *Mastering STM32*, 2017
- [3] Neil Kolban, *Kolban's Book on ESP32*, 2017
- [4] Μιχάλης Ν. Σπάσος, *Αναλογική Επεξεργασία Σημάτων Αισθητηρίων*, Εκδόσεις ΑΪΒΑΖΗ, 2018

Manuals

- [5] Aosong(Guangzhou) Electronics Co. Ltd, “*Temperature and humidity module AM2302 Product Manual*”. [Online]. Available: https://files.seeedstudio.com/wiki/Grove-Temperature_and_Humidity_Sensor_Pro/res/AM2302-EN.pdf
- [6] Sandboxelectronics, “*MG-811 CO2 Sensor*”. [Online]. Available: <https://sandboxelectronics.com/files/SEN-000007/MG811.pdf>
- [7] Zhengzhou Winsen Electronics Technology Co., Ltd, “*Solid Electrolyte CO2 Gas Sensor*”, 2015. [Online]. Available: <https://www.winsen-sensor.com/d/files/solid-electrolyte/mg811.pdf>
- [8] OASIS, “*MQTT Version 3.1.1*”, OASIS Standard, 2014

Data Sheets

- [9] Core-electronics, “*GS1662*”. [Online]. Available: https://core-electronics.com.au/attachments/localcontent/GS1662Datasheet_38416f81d32.pdf
- [10] STMicroelectronics, “*ULN2003*”.2002. [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/25575/STMICROELECTRONICS/ULN2003.html>
- [11] Micrel Semiconductor, “*MIC29302WT*”, 2010. [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/407729/MICREL/MIC29302WT.html>
- [12] STMicroelectronics, “*LF33CV*”, 2003. [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/22819/STMICROELECTRONICS/LF33CV.html>
- [13] Sitronix Technology Co., Ltd., “*ST7735*”, 2009. [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/326213/SITRONIX/ST7735.html>
- [14] Mean Well, “*HDR-60*”, 2024. [Online]. Available: <https://www.meanwell-web.com/content/files/pdfs/productPdfs/MW/HDR-60/HDR-60-spec.pdf>
- [15] Tyco Electronics, “*PE014005*”, 2007. [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/239843/MACOM/PE014005.html>

- [16] Fairchild Semiconductor, “*BC337*”, 2002. [Online]. Available:
<https://pdf1.alldatasheet.com/datasheet-pdf/view/643709/FAIRCHILD/BC337.html>
- [17] SIMCom, “*SIM800L*”, 2015, [Online]. Available:
<https://pdf1.alldatasheet.com/datasheet-pdf/view/1741389/SIMCOM/SIM800L.html>
- [18] Espressif Systems, “*ESP32*”. [Online]. Available:
https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

Internet Sites

- [19] Wikipedia, “*Greenhouse*”. [Online]. Available:
<https://en.wikipedia.org/wiki/Greenhouse>
- [20] HiveMQ, “*The Origin of MQTT*”. [Online]. Available:
<https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>
- [21] HiveMQ, “*MQTT Publish/Subscribe Architecture*”. [Online]. Available:
<https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe/>
- [22] HiveMQ, “*MQTT Client, MQTT Broker, and MQTT Server Connection Establishment Explained*”. [Online]. Available:
<https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/>
- [23] HiveMQ, “*MQTT Publish, MQTT Subscribe & Unsubscribe*”. [Online]. Available:
<https://www.hivemq.com/blog/mqtt-essentials-part-4-mqtt-publish-subscribe-unsubscribe/>
- [24] HiveMQ, “*MQTT Topics, Wildcards, & Best Practices*”. [Online]. Available:
<https://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices/>
- [25] HiveMQ, “*What is MQTT Quality of Service (QoS) 0, 1, & 2*”. [Online]. Available:
<https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/>
- [26] HiveMQ, “*Understanding Persistent Sessions and Clean Sessions*”. [Online]. Available:
<https://www.hivemq.com/blog/mqtt-essentials-part-7-persistent-session-queuing-messages/>
- [27] HiveMQ, “*What are Retained Messages in MQTT*”. [Online]. Available:
<https://www.hivemq.com/blog/mqtt-essentials-part-8-retained-messages/>
- [28] HiveMQ, “*What is MQTT Last Will and Testament (LWT)*”. [Online]. Available:
<https://www.hivemq.com/blog/mqtt-essentials-part-9-last-will-and-testament/>
- [29] HiveMQ, “*What Is MQTT Keep Alive*”. [Online]. Available:
<https://www.hivemq.com/blog/mqtt-essentials-part-10-alive-client-take-over/>
- [30] HiveMQ, “*MQTT Vs. HTTP for IoT*”. [Online]. Available:
<https://www.hivemq.com/article/mqtt-vs-http-protocols-in-iiot/>
- [31] HiveMQ, “*MQTT vs AMQP for IoT*”. [Online]. Available:
<https://www.hivemq.com/article/mqtt-vs-amqp-for-iiot/>
- [32] HiveMQ, “*MQTT vs CoAP for IoT*”. [Online]. Available:
<https://www.hivemq.com/article/mqtt-vs-coap-for-iiot/>

- [33] HiveMQ, “MQTT vs ZeroMQ for IoT”. [Online]. Available:
<https://www.hivemq.com/article/mqtt-vs-zeromq-for-iot/>
- [34] DFROBOT, “SEN0159”. [Online]. Available:
https://wiki.dfrobot.com/CO2_Sensor_SKU_SEN0159
- [35] DFROBOT, “SEN0159 Sensor”. [Online]. Available:
<https://image.dfrobot.com/image/data/SEN0159/CO2%20Sensor%20V1%20SCH.pdf>
- [36] ElectronicWings, “Soil Moisture Sensor Interfacing with ESP32”. [Online]. Available:
<https://www.electronicwings.com/esp32/soil-moisture-sensor-interfacing-with-esp32>
- [37] ESP32IO, “ESP32-Rain Sensor”. [Online]. Available:
https://esp32io.com/tutorials/esp32-rain-sensor?utm_content=cmp-true
- [38] Last Minute Engineers, “Control 28BYJ-48 Stepper Motor with ULN2003 Driver & Arduino”. [Online]. Available:
https://lastminuteengineers.com/28byj48-stepper-motor-arduino-tutorial/?utm_content=cmp-true
- [39] Controllers Tech, “ST7735” 1.8 TFT Display with STM32”. [Online]. Available:
<https://controllerstech.com/st7735-1-8-tft-display-with-stm32/>
- [40] Last Minute Engineers, “Send Receive SMS & Call with SIM800L GSM Module & Arduino”. [Online]. Available:
<https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/>
- [41] STMicroelectronics, “STM32F103C8T6”. [Online]. Available:
<https://www.st.com/en/microcontrollers-microprocessors/stm32f103c8.html>
- [42] Wikipedia, “HTML”, [Online]. Available:
<https://en.wikipedia.org/wiki/HTML>
- [43] Wikipedia, “CSS”, [Online]. Available:
<https://en.wikipedia.org/wiki/CSS>
- [44] Wikipedia, “MySQL”, [Online]. Available:
<https://en.wikipedia.org/wiki/MySQL>
- [45] Wikipedia, “PHP”, [Online]. Available:
<https://en.wikipedia.org/wiki/PHP>
- [46] Wikipedia, “Python”, [Online]. Available:
[https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- [47] Wikipedia, “JavaScript”, [Online]. Available:
<https://en.wikipedia.org/wiki/JavaScript>
- [48] <https://www.pexels.com/photo/field-of-plants-in-greenhouse-2886937/>
- [49] Controllers Tech, “How to use DHT 22 with STM32”. [Online]. Available:
<https://controllerstech.com/temperature-measurement-using-dht22-in-stm32/>
- [50] Steve’s Internet Guide, “MQTT and Python Beginners”. [Online]. Available:
<http://www.steves-internet-guide.com/mqtt-python-beginners-course/>
- [51] Random Nerd Tutorials, “ESP32 MQTT – Publish and Subscribe with Arduino IDE”, [Online]. Available:
<https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/>

- [52] https://www.google.com/search?sca_esv=601079603&sxsrf=ACQVn0_Rm6RjAHGY1oMQCVtICjMeojq1KQ:1706105298039&q=moisture+sensor&tbm=isch&source=lnms&sa=X&ved=2ahUKEwjftvTBmfaDaxXzSkEAHWvhC1AQ0pQJegQIDhAB&biw=1920&bih=919&dpr=1#imgrc=_zvI4meZoLLx8M
- [53] https://www.google.com/search?q=rain+sensor&tbm=isch&ved=2ahUKEwiyz9D1mvaDaxXPgv0HHZJ4DGwQ2-cCegQIABAA&oq=rain+sensor&gs_lcp=CgNpbWcQAzIFCAAQgAQyBggAEAcQHjIGCAAQBxAeMgYIABAHEB4yBggAEAcQHjIGCAAQBxAeMgYIABAHEB4yBggAEAcQHjIECAAQHjIECAAQHjoHCAAQgAQQEzoICAAQBxAeEBM6CAgAEAUQBxAeOggIABAIEAcQHIDBClrP2CbRmgLcAB4AIABuwGIAa8RkgEENC4xNZgBAKABAaoBC2d3cy13aXotaW1nwAEB&sclient=img&ei=ShuxZbKCOc-F9u8PkvGx4AY&bih=919&biw=1920
- [54] https://www.google.com/search?q=switch+tamper+electronics&tbm=isch&ved=2ahUKEwivulaXsvaDaxWY_rsIHVrSDIUQ2-cCegQIABAA&oq=switch+tamper+electronics&gs_lcp=CgNpbWcQAzoECCMQJzoGCAAQCBAAeUNYCWKAkYIUmaABwAHgAgAGXAYgBIAuSAQM1LjiYAQCgAQGqAQtn d3Mtd2l6LWltZ8ABAQ&sclient=img&ei=rzOxZe-kApj97_UP2qS7qAU&bih=919&biw=1920&rlz=1C1GCEA_enGR918GR918
- [55] https://www.google.com/search?sca_esv=601135296&rlz=1C1GCEA_enGR918GR918&sxsrf=ACQVn09RCdfSmtHE09RlgDTgQMySVDqQeQ:1706119834045&q=28BYJ-48&tbm=isch&source=lnms&sa=X&ved=2ahUKEwiW45vVz_aDaxUZ8wIHHSxjA3gQ0pQJegQIEBAB&biw=1920&bih=919&dpr=1#imgrc=i-vQsvv5lNo1gM
- [56] https://www.google.com/search?q=uln2003an&tbm=isch&ved=2ahUKEwjm1O_W2faDaxXJk_0HHbBCDcsQ2-cCegQIABAA&oq=uln2003an&gs_lcp=CgNpbWcQAzIECCMQJzIHCAAQgAQQEzIHCAAQgAQQEzIHCAAQgAQQEzIHCAAQgAQQEzIGCAAQHhATMggIABAFEB4QE1CoBlj3FGCvGmgAcAB4AIABmQGIAe0HkgEDMS43mAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=GV2xZabkNMmn9u8PsIW12Aw&bih=919&biw=1920&rlz=1C1GCEA_enGR918GR918#imgrc=BunrzEkbSUnsdM
- [57] https://www.google.com/search?q=uln2003an&tbm=isch&ved=2ahUKEwjm1O_W2faDaxXJk_0HHbBCDcsQ2-cCegQIABAA&oq=uln2003an&gs_lcp=CgNpbWcQAzIECCMQJzIHCAAQgAQQEzIHCAAQgAQQEzIHCAAQgAQQEzIHCAAQgAQQEzIGCAAQHhATMggIABAFEB4QE1CoBlj3FGCvGmgAcAB4AIABmQGIAe0HkgEDMS43mAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=GV2xZabkNMmn9u8PsIW12Aw&bih=919&biw=1920&rlz=1C1GCEA_enGR918GR918
- [58] https://www.google.com/search?q=mic29302wt+pinout&tbm=isch&ved=2ahUKEwjOtpO A4vaDaxU8g_0HHTdzBC4Q2-cCegQIABAA&oq=mic29302wt+pinout&gs_lcp=CgNpbWcQAzoECCMQJ1CcBFi-GGCDHGgAcAB4AIABrWKIAC0JkgEHMi40LjEuMZgBAKABAaoBC2d3cy13aXotaW1nwAEB&sclient=img&ei=02WxZY6BOLyG9u8Pt-aR8AI&bih=919&biw=1920&rlz=1C1GCEA_enGR918GR918

- [59] https://www.google.com/search?q=lf33cv&tbm=isch&ved=2ahUKEwiJ2MWP6faDAXd9wIHHUmpCIAQ2-cCegQIABAA&oq=lf33cv&gs_lcp=CgNpbWcQAzIECCMQJ1AAWABgmBBoAXAAeAACAAQCIACSAQCQAQnd3Mtd2l6LWltZ8ABAQ&scient=img&ei=S22xZcmGD93vi-gPydKigAg&bih=919&biw=1920&rlz=1C1GCEA_enGR918GR918#imgrc=lcM1W4KgsdSS7M
- [60] https://www.google.com/search?q=st7735+tft+display&tbm=isch&ved=2ahUKEwjdp-em7_aDAXXD6QIHHab_CZoQ2-cCegQIABAA&oq=st7735+tft&gs_lcp=CgNpbWcQARgAMgcIABCABBATMggIABAIEB4QE1AAWNMkYPwgaABwAHgAgAFniAGGA5IBAzluMpgBAKABAaoBC2d3cy13aXotaW1nwAEB&scient=img&ei=xnOxZd2sJ8PTi-gPpv-n0Ak&bih=919&biw=1920&rlz=1C1GCEA_enGR918GR918
- [61] https://www.google.com/search?sca_esv=601184441&rlz=1C1GCEA_enGR918GR918&hl=el&sxsrf=ACQVn0_LZKR9J_VOJ8Tv-MYFqRvkWLbz0w:1706129759205&q=mean+well+hdr-60-5&tbm=isch&source=lnms&sa=X&ved=2ahUKEwjxvPR9PaDAXUtzAIHHZnSD_YQ0pQJegQIDBAB&bih=1920&bih=919&dpr=1#imgrc=I9qhumADTVKuMM
- [62] https://www.google.com/search?sca_esv=601184441&rlz=1C1GCEA_enGR918GR918&hl=el&sxsrf=ACQVn09BuvlZTQAZnnVhQCW2_7Uoj5kGIg:1706131068667&q=pe014005&tbm=isch&source=lnms&sa=X&ved=2ahUKEwiC2qbC-faDAXUj7AIHHb3dDY0Q0pQJegQIDhAB&bih=1920&bih=919&dpr=1#imgrc=QQ-dmOOS-svEzM
- [63] [https://www.google.com/search?q=bc337+pinout&tbm=isch&ved=2ahUKEwjpp5q8_PaDAXWb5QIHHZH4BCMq2-cCegQIABAA&oq=bc337+&gs_lcp=CgNpbWcQARgBMgUIABCABDIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjoECCMQJzoGCAAQBxAeUMUCWMUCYOUPaABwAHgAgAGeAYgB-wGSAQMxLjGYAQCgAQGqAQnd3Mtd2l6LWltZ8ABAQ&scient=img&ei=1YGxZamJC5vLi-gPkgFTmAI&bih=919&biw=1920&rlz=1C1GCEA_enGR918GR918&hl=el#imgrc=Wd56yewbQCOuLM](https://www.google.com/search?q=bc337+pinout&tbm=isch&ved=2ahUKEwjpp5q8_PaDAXWb5QIHHZH4BCMq2-cCegQIABAA&oq=bc337+&gs_lcp=CgNpbWcQARgBMgUIABCABDIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjoECCMQJzoGCAAQBxAeUMUCWMUCYOUPaABwAHgAgAGeAYgB-wGSAQMxLjGYAQCgAQGqAQnd3Mtd2l6LWltZ8ABAQ&scient=img&ei=1YGxZamJC5vLi-gPkgFTmAI&bih=919&biw=1920&rlz=1C1GCEA_enGR918GR918&hl=el#imgrc=Wd56yewbQCOuLM)
- [64] https://www.google.com/search?sca_esv=601452934&rlz=1C1GCEA_enGR918GR918&sxsrf=ACQVn0_QOnli-Hpm-8mRGcSuR7sRw1frKQ:1706209843721&q=stm32f103c8t6&tbm=isch&source=lnms&sa=X&sqi=2&ved=2ahUKEwjZqJb9nvmDAXWsgf0HHYQFDugQ0pQJegQIEBAB&bih=1920&bih=919&dpr=1#imgrc=ppjg2fadaCDZ_M
- [65] https://www.google.com/search?q=esp32+pinout&tbm=isch&ved=2ahUKEwisY3GsvmDAXUbuP0HHXWKAHEQ2-cCegQIABAA&oq=esp32+&gs_lcp=CgNpbWcQARgBMgQIIxAnMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjoHCAAQgAQQEzoICAAQBRAeEBM6CAGAEAgQHhATOgYIABAeEBNqZgdYxhBglhxoAHAAeACAAAYIBiAHXB5IBAzcuM5gBAKABAaoBC2d3cy13aXotaW1nwAEB

&scient=img&ei=uMayZayNOZvw9u8P9ZSCiAc&bih=919&biw=1920&rlz=1C1GCEA_enGR918GR918#imgrc=IZ0MUu-_ippJ6M

Papers in Conference Proceedings

- [66] Shatadru Bipasha Biswas, M. Tariq Iqbal, “*Solar water pumping system control using a low cost ESP32 microcontroller*”, IEEE Canadian Conference on Electrical & Computer Engineering, 2018
- [67] Kunja Bihari Swain, G. Santamanyu, Amiya Ranjan Senapati, “*Smart industry pollution monitoring and controlling using LabVIEW based IoT*”, IEEE 3rd International Conference on Sensing, Signal Processing and Security, 2017
- [68] Dan Dragomir, Laura Gheorghe, Sergiu Costea, Alexandru Radovici, “*A Survey on Secure Communication Protocols for IoT Systems*”, IEEE 2016 International Workshop on Secure Internet of Things

Journal Articles

- [69] Vijitha Ananthi, “*Automation using IoT in greenhouse environment*”, Journal of Information Technology and Digital World, Vol.01/ No. 01, Pages: 38-47, 2019
- [70] Dia Mehdi Faris, Mahmood Basil Mahmood, “*Data acquisition of greenhouse using Arduino*”, Journal of Babylon University/Pure and Applied Sciences, No.(7)/ Vol.(22), 2014

ΠΑΡΑΡΤΗΜΑ Α: ΚΩΔΙΚΑΣ STM32

```
/* USER CODE BEGIN Header */
/**
 * *****
 * @file      : main.c
 * @brief     : Main program body
 * *****
 * @attention
 *
 * <h2><center>&copy; Copyright (c) 2023 STMicroelectronics.
 * All rights reserved.</center></h2>
 *
 * This software component is licensed by ST under BSD 3-Clause license,
 * the "License"; You may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *
 *             opensource.org/licenses/BSD-3-Clause
 *
 * *****
 */
/* USER CODE END Header */
/* Includes -----*/
#include "main.h"

/* Private includes -----*/
/* USER CODE BEGIN Includes */
#include "stdio.h"
#include "string.h"
#include "ST7735.h"
#include "GFX_FUNCTIONS.h"
#include "fonts.h"
#include "math.h"

/* USER CODE END Includes */

/* Private typedef -----*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----*/
/* USER CODE BEGIN PD */
#define DHT22_PORT GPIOA
#define DHT22_PIN GPIO_PIN_12
#define epanalipseis_moter 650 //2600 einai mia pliri peristrofi
#define READ_SAMPLE_INTERVAL 50
#define READ_SAMPLE_TIMES 5
#define DC_GAIN 8.5 //define the DC gain of amplifier

/* USER CODE END PD */

/* Private macro -----*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----*/
ADC_HandleTypeDef hadc1;
```

```

SPI_HandleTypeDef hspi1;

TIM_HandleTypeDef htim1;

UART_HandleTypeDef huart1;
UART_HandleTypeDef huart2;

/* USER CODE BEGIN PV */

/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_SPI1_Init(void);
static void MX_TIM1_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_USART2_UART_Init(void);
static void MX_ADC1_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code -----*/
/* USER CODE BEGIN 0 */

/***** Metavlites *****/
uint8_t Rh_byte1, Rh_byte2, Temp_byte1, Temp_byte2;
uint16_t SUM, RH, TEMP;
float Temperature = 0;
float Humidity = 0;
uint8_t Presence = 0;
int ppm=0;
char str[120];
int moisture_sensor = 0;//moisture_sensor
int moisture_sensor_2 = 0;//moisture_sensor_2
int rain_sensor = 0;//rain_sensor
int i=0;
uint16_t ADC_VAL[4]; // gia tous ADC, na mpoun se pinaka kai na staloun mesw tis
seiriakis
float sensor1 = 0; //CO2_sensor
float CO2_sensor = 0;
float voltage_CO2;
uint8_t received_Data_ESP32[10]; //gia anagnwsi apo seiriaki thira
int leitourgia_seiriakis=0;
int flag_seiriakis=0;
int flag_potismatos=0;
int flag_psiksis=0;
int flag_thermansis=0;
int flag_CO2=0;
int flag_anoigma_parathirwn=0;
int flag_stop_motor_A=0;
int flag_stop_motor_B=0;
int flag_SIM800L_restart=0;
int flag_SIM800L_high_temperature=0;
int flag_SIM800L_high_temperature_disable=0;
int flag_SIM800L_low_temperature=0;

```

```

int flag_SIM800L_low_temperature_disable=0;
int flag_SIM800L_rain=0;
int flag_SIM800L_rain_disable=0;
float ZERO_POINT_VOLTAGE=0.295; //output of the sensor in volts when the
concentration of CO2 is 400PPM (1,058volts)
float REACTION_VOLTAGE=0.030;
int windows=0;
int potisma=0;
int psiksi=0;
int therm=0;
int dioxide=0;
//float CO2Curve[3] = {2.602,ZERO_POINT_VOLTAGE,(REACTION_VOLTAGE/(2.602-3))}
//two points are taken from the curve.
//with these two points, a line is formed which is
//"approximately equivalent" to the original curve.
//data format:{ x, y, slope}; point1: (lg400, 0.324), point2: (lg4000, 0.280)
//slope = ( reaction voltage ) / (log400 -log1000)

/***** DHT22 FUNCTIONS *****/
/***** Delay in microseconds for DHT22 *****/
void delay (uint16_t time){
    /* change your code here for the delay in microseconds */
    __HAL_TIM_SET_COUNTER(&htim1, 0);
    while ((__HAL_TIM_GET_COUNTER(&htim1))<time);
}

/***** Configuration of pin for DHT_22 *****/
void Set_Pin_Output (GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin){
    GPIO_InitTypeDef GPIO_InitStructure = {0};
    GPIO_InitStructure.Pin = GPIO_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOx, &GPIO_InitStructure);
}

void Set_Pin_Input (GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin){
    GPIO_InitTypeDef GPIO_InitStructure = {0};
    GPIO_InitStructure.Pin = GPIO_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
    GPIO_InitStructure.Pull = GPIO_PULLUP;
    HAL_GPIO_Init(GPIOx, &GPIO_InitStructure);
}

/***** Start DHT_22 *****/
void DHT22_Start (void){
    Set_Pin_Output(DHT22_PORT, DHT22_PIN); // set the pin as output
    HAL_GPIO_WritePin(DHT22_PORT, DHT22_PIN, 0); // pull the pin low
    delay(1200); // wait for > 1ms
    HAL_GPIO_WritePin(DHT22_PORT, DHT22_PIN, 1); // pull the pin high
    delay(20); // wait for 30us ( 20-40)
    Set_Pin_Input(DHT22_PORT, DHT22_PIN); // set as input
}

/***** Elegxos antapokrisis DHT_22 *****/

```



```

uint8_t DHT22_Check_Response (void){
    Set_Pin_Input(DHT22_PORT, DHT22_PIN);    // set as input
    uint8_t Response = 0;
    delay(40); // wait for 40us
    if (!(HAL_GPIO_ReadPin(DHT22_PORT, DHT22_PIN))){ // if the pin is low
        delay(80); // wait for 80us
        if ((HAL_GPIO_ReadPin(DHT22_PORT, DHT22_PIN))){ //if the pin is
high,response is ok
            Response = 1;
        }
        else{
            Response = -1;
        }
    }
    while ((HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN))); // wait for the pin
to go low
    return Response;
}

/***** Read data from DHT_22 *****/
uint8_t DHT22_Read (void){
    uint8_t i,j;
    for (j=0;j<8;j++){
        while (!(HAL_GPIO_ReadPin(DHT22_PORT, DHT22_PIN))); // wait for the
pin to go high
        delay(40); // wait for 40 us
        if (!(HAL_GPIO_ReadPin(DHT22_PORT, DHT22_PIN))){ // if the pin is
low
            i|= ~(1<<(7-j)); // write 0
        }
        else{
            i|= (1<<(7-j)); // if the pin is high, write 1
        }
        while ((HAL_GPIO_ReadPin(DHT22_PORT, DHT22_PIN))); // wait for the
pin to go low
    }
    return i;
}

/***** Stepper Motors *****/
/***** Stepper Motor A *****/
void Motor_A_Open(void){

    if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_8) == 1){
        for( i=0; i<=epanelipseis_moter; i++){
            if(i>epanelipseis_moter){
                break;
            }
            HAL_GPIO_WritePin(GPIOC,STEP_A1_Pin, GPIO_PIN_SET); //1
            HAL_GPIO_WritePin(GPIOC,STEP_A2_Pin, GPIO_PIN_SET); //1
            HAL_GPIO_WritePin(GPIOC,STEP_A3_Pin, GPIO_PIN_RESET);//0
            HAL_GPIO_WritePin(GPIOB,STEP_A4_Pin, GPIO_PIN_RESET);//0
            HAL_Delay(5);
            i++;
        }
    }
}

```

```

        if(i>epanalipseis_moter){
            break;
        }
        HAL_GPIO_WritePin(GPIOC,STEP_A1_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOC,STEP_A2_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOC,STEP_A3_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOB,STEP_A4_Pin, GPIO_PIN_RESET);//0
        HAL_Delay(5);
        i++;
        if(i>epanalipseis_moter){
            break;
        }
        HAL_GPIO_WritePin(GPIOC,STEP_A1_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOC,STEP_A2_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOC,STEP_A3_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOB,STEP_A4_Pin, GPIO_PIN_SET); //1
        HAL_Delay(5);
        i++;
        if(i>epanalipseis_moter){
            break;
        }
        HAL_GPIO_WritePin(GPIOC,STEP_A1_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOC,STEP_A2_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOC,STEP_A3_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOB,STEP_A4_Pin, GPIO_PIN_SET); //1
        HAL_Delay(5);
        i++;
        if(i>epanalipseis_moter){
            break;
        }
    }
}
}

```

```

void Motor_A_Close(void){
    flag_stop_motor_A=1;
    while(flag_stop_motor_A==1){

        if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_8) == 1){
            flag_stop_motor_A=0;
            break;
        }
        HAL_GPIO_WritePin(GPIOC,STEP_A1_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOC,STEP_A2_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOC,STEP_A3_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOB,STEP_A4_Pin, GPIO_PIN_RESET);//0
        HAL_Delay(5);

        if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_8) == 1){
            flag_stop_motor_A=0;
            break;
        }
        HAL_GPIO_WritePin(GPIOC,STEP_A1_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOC,STEP_A2_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOC,STEP_A3_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOB,STEP_A4_Pin, GPIO_PIN_SET); //1
        HAL_Delay(5);
    }
}

```

```

    if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_8) == 1){
        flag_stop_motor_A=0;
        break;
    }
    HAL_GPIO_WritePin(GPIOC,STEP_A1_Pin, GPIO_PIN_RESET);//0
    HAL_GPIO_WritePin(GPIOC,STEP_A2_Pin, GPIO_PIN_RESET);//0
    HAL_GPIO_WritePin(GPIOC,STEP_A3_Pin, GPIO_PIN_SET); //1
    HAL_GPIO_WritePin(GPIOB,STEP_A4_Pin, GPIO_PIN_SET); //1
    HAL_Delay(5);

    if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_8) == 1){
        flag_stop_motor_A=0;
        break;
    }
    HAL_GPIO_WritePin(GPIOC,STEP_A1_Pin, GPIO_PIN_RESET);//0
    HAL_GPIO_WritePin(GPIOC,STEP_A2_Pin, GPIO_PIN_SET); //1
    HAL_GPIO_WritePin(GPIOC,STEP_A3_Pin, GPIO_PIN_SET); //1
    HAL_GPIO_WritePin(GPIOB,STEP_A4_Pin, GPIO_PIN_RESET);//0
    HAL_Delay(5);
}
}
}

```

/****** Stepper Motor B *****/

```

void Motor_B_Open(void){
    if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_15) == 1){
        for( i=0; i<=epanalipseis_moter; i++){
            if(i>epanalipseis_moter){
                break;
            }
            HAL_GPIO_WritePin(GPIOB,STEP_B1_Pin, GPIO_PIN_SET); //1
            HAL_GPIO_WritePin(GPIOB,STEP_B2_Pin, GPIO_PIN_SET); //1
            HAL_GPIO_WritePin(GPIOB,STEP_B3_Pin, GPIO_PIN_RESET);//0
            HAL_GPIO_WritePin(GPIOB,STEP_B4_Pin, GPIO_PIN_RESET);//0
            HAL_Delay(5);
            i++;
            if(i>epanalipseis_moter){
                break;
            }
            HAL_GPIO_WritePin(GPIOB,STEP_B1_Pin, GPIO_PIN_SET); //1
            HAL_GPIO_WritePin(GPIOB,STEP_B2_Pin, GPIO_PIN_RESET);//0
            HAL_GPIO_WritePin(GPIOB,STEP_B3_Pin, GPIO_PIN_RESET);//0
            HAL_GPIO_WritePin(GPIOB,STEP_B4_Pin, GPIO_PIN_SET); //1
            HAL_Delay(5);
            i++;
            if(i>epanalipseis_moter){
                break;
            }
            HAL_GPIO_WritePin(GPIOB,STEP_B1_Pin, GPIO_PIN_RESET);//0
            HAL_GPIO_WritePin(GPIOB,STEP_B2_Pin, GPIO_PIN_RESET);//0
            HAL_GPIO_WritePin(GPIOB,STEP_B3_Pin, GPIO_PIN_SET); //1
            HAL_GPIO_WritePin(GPIOB,STEP_B4_Pin, GPIO_PIN_SET); //1
            HAL_Delay(5);
            i++;
            if(i>epanalipseis_moter){
                break;
            }
        }
    }
}

```

```

    }
    HAL_GPIO_WritePin(GPIOB,STEP_B1_Pin, GPIO_PIN_RESET);//0
    HAL_GPIO_WritePin(GPIOB,STEP_B2_Pin, GPIO_PIN_SET); //1
    HAL_GPIO_WritePin(GPIOB,STEP_B3_Pin, GPIO_PIN_SET); //1
    HAL_GPIO_WritePin(GPIOB,STEP_B4_Pin, GPIO_PIN_RESET);//0
    HAL_Delay(5);
    i++;
    if(i>epanalipseis_moter){
        break;
    }
}
}
}

```

```

void Motor_B_Close(void){
    flag_stop_motor_B=1;
    while(flag_stop_motor_B==1){
        if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_15) == 1){
            flag_stop_motor_B=0;
            break;
        }
        HAL_GPIO_WritePin(GPIOB,STEP_B1_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOB,STEP_B2_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOB,STEP_B3_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOB,STEP_B4_Pin, GPIO_PIN_RESET);//0
        HAL_Delay(5);

        if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_15) == 1){
            flag_stop_motor_B=0;
            break;
        }
        HAL_GPIO_WritePin(GPIOB,STEP_B1_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOB,STEP_B2_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOB,STEP_B3_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOB,STEP_B4_Pin, GPIO_PIN_RESET);//0
        HAL_Delay(5);

        if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_15) == 1){
            flag_stop_motor_B=0;
            break;
        }
        HAL_GPIO_WritePin(GPIOB,STEP_B1_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOB,STEP_B2_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOB,STEP_B3_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOB,STEP_B4_Pin, GPIO_PIN_SET); //1
        HAL_Delay(5);

        if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_15) == 1){
            flag_stop_motor_B=0;
            break;
        }
        HAL_GPIO_WritePin(GPIOB,STEP_B1_Pin, GPIO_PIN_SET); //1
        HAL_GPIO_WritePin(GPIOB,STEP_B2_Pin, GPIO_PIN_RESET);//0
        HAL_GPIO_WritePin(GPIOB,STEP_B3_Pin, GPIO_PIN_RESET);//0

```

```

        HAL_GPIO_WritePin(GPIOB,STEP_B4_Pin, GPIO_PIN_SET); //1
        HAL_Delay(5);
    }
}

/***** STOP MOTOR A *****/
void Motor_A_Stop(void){
    HAL_GPIO_WritePin(GPIOC,STEP_A1_Pin, GPIO_PIN_RESET); //0
    HAL_GPIO_WritePin(GPIOC,STEP_A2_Pin, GPIO_PIN_RESET); //0
    HAL_GPIO_WritePin(GPIOC,STEP_A3_Pin, GPIO_PIN_RESET); //0
    HAL_GPIO_WritePin(GPIOB,STEP_A4_Pin, GPIO_PIN_RESET); //0
}

/***** STOP MOTOR B *****/
void Motor_B_Stop(void){
    HAL_GPIO_WritePin(GPIOB,STEP_B1_Pin, GPIO_PIN_RESET); //0
    HAL_GPIO_WritePin(GPIOB,STEP_B2_Pin, GPIO_PIN_RESET); //0
    HAL_GPIO_WritePin(GPIOB,STEP_B3_Pin, GPIO_PIN_RESET); //0
    HAL_GPIO_WritePin(GPIOB,STEP_B4_Pin, GPIO_PIN_RESET); //0
}

/***** Analogikes eisodoi *****/
/***** 1os ADC *****/
void ADC_Select_CH0 (void){
    ADC_ChannelConfTypeDef sConfig = {0};
    sConfig.Channel = ADC_CHANNEL_0;
    sConfig.Rank = ADC_REGULAR_RANK_1;
    sConfig.SamplingTime = ADC_SAMPLETIME_28CYCLES_5;
    // sConfig.SingleDiff = ADC_SINGLE_ENDED;
    // sConfig.OffsetNumber = ADC_OFFSET_NONE;
    // sConfig.Offset = 0;
    // sConfig.OffsetSignedSaturation = DISABLE;
    if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK){
        Error_Handler();
    }
}

/***** 2os ADC *****/
void ADC_Select_CH1 (void){
    ADC_ChannelConfTypeDef sConfig = {0};
    sConfig.Channel = ADC_CHANNEL_1;
    sConfig.Rank = ADC_REGULAR_RANK_1;
    sConfig.SamplingTime = ADC_SAMPLETIME_41CYCLES_5;
    // sConfig.SingleDiff = ADC_SINGLE_ENDED;
    // sConfig.OffsetNumber = ADC_OFFSET_NONE;
    // sConfig.Offset = 0;
    // sConfig.OffsetSignedSaturation = DISABLE;
    if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK){
        Error_Handler();
    }
}

```

```

/***** 30s ADC *****/
void ADC_Select_CH4 (void){
    ADC_ChannelConfTypeDef sConfig = {0};
    sConfig.Channel = ADC_CHANNEL_4;
    sConfig.Rank = ADC_REGULAR_RANK_1;
    sConfig.SamplingTime = ADC_SAMPLETIME_55CYCLES_5;
    // sConfig.SingleDiff = ADC_SINGLE_ENDED;
    // sConfig.OffsetNumber = ADC_OFFSET_NONE;
    // sConfig.Offset = 0;
    // sConfig.OffsetSignedSaturation = DISABLE;
    if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK){
        Error_Handler();
    }
}

/***** 40s ADC *****/
void ADC_Select_CH6 (void){
    ADC_ChannelConfTypeDef sConfig = {0};
    sConfig.Channel = ADC_CHANNEL_6;
    sConfig.Rank = ADC_REGULAR_RANK_1;
    sConfig.SamplingTime = ADC_SAMPLETIME_71CYCLES_5;
    // sConfig.SingleDiff = ADC_SINGLE_ENDED;
    // sConfig.OffsetNumber = ADC_OFFSET_NONE;
    // sConfig.Offset = 0;
    // sConfig.OffsetSignedSaturation = DISABLE;
    if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK){
        Error_Handler();
    }
}

/***** MG811 functions *****/
float MG811_Read(){
    int i;
    float sensor1 = 0;
    float v = 0;
    for (i = 0; i < READ_SAMPLE_TIMES; i++) {
        ADC_Select_CH0();
        HAL_ADC_Start(&hadc1);
        HAL_ADC_PollForConversion(&hadc1, 1000);
        sensor1 = HAL_ADC_GetValue(&hadc1);
        HAL_ADC_Stop(&hadc1);
        v = v + sensor1;
        HAL_Delay(READ_SAMPLE_INTERVAL);
    }
    v = (v / READ_SAMPLE_TIMES) * 3.3 / 4096;
    return v;
}

/***** Silogi metrisewn apo aisthitires *****/
void Check_Sensors(void){
    /***** DHT_22 *****/
    DHT22_Start();
    Presence = DHT22_Check_Response();
    Rh_byte1 = DHT22_Read (); //akeraio meros ugrasias
    Rh_byte2 = DHT22_Read (); //dekadiko meros ugrasias
}

```

```

Temp_byte1 = DHT22_Read (); //akeraio meros thermokrasias
Temp_byte2 = DHT22_Read (); //dekadiko meros thermokrasias
SUM = DHT22_Read(); //checksum, elegxos isotimias
TEMP = ((Temp_byte1<<8)|Temp_byte2);
RH = ((Rh_byte1<<8)|Rh_byte2);
Temperature = (float) (TEMP/10.0);
Humidity = (float) (RH/10.0);

/***** 1os ADC *****/ //CO2_sensor
voltage_CO2=0;
voltage_CO2 = MG811_Read();
//ST7735_SetRotation(1);
//sprintf(str, "%.3f", voltage_CO2);
//ST7735_WriteString(10, 80, &str, Font_7x10, GREEN,BLACK);

if ((voltage_CO2 / DC_GAIN) >= ZERO_POINT_VOLTAGE) {
    ppm = -1;
}
else{
    ppm = pow(10, ((voltage_CO2 / DC_GAIN) - ZERO_POINT_VOLTAGE) /
(REACTION_VOLTAGE/(2.602 - 3)) + 2.602);
    //ST7735_SetRotation(1);
    //sprintf(str, "%d", ppm);
    //ST7735_WriteString(10, 95, &str, Font_7x10, GREEN,BLACK);
}

/***** 2os ADC *****/ // moisture_sensor
ADC_Select_CH1();
HAL_ADC_Start(&hadc1);
HAL_ADC_PollForConversion(&hadc1, 1000);
moisture_sensor = HAL_ADC_GetValue(&hadc1);
HAL_ADC_Stop(&hadc1);
HAL_Delay(10);

/***** 3os ADC *****/ // moisture_sensor_2
ADC_Select_CH4();
HAL_ADC_Start(&hadc1);
HAL_ADC_PollForConversion(&hadc1, 1000);
moisture_sensor_2 = HAL_ADC_GetValue(&hadc1);
HAL_ADC_Stop(&hadc1);
HAL_Delay(10);

/***** 4os ADC *****/ //rain_sensor
ADC_Select_CH6();
HAL_ADC_Start(&hadc1);
HAL_ADC_PollForConversion(&hadc1, 1000);
rain_sensor = HAL_ADC_GetValue(&hadc1);
HAL_ADC_Stop(&hadc1);
HAL_Delay(10);
}

```

```

/***** Apikonisi stin TFT
*****/
void TFT_Display(void){
    ST7735_SetRotation(1);

    ST7735_WriteString(0, 0, "Temp: ", Font_7x10, GREEN,BLACK);
    sprintf(str, "%.2f", Temperature);
    //ST7735_WriteString(50, 0, "      ", Font_7x10, GREEN,BLACK);
//katharismos apo skoupidia
    ST7735_WriteString(50, 0, &str, Font_7x10, GREEN,BLACK);
    ST7735_WriteString(95, 0, "C", Font_7x10, GREEN,BLACK);

    ST7735_WriteString(0, 12, "Hum: ", Font_7x10, GREEN,BLACK);
    sprintf(str, "%.2f", Humidity);
    //ST7735_WriteString(50, 12, "      ", Font_7x10, GREEN,BLACK);
    ST7735_WriteString(50, 12, &str, Font_7x10, GREEN,BLACK);
    ST7735_WriteString(95, 12, "%", Font_7x10, GREEN,BLACK);

    ST7735_WriteString(0,24, "CO2:", Font_7x10, GREEN,BLACK);
    //memset(str, '\0', sizeof(str));
    sprintf(str, "%d", ppm);
    ST7735_WriteString(50, 24, "      ", Font_7x10, GREEN,BLACK);
    ST7735_WriteString(50, 24, &str, Font_7x10, GREEN,BLACK);
    ST7735_WriteString(95, 24, "ppm", Font_7x10, GREEN,BLACK);

    ST7735_WriteString(0,36, "Moi1:", Font_7x10, GREEN,BLACK);
    if((moisture_sensor <= 4095) && (moisture_sensor >= 3500)){
        ST7735_WriteString(50,36, "Dry soil      ", Font_7x10, GREEN,BLACK);
    }
    else if((moisture_sensor <= 3499) && (moisture_sensor >= 2000)){
        ST7735_WriteString(50,36, "Humid soil   ", Font_7x10, GREEN,BLACK);
    }
    else if((moisture_sensor <= 1999) && (moisture_sensor >= 0)){
        ST7735_WriteString(50,36, "Full of water", Font_7x10, GREEN,BLACK);
    }
    }

    ST7735_WriteString(0,48, "Moi2:", Font_7x10, GREEN,BLACK);
    if((moisture_sensor_2 <= 4095) && (moisture_sensor_2 >= 3500)){
        ST7735_WriteString(50,48, "Dry soil      ", Font_7x10, GREEN,BLACK);
    }
    else if((moisture_sensor_2 <= 3499) && (moisture_sensor_2 >= 2000)){
        ST7735_WriteString(50,48, "Humid soil   ", Font_7x10, GREEN,BLACK);
    }
    else if((moisture_sensor_2 <= 1999) && (moisture_sensor_2 >= 0)){
        ST7735_WriteString(50,48, "Full of water", Font_7x10, GREEN,BLACK);
    }
    }

    ST7735_WriteString(0,60, "Rain:", Font_7x10, GREEN,BLACK);
    if((rain_sensor <= 4095) && (rain_sensor >= 3500)){
        ST7735_WriteString(50,60, "No rain      ", Font_7x10, GREEN,BLACK);
    }
    else if((rain_sensor <= 3499) && (rain_sensor >= 2500)){
        ST7735_WriteString(50,60, "Light rain", Font_7x10, GREEN,BLACK);
    }
    }

```



```

else if((rain_sensor <= 2499) && (rain_sensor >= 1500)){
    ST7735_WriteString(50,60, "Rain      ", Font_7x10, GREEN, BLACK);
}
else if((rain_sensor <= 1499) && (rain_sensor >= 0)){
    ST7735_WriteString(50,60, "Heavy rain", Font_7x10, GREEN, BLACK);
}
}

```

```

/***** Apostoli_sto_USART1
*****/
void Apostoli_sto_USART1(void){
    HAL_UART_Transmit(&huart1, (uint8_t*)str, sprintf(str, "@Temp=%.2f#\r\n",
Temperature), 1000);
    HAL_UART_Transmit(&huart1, (uint8_t*)str, sprintf(str, "@Humi=%.2f#\r\n",
Humidity), 1000);
    HAL_UART_Transmit(&huart1, (uint8_t*)str, sprintf(str, "@CO2=%d#\r\n",
ppm), 1000);
    HAL_UART_Transmit(&huart1, (uint8_t*)str, sprintf(str, "@Moi1=%d#\r\n",
moisture_sensor), 1000);
    HAL_UART_Transmit(&huart1, (uint8_t*)str, sprintf(str, "@Moi2=%d#\r\n",
moisture_sensor_2), 1000);
    HAL_UART_Transmit(&huart1, (uint8_t*)str, sprintf(str, "@Rain=%d#\r\n",
rain_sensor), 1000);
    HAL_UART_Transmit(&huart1, (uint8_t*)str, sprintf(str, "@Wind=%d#\r\n",
windows), 1000);
    HAL_UART_Transmit(&huart1, (uint8_t*)str, sprintf(str, "@Poti=%d#\r\n",
potisma), 1000);
    HAL_UART_Transmit(&huart1, (uint8_t*)str, sprintf(str, "@Psik=%d#\r\n",
psiksi), 1000);
    HAL_UART_Transmit(&huart1, (uint8_t*)str, sprintf(str, "@Ther=%d#\r\n",
therm), 1000);
    HAL_UART_Transmit(&huart1, (uint8_t*)str, sprintf(str, "@Diox=%d#\r\n",
dioxide), 1000);
}

```

```

/***** elegxos leitourgiwn thermokipiou
*****/
void Elegxos_leitourgiwn_thermokipiou(void){
    if(flag_seiriakis==1){

        /*****energopoiisi flag analoga tin peritpwsii
*****/

        /*****potisma*****/
        if(received_Data_ESP32[0]=='0' &&
received_Data_ESP32[1]=='0' && received_Data_ESP32[2]=='0' &&
received_Data_ESP32[3]=='0'){
            leitourgia_seiriakis=1;
            flag_potismatos=1;
        }
        else if(received_Data_ESP32[0]=='1' &&
received_Data_ESP32[1]=='1' && received_Data_ESP32[2]=='1' &&
received_Data_ESP32[3]=='1'){
            leitourgia_seiriakis=1;
            flag_potismatos=0;
        }
    }
}

```

```

    }

    /*****psiksi*****/
    else if(received_Data_ESP32[0]=='2' &&
received_Data_ESP32[1]=='2' && received_Data_ESP32[2]=='2' &&
received_Data_ESP32[3]=='2'){
        leitourgia_seiriakis=2;
        flag_psiksis=1;
    }
    else if(received_Data_ESP32[0]=='3' &&
received_Data_ESP32[1]=='3' && received_Data_ESP32[2]=='3' &&
received_Data_ESP32[3]=='3'){
        leitourgia_seiriakis=2;
        flag_psiksis=0;
    }

    /*****thermansis*****/
    else if(received_Data_ESP32[0]=='4' &&
received_Data_ESP32[1]=='4' && received_Data_ESP32[2]=='4' &&
received_Data_ESP32[3]=='4'){
        leitourgia_seiriakis=3;
        flag_thermansis=1;
    }
    else if(received_Data_ESP32[0]=='5' &&
received_Data_ESP32[1]=='5' && received_Data_ESP32[2]=='5' &&
received_Data_ESP32[3]=='5'){
        leitourgia_seiriakis=3;
        flag_thermansis=0;
    }

    /*****prosthiki CO2*****/
    else if(received_Data_ESP32[0]=='6' &&
received_Data_ESP32[1]=='6' && received_Data_ESP32[2]=='6' &&
received_Data_ESP32[3]=='6'){
        leitourgia_seiriakis=4;
        flag_CO2=1;
    }
    else if(received_Data_ESP32[0]=='7' &&
received_Data_ESP32[1]=='7' && received_Data_ESP32[2]=='7' &&
received_Data_ESP32[3]=='7'){
        leitourgia_seiriakis=4;
        flag_CO2=0;
    }

    /*****anoigma-kleisimo parathirwn*****/
    else if(received_Data_ESP32[0]=='8' &&
received_Data_ESP32[1]=='8' && received_Data_ESP32[2]=='8' &&
received_Data_ESP32[3]=='8'){
        leitourgia_seiriakis=5;
        flag_anoigma_parathirwn=1; //anoigma
    }
    else if(received_Data_ESP32[0]=='9' &&
received_Data_ESP32[1]=='9' && received_Data_ESP32[2]=='9' &&
received_Data_ESP32[3]=='9'){
        leitourgia_seiriakis=5;
        flag_anoigma_parathirwn=0; //kleisimo
    }
}

```

```

    //memset(received_Data_ESP32, 'A', sizeof(received_Data_ESP32));
    flag_seiriakis=0;

    if(leitourgia_seiriakis==1 && flag_potismatos==1){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, 1);
    }
    if(leitourgia_seiriakis==1 && flag_potismatos==0){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, 0);
    }
    if(leitourgia_seiriakis==2 && flag_psiksis==1){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, 1);
    }
    if(leitourgia_seiriakis==2 && flag_psiksis==0){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, 0);
    }
    if(leitourgia_seiriakis==3 && flag_thermansis==1){
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 1);
    }
    if(leitourgia_seiriakis==3 && flag_thermansis==0){
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 0);
    }
    if(leitourgia_seiriakis==4 && flag_CO2==1){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3, 1);
    }
    if(leitourgia_seiriakis==4 && flag_CO2==0){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3, 0);
    }
    if(leitourgia_seiriakis==5 && flag_anoigma_parathirwn==1){
        Motor_A_Open();
        Motor_A_Stop();
        Motor_B_Open();
        Motor_B_Stop();
    }
    if(leitourgia_seiriakis==5 && flag_anoigma_parathirwn==0){
        Motor_A_Close();
        Motor_A_Stop();
        Motor_B_Close();
        Motor_B_Stop();
    }
}
}

/***** Seiriaki
*****/
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    flag_seiriakis=1;

    HAL_UART_Receive_IT(&huart1, (uint8_t *)&received_Data_ESP32,4);
}

```

```

/***** SIM800L
*****/
void Send_SMS(void){
    char mobile_Number[] = "+306912345678";
    char AT_command[80];
    uint8_t buffer_sim800l[30] = {0};
    uint8_t AT_is_OK = 0;
    while(!AT_is_OK){
        sprintf(AT_command,"AT\r\n");
        HAL_UART_Transmit(&huart2,(uint8_t
*)AT_command,strlen(AT_command),1000);
        HAL_UART_Receive(&huart2, buffer_sim800l, 30, 1000);
        HAL_Delay(1000);
        if(strstr((char *)buffer_sim800l,"OK")){ //vriskei tin prwti
emfanisi tou s2 sto s1.
            AT_is_OK = 1;
        }
        HAL_Delay(1000);
        memset(buffer_sim800l,0,sizeof(buffer_sim800l));
    }
    sprintf(AT_command,"AT+CFUN=1\r\n"); //1 full functionality
    HAL_UART_Transmit(&huart2,(uint8_t *)AT_command,strlen(AT_command),1000);
    HAL_UART_Receive(&huart2, buffer_sim800l, 30, 1000);
    HAL_Delay(1000);
    memset(buffer_sim800l,0,sizeof(buffer_sim800l));
    sprintf(AT_command,"AT+CREG=1\r\n"); //1 register to the network
    HAL_UART_Transmit(&huart2,(uint8_t *)AT_command,strlen(AT_command),1000);
    HAL_UART_Receive(&huart2, buffer_sim800l, 30, 1000);
    HAL_Delay(1000);
    memset(buffer_sim800l,0,sizeof(buffer_sim800l));
    sprintf(AT_command,"AT+CMGF=1\r\n"); //Selects the SMS message format as
text
    HAL_UART_Transmit(&huart2,(uint8_t *)AT_command,strlen(AT_command),1000);
    HAL_UART_Receive(&huart2, buffer_sim800l, 30, 1000);
    HAL_Delay(1000);
    memset(buffer_sim800l,0,sizeof(buffer_sim800l));
    sprintf(AT_command,"AT+CMGS=\"%s\"\r\n",mobile_Number);
    HAL_UART_Transmit(&huart2,(uint8_t *)AT_command,strlen(AT_command),1000);
    HAL_Delay(100);

    /***** Epanekkinisi sistimatos *****/
    if(flag_SIM800L_restart==1){
        flag_SIM800L_restart=0;
        sprintf(AT_command,"The device has been restarted%c",0x1a);
        HAL_UART_Transmit(&huart2,(uint8_t
*)AT_command,strlen(AT_command),1000);
        HAL_UART_Receive(&huart2, buffer_sim800l, 30, 1000);
        memset(buffer_sim800l,0,sizeof(buffer_sim800l));
    }

    /***** Ipsili thermokrasia *****/
    if(flag_SIM800L_high_temperature==1){
        flag_SIM800L_high_temperature=0;
        flag_SIM800L_high_temperature_disable=1;
        sprintf(AT_command,"High temperature at Greenhouse%c",0x1a);

```

```

        HAL_UART_Transmit(&huart2,(uint8_t
*)AT_command,strlen(AT_command),1000);
        HAL_UART_Receive(&huart2, buffer_sim8001, 30, 1000);
        memset(buffer_sim8001,0,sizeof(buffer_sim8001));
    }

    /***** Xamili thermokrasia *****/
    if(flag_SIM800L_low_temperature==1){
        flag_SIM800L_low_temperature=0;
        flag_SIM800L_low_temperature_disable=1;
        sprintf(AT_command,"Low temperature at Greenhouse%c",0x1a);
        HAL_UART_Transmit(&huart2,(uint8_t
*)AT_command,strlen(AT_command),1000);
        HAL_UART_Receive(&huart2, buffer_sim8001, 30, 1000);
        memset(buffer_sim8001,0,sizeof(buffer_sim8001));
    }

    /***** Vrexiei sto thermokipio *****/
    if(flag_SIM800L_rain==1){
        flag_SIM800L_rain=0;
        flag_SIM800L_rain_disable=1;
        sprintf(AT_command,"It's raining at Greenhouse%c",0x1a);
        HAL_UART_Transmit(&huart2,(uint8_t
*)AT_command,strlen(AT_command),1000);
        HAL_UART_Receive(&huart2, buffer_sim8001, 30, 1000);
        memset(buffer_sim8001,0,sizeof(buffer_sim8001));
    }

    HAL_Delay(3000);
}

/***** check for sending SMS *****/
/***** check for sending SMS *****/
void Check_for_SMS(void){
    /***** High Temperature *****/
    if(Temperature<27){
        flag_SIM800L_high_temperature_disable=0;
    }
    if((Temperature>=27) && (flag_SIM800L_high_temperature_disable==0)){
        flag_SIM800L_high_temperature=1; //gia apostoli katallilou minimatos
        Send_SMS();
    }

    /***** Low Temperature *****/
    if(Temperature>23){
        flag_SIM800L_low_temperature_disable=0;
    }
    if((Temperature<=23) && (flag_SIM800L_low_temperature_disable==0)){
        flag_SIM800L_low_temperature=1; //gia apostoli katallilou minimatos
        Send_SMS();
    }

    /***** Raining *****/
    if(rain_sensor>3499){
        flag_SIM800L_rain_disable=0;
    }
    //otan einai anoixta ta parathira kai vrexiei

```

```

        if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_8)==0 && HAL_GPIO_ReadPin(GPIOB,
GPIO_PIN_15)==0 && rain_sensor<=3499 && flag_SIM800L_rain_disable==0){
            flag_SIM800L_rain=1; //gia apostoli katallilou minimatos
            Send_SMS();
        }
    }
}

```

```

/***** elegxos_trexousas_katastasis_twn_leitourgiwn
*****/

```

```

void Elegxos_trexousas_katastasis(void){

```

```

    /***** parathira *****/

```

```

        if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_8)==1 && HAL_GPIO_ReadPin(GPIOB,
GPIO_PIN_15)==1){
            windows=0;//closed
        }
        else{
            windows=1;//opened
        }

```

```

    /***** potisma *****/

```

```

        if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_7)==1){
            potisma=1;//opened
        }
        else{
            potisma=0;//closed
        }

```

```

    /***** psiksi *****/

```

```

        if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_9)==1){
            psiksi=1;//opened
        }
        else{
            psiksi=0;//closed
        }

```

```

    /***** thermans *****/

```

```

        if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_15)==1){
            therm=1;//opened
        }
        else{
            therm=0;//closed
        }

```

```

    /***** CO2 *****/

```

```

        if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_3)==1){
            dioxide=1;//opened
        }
        else{
            dioxide=0;//closed
        }

```

```

}

```

```

/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_SPI1_Init();
    MX_TIM1_Init();
    MX_USART1_UART_Init();
    MX_USART2_UART_Init();
    MX_ADC1_Init();
    /* USER CODE BEGIN 2 */
    HAL_TIM_Base_Start(&htim1); //Start timer_1
    HAL_UART_Receive_IT(&huart1, (uint8_t *)&received_Data_ESP32,4);
    ST7735_Init(0);
    fillScreen(BLACK);
    //HAL_Delay(10000);
    Motor_A_Close();
    Motor_A_Stop();
    Motor_B_Close();
    Motor_B_Stop();
    flag_SIM800L_restart=1; //gia apostoli katalilou minimatos
    Send_SMS();

    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {

```

```

/***** Sensors
*****/
    Check_Sensors();

/***** elegxos leitourgiwn thermokipiou pou tha ektelestoun
*****/
    Elegxos_leitourgiwn_thermokipiou();

/***** elegxos trexousas katastasis twn leitourgiwn
*****/
    Elegxos_trexousas_katastasis();

/***** Apostoli sto USART1
*****/
    Apostoli_sto_USART1();

/***** Sending SMS
*****/
    Check_for_SMS();

/***** Apikonisi stin TFT
*****/
    TFT_Display();

    HAL_Delay(1000);

    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};

    /** Initializes the RCC Oscillators according to the specified parameters
     * in the RCC_OscInitTypeDef structure.
     */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSIState = RCC_HSI_ON;

```



```

RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI_DIV2;
RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL16;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    Error_Handler();
}
/** Initializes the CPU, AHB and APB buses clocks
 */
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                               |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
{
    Error_Handler();
}
PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_ADC;
PeriphClkInit.AdcClockSelection = RCC_ADCCLK2_DIV8;
if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
{
    Error_Handler();
}
}

/**
 * @brief ADC1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_ADC1_Init(void)
{
    /* USER CODE BEGIN ADC1_Init 0 */

    /* USER CODE END ADC1_Init 0 */

    ADC_ChannelConfTypeDef sConfig = {0};

    /* USER CODE BEGIN ADC1_Init 1 */

    /* USER CODE END ADC1_Init 1 */
    /** Common config
     */
    hadc1.Instance = ADC1;
    hadc1.Init.ScanConvMode = ADC_SCAN_ENABLE;
    hadc1.Init.ContinuousConvMode = ENABLE;
    hadc1.Init.DiscontinuousConvMode = DISABLE;
    hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
    hadc1.Init.NbrOfConversion = 1;
    if (HAL_ADC_Init(&hadc1) != HAL_OK)
    {
        Error_Handler();
    }
}

```

```

// /** Configure Regular Channel
// */
// sConfig.Channel = ADC_CHANNEL_0;
// sConfig.Rank = ADC_REGULAR_RANK_1;
// sConfig.SamplingTime = ADC_SAMPLETIME_28CYCLES_5;
// if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
// {
//     Error_Handler();
// }
// /** Configure Regular Channel
// */
// sConfig.Channel = ADC_CHANNEL_1;
// sConfig.Rank = ADC_REGULAR_RANK_2;
// sConfig.SamplingTime = ADC_SAMPLETIME_41CYCLES_5;
// if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
// {
//     Error_Handler();
// }
// /** Configure Regular Channel
// */
// sConfig.Channel = ADC_CHANNEL_4;
// sConfig.Rank = ADC_REGULAR_RANK_3;
// sConfig.SamplingTime = ADC_SAMPLETIME_55CYCLES_5;
// if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
// {
//     Error_Handler();
// }
// /** Configure Regular Channel
// */
// sConfig.Channel = ADC_CHANNEL_6;
// sConfig.Rank = ADC_REGULAR_RANK_4;
// sConfig.SamplingTime = ADC_SAMPLETIME_71CYCLES_5;
// if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
// {
//     Error_Handler();
// }
/* USER CODE BEGIN ADC1_Init 2 */

/* USER CODE END ADC1_Init 2 */

}

/**
 * @brief SPI1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_SPI1_Init(void)
{
    /* USER CODE BEGIN SPI1_Init 0 */

    /* USER CODE END SPI1_Init 0 */

    /* USER CODE BEGIN SPI1_Init 1 */

    /* USER CODE END SPI1_Init 1 */
    /* SPI1 parameter configuration*/
    hspi1.Instance = SPI1;

```

```

hspi1.Init.Mode = SPI_MODE_MASTER;
hspi1.Init.Direction = SPI_DIRECTION_1LINE;
hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;
hspi1.Init.NSS = SPI_NSS_SOFT;
hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_16;
hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
hspi1.Init.CRCPolynomial = 10;
if (HAL_SPI_Init(&hspi1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN SPI1_Init 2 */

/* USER CODE END SPI1_Init 2 */

}

/**
 * @brief TIM1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM1_Init(void)
{
    /* USER CODE BEGIN TIM1_Init 0 */

    /* USER CODE END TIM1_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};

    /* USER CODE BEGIN TIM1_Init 1 */

    /* USER CODE END TIM1_Init 1 */
    htim1.Instance = TIM1;
    htim1.Init.Prescaler = 63;
    htim1.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim1.Init.Period = 65535;
    htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim1.Init.RepetitionCounter = 0;
    htim1.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim1) != HAL_OK)
    {
        Error_Handler();
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim1, &sClockSourceConfig) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim1, &sMasterConfig) != HAL_OK)
    {

```

```

    Error_Handler();
}
/* USER CODE BEGIN TIM1_Init 2 */

/* USER CODE END TIM1_Init 2 */

}

/**
 * @brief USART1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_USART1_UART_Init(void)
{
    /* USER CODE BEGIN USART1_Init 0 */

    /* USER CODE END USART1_Init 0 */

    /* USER CODE BEGIN USART1_Init 1 */

    /* USER CODE END USART1_Init 1 */
    huart1.Instance = USART1;
    huart1.Init.BaudRate = 9600;
    huart1.Init.WordLength = UART_WORDLENGTH_8B;
    huart1.Init.StopBits = UART_STOPBITS_1;
    huart1.Init.Parity = UART_PARITY_NONE;
    huart1.Init.Mode = UART_MODE_TX_RX;
    huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart1.Init.OverSampling = UART_OVERSAMPLING_16;
    if (HAL_UART_Init(&huart1) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN USART1_Init 2 */

    /* USER CODE END USART1_Init 2 */

}

/**
 * @brief USART2 Initialization Function
 * @param None
 * @retval None
 */
static void MX_USART2_UART_Init(void)
{
    /* USER CODE BEGIN USART2_Init 0 */

    /* USER CODE END USART2_Init 0 */

    /* USER CODE BEGIN USART2_Init 1 */

    /* USER CODE END USART2_Init 1 */
    huart2.Instance = USART2;
    huart2.Init.BaudRate = 9600;
    huart2.Init.WordLength = UART_WORDLENGTH_8B;

```

```

huart2.Init.StopBits = UART_STOPBITS_1;
huart2.Init.Parity = UART_PARITY_NONE;
huart2.Init.Mode = UART_MODE_TX_RX;
huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
huart2.Init.OverSampling = UART_OVERSAMPLING_16;
if (HAL_UART_Init(&huart2) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN USART2_Init 2 */

/* USER CODE END USART2_Init 2 */

}

/**
 * @brief GPIO Initialization Function
 * @param None
 * @retval None
 */
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOC, STEP_A3_Pin|STEP_A2_Pin|STEP_A1_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOB, STEP_B4_Pin|STEP_B3_Pin|STEP_B2_Pin|STEP_B1_Pin
        |GPIO_PIN_12|GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_3
        |GPIO_PIN_7|STEP_A4_Pin|GPIO_PIN_9, GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, GPIO_PIN_RESET);

    /*Configure GPIO pins : STEP_A3_Pin STEP_A2_Pin STEP_A1_Pin */
    GPIO_InitStruct.Pin = STEP_A3_Pin|STEP_A2_Pin|STEP_A1_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

    /*Configure GPIO pins : STEP_B4_Pin STEP_B3_Pin STEP_B2_Pin STEP_B1_Pin
        PB12 PB13 PB14 PB3
        PB7 STEP_A4_Pin PB9 */
    GPIO_InitStruct.Pin = STEP_B4_Pin|STEP_B3_Pin|STEP_B2_Pin|STEP_B1_Pin
        |GPIO_PIN_12|GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_3
        |GPIO_PIN_7|STEP_A4_Pin|GPIO_PIN_9;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

    /*Configure GPIO pin : PB15 */

```

```

GPIO_InitStruct.Pin = GPIO_PIN_15;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/*Configure GPIO pins : PA8 PA12 */
GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_12;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/*Configure GPIO pin : PA15 */
GPIO_InitStruct.Pin = GPIO_PIN_15;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state */
    __disable_irq();
    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line number,
    ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

/***** (C) COPYRIGHT STMicroelectronics *****/

```

ΠΑΡΑΡΤΗΜΑ Β: ΚΩΔΙΚΑΣ ESP32

```
#include <ArduinoJson.h>
#include <WiFi.h>
#include <PubSubClient.h>

//Variables for serial
static boolean receive_in_progress = false;
int ndx = 0;
char start_marker = '@';
char end_marker = '#';
char received_char;
const byte numChars = 32;
char receivedChars[numChars];
char buf[30];
boolean newData = false;
int k=0, m=0;

//Variables for sensors
int i, CO2, Moisture_1, Moisture_2, Rain, Windows, Potisma, Psiksi, Thermansi, Dioxide;
float Temperature, Humidity;

//WiFi
const char *ssid = "*****";
const char *password = "*****";

//MQTT Broker - Topic
const char *mqtt_broker = "192.168.1.8";
const int mqtt_port = 1883;
```

```

const char *mqtt_username = "*****";
const char *mqtt_password = "*****";

const char *topic_pub = "Greenhouse/VD_GRH001/Measurements";
//Greenhouse/Serial_Number_of_device/Measurements

const char *topic_pub_status = "Greenhouse/VD_GRH001/Status";
//Greenhouse/Serial_Number_of_device/Status

const char *topic_sub = "Greenhouse/VD_GRH001/Commands";
//Greenhouse/Serial_Number_of_device/Commands

String client_id = "esp32-client-";

long lastMsg1 = 0;
long lastMsg2 = 0;

```

```

WiFiClient espClient;
PubSubClient client(espClient);

```

```

StaticJsonDocument<256> doc1;
StaticJsonDocument<256> doc2;
char JSONmessageBuffer[256];

```

```

void clear_the_buf(){
    for(i=0; i<30; i++){
        buf[i]=0;
    }
}

```

```

void clear_the_json_buf(){
    for(i=0; i<256; i++){
        JSONmessageBuffer[i]=0;
    }
}

```



```

void check_serial(){
  while(Serial.available(>0) && newData == false){
    received_char = Serial.read();
    if(received_char == start_marker){
      receive_in_progress = true;
    }
    if(receive_in_progress == true){
      if(received_char != end_marker){
        receivedChars[ndx] = received_char;
        ndx++;
      }
      else{
        receivedChars[ndx] = '\0'; // terminate the string
        receive_in_progress = false;
        k = ndx;
        ndx = 0;
        newData = true;
      }
    }
  }

  if(newData == true){
    newData = false;
  }
}

void check_sensors(){
  for(i=1; i<k; i++){

```

```
/******Temperature
******/
```

```
    if((receivedChars[1]=='T') && (receivedChars[2]=='e') && (receivedChars[3]=='m') &&
(receivedChars[4]=='p') && (receivedChars[5]=='=')){
        clear_the_buf();
        m=0;
        for(i=6; i<k; i++){
            buf[m]=receivedChars[i];
            m++;
        }
        Temperature = atof(buf);
        if(i==k){
            // Serial.print("Temperature = ");
            // Serial.print(Temperature);
            // Serial.println(" C");
        }
    }
}
```

```
/******Humidity
******/
```

```
    if((receivedChars[1]=='H') && (receivedChars[2]=='u') && (receivedChars[3]=='m') &&
(receivedChars[4]=='i') && (receivedChars[5]=='=')){
        clear_the_buf();
        m=0;
        for(i=6; i<k; i++){
            buf[m]=receivedChars[i];
            m++;
        }
        Humidity = atof(buf);
        if(i==k){
            // Serial.print("Humidity = ");
            // Serial.print(Humidity);
            // Serial.println(" %");
        }
    }
}
```

```

/*****
*****/
CO2

if((receivedChars[1]=='C') && (receivedChars[2]=='O') && (receivedChars[3]=='2') &&
(receivedChars[4]=='=')){

    clear_the_buf();

    m=0;
    for(i=5; i<k; i++){
        buf[m]=receivedChars[i];
        m++;
    }
    CO2 = atoi(buf);
    if(i==k){
        // Serial.print("CO2 = ");
        // Serial.print(CO2);
        // Serial.println(" ppm");
    }
}

```

```

/*****
*****/
Moisture_1

if((receivedChars[1]=='M') && (receivedChars[2]=='o') && (receivedChars[3]=='i') &&
(receivedChars[4]=='1') && (receivedChars[5]=='=')){

    clear_the_buf();

    m=0;
    for(i=6; i<k; i++){
        buf[m]=receivedChars[i];
        m++;
    }
    Moisture_1 = atoi(buf);
    if(i==k){
        // Serial.print("Moisture_1 = ");
        // Serial.println(Moisture_1);
    }
}

```

```

/*****
*****/
Moisture_2

if((receivedChars[1]=='M') && (receivedChars[2]=='o') && (receivedChars[3]=='i') &&
(receivedChars[4]=='2') && (receivedChars[5]=='=')){

  clear_the_buf();

  m=0;
  for(i=6; i<k; i++){
    buf[m]=receivedChars[i];
    m++;
  }
  Moisture_2 = atoi(buf);
  if(i==k){
    // Serial.print("Moisture_2 = ");
    // Serial.println(Moisture_2);
  }
}

```

```

/*****
*****/
Rain

if((receivedChars[1]=='R') && (receivedChars[2]=='a') && (receivedChars[3]=='i') &&
(receivedChars[4]=='n') && (receivedChars[5]=='=')){

  clear_the_buf();

  m=0;
  for(i=6; i<k; i++){
    buf[m]=receivedChars[i];
    m++;
  }
  Rain = atoi(buf);
  if(i==k){
    // Serial.print("Rain = ");
    // Serial.println(Rain);
  }
}

```

```

/*****
*****/
Windows

```

```

    if((receivedChars[1]=='W') && (receivedChars[2]=='i') && (receivedChars[3]=='n') &&
(receivedChars[4]=='d') && (receivedChars[5]=='=')){
        clear_the_buf();
        m=0;
        for(i=6; i<k; i++){
            buf[m]=receivedChars[i];
            m++;
        }
        Windows = atof(buf);
        if(i==k){
            // Serial.print("Temperature = ");
            // Serial.print(Temperature);
            // Serial.println(" C");
        }
    }
}

```

```

/***** Potisma *****/

```

```

    if((receivedChars[1]=='P') && (receivedChars[2]=='o') && (receivedChars[3]=='t') &&
(receivedChars[4]=='i') && (receivedChars[5]=='=')){
        clear_the_buf();
        m=0;
        for(i=6; i<k; i++){
            buf[m]=receivedChars[i];
            m++;
        }
        Potisma = atof(buf);
        if(i==k){
            // Serial.print("Temperature = ");
            // Serial.print(Temperature);
            // Serial.println(" C");
        }
    }
}

```

```

/***** Psiksi *****/

```

```

    if((receivedChars[1]=='P') && (receivedChars[2]=='s') && (receivedChars[3]=='i') &&
(receivedChars[4]=='k') && (receivedChars[5]=='=')){
        clear_the_buf();
        m=0;
        for(i=6; i<k; i++){
            buf[m]=receivedChars[i];
            m++;
        }
        Psiksi = atof(buf);
        if(i==k){
            // Serial.print("Temperature = ");
            // Serial.print(Temperature);
            // Serial.println(" C");
        }
    }
}

```

```

/***** Thermansi *****/

```

```

    if((receivedChars[1]=='T') && (receivedChars[2]=='h') && (receivedChars[3]=='e') &&
(receivedChars[4]=='r') && (receivedChars[5]=='=')){
        clear_the_buf();
        m=0;
        for(i=6; i<k; i++){
            buf[m]=receivedChars[i];
            m++;
        }
        Thermansi = atof(buf);
        if(i==k){
            // Serial.print("Temperature = ");
            // Serial.print(Temperature);
            // Serial.println(" C");
        }
    }
}

```

```

/***** Dioxide *****/

```

```

    if((receivedChars[1]=='D') && (receivedChars[2]=='i') && (receivedChars[3]=='o') &&
(receivedChars[4]=='x') && (receivedChars[5]=='=')){
        clear_the_buf();
        m=0;
        for(i=6; i<k; i++){
            buf[m]=receivedChars[i];
            m++;
        }
        Dioxide = atof(buf);
        if(i==k){
            // Serial.print("Temperature = ");
            // Serial.print(Temperature);
            // Serial.println(" C");
        }
    }
    k = 0;
}

```

```

void setup_wifi(){
    delay(10);
    //Serial.println();
    //Serial.print("Connecting to ");
    //Serial.println(ssid);
    WiFi.begin(ssid, password);
    while(WiFi.status()!= WL_CONNECTED) {
        delay(500);
        //Serial.print(".");
    }
    //Serial.println("");
    //Serial.println("WiFi connected");
}

```

```
//Serial.print("IP address: ");  
//Serial.println(WiFi.localIP());  
}
```

```
void reconnect(){  
  while(!client.connected()){ //Loop until to reconnect  
    //Serial.print("Attempting MQTT reconnection...");  
    // Attempt to reconnect  
    if(client.connect(client_id.c_str(), mqtt_username, mqtt_password)){  
      //Serial.println("connected");  
      client.subscribe(topic_sub); //Subscribe  
    }  
    else{  
      //Serial.print("failed, rc=");  
      //Serial.println(client.state());  
      //Serial.println("try again in 5 seconds");  
      delay(5000);  
    }  
  }  
}
```

```
void callback(char *topic, byte *payload, unsigned int length){  
  String payloadTemp;  
  //Serial.print("Message arrived on topic: ");  
  //Serial.println(topic);  
  //Serial.print("Message:");  
  for (int i = 0; i < length; i++){  
    payloadTemp += (char)payload[i];  
  }  
}
```



```

Serial.print(payloadTemp);
//Serial.println();
//Serial.println("-----");
}

void setup(){
  //btStop();
  Serial.begin(9600);
  setup_wifi(); // connecting to a WiFi network
  client.setServer(mqtt_broker, mqtt_port); //connecting to a mqtt broker
  client.setCallback(callback);
  while(!client.connected()){
    client_id += String(WiFi.macAddress());
    //Serial.printf("The client %s connects to the mqtt broker\n", client_id.c_str());
    if(client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
      //Serial.println("mqtt broker connected");
      client.subscribe(topic_sub); //Subscribe
    }
    else{
      //Serial.print("failed with state ");
      //Serial.println(client.state());
      delay(2000);
    }
  }
}

void loop(){
  check_serial();
  check_sensors();
}

```

```

if(!client.connected()){
    reconnect();
}

client.loop();

long now1 = millis();
long now2 = millis();

if(now1 - lastMsg1 > 5000){

    lastMsg1 = now1;
    clear_the_json_buf();

    doc1["Windows"] = Windows;
    doc1["Potisma"] = Potisma;
    doc1["Psiksi"] = Psiksi;
    doc1["Thermansi"] = Thermansi;
    doc1["Dioxide"] = Dioxide;

    serializeJson(doc1,JSONmessageBuffer);
    //Serial.printf("Sending message to MQTT topic: %s\n", topic_pub_status);
    //Serial.println(JSONmessageBuffer);

    if(client.publish(topic_pub_status,          JSONmessageBuffer)          ==          true){
//client.publish(topic_pub_status, Status);
        //Serial.println("Message status sent");
    }
    else{
        //Serial.println("Message status error");
    }
}
}

```

```

if(now2 - lastMsg2 > 60000){

    lastMsg2 = now2;
    clear_the_json_buf();

    doc2["Temperature"] = Temperature;
    doc2["Humidity"] = Humidity;
    doc2["CO2"] = CO2;
    doc2["Moisture_1"] = Moisture_1;
    doc2["Moisture_2"] = Moisture_2;
    doc2["Rain"] = Rain;

    serializeJson(doc2,JSONmessageBuffer);
    //Serial.printf("Sending message to MQTT topic: %s\n", topic_pub);
    //Serial.println(JSONmessageBuffer);

    if(client.publish(topic_pub, JSONmessageBuffer) == true){           //client.publish(topic_pub,
Measurements);
        //Serial.println("Message sent");
    }
    else{
        //Serial.println("Message error");
    }
}
}

```

ΠΑΡΑΡΤΗΜΑ Γ: ΚΩΔΙΚΑΣ PYTHON

```
import paho.mqtt.client as mqtt
import time
import mysql.connector
import json
import datetime

##### στοιχεία για database #####
mydb = mysql.connector.connect(
    host = "127.0.0.1",
    user = "ntamos",
    password = "ntamos",
    database = "thermokipio"
)

##### στοιχεία για MQTT #####
broker = "192.168.1.8"
port = 1883
Serial_Number_of_device = ""
topic_sub1 = ""
topic_sub2 = ""

##### MQTT flags #####
def init_flags():
    mqtt.Client.connected_flag = False
    mqtt.Client.bad_connection_flag = False
    mqtt.Client.disconnect_flag = False
    mqtt.Client.suback_flag = False
```

```
##### Callback functions #####
```

```
def on_connect(client, userdata, flags, rc):
```

```
    if rc==0:
```

```
        client.connected_flag = True
```

```
        client.bad_connection_flag = False
```

```
        print("Connected OK Returned code = ", rc)
```

```
    else:
```

```
        client.connected_flag = False
```

```
        client.bad_connection_flag = True
```

```
        print("Bad connection Returned code = ", rc)
```

```
        #on_disconnect()
```

```
def on_subscribe(client, userdata, mid, qos):
```

```
    print("Subscribe on topic {} OK".format(topic_sub1))
```

```
    print("Subscribe on topic {} OK".format(topic_sub2))
```

```
def on_message(client, userdata, message):
```

```
    print("Message topic = ", message.topic)
```

```
    print("Message qos = ", message.qos)
```

```
    print("message retain flag = ", message.retain)
```

```
    print("Message received : ", str(message.payload.decode("utf-8")))
```

```
    received_payload = str(message.payload.decode("utf-8"))
```

```
    y = json.loads(received_payload)
```

```
    txt = message.topic
```

```

x = txt.rsplit("/")
Greenhouse_serial_number = x[1]
z = x[2]

if z == "Measurements":
    Temperature = y["Temperature"]
    Humidity = y["Humidity"]
    CO2 = y["CO2"]
    if y["Moisture_1"]<=4095 and y["Moisture_1"]>=3500:
        Moisture_1 = 1
    elif y["Moisture_1"]<=3499 and y["Moisture_1"]>=2000:
        Moisture_1 = 2
    elif y["Moisture_1"]<=1999 and y["Moisture_1"]>=0:
        Moisture_1 = 3

    if y["Moisture_2"]<=4095 and y["Moisture_2"]>=3500:
        Moisture_2 = 1
    elif y["Moisture_2"]<=3499 and y["Moisture_2"]>=2000:
        Moisture_2 = 2
    elif y["Moisture_2"]<=1999 and y["Moisture_2"]>=0:
        Moisture_2 = 3

    if y["Rain"]<=4095 and y["Rain"]>=3500:
        Rain = 1
    elif y["Rain"]<=3499 and y["Rain"]>=2500:
        Rain= 2
    elif y["Rain"]<=2499 and y["Rain"]>=1500:
        Rain = 3
    elif y["Rain"]<=1499 and y["Rain"]>=0:
        Rain = 4

```

```

mycursor = mydb.cursor()

Datetime = datetime.datetime.now().strftime("Ωρα:%H:%M Ημερ:%d-%m-%Y")

sql = """INSERT INTO measurements (Temperature, Humidity, Carbon_Dioxide,
Soil_Moisture_1, Soil_Moisture_2, Rain, Datetime, Greenhouse_serial_number) VALUES (%s, %s,
%s, %s, %s, %s, %s, %s)"""

input_data = (Temperature, Humidity, CO2, Moisture_1, Moisture_2, Rain, Datetime,
Greenhouse_serial_number)

mycursor.execute(sql, input_data)

mydb.commit()

```

```

if z == "Status":

```

```

    Windows = y["Windows"]

```

```

    Watering = y["Potisma"]

```

```

    Cooling = y["Psiksi"]

```

```

    Heating = y["Thermansi"]

```

```

    Dioxide = y["Dioxide"]

```

```

mycursor = mydb.cursor()

```

```

sql = """UPDATE greenhouses SET Windows=%s, Watering=%s, Cooling=%s, Heating=%s,
Dioxide=%s WHERE Greenhouse_serial_number=%s"""

```

```

input_data = (Windows, Watering, Cooling, Heating, Dioxide, Greenhouse_serial_number)

```

```

mycursor.execute(sql, input_data)

```

```

mydb.commit()

```

```

def on_publish(client, userdata, result):

```

```

    print("data published \n")

```

```

def on_disconnect(client, userdata, rc):
    client.connected_flag = False
    client.disconnect_flag = True
    client.loop_stop()
    print("Client disconnected ok")

#def on_log(client, userdata, level, buf):
#    print("log: ",buf)

##### Other functions #####

def connect():
    while not client.connected_flag and not client.bad_connection_flag:
        try:
            client.connect(broker, port)
            break
        except Exception as e:
            print("Connection failed. Error")
            print("Trying again to connect...")
            time.sleep(1)

def subscribe_to_topics():
    mycursor = mydb.cursor()
    sql = "SELECT Greenhouse_serial_number FROM greenhouses"
    mycursor.execute(sql)
    for row in mycursor:
        str = ".join(row)
        a = "Greenhouse/"

```



```

Serial_Number_of_device = str
b = "/Measurements"
c = "/Status"
topic_sub1 = a + Serial_Number_of_device + b
topic_sub2 = a + Serial_Number_of_device + c
client.subscribe(topic_sub1, qos=2)
client.subscribe(topic_sub2, qos=2)
#print(topic_sub1)
#print(topic_sub2)
mydb.commit()

```

```

#####
#####

```

Main

Program

```

##### Create client #####
print("Creating new instance")
init_flags()
client = mqtt.Client("P1")

```

```

##### Attach functions to callback functions #####
client.on_connect = on_connect
client.on_message = on_message
client.on_publish = on_publish
client.on_subscribe = on_subscribe
client.on_disconnect = on_disconnect
#client.on_log = on_log

```

```

client.loop_start()

```

```
print("Connecting to broker")
connect()
time.sleep(4)
subscribe_to_topics()

while True:
    flag_counter=1
    check_time = datetime.datetime.now()
    Hour = check_time.strftime("%H")
    Minute = check_time.strftime("%M")
    while (Hour=="00" and Minute=="00"):
        if flag_counter==1:
            flag_counter=0
            subscribe_to_topics()
        check_time = datetime.datetime.now()
        Hour = check_time.strftime("%H")
        Minute = check_time.strftime("%M")
```

ΠΑΡΑΡΤΗΜΑ Δ: ΚΩΔΙΚΑΣ SQL

```
-- phpMyAdmin SQL Dump
-- version 5.0.4
-- https://www.phpmyadmin.net/
--
-- Φιλοξενητής: 127.0.0.1
-- Χρόνος δημιουργίας: 24 Φεβ 2024 στις 16:25:51
-- Έκδοση διακομιστή: 10.4.17-MariaDB
-- Έκδοση PHP: 7.4.13

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Βάση δεδομένων: `thermokipio`
--
-----

--
-- Δομή πίνακα για τον πίνακα `greenhouses`
--

CREATE TABLE `greenhouses` (
  `Email` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
```

```

`Greenhouse_name` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
`Greenhouse_serial_number` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
`Topic` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
`Windows` varchar(10) COLLATE utf8_unicode_ci NOT NULL,
`Watering` varchar(10) COLLATE utf8_unicode_ci NOT NULL,
`Cooling` varchar(10) COLLATE utf8_unicode_ci NOT NULL,
`Heating` varchar(10) COLLATE utf8_unicode_ci NOT NULL,
`Dioxide` varchar(10) COLLATE utf8_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--
-- Άδειασμα δεδομένων του πίνακα `greenhouses`
--

INSERT INTO `greenhouses` (`Email`, `Greenhouse_name`, `Greenhouse_serial_number`, `Topic`,
`Windows`, `Watering`, `Cooling`, `Heating`, `Dioxide`) VALUES
('ntamotsidisvasilis@gmail.com', 'Πολίχνη', 'VD_GRH001', 'Greenhouse/VD_GRH001/Commands',
'0', '0', '0', '0', '0'),
('ntamotsidisvasilis@gmail.com', 'Πολίχνη', 'VD_GRH002', 'Greenhouse/VD_GRH002/Commands',
'0', '0', '0', '0', '0'),
('ntamotsidisvasilis@gmail.com', 'Χαλκιδική', 'VD_GRH003', 'Greenhouse/VD_GRH003/Commands',
'0', '0', '0', '0', '0'),
('ntamotsidisvasilis@gmail.com', 'Χαλκιδική', 'VD_GRH004', 'Greenhouse/VD_GRH004/Commands',
'0', '0', '0', '0', '0'),
('giannis123@gmail.com', 'Χαλκιδική', 'VD_GRH100', 'Greenhouse/VD_GRH100/Commands', '0', '0',
'0', '0', '0'),
('xristos123@gmail.com', 'Λάρισα', 'VD_GRH111', 'Greenhouse/VD_GRH111/Commands', '0', '0', '0',
'0', '0');

-----

--
-- Δομή πίνακα για τον πίνακα `measurements`
--

```

```

CREATE TABLE `measurements` (
  `Email` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
  `Temperature` float(10,2) NOT NULL,
  `Humidity` float(10,2) NOT NULL,
  `Carbon_Dioxide` smallint(6) NOT NULL,
  `Soil_Moisture_1` mediumint(9) NOT NULL,
  `Soil_Moisture_2` mediumint(9) NOT NULL,
  `Rain` mediumint(9) NOT NULL,
  `Datetime` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
  `Greenhouse_serial_number` varchar(100) COLLATE utf8_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--
-- Αδειασμα δεδομένων του πίνακα `measurements`
--

INSERT INTO `measurements` (`Email`, `Temperature`, `Humidity`, `Carbon_Dioxide`,
`Soil_Moisture_1`, `Soil_Moisture_2`, `Rain`, `Datetime`, `Greenhouse_serial_number`) VALUES
('ntamotsidisvasilis@gmail.com', 0.00, 0.00, 0, 0, 0, 0, '', 'VD_GRH001'),
('ntamotsidisvasilis@gmail.com', 0.00, 0.00, 0, 0, 0, 0, '', 'VD_GRH002'),
('ntamotsidisvasilis@gmail.com', 0.00, 0.00, 0, 0, 0, 0, '', 'VD_GRH003'),
('ntamotsidisvasilis@gmail.com', 0.00, 0.00, 0, 0, 0, 0, '', 'VD_GRH004'),
('giannis123@gmail.com', 0.00, 0.00, 0, 0, 0, 0, '', 'VD_GRH100'),
('xristos123@gmail.com', 0.00, 0.00, 0, 0, 0, 0, '', 'VD_GRH111'),

-----

--
-- Δομή πίνακα για τον πίνακα `personal_data`
--

CREATE TABLE `personal_data` (
  `Name` varchar(30) COLLATE utf8_unicode_ci NOT NULL,

```

```

`Surname` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
`Email` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
`Password` varchar(500) COLLATE utf8_unicode_ci NOT NULL,
`Must_change_password` varchar(20) COLLATE utf8_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--
-- Άδειασμα δεδομένων του πίνακα `personal_data`
--

INSERT INTO `personal_data` (`Name`, `Surname`, `Email`, `Password`, `Must_change_password`)
VALUES
('Βασίλης',          'Νταμοτσίδης',          'ntamotsidisvasilis@gmail.com',
'$2y$10$IjFSI/NBUpn5.zbaLhJrs.9KO7UojktJk6yXx0d/IJz3nHKHrW4Ka', ''),
('Γιάννης',          'Γιάννου',          'giannis123@gmail.com',
'$2y$10$Ad2Rsp0YnEhGRKP5hjrGuJix2xZjdMnMZRQHfSECxV5EFxY6Vw7K', ''),
('Χρήστος',          'Χρήστου',          'xristos123@gmail.com',
'$2y$10$epK1cHbGvbxTnPCv9gV4AOYm5eRUdIAzHAFc2szDQgR.ssikovHW6', ''),
('Σωτήρης',          'Σωτηρίου',          'sotiris123@gmail.com',
'$2y$10$aJrtCDMdexJ7W8CsV/46HuNZ5Ikpo/5/exVUX7I3rz3IVFfs0TuY6', ''),
('Νίκος',            'Νικολάου',          'nikos123@gmail.com',
'$2y$10$RbvflTU3VPozuoTb6gAcYe.TDaKZifIEGB6OgVT8Br0XuKTnL/W6', '');

-- Ευρετήρια για άχρηστους πίνακες

-- Ευρετήρια για πίνακα `personal_data`
--
ALTER TABLE `personal_data` ADD FULLTEXT KEY `Name` (`Name`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

ΠΑΡΑΡΤΗΜΑ Ε: ΚΩΔΙΚΕΣ ΙΣΤΟΣΕΛΙΔΑΣ

```
/****** account_settings.css *****/
```

```
.container{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 1fr 1fr 0.1fr;
  gap: 0px 0px;
  grid-auto-flow: row;
  grid-template-areas:
    ". Change-personal-data ."
    ". Delete-account ."
    ". . .";
}

.header-1{
  text-align: center;
}

body{
  font-family: Arial,Helvetica,sans-serif;
  font-size: 1em;
  background-color: rgb(247, 247, 247);
}

input::placeholder{
  opacity: 0.4;
  font-family: Arial,Helvetica,sans-serif;
  font-size: 1em;
  font-style: italic;
}

/***** Change Data *****/
.Change-personal-data{
  grid-area: Change-personal-data;
  border: 2px solid rgb(223, 223, 223);
  border-radius: 10px;
  box-shadow: 0 0 5px 1px rgb(223, 223, 223);
  height: 310px;
```

```

width: 600px;
/* margin: 70px 0px 5px 60px; */
margin: auto;
margin-top: 70px;
background-color: white;
}

/***** form-control-change *****/
.form-control-change{
  font-family: Arial,Helvetica,sans-serif;
  font-size: 1em;
  border: 1px solid rgb(223, 223, 223);
  border-radius: 5px;
  width: 300px;
  margin: 5px 5px 5px 145px;
  padding:7px 5px 5px 5px;
  display: inline;
}

.form-control-change:focus{
  outline: none;
  border-color: #0275d8;
}

.form-control-change:hover{
  box-shadow: 0 0 5px 1px rgb(223, 223, 223);
}

/***** submit-control-change *****/
.submit-control-change{
  font-family: Arial,Helvetica,sans-serif;
  font-size: 14px;
  border: 2px solid #0275d8;
  border-radius: 5px;
  margin: 0px 50px 10px 245px;
  padding: 8px 25px 8px 25px;
  background-color: #0275d8;
  color: white;
}

.submit-control-change:focus{
  border: 2px solid #178df5;
  background-color: #178df5;
}

```



```

.submit-control-change:hover{
    box-shadow: 0px 0px 3px 2px rgb(223, 223, 223);
}

/***** Delete Account *****/
.Delete-account{
    grid-area: Delete-account;
    border: 2px solid rgb(223, 223, 223);
    border-radius: 10px;
    box-shadow: 0 0 5px 1px rgb(223, 223, 223);
    height: 260px;
    width: 600px;
    /* margin: 50px 0px 5px 60px; */
    margin: auto;
    margin-top: 50px;
    background-color: white;
}

/***** form-control-delete *****/
.form-control-delete{
    font-family: Arial,Helvetica,sans-serif;
    font-size: 1em;
    border: 1px solid rgb(223, 223, 223);
    border-radius: 5px;
    width: 300px;
    margin: 5px 5px 5px 145px;
    padding:7px 5px 5px 5px;
}

.form-control-delete:focus{
    outline: none;
    border-color: #0275d8;
}

.form-control-delete:hover{
    box-shadow: 0 0 5px 1px rgb(223, 223, 223);
}

.header-2{
    text-align: center;
    color: red;
    margin: 15px 0px 0px 0px;
}

```

```

/***** submit-control-delete *****/
.submit-control-delete{
    font-family: Arial,Helvetica,sans-serif;
    font-size: 14px;
    border: 2px solid #0275d8;
    border-radius: 5px;
    margin: 0px 50px 10px 245px;
    padding: 8px 25px 8px 25px;
    background-color: #0275d8;
    color: white;
}

.submit-control-delete:focus{
    border: 2px solid #178df5;
    background-color: #178df5;
}

.submit-control-delete:hover{
    box-shadow: 0px 0px 3px 2px rgb(223, 223, 223);
}

```

```

/***** account_settings.php *****/

```

```

<?php
    session_start();
    include 'server_connection.php'; //sindesi me tin vasi dedomenwn
?>

<!DOCTYPE html>

<html lang="el" lang="en">

    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="icon" type="image/x-icon" href="images/favicon.ico">
        <title>Account settings</title>
        <link rel="stylesheet" href="account_settings.css">
        <script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
        </head>

    <body>

        <?php

```

```

function change_password_and_deletion_form(){
?>

<!-- Header -->
<div>
    <header class=header_private>
        <?php include 'header_private.php'; ?>
    </header>
</div>

<div class="container">
    <!------- Change personal data ----->
    <div class="Change-personal-data">
        <form class="personal-data-form" method="POST"
autocomplete="off">
            <div>
                <h3 class="header-1"> <u>Αλλαγή κωδικού
πρόσβασης</u></h3>
            </div>

            <!-- Email -->
            <div>
                <input class="form-control-change" type="email"
id="email" name="email" placeholder="Email" required/><br>
            </div>

            <!-- Old password input -->
            <div>
                <input class="form-control-change" type="password"
id="old_password" name="old_password" placeholder="Παλιός κωδικός πρόσβασης"
required/><br>
            </div>

            <!-- Password input -->
            <div>
                <input class="form-control-change" type="password"
id="new_password" name="new_password" placeholder="Νέος κωδικός πρόσβασης"
required/><br>
            </div>

            <!-- Password input confirmation -->
            <div>
                <input class="form-control-change" type="password"
id="password_confirm" name="password_confirm" placeholder="Επιβεβαίωση νέου
κωδικού πρόσβασης" required/><br>
            </div>

            <!-- Submit button -->

```

```

        <div>
            <br>
            <button class="submit-control-change"
type="submit" name="Submit_1" value="Υποβολή">Υποβολή</button>
        </div>
    </form>
</div>

<!------- Delete account ----->
<div class="Delete-account">
    <form class="personal-data-form-delete" method="POST"
autocomplete="off">
        <div>
            <h3 class="header-1"> <u>Διαγραφή
λογαριασμού</u></h3>
        </div>

        <!-- Email input -->
        <div>
            <input class="form-control-delete" type="email"
id="email" name="email" placeholder="Email" required/><br>
        </div>

        <!-- Password input -->
        <div>
            <input class="form-control-delete" type="password"
id="password" name="password" placeholder="Κωδικός πρόσβασης" required/><br>
        </div>

        <div>
            <h4 class="header-2"> * Προσοχή η ενέργεια αυτή
είναι μη αναστρέψιμη</h4>
        </div>

        <!-- Submit button -->
        <div>
            <button class="submit-control-delete"
type="submit" name="Submit_2" value="Υποβολή">Υποβολή</button>
        </div>
    </form>
</div>
</div>

<!-- Footer -->
<div>
    <footer>
        <?php include 'footer.php'; ?>

```

```

        </footer>
    </div>

<?php
}
?>

<?php
    if(isset($_SESSION['email'])){
        header("Location: login_check.php");
        exit();
    }

    // For changing password
    if(isset($_POST['Submit_1'])){
        // elegxos stoixeiwn kai ektelesi diadikasias
        if(isset($_POST['email']) && isset($_POST['old_password']) &&
isset($_POST['new_password']) && isset($_POST['password_confirm'])){
            try{
                if($_POST['new_password']===$_POST['password_confirm']
){
                    $stmt=$pdo->prepare("SELECT * FROM
personal_data WHERE email=?");
                    $stmt->execute([$_POST['email']]);
                    if($stmt->rowCount()>0){
                        $user = $stmt->fetch(); //fernei mono mia
eggrafi kai o deiktis paei stin epomeni grammi, etsi otan tin kalesoume ksana
na mas ferei tin epomeni eggrafi
                        if
(password_verify($_POST['old_password'], $user['Password'])){
                            $hashed_password =
password_hash($_POST['new_password'], PASSWORD_DEFAULT);
                            $email = $user['Email'];
                            $stmt=$pdo->prepare("UPDATE
personal_data SET Password='$hashed_password' WHERE Email='$email'");
                            $stmt->execute([$hashed_password]);
                            change_password_and_deletion_form();
                            ?>
                            <script>
                                swal({
                                    text: "Ο κωδικός σας
άλλαξε με επιτυχία!",
                                    icon: "success",

```

```

    });
    </script>
    <?php
    }
    }
}
else{
    change_password_and_deletion_form();
    ?>
    <script>
        swal({
            title: "Σφάλμα!",
            text: "Παρακαλούμε
πληκτρολογήστε\ησωστά τον νέο κωδικό!",
            icon: "error",
        });
    </script>
    <?php
    }
}
catch(PDOException $e){
    change_password_and_deletion_form();
    ?>
    <script>
        swal({
            title: "Σφάλμα!",
            text: "Παρουσιάστηκε πρόβλημα, δοκιμάστε
ξανά!",
            icon: "error",
        });
    </script>
    <?php
    }
}
}

//For account deletion
else if(isset($_POST['Submit_2'])){
    // elegxos stoxeiwn kai ektelesi diadikias
    if(isset($_POST['email']) && isset($_POST['password'])){
        try{
            $stmt=$pdo->prepare("SELECT * FROM personal_data WHERE
email=?");
            $stmt->execute([$_POST['email']]);
            if($stmt->rowCount()>0){
                $user = $stmt->fetch();
                if (password_verify($_POST['password'],
$user['Password'])){
                    $email = $user['Email'];

```

```

WHERE Email='$email');
$stmt=$pdo->prepare("DELETE FROM personal_data
WHERE Email='$email');
$stmt->execute();
$stmt=$pdo->prepare("DELETE FROM measurements
WHERE Email='$email');
$stmt->execute();
session_destroy();
header("Location: login.php");
}
else{
change_password_and_deletion_form();
?>
<script>
swal({
title: "Σφάλμα!",
text: "Ο κωδικός που δώσατε\ηδεν
αντιστοιχεί με το email!",
icon: "error",
});
</script>
<?php
}
}
}
}
catch(PDOException $e){
change_password_and_deletion_form();
?>
<script>
swal({
title: "Σφάλμα!",
text: "Παρουσιάστηκε πρόβλημα, δοκιμάστε
ξανά!",
icon: "error",
});
</script>
<?php
}
}
}
}
else{
change_password_and_deletion_form();
}
?>
</body>
</html>

```

```
/****** change_the_password.css *****/
```

```
.container{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 0.3fr 1fr 0.8fr;
  gap: 0px 0px;
  grid-auto-flow: row;
  grid-template-areas:
    ". . ."
    ". personal-data-form ."
    ". . .";
}

.container_2{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 2fr 1fr 1fr;
  gap: 0px 0px;
  grid-auto-flow: row;
  grid-template-areas:
    ". . ."
    ". alert_message ."
    ". . .";
}

.header-1{
  text-align: center;
}

body{
  font-family: Arial,Helvetica,sans-serif;
  font-size: 1em;
  background-color: rgb(247, 247, 247);
}

input::placeholder{
  opacity: 0.4;
  font-family: Arial,Helvetica,sans-serif;
  font-size: 1em;
  font-style: italic;
}
```



```

.alert_message{
  grid-area: alert_message;
  text-align: center;
  color: red;
}

/***** Change Data *****/

.personal-data-form{
  grid-area: personal-data-form;
  border: 2px solid rgb(223, 223, 223);
  border-radius: 10px;
  box-shadow: 0 0 5px 1px rgb(223, 223, 223);
  height: 310px;
  width: 600px;
  margin: 70px 0px 5px 60px;
  background-color: white;
}

/***** form-control-change *****/

.form-control-change{
  font-family: Arial,Helvetica,sans-serif;
  font-size: 1em;
  border: 1px solid rgb(223, 223, 223);
  border-radius: 5px;
  width: 300px;
  margin: 5px 5px 5px 145px;
  padding:7px 5px 5px 5px;
  display: inline;
}

.form-control-change:focus{
  outline: none;
  border-color: #0275d8;
}

.form-control-change:hover{
  box-shadow: 0 0 5px 1px rgb(223, 223, 223);
}

/***** submit-control-change *****/

.submit-control-change{
  font-family: Arial,Helvetica,sans-serif;

```

```

font-size: 14px;
border: 2px solid #0275d8;
border-radius: 5px;
margin: 0px 50px 10px 245px;
padding: 8px 25px 8px 25px;
background-color: #0275d8;
color: white;
}

.submit-control-change:focus{
border: 2px solid #178df5;
background-color: #178df5;
}

.submit-control-change:hover{
box-shadow: 0px 0px 3px 2px rgb(223, 223, 223);
}

```

/****** change_the_password.php *****/

```

<?php
    session_start();
    include 'server_connection.php'; //sindesi me tin vasi dedomenwn
?>

<!DOCTYPE html>

<html lang="el" lang="en">

    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="icon" type="image/x-icon" href="images/favicon.ico">
        <title>Password changing</title>
        <link rel="stylesheet" href="change_the_password.css">
        <script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
        </head>

    <body>

        <?php
            function change_password_form(){
                ?>

                <!-- Change personal data -->
                <div class="container">

```

```

        <form class="personal-data-form" method="POST"
autocomplete="off">
            <div>
                <h3 class="header-1"> <u>Αλλαγή κωδικού
πρόσβασης</u></h3>
            </div>

            <!-- Email -->
            <div>
                <input class="form-control-change" type="email"
id="email" name="email" placeholder="Email" required/><br>
            </div>

            <!-- Old password input -->
            <div>
                <input class="form-control-change" type="password"
id="old_password" name="old_password" placeholder="Τρέχων κωδικός πρόσβασης"
required/><br>
            </div>

            <!-- Password input -->
            <div>
                <input class="form-control-change" type="password"
id="new_password" name="new_password" placeholder="Νέος κωδικός πρόσβασης"
required/><br>
            </div>

            <!-- Password input confirmation -->
            <div>
                <input class="form-control-change" type="password"
id="password_confirm" name="password_confirm" placeholder="Επιβεβαίωση νέου
κωδικού πρόσβασης" required/><br>
            </div>

            <!-- Submit button -->
            <div>
                <br>
                <button class="submit-control-change"
type="submit" name="Submit" value="Υποβολή">Υποβολή</button>
            </div>
        </form>
    </div>

    <?php
    }
    ?>

```

```

<?php
    // For changing password
    if(isset($_POST['Submit'])){
        // elegxos stoxeiwn kai ektelesi diadikasias
        if(isset($_POST['email']) && isset($_POST['old_password']) &&
isset($_POST['new_password']) && isset($_POST['password_confirm'])){
            try{
                if($_POST['new_password']===$_POST['password_confirm']
){
                    $stmt=$pdo->prepare("SELECT * FROM personal_data
WHERE email=?");
                    $stmt->execute([$_POST['email']]);
                    if($stmt->rowCount(>0){
                        $user = $stmt->fetch(); //fernei mono mia
eggrafi kai o deiktis paei stin epomeni grammi, etsi otan tin kalesoume ksana
na mas ferei tin epomeni eggrafi
                        if(password_verify($_POST['old_password'],
$user['Password'])){
                            $hashed_password =
password_hash($_POST['new_password'], PASSWORD_DEFAULT);
                            $email = $user['Email'];
                            $stmt=$pdo->prepare("UPDATE personal_data
SET Password='$hashed_password' WHERE Email='$email'");
                            $stmt->execute([$hashed_password]);
                            $stmt=$pdo->prepare("UPDATE personal_data
SET Must_change_password=NULL WHERE Email='$email'");
                            $stmt->execute();
                            session_destroy();
                            ?>
                            <div class="container_2">
                                <div class="alert_message">
                                    <h2><br><br>
                                        <a
href="login.php">Συνδεθείτε στο λογαριασμό σας</a>
                                    </h2>
                                </div>
                            </div>
                            <script>
                                swal({
                                    title: "Ο κωδικός σας άλλαξε
με επιτυχία!",
                                    text: "Συνδεθείτε στο
λογαριασμό σας!",
                                    icon: "success",
                                });
                            </script>

```

```

        <?php
        }
        else{
            change_password_form();
            ?>
            <script>
                swal({
                    title: "Λανθανσμένος τρέχων
κωδικός!",
                    text: "Δοκιμάστε ξανά!",
                    icon: "error",
                });
            </script>
            <?php
            }
        }
    }
    else{
        change_password_form();
        ?>
        <script>
            swal({
                title: "Παρακαλούμε πληκτρολογήστε
σωστά τον νέο κωδικό!",
                text: "Δοκιμάστε ξανά!",
                icon: "error",
            });
        </script>
        <?php
        }
    }
    catch(PDOException $e){
        change_password_form();
        ?>
        <script>
            swal({
                title: "Παρουσιάστηκε πρόβλημα!",
                text: "Δοκιμάστε ξανά!",
                icon: "error",
            });
        </script>
        <?php
        }
    }
}
else{
    change_password_form();
}
?>

```



```

grid-template-areas:
  ". . ."
  ". forget-password-form ."
  ". . .";
}

.container_after_sending_email{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 1.5fr 1fr 0.5fr;
  gap: 0px 0px;
  grid-template-areas:
    ". . ."
    ". after_sending_email ."
    ". . .";
}

body{
  font-family: Arial,Helvetica,sans-serif;
  font-size: 1em;
  background-color: rgb(247, 247, 247);
}

h3{
  margin-top: 50px;
  color: rgb(143, 143, 143);
  text-align: center;
  font-weight: normal;
}

.after_sending_email{
  grid-area: after_sending_email;
  font-size: 25px;
  color: black;
  background-color: rgb(247, 247, 247);
  text-align:center;
}

.forget-password-form{
  grid-area: forget-password-form;
  border: 2px solid rgb(223, 223, 223);
  border-radius: 10px;
  box-shadow: 0 0 5px 1px rgb(223, 223, 223);
  height: 350px;
  width: 450px;
}

```

```

margin: 30px 0px 5px 60px;
/* text-align:justify; */
background-color: white;
}

input::placeholder{
  opacity: 0.4;
  font-family: Arial,Helvetica,sans-serif;
  font-size: 1em;
  font-style: italic;
}

/***** form-label *****/
.form-label{
  display: inline-block;
  margin-top: 1em;
  background-color: white;
}

/***** form-control *****/
.form-control{
  font-family: Arial,Helvetica,sans-serif;
  font-size: 1em;
  border: 1px solid rgb(223, 223, 223);
  border-radius: 5px;
  width: 350px;
  margin: 0px 5px 5px 50px;
  padding:7px 5px 5px 5px;
}

.form-control:focus{
  outline: none;
  border-color: #0275d8;
}

.form-control:hover{
  box-shadow: 0 0 5px 1px rgb(223, 223, 223);
}

/***** submit *****/
.submit-control{
  display: inline;
  font-family: Arial,Helvetica,sans-serif;
  font-size: 14px;
  border: 2px solid #0275d8;
  border-radius: 5px;
}

```



```

margin: 50px 0px 5px 140px;
padding: 8px 25px 8px 25px;
color: white;
background-color: #0275d8;
}

.submit-control:focus{
border: 2px solid #178df5;
background-color: #178df5;
}

.submit-control:hover{
box-shadow: 0px 0px 3px 2px rgb(223, 223, 223);
}

```

/****** forget_password.php *****/

```

<?php
include 'server_connection.php'; //sindesi me tin vasi dedomenwn
?>

<!DOCTYPE html>
<html lang="el" lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="icon" type="image/x-icon" href="images/favicon.ico">
    <title>My Greenhouse</title>
    <link rel="stylesheet" href="forget_password.css">
    <script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
  </head>

  <body>

    <?php
function forget_password_form(){
?>

    <!-- Header -->
    <div>
      <header>
        <?php include 'header_public.php'; ?>
      </header>
    </div>

```

```

        <div class="container">
            <form class="forget-password-form" method="POST"
autocomplete="off">
                <div>
                    <h3>Για να επαναφέρετε τον κωδικό<br>
                    πρόσβασής σας, εισάγετε το email σας<br>
                    και πατήστε "Αποστολή κωδικού".<br>
                    </h3>
                </div>

                <!-- Email input -->
                <div>
                    <label class="form-label" for="email"></label><br>
                    <input class="form-control" type="email"
id="email" name="email" placeholder="E-mail" autofocus required/><br>
                </div>

                <!-- Submit button -->
                <div>
                    <button class="submit-control" type="submit"
name="Submit" value="Αποστολή κωδικού">Αποστολή κωδικού</button>
                </div>
            </form>
        </div>

        <!-- Footer -->
        <div>
            <footer>
                <?php include 'footer.php'; ?>
            </footer>
        </div>

<?php
}
?>

<?php
if(isset($_POST['Submit'])){
    if(isset($_POST['email'])){
        $stmt = $pdo->prepare("SELECT * FROM personal_data WHERE
email=?"); // elegxos an iparxei to email sti vasi dedomenwn
        $stmt->execute([$_POST['email']]);
        if($stmt->rowCount()>0){

```

```

        $user = $stmt->fetch(); //fernei mono mia eggrafi kai
o deiktis paei stin epomeni grammi, etsi otan tin kalesoume ksana na mas ferei
tin epomeni eggrafi

        $email = $user['Email'];
        $generated_password=0;
        for($i=0; $i<=5; $i++){ //dimiourgia neou kwdikou
            $number = rand(1,9);
            $generated_password = $generated_password +
($number*(10**$i));
        }
        $hashed_password = password_hash($generated_password,
PASSWORD_DEFAULT);

        //oi epomenes grammes exoun na kanoun me tin apostoli
του email

        ini_set('display_errors', '1');
        error_reporting(E_ALL);
        $from = "ntamotsidisvasilis@gmail.com";
        $to = $email;
        $subject = "Greenhouse";
        $message = "Ο κωδικός σας είναι:
".$generated_password;

        $headers = array(
            "MIME-Version" => "1.0",
            "Content-Type" => "text/html;charset=UTF-8",
            "From" => "<ntamotsidisvasilis@gmail.com>"
        );
        //kataxwrisi tou neou kwdikou sti vasi dedomenwn
        $stmt=$pdo->prepare("UPDATE personal_data SET
Password='$hashed_password' WHERE Email='$email'");
        $stmt->execute([$hashed_password]);
        $stmt=$pdo->prepare("UPDATE personal_data SET
Must_change_password='Yes' WHERE Email='$email'");
        $stmt->execute();
        if(mail($to, $subject, $message, $headers)){
            ?>
            <div class="container_after_sending_email">
                <div class="after_sending_email">
                    <br><br>
                    <p><a href="login.php">Συνδεθείτε στο
λογαριασμό σας</a></p>
                </div>
            </div>
            <script>
                swal({
                    title: "Στάλθηκε κωδικός στο email που
δηλώσατε για την ανάκτηση του λογαριασμού σας!",
                    text: "Συνδεθείτε στο λογαριασμό σας",
                    icon: "success",
                });

```

```

        </script>
        <?php
        }
        else{
            forget_password_form();
        ?>
        <script>
            swal({
                title: "Η αποστολή απέτυχε!",
                text: "Δοκιμάστε ξανά!",
                icon: "error",
            });
        </script>
        <?php
        }
    }
    else{
        ?>
        <script>
            swal({
                title: "Σφάλμα!",
                text: "Δεν υπάρχει χρήστης με αυτό το
email!",
                icon: "error",
            });
        </script>
        <?php
            forget_password_form();
        }
        exit();
    }
}
else{
    forget_password_form();
}
?>

</body>

</html>

```

```

/***** greenhouse_main.css *****/

```

```

body{
    font-family: Arial,Helvetica,sans-serif;
    font-size: 1em;
    background-color: rgb(247, 247, 247);
}

```

```

.container{
  display: grid;
  grid-template-columns: 1.5fr 1.5fr;
  grid-template-rows: 0.7fr 0.7fr 0.7fr 0.7fr;
  gap: 0px 0px;
  /* grid-auto-flow: row; */
  grid-template-areas:
    "ChartBoxTemperature ChartBoxHumidity"
    "ChartBoxMoisture1 ChartBoxMoisture2"
    "ChartBoxDioxide ChartBoxRain"
    "commands commands";
}

/***** Charts *****/
/***** Temperature chart *****/
.ChartBoxTemperature{
  grid-area: ChartBoxTemperature;
  width: 800px;
  height: 300px;
  margin: 0px 0px 5px 80px;
}

/***** Temperature chart *****/
.ChartBoxHumidity{
  grid-area: ChartBoxHumidity;
  width: 800px;
  height: 300px;
  margin: 0px 0px 5px 80px;
}

/***** Temperature chart *****/
.ChartBoxMoisture1{
  grid-area: ChartBoxMoisture1;
  width: 800px;
  height: 300px;
  margin: 0px 0px 5px 80px;
}

/***** Temperature chart *****/
.ChartBoxMoisture2{
  grid-area: ChartBoxMoisture2;
  width: 800px;
  height: 300px;
  margin: 0px 0px 5px 80px;
}

```

```

}

/***** Temperature chart *****/
.ChartBoxDioxide{
  grid-area: ChartBoxDioxide;
  width: 800px;
  height: 300px;
  margin: 0px 0px 5px 80px;
}

/***** Temperature chart *****/
.ChartBoxRain{
  grid-area: ChartBoxRain;
  width: 800px;
  height: 300px;
  margin: 0px 0px 5px 80px;
}

/***** Available greenhouses *****/
/* .available-greenhouses{
  grid-area: available-greenhouses;
}

.header-3{
  text-align: center;
}

.message_available_greenhouses{
  text-align: center;
} */

/***** Buttons - Entoles thermokipiou *****/
.commands{
  display: grid;
  grid-area: commands;
  grid-template-columns: 0.9fr 0.9fr 0.9fr 0.9fr 0.9fr 1.5fr;
  grid-template-rows: 0.7fr 0.7fr 0.7fr;
  gap: 0px 0px;
  /* grid-auto-flow: row; */
  grid-template-areas:
    "output1 output2 output3 output4 output5 select-greenhouse"
    "Button1 Button3 Button5 Button7 Button9 select-greenhouse"
    "Button2 Button4 Button6 Button8 Button10 select-greenhouse";
}

```

```
}

.output1{
  grid-area: output1;
  margin:auto;
  margin-top: 30px;
}

.output2{
  grid-area: output2;
  margin:auto;
  margin-top: 30px;
}

.output3{
  grid-area: output3;
  margin:auto;
  margin-top: 30px;
}

.output4{
  grid-area: output4;
  margin:auto;
  margin-top: 30px;
}

.output5{
  grid-area: output5;
  margin:auto;
  margin-top: 30px;
}

.Button1{
  grid-area: Button1;
  margin:auto;
}

.Button2{
  grid-area: Button2;
  margin:auto;
}

.Button3{
  grid-area: Button3;
  margin:auto;
}

.Button4{
  grid-area: Button4;
  margin:auto;
}
```

```

.Button5{
  grid-area: Button5;
  margin:auto;
}
.Button6{
  grid-area: Button6;
  margin:auto;
}
.Button7{
  grid-area: Button7;
  margin:auto;
}
.Button8{
  grid-area: Button8;
  margin:auto;
}
.Button9{
  grid-area: Button9;
  margin:auto;
}
.Button10{
  grid-area: Button10;
  margin:auto;
}

.Button-control{
  display: inline;
  font-family: Arial,Helvetica,sans-serif;
  font-size: 15px;
  border: 2px solid #0275d8;
  border-radius: 5px;
  /* margin: 30px 0px 5px 40px; */
  padding: 8px 5px 8px 5px;
  color: white;
  background-color: #0275d8;
  width: 130px;
}

.Button-control:focus{
  border: 2px solid #178df5;
  background-color: #178df5;
}

.Button-control:hover{
  box-shadow: 0px 0px 5px 5px rgb(223, 223, 223);
}

```



```

}

/***** select_greenhouse *****/
.select-greenhouse{
  grid-area: select-greenhouse;
  display: block;
  border: 2px solid rgb(223, 223, 223);
  border-radius: 10px;
  box-shadow: 0 0 5px 1px rgb(223, 223, 223);
  height: 120px;
  width: 350px;
  /* margin: 30px 0px 5px 40px; */
  margin:auto;
  background-color: white;
}

.select-greenhouse-form{
  text-align: center;
  margin-top: 10px;
}

/* .select-greenhouse-control{
  border: 1px solid rgb(223, 223, 223);
  border-radius: 5px;
  width: 250px;
  margin: 10px 5px 5px 20px;
  padding:7px 5px 5px 5px;
}

input::placeholder{
  opacity: 0.8;
  font-family: Arial,Helvetica,sans-serif;
  font-size: 15px;
  font-style: italic;
}

.select-greenhouse-control:focus{
  outline: none;
  border-color: #0275d8;
}

.select-greenhouse-control:hover{
  box-shadow: 0 0 5px 1px rgb(223, 223, 223);
} */

/***** submit button *****/

```

```

.select-greenhouse-button-control{
  display: inline;
  font-family: Arial,Helvetica,sans-serif;
  font-size: 15px;
  border: 2px solid #0275d8;
  border-radius: 5px;
  /* margin: 5px 0px 5px 90px; */
  padding: 8px 5px 8px 5px;
  color: white;
  background-color: #0275d8;
  width: 130px;
}

.select-greenhouse-button-control:focus{
  border: 2px solid #178df5;
  background-color: #178df5;
}

.select-greenhouse-button-control:hover{
  box-shadow: 0px 0px 3px 3px rgb(223, 223, 223);
}

.output{
  text-align: center;
  margin-top: 10px;
}

```

/****** greenhouse_main.php *****/

```

<?php
  session_start();
  include 'server_connection.php'; //sindesi me tin vasi dedomenwn
?>

<!DOCTYPE html>

<html lang="el" lang="en">

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="icon" type="image/x-icon" href="images/favicon.ico">
    <title>Greenhouse main</title>
    <link rel="stylesheet" href="greenhouse_main.css">
    <script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  </head>

```

```

<body>

  <?php
    function greenhouse_main_form(){
  ?>

  <!-- Header -->
  <div>
    <header class=header_private>
      <?php include 'header_private.php'; ?>
    </header>
  </div>

  <div class="container">

    <!-- Temperature Chart -->
    <div class="ChartBoxTemperature">
      <canvas id="TemperatureChart"></canvas>
    </div>

    <!-- Humidity Chart -->
    <div class="ChartBoxHumidity">
      <canvas id="HumidityChart"></canvas>
    </div>

    <!-- Soil Moisture 1 Chart -->
    <div class="ChartBoxMoisture1">
      <canvas id="Moisture1Chart"></canvas>
    </div>

    <!-- Soil Moisture 2 Chart -->
    <div class="ChartBoxMoisture2">
      <canvas id="Moisture2Chart"></canvas>
    </div>

    <!-- Carbon Dioxide Chart -->
    <div class="ChartBoxDioxide">
      <canvas id="DioxideChart"></canvas>
    </div>

```



```

        ?>
        <!-- <div
class=message_available_greenhouses> -->
        <?php
            // echo "Παρουσιάστηκε πρόβλημα στην
<br>";
            // echo "εμφάνιση των θερμοκηπίων";
        ?>
        <!-- </div> -->
        <?php
            // }
        ?>
        <!-- </div> -->
    <!-- </div> -->

    <!-- Entoles gia to thermokipio -->
    <div class="commands">

        <div class=output1>
            <output id="demo1"></output>
        </div>

        <div class=output2>
            <output id="demo2"></output>
        </div>

        <div class=output3>
            <output id="demo3"></output>
        </div>

        <div class=output4>
            <output id="demo4"></output>
        </div>

        <div class=output5>
            <output id="demo5"></output>
        </div>

        <div class="Button1">
            <form class="Button-form" method="POST">
                <div>
                    <button class="Button-control"
type="submit" name="PotismaON" value="Πότισμα ON"> Πότισμα ON</button>
                </div>
            </form>

```

```

        </div>

        <div class="Button3">
            <form class="Button-form" method="POST">
                <div>
                    <button class="Button-control"
type="submit" name="CoolingON" value="Ψύξη ON"> Ψύξη ON</button>
                </div>
            </form>
        </div>

        <div class="Button5">
            <form class="Button-form" method="POST">
                <div>
                    <button class="Button-control"
type="submit" name="HeatingON" value="Θέρμανση ON"> Θέρμανση ON</button>
                </div>
            </form>
        </div>

        <div class="Button7">
            <form class="Button-form" method="POST">
                <div>
                    <button class="Button-control"
type="submit" name="CO2ON" value="CO2 ON"> CO2 ON</button>
                </div>
            </form>
        </div>

        <div class="Button9">
            <form class="Button-form" method="POST">
                <div>
                    <button class="Button-control"
type="submit" name="WindowsON" value="Παράθυρα ON"> Παράθυρα ON</button>
                </div>
            </form>
        </div>

        <div class="Button2">
            <form class="Button-form" method="POST">
                <div>
                    <button class="Button-control"
type="submit" name="PotismaOFF" value="Πότισμα OFF"> Πότισμα OFF</button>
                </div>
            </form>
        </div>

        <div class="Button4">
            <form class="Button-form" method="POST">

```

```

        <div>
            <button class="Button-control"
type="submit" name="CoolingOFF" value="Ψύξη OFF"> Ψύξη OFF</button>
        </div>
    </form>
</div>

<div class="Button6">
    <form class="Button-form" method="POST">
        <div>
            <button class="Button-control"
type="submit" name="HeatingOFF" value="Θέρμανση OFF"> Θέρμανση OFF</button>
        </div>
    </form>
</div>

<div class="Button8">
    <form class="Button-form" method="POST">
        <div>
            <button class="Button-control"
type="submit" name="CO2OFF" value="CO2 OFF"> CO2 OFF</button>
        </div>
    </form>
</div>

<div class="Button10">
    <form class="Button-form" method="POST">
        <div>
            <button class="Button-control"
type="submit" name="WindowsOFF" value="Παράθυρα OFF"> Παράθυρα OFF</button>
        </div>
    </form>
</div>

<!-- Select Greenhouse -->
<div class="select-greenhouse">

    <form class="select-greenhouse-form" method="GET"
autocomplete="off">

        <label for="sel-grn">Επιλογή θερμοκηπίου</label>
        <?php
            try{
                $sql = "SELECT * FROM greenhouses
WHERE email='$email'";

                $result = $pdo->query($sql);
            }?>

```

```

        <select name="greenhouse_SN"
id="sel-grn">
        <option value=''></option>
        <?php
            foreach ($result as $row){
                echo '<option
value='. $row['Greenhouse_serial_number']. '>'. $row['Greenhouse_serial_number'].
'</option>';
            }
        ?>
        </select>
        <?php
        }
        catch(PDOException $e){
            ?>
            <div
class=message_available_greenhouses>
                <?php
                    echo "Παρουσιάστηκε
πρόβλημα<br>";
                    echo "δοκιμάστε ξανά";
                ?>
            </div>
            <?php
            }
        ?>
        <br><br>
        <button class="select-greenhouse-button-control"
type="submit" name="select_greenhouse_button"
value="Εκτέλεση">Εκτέλεση</button>
        <div class=output>
            <p id="demo"></p>
        </div>
    </form>
</div>
</div>

</div>

<!-- Footer -->
<div>
    <footer>
        <?php include 'footer.php'; ?>
    </footer>
</div>

```



```

<?php
}
?>

<?php
include 'server_connection.php'; //sindesi me tin vasi dedomenwn
/***** settings for MQTT *****/
require('vendor/autoload.php');
use \PhpMqtt\Client\MqttClient;
use \PhpMqtt\Client\ConnectionSettings;

$server = '192.168.1.8';
$port = 1883;
$clientId = $_SESSION['Email'];
$username = 'vasilis';
$password = 'ntamo';
$clean_session = true;
$mqtt_version = MqttClient::MQTT_3_1_1;

$connectionSettings = (new ConnectionSettings)
->setUsername($username)
->setPassword($password)
->setKeepAliveInterval(60);

$mqtt = new MqttClient($server, $port, $clientId,
$mqtt_version); // Create a new instance of the MQTT client

/***** select greenhouse *****/
if(isset($_GET['select_greenhouse_button'])){
    $Serial_Number_of_device = $_GET['greenhouse_SN'];
    ?>
        <script>
            let Serial_Number_of_device = "<?php
echo"$Serial_Number_of_device"?>";
            document.getElementById("demo").innerHTML =
"Επιλεγμένο θερμοκήπιο: "+Serial_Number_of_device;
        </script>
        <?php
        try{
            $stmt=$pdo->prepare("SELECT * FROM measurements WHERE
greenhouse_serial_number=?");
            $stmt->execute([$_GET['greenhouse_SN']]);
            if($stmt->rowCount()>0){
                $temperature_array = array();
                $humidity_array = array();

```

```

        $soilmoisture1_array = array();
        $soilmoisture2_array = array();
        $dioxide_array = array();
        $rain_array = array();
        while($row = $stmt->fetch()){
            $temperature_array[] = $row["Temperature"];
            $humidity_array[] = $row["Humidity"];
            $soilmoisture1_array[] = $row["Soil_Moisture_1"];
            $soilmoisture2_array[] = $row["Soil_Moisture_2"];
            $dioxide_array[] = $row["Carbon_Dioxide"];
            $rain_array[] = $row["Rain"];
            $datetime_array[] = $row["Datetime"];
        }
    }
    $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
    $stmt->execute([$GET['greenhouse_SN']]);
    if($stmt->rowCount()>0){
        $trexousa_katastasi = $stmt->fetch();
        if($trexousa_katastasi['Windows']=="0"){
            ?><script> let Windows = "<?php echo"OFF";?>";
</script><?php
        }else{
            ?><script> let Windows = "<?php echo"ON";?>";
</script><?php
        }

        if($trexousa_katastasi['Watering']=="0"){
            ?><script> let Watering = "<?php echo"OFF";?>";
</script><?php
        }else{
            ?><script> let Watering = "<?php echo"ON";?>";
</script><?php
        }

        if($trexousa_katastasi['Cooling']=="0"){
            ?><script> let Cooling = "<?php echo"OFF";?>";
</script><?php
        }else{
            ?><script> let Cooling = "<?php echo"ON";?>";
</script><?php
        }

        if($trexousa_katastasi['Heating']=="0"){
            ?><script> let Heating = "<?php echo"OFF";?>";
</script><?php
        }else{
            ?><script> let Heating = "<?php echo"ON";?>";
</script><?php

```

```

    }

    if($trexousa_katastasi['Dioxide']=="0"){
        ?><script> let Dioxide = "<?php echo"OFF";?>";
</script><?php

    }else{
        ?><script> let Dioxide = "<?php echo"ON";?>";
</script><?php

    }
}
}
catch(PDOException $e){
    ?>
    <script>
        swal({
            text: "Δεν φορτώθηκαν δεδομένα!",
            icon: "error",
        });
    </script>
    <?php
    greenhouse_main_form();
}

}
/***** Apostoli entolis ston broker otan patitheo to button
*****/
if(isset($_POST['PotismaON'])){
    try{
        $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
        $stmt->execute([$_GET['greenhouse_SN']]);
        if($stmt->rowCount()>0){
            $user = $stmt->fetch();
            if($user['Watering']=="0"){
                $topic =
'Greenhouse/' .$_GET['greenhouse_SN']. '/Commands';
                $payload = '0000';
                $mqtt->connect($connectionSettings,
$clean_session);

                $mqtt->publish($topic, $payload, 2, true); //
topic, payload, qos, retain
                sleep(1);
                $mqtt->loop(true, true);
                $mqtt->disconnect();
            }
            else if($user['Watering']=="1"){
                ?>
                <script>
                    swal({

```

```

                text: "Είναι ενεργοποιημένο!",
                icon: "warning",
            });
        </script>
    <?php
    }
    }
    greenhouse_main_form();
}
catch(PDOException $e){
    greenhouse_main_form();
}
}
else if(isset($_POST['PotismaOFF'])){
    try{
        $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
        $stmt->execute([$_GET['greenhouse_SN']]);
        if($stmt->rowCount()>0){
            $user = $stmt->fetch();
            if($user['Watering']=="1"){
                $topic =
'Greenhouse/'. $_GET['greenhouse_SN']. '/Commands';
                $payload = '1111';
                $mqtt->connect($connectionSettings,
$clean_session);
                $mqtt->publish($topic, $payload, 2, true); //
topic, payload, qos, retain
                sleep(1);
                $mqtt->loop(true, true);
                $mqtt->disconnect();
            }
            else if($user['Watering']=="0"){
                ?>
                <script>
                    swal({
                        text: "Είναι απενεργοποιημένο!",
                        icon: "warning",
                    });
                </script>
                <?php
            }
        }
        greenhouse_main_form();
    }
    catch(PDOException $e){
        greenhouse_main_form();
    }
}
}

```

```

        else if(isset($_POST['CoolingON'])){
            try{
                $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
                $stmt->execute($_GET['greenhouse_SN']);
                if($stmt->rowCount()>0){
                    $user = $stmt->fetch();
                    if($user['Cooling']=="0"){
                        $topic =
'Greenhouse/'. $_GET['greenhouse_SN']. '/Commands';
                        $payload = '2222';
                        $mqtt->connect($connectionSettings,
$clean_session);
                        $mqtt->publish($topic, $payload, 2, true); //
topic, payload, qos, retain
                        sleep(1);
                        $mqtt->loop(true, true);
                        $mqtt->disconnect();
                    }
                    else if($user['Cooling']=="1"){
                        ?>
                        <script>
                            swal({
                                text: "Είναι ενεργοποιημένο!",
                                icon: "warning",
                            });
                        </script>
                        <?php
                    }
                }
                greenhouse_main_form();
            }
            catch(PDOException $e){
                greenhouse_main_form();
            }
        }
        else if(isset($_POST['CoolingOFF'])){
            try{
                $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
                $stmt->execute($_GET['greenhouse_SN']);
                if($stmt->rowCount()>0){
                    $user = $stmt->fetch();
                    if($user['Cooling']=="1"){
                        $topic =
'Greenhouse/'. $_GET['greenhouse_SN']. '/Commands';
                        $payload = '3333';
                        $mqtt->connect($connectionSettings,
$clean_session);

```

```

        $mqtt->publish($topic, $payload, 2, true); //
topic, payload, qos, retain
        sleep(1);
        $mqtt->loop(true, true);
        $mqtt->disconnect();
    }
    else if($user['Cooling']=="0"){
        ?>
        <script>
            swal({
                text: "Είναι απενεργοποιημένο!",
                icon: "warning",
            });
        </script>
        <?php
    }
    }
    greenhouse_main_form();
}
catch(PDOException $e){
    greenhouse_main_form();
}
}
else if(isset($_POST['HeatingON'])){
    try{
        $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
        $stmt->execute([$_GET['greenhouse_SN']]);
        if($stmt->rowCount()>0){
            $user = $stmt->fetch();
            if($user['Heating']=="0"){
                $topic =
'Greenhouse/' . $_GET['greenhouse_SN'] . '/Commands';
                $payload = '4444';
                $mqtt->connect($connectionSettings,
$clean_session);
                $mqtt->publish($topic, $payload, 2, true); //
topic, payload, qos, retain
                sleep(1);
                $mqtt->loop(true, true);
                $mqtt->disconnect();
            }
            else if($user['Heating']=="1"){
                ?>
                <script>
                    swal({
                        text: "Είναι ενεργοποιημένο!",
                        icon: "warning",
                    });
                </script>
            }
        }
    }
}

```

```

        </script>
        <?php
            }
        }
        greenhouse_main_form();
    }
    catch(PDOException $e){
        greenhouse_main_form();
    }
}
else if(isset($_POST['HeatingOFF'])){
    try{
        $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
        $stmt->execute([$_GET['greenhouse_SN']]);
        if($stmt->rowCount()>0){
            $user = $stmt->fetch();
            if($user['Heating']=="1"){
                $topic =
'Greenhouse/' .$_GET['greenhouse_SN']. '/Commands';
                $payload = '5555';
                $mqtt->connect($connectionSettings,
$clean_session);
                $mqtt->publish($topic, $payload, 2, true); //
topic, payload, qos, retain
                sleep(1);
                $mqtt->loop(true, true);
                $mqtt->disconnect();
            }
            else if($user['Heating']=="0"){
                ?>
                <script>
                    swal({
                        text: "Είναι απενεργοποιημένο!",
                        icon: "warning",
                    });
                </script>
            <?php
                }
            }
            greenhouse_main_form();
        }
        catch(PDOException $e){
            greenhouse_main_form();
        }
    }
}
else if(isset($_POST['CO2ON'])){
    try{

```

```

        $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
        $stmt->execute([$GET['greenhouse_SN']]);
        if($stmt->rowCount(>0){
            $user = $stmt->fetch();
            if($user['Dioxide']=="0"){
                $topic =
'Greenhouse/'. $GET['greenhouse_SN']. '/Commands';
                $payload = '6666';
                $mqtt->connect($connectionSettings,
$clean_session);
                $mqtt->publish($topic, $payload, 2, true); //
topic, payload, qos, retain
                sleep(1);
                $mqtt->loop(true, true);
                $mqtt->disconnect();
            }
            else if($user['Dioxide']=="1"){
                ?>
                <script>
                    swal({
                        text: "Είναι ενεργοποιημένο!",
                        icon: "warning",
                    });
                </script>
                <?php
            }
        }
        greenhouse_main_form();
    }
    catch(PDOException $e){
        greenhouse_main_form();
    }
}
else if(isset($_POST['CO2OFF'])){
    try{
        $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
        $stmt->execute([$GET['greenhouse_SN']]);
        if($stmt->rowCount(>0){
            $user = $stmt->fetch();
            if($user['Dioxide']=="1"){
                $topic =
'Greenhouse/'. $GET['greenhouse_SN']. '/Commands';
                $payload = '7777';
                $mqtt->connect($connectionSettings,
$clean_session);
                $mqtt->publish($topic, $payload, 2, true); //
topic, payload, qos, retain

```



```

        sleep(1);
        $mqtt->loop(true, true);
        $mqtt->disconnect();
    }
    else if($user['Dioxide']=="0"){
        ?>
        <script>
            swal({
                text: "Το είναι απενεργοποιημένο!",
                icon: "warning",
            });
        </script>
        <?php
    }
}
greenhouse_main_form();
}
catch(PDOException $e){
    greenhouse_main_form();
}
}
else if(isset($_POST['WindowsON'])){
    try{
        $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
        $stmt->execute([$_GET['greenhouse_SN']]);
        if($stmt->rowCount()>0){
            $user = $stmt->fetch();
            if($user['Windows']=="0"){
                $topic =
'Greenhouse/' . $_GET['greenhouse_SN'] . '/Commands';
                $payload = '8888';
                $mqtt->connect($connectionSettings,
$clean_session);
                $mqtt->publish($topic, $payload, 2, true); //
topic, payload, qos, retain
                sleep(1);
                $mqtt->loop(true, true);
                $mqtt->disconnect();
            }
        }
        else if($user['Windows']=="1"){
            ?>
            <script>
                swal({
                    text: "Τα παράθυρα είναι ανοιχτά!",
                    icon: "warning",
                });
            </script>
            <?php

```

```

        }
    }
    greenhouse_main_form();
}
catch(PDOException $e){
    greenhouse_main_form();
}
}
else if(isset($_POST['WindowsOFF'])){
    try{
        $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
        $stmt->execute([$_GET['greenhouse_SN']]);
        if($stmt->rowCount()>0){
            $user = $stmt->fetch();
            if($user['Windows']=="1"){
                $topic =
'Greenhouse/'. $_GET['greenhouse_SN']. '/Commands';
                $payload = '9999';
                $mqtt->connect($connectionSettings,
$clean_session);
                $mqtt->publish($topic, $payload, 2, true); //
topic, payload, qos, retain
                sleep(1);
                $mqtt->loop(true, true);
                $mqtt->disconnect();
            }
            else if($user['Windows']=="0"){
                ?>
                <script>
                    swal({
                        text: "Τα παράθυρα είναι κλειστά",
                        icon: "warning",
                    });
                </script>
                <?php
            }
        }
        greenhouse_main_form();
    }
    catch(PDOException $e){
        greenhouse_main_form();
    }
}
else{
    greenhouse_main_form();
}
?>
<script>

```

```

        document.getElementById("demo").innerHTML = "Επιλεγμένο
Θερμοκήπιο: "+Serial_Number_of_device;
    </script>
    <?php

?>

<!-- ***** javascript for charts
***** -->
    <script>
        const datetime = <?php echo json_encode($datetime_array);?>;

        document.getElementById("demo1").innerHTML =
"<h2>"+Watering+"</h2>";
        document.getElementById("demo2").innerHTML =
"<h2>"+Cooling+"</h2>";
        document.getElementById("demo3").innerHTML =
"<h2>"+Heating+"</h2>";
        document.getElementById("demo4").innerHTML =
"<h2>"+Dioxide+"</h2>";
        document.getElementById("demo5").innerHTML =
"<h2>"+Windows+"</h2>";

    </script>

<!-- Temperature Chart -->
    <script>
        const temperature = <?php echo
json_encode($temperature_array);?>;
        const ctx1 = document.getElementById('TemperatureChart');

        new Chart(ctx1, {
            type: 'line',
            data: {
                labels: datetime,
                datasets: [{
                    label: 'Θερμοκρασία',
                    data: temperature,
                    borderWidth: 1,
                    borderColor: 'rgb(255, 0, 0)',
                    tension: 1
                }]
            },
            options: {
                scales: {
                    y: {
                        beginAtZero: true,
                        grid:{

```

```

        //display: false
    },
    title: {
        display: true,
        text: 'Θερμοκρασία [C]'
    }
},
x: {
    ticks: {
        display: false //this will remove only the
label
    },
    grid:{
        display: false
    },
    title: {
        display: true,
        text: 'Χρονική στιγμή'
    }
}
}
}
});
</script>

```

```

<!-- Humidity Chart -->
<script>
    const humidity = <?php echo json_encode($humidity_array);?>;
    const ctx2 = document.getElementById('HumidityChart');

    new Chart(ctx2, {
        type: 'line',
        data: {
            labels: datetime,
            datasets: [{
                label: 'Υγρασία',
                data: humidity,
                borderWidth: 1,
                borderColor: 'rgb(255, 128, 0)',
                tension: 0.5
            }]
        },
        options: {
            scales: {
                y: {
                    beginAtZero: true,
                    grid:{
                        //display: false

```

```

        },
        title: {
            display: true,
            text: 'Υγρασία [%]'
        }
    },
    x: {
        ticks: {
            display: false //this will remove only the
label
        },
        grid:{
            display: false
        },
        title: {
            display: true,
            text: 'Χρονική στιγμή'
        }
    }
}
}
});
</script>

```

```

<!-- Soil Moisture 1 Chart -->
<script>
    const soilmoisture1 = <?php echo
json_encode($soilmoisture1_array);?>;
    const ctx3 = document.getElementById('Moisture1Chart');

    new Chart(ctx3, {
        type: 'line',
        data: {
            labels: datetime,
            datasets: [{
                label: 'Υγρασία εδάφους A',
                data: soilmoisture1,
                borderWidth: 1,
                borderColor: 'rgb(0, 0, 255)',
                tension: 0.5
            }]
        },
        options: {
            scales: {
                y: {
                    beginAtZero: true,
                    grid:{
                        //display: false

```

```

        },
        title: {
            display: true,
            text: 'Υγρασία'
        }
    },
    x: {
        ticks: {
            display: false //this will remove only the
label
        },
        grid:{
            display: false
        },
        title: {
            display: true,
            text: 'Χρονική στιγμή'
        }
    }
}
}
});
</script>

<!-- Soil Moisture 2 Chart -->
<script>
    const soilmoisture2 = <?php echo
json_encode($soilmoisture2_array);?>;
    const ctx4 = document.getElementById('Moisture2Chart');

    new Chart(ctx4, {
        type: 'line',
        data: {
            labels: datetime,
            datasets: [{
                label: 'Υγρασία εδάφους B',
                data: soilmoisture2,
                borderColor: 'rgb(0, 0, 255)',
                borderWidth: 1,
                tension: 0.5
            }]
        },
        options: {
            scales: {
                y: {
                    beginAtZero: true,
                    grid:{
                        //display: false

```

```

        },
        title: {
            display: true,
            text: 'Υγρασία'
        }
    },
    x: {
        ticks: {
            display: false //this will remove only the
label
        },
        grid:{
            display: false
        },
        title: {
            display: true,
            text: 'Χρονική στιγμή'
        }
    }
}
});
</script>

```

```

<!-- Carbon Dioxide Chart -->

```

```

<script>
    const dioxide = <?php echo json_encode($dioxide_array);?>;
    const ctx5 = document.getElementById('DioxideChart');

    new Chart(ctx5, {
        type: 'line',
        data: {
            labels: datetime,
            datasets: [{
                label: 'CO2',
                data: dioxide,
                borderColor: 'rgb(0, 255, 0)',
                borderWidth: 1,
                tension: 0.5
            }]
        },
        options: {
            scales: {
                y: {
                    beginAtZero: true,
                    grid:{
                        //display: false
                    },

```

```

        title: {
            display: true,
            text: 'CO2 [ppm]'
        }
    },
    x: {
        ticks: {
            display: false //this will remove only the
label
        },
        grid:{
            display: false
        },
        title: {
            display: true,
            text: 'Χρονική στιγμή'
        }
    }
}
});
</script>

```

```

<!-- Rain Chart -->
<script>
    const rain = <?php echo json_encode($rain_array);?>;
    const ctx6 = document.getElementById('RainChart');

    new Chart(ctx6, {
        type: 'line',
        data: {
            labels: datetime,
            datasets: [{
                label: 'Βροχή',
                data: rain,
                borderWidth: 1,
                borderColor: 'rgb(255, 0, 255)',
                tension: 0.5
            }]
        },
        options: {
            scales: {
                y: {
                    beginAtZero: true,
                    grid:{
                        // display: false
                    },
                    title: {

```



```

        display: true,
        text: 'Βροχή'
    }
},
x: {
    ticks: {
        display: false //this will remove only the
label
    },
    grid:{
        display: false
    },
    title: {
        display: true,
        text: 'Χρονική στιγμή'
    }
}
}
}
});
</script>

</body>

</html>

```

/****** greenhouse_settings.css *****/

```

.container{
    display: grid;
    grid-template-columns: 0.6fr 1.2fr 1.2fr;
    grid-template-rows: 1fr 1fr 0.45fr;
    gap: 0px 0px;
    grid-auto-flow: row;
    grid-template-areas:
        ". greenhouse-addition available-greenhouses"
        ". greenhouse-deletion available-greenhouses"
        ". . .";
}

.greenhouse-addition{
    grid-area: greenhouse-addition;
    border: 2px solid rgb(223, 223, 223);
    border-radius: 10px;
    box-shadow: 0 0 5px 1px rgb(223, 223, 223);
    height: 260px;
    width: 400px;
}

```

```

    /* margin: 70px 0px 5px 60px; */
    margin: auto;
    margin-top: 70px;
    background-color: white;
}

.greenhouse-deletion{
    grid-area: greenhouse-deletion;
    border: 2px solid rgb(223, 223, 223);
    border-radius: 10px;
    box-shadow: 0 0 5px 1px rgb(223, 223, 223);
    height: 260px;
    width: 400px;
    /* margin: 70px 0px 5px 60px; */
    margin: auto;
    margin-top: 70px;
    background-color: white;
}

.available-greenhouses{
    grid-area: available-greenhouses;
}

body{
    font-family: Arial,Helvetica,sans-serif;
    font-size: 1em;
    background-color: rgb(247, 247, 247);
}

input::placeholder{
    opacity: 0.4;
    font-family: Arial,Helvetica,sans-serif;
    font-size: 1em;
    font-style: italic;
}

/***** Greenhouse addition *****/
/***** greenhouse-addition-form *****/

.header-1{

```

```

    text-align: center;
}

.form-control-addition{
    font-family: Arial,Helvetica,sans-serif;
    font-size: 1em;
    border: 1px solid rgb(223, 223, 223);
    border-radius: 5px;
    width: 300px;
    margin: 5px 5px 5px 45px;
    padding:7px 5px 5px 5px;
    display: inline;
}

.form-control-addition:focus{
    outline: none;
    border-color: #0275d8;
}

.form-control-addition:hover{
    box-shadow: 0 0 5px 1px rgb(223, 223, 223);
}

/***** submit-control-addition *****/
.submit-control-addition{
    font-family: Arial,Helvetica,sans-serif;
    font-size: 14px;
    border: 2px solid #0275d8;
    border-radius: 5px;
    margin: 0px 50px 10px 145px;
    padding: 8px 25px 8px 25px;
    background-color: #0275d8;
    color: white;
}

.submit-control-addition:focus{
    border: 2px solid #178df5;
    background-color: #178df5;
}

.submit-control-addition:hover{
    box-shadow: 0px 0px 3px 2px rgb(223, 223, 223);
}

```

```

/***** Greenhouse deletion *****/
/***** greenhouse-deletion-form *****/

.header-2{
  text-align: center;
}

.form-control-deletion{
  font-family: Arial,Helvetica,sans-serif;
  font-size: 1em;
  border: 1px solid rgb(223, 223, 223);
  border-radius: 5px;
  width: 300px;
  margin: 5px 5px 5px 45px;
  padding: 7px 5px 5px 5px;
  display: inline;
}

.form-control-deletion:focus{
  outline: none;
  border-color: #0275d8;
}

.form-control-deletion:hover{
  box-shadow: 0 0 5px 1px rgb(223, 223, 223);
}

/***** submit-control-deletion *****/
.submit-control-deletion{
  font-family: Arial,Helvetica,sans-serif;
  font-size: 14px;
  border: 2px solid #0275d8;
  border-radius: 5px;
  margin: 0px 50px 10px 145px;
  padding: 8px 25px 8px 25px;
  background-color: #0275d8;
  color: white;
}

.submit-control-deletion:focus{
  border: 2px solid #178df5;
  background-color: #178df5;
}

.submit-control-deletion:hover{
  box-shadow: 0px 0px 3px 2px rgb(223, 223, 223);
}

```

```

}

/***** Available greenhouses *****/

.header-3{
    text-align: center;
}

table{
    width: 50%;
    text-align: center;
    margin-left: auto;
    margin-right: auto;
}

tr{
    height: 30px;
    border-bottom: 1px solid #ce1515;
}

tr:nth-child(even) {
    background-color: #d8d7d7;
}

```

```

/***** greenhouse_settings.php *****/

```

```

<?php
    session_start();
    include 'server_connection.php'; //sindesi me tin vasi dedomenwn
?>

<!DOCTYPE html>

<html lang="el" lang="en">

    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="icon" type="image/x-icon" href="images/favicon.ico">
        <title>Greenhouse settings</title>
        <link rel="stylesheet" href="greenhouse_settings.css">
        <script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
        </head>

```

```

<body>

    <?php
        function addition_deletion_form(){
        ?>

        <!-- Header -->
        <div>
            <header class=header_private>
                <?php include 'header_private.php'; ?>
            </header>
        </div>

        <div class="container">

            <!-- Greenhouse addition -->
            <div class="greenhouse-addition">
                <form class="greenhouse-addition-form" method="POST"
autocomplete="off">
                    <div>
                        <h3 class="header-1"> <u>Προσθήκη
θερμοκηπίου</u></h3>
                    </div>

                    <!-- Email -->
                    <div>
                        <input class="form-control-addition" type="email"
id="email" name="email" placeholder="Email" required/><br>
                    </div>

                    <!-- Greenhouse name -->
                    <div>
                        <input class="form-control-addition" type="text"
id="greenhouse_name" name="greenhouse_name" placeholder="Όνομα θερμοκηπίου"
required/><br>
                    </div>

                    <!-- Serial number of greenhouse -->
                    <div>
                        <input class="form-control-addition" type="text"
id="greenhouse_serial_number" name="greenhouse_serial_number"
placeholder="Serial number θερμοκηπίου" required/><br>
                    </div>

                    <!-- Submit button -->
                    <div>
                        <br>
                    </div>
                </form>
            </div>
        </div>
    </body>

```

```

                <button class="submit-control-addition"
type="submit" name="Submit_1" value="Υποβολή">Υποβολή</button>
            </div>
        </form>
    </div>

    <!-- Greenhouse deletion -->
    <div class="greenhouse-deletion">
        <form class="greenhouse-deletion-form" method="POST"
autocomplete="off">
            <div>
                <h3 class="header-2"> <u>Αφαίρεση
θερμοκηπίου</u></h3>
            </div>

            <!-- Email -->
            <div>
                <input class="form-control-deletion" type="email"
id="email" name="email" placeholder="Email" required/><br>
            </div>

            <!-- Greenhouse name -->
            <div>
                <input class="form-control-deletion" type="text"
id="greenhouse_name" name="greenhouse_name" placeholder="Όνομα θερμοκηπίου"
required/><br>
            </div>

            <!-- Serial number of greenhouse -->
            <div>
                <input class="form-control-deletion" type="text"
id="greenhouse_serial_number" name="greenhouse_serial_number"
placeholder="Serial number θερμοκηπίου" required/><br>
            </div>

            <!-- Submit button -->
            <div>
                <br>
                <button class="submit-control-deletion"
type="submit" name="Submit_2" value="Υποβολή">Υποβολή</button>
            </div>
        </form>
    </div>

    <!-- Available greenhouses-->
    <div class="available-greenhouses">
        <br><br><br>
    </div>

```

```

<h2 class="header-3"><u>Διαθέσιμα θερμοκήπια</u></h2>
<?php
    $server = "localhost";
    $username = "ntamos";
    $password = "ntamos";
    $schema = "thermokipio";
    $pdo = new PDO('mysql:dbname=' . $schema .
';host=' . $server, $username, $password,
[PDO::ATTR_ERRMODE=>PDO::ERRMODE_EXCEPTION, PDO::ATTR_DEFAULT_FETCH_MODE =>
PDO::FETCH_ASSOC, PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"]);

    if(isset($_SESSION['Email'])){
        $email = $_SESSION['Email'];
    }

    try{
        $sql = "SELECT * FROM greenhouses WHERE
email='$email'";

        $result = $pdo->query($sql);
        ?>
        <div class=message_available_greenhouses>
            <?php
                echo
'<table><tr"><th><b><h3>Περιοχή</h3></b></th><th><b><h3>Serial
Number</h3></b></th></tr>';

                foreach ($result as $row){
                    echo
'<tr><td>' . $row['Greenhouse_name'] . '</td><td>' . $row['Greenhouse_serial_number']
.'</td></tr>';
                }
                echo '</table>';
            ?>
        </div>
        <?php
        }
        catch(PDOException $e){
            ?>
            <div class=message_available_greenhouses>
                <?php
                    echo "Παρουσιάστηκε πρόβλημα στην
<br>";

                    echo "εμφάνιση των θερμοκηπίων";
                ?>
            </div>
        <?php
        }
    }
    ?>
</div>
</div>

```



```

<!-- Footer -->
<div>
    <footer>
        <?php include 'footer.php'; ?>
    </footer>
</div>

<?php
}
?>

<?php
if(isset($_SESSION['email'])){
    header("Location: login_check.php");
    exit();
}

// Greenhouse addition
if(isset($_POST['Submit_1'])){
    if(isset($_POST['email']) && isset($_POST['greenhouse_name'])
&& isset($_POST['greenhouse_serial_number'])){
        try{
            $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
            $stmt->execute([\$_POST['greenhouse_serial_number']]);
            if($stmt->rowCount()>0 ){
                ?>
                <script>
                    swal({
                        title: "Σφάλμα!",
                        text: "Η συσκευή υπάρχει και δεν
γίνετε να προστεθεί ξανά!",
                        icon: "error",
                    });
                </script>
                <?php
            }
            else{
                $stmt=$pdo->prepare("INSERT INTO measurements
(email, greenhouse_serial_number) VALUES (?, ?)");
                $stmt->execute([\$_POST['email'],
$_POST['greenhouse_serial_number']]);

```

```

        $topic =
'Greenhouse/' . $_POST['greenhouse_serial_number'] . '/Commands';
        $stmt=$pdo->prepare("INSERT INTO greenhouses
(email, greenhouse_name, greenhouse_serial_number, topic) VALUES (?, ?, ?,
?)");
        $stmt->execute([\$_POST['email'],
$_POST['greenhouse_name'], $_POST['greenhouse_serial_number'], $topic]);
        ?>
        <script>
            swal({
                text: "Η προσθήκη έγινε με επιτυχία!",
                icon: "success",
            });
        </script>
    <?php
    }
    addition_deletion_form();
}
catch(PDOException $e){
    ?>
    <script>
        swal({
            title: "Σφάλμα!",
            text: "Παρουσιάστηκε πρόβλημα,
προσπαθείστε ξανά!",
            icon: "error",
        });
    </script>
    <?php
    addition_deletion_form();
}
}
}
// Greenhouse deletion
else if(isset($_POST['Submit_2'])){
    if(isset($_POST['email']) && isset($_POST['greenhouse_name'])
&& isset($_POST['greenhouse_serial_number'])){
        try{
            $stmt=$pdo->prepare("SELECT * FROM greenhouses WHERE
greenhouse_serial_number=?");
            $stmt->execute([\$_POST['greenhouse_serial_number']]);
            if($stmt->rowCount()>0){
                $user = $stmt->fetch();
                if($_POST['email']===$user['Email'] &&
$_POST['greenhouse_name']===$user['Greenhouse_name'] &&
$_POST['greenhouse_serial_number']===$user['Greenhouse_serial_number']){
                    $email = $user['Email'];
                    $greenhouse_name = $user['Greenhouse_name'];

```

```

        $greenhouse_serial_number =
$user['Greenhouse_serial_number'];
        $stmt=$pdo->prepare("DELETE FROM measurements
WHERE Email='$email' AND
Greenhouse_serial_number='$greenhouse_serial_number'");
        $stmt->execute();
        $stmt=$pdo->prepare("DELETE FROM greenhouses
WHERE Greenhouse_name='$greenhouse_name' AND
Greenhouse_serial_number='$greenhouse_serial_number'");
        $stmt->execute();
        ?>
        <script>
            swal({
                text: "Η διαγραφή έγινε με
επιτυχία!",
                icon: "success",
            });
        </script>
        <?php
    }
    else{
        ?>
        <script>
            swal({
                title: "Σφάλμα!",
                text: "Έχετε δώσει λάθος
στοιχεία!",
                icon: "error",
            });
        </script>
        <?php
    }
}
else{
    ?>
    <script>
        swal({
            title: "Σφάλμα!",
            text: "Δεν υπάρχει στη βάση δεδομένων
τετοιο θερμοκήπιο!",
            icon: "error",
        });
    </script>
    <?php
    }
    addition_deletion_form();
}
catch(PDOException $e){
    ?>

```

```

        <script>
            swal({
                title: "Σφάλμα!",
                text: "Παρουσιάστηκε πρόβλημα,
προσπαθείστε ξανά!",
                icon: "error",
            });
        </script>
        <?php
            addition_deletion_form();
        }
    }
}
else{
    addition_deletion_form();
}
?>

</body>

</html>

```

/****** header_private.css *****/

```

body{
    font-family: Arial,Helvetica,sans-serif;
    font-size: 1em;
}

.home_link, .thermokipio_link, .account_link, .logout_link,
.greenhouse_settings, .account_settings{
    text-decoration: none;
    color: azure;
    font-size: 17px;
}

ul{
    display: flex;
    background-color: #2C394B;
    list-style: none;
    width: 100%;
    margin: 0 auto;
    padding: 0;
    align-items: center;
}

```

```

li{
  color: azure;
  padding: 1em;
  transition: background-color;
  transition-duration: 0.4s;
  display: flex;
  align-items: center;
  flex-basis: 100%;
}

li:hover{
  background-color: #0275d8;
}

/* Hide dropdown by default */
nav ul ul{
  display: none;
  position: absolute;
  top: 62px; /* the height from the main nav */
}

/* Display dropdown on hover */
nav ul li:hover > ul{
  display: block;
  width: auto;
}

/* Fisrt Tier Dropdown */
nav ul ul li {
  width: 220px;
  float:none;
  display: list-item;
  position: relative;
  text-align: left;
}

```

```

/***** header_private.php *****/

```

```

<!DOCTYPE html>

<html lang="el" lang="en">

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="header_private.css">
  </head>

```

```

<body>

    <nav>
        <ul>
            <li><a class=home_link
href="index_private.php">Αρχική</a></li>
            <li><a class=thermokipio_link href="greenhouse_main.php">To
θερμοκήπιο μου</a></li>
            <li class=account_link>Λογαριασμός
                <ul>
                    <li><a class="greenhouse_settings"
href="greenhouse_settings.php">Ρυθμίσεις θερμοκηπίου</a></li>
                    <li><a class="account_settings"
href="account_settings.php">Ρυθμίσεις Λογαριασμού</a></li>
                </ul>
            </li>
            <li><a class=logout_link href="logout.php">Αποσύνδεση</a></li>
            <li>Συνδεθήκατε ως&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<?php echo
$_SESSION['Name'];?>&nbsp;&nbsp;&nbsp;<?php echo $_SESSION['Surname'];?></li>
        </ul>
    </nav>

</body>

</html>

```

```

/***** header_public.css *****/

```

```

body {
    font-family: Arial,Helvetica,sans-serif;
    font-size: 1em;
}

.login_link, .register_link, .home_link {
    text-decoration: none;
    color: azure;
    font-size: 17px;
}

ul {
    display: flex;
    background-color: #2C394B;
    list-style: none;
    width: 100%;
    margin: 0 auto;
    padding: 0;
    align-items: center;
}

```

```

li {
  color: azure;
  padding: 1em;
  transition: background-color;
  transition-duration: 0.4s;
  display: flex;
  align-items: center;
  flex-basis: 100%;
}

li:hover {
  background-color: #0275d8;
}

```

/****** header_public.php *****/

```

<!DOCTYPE html>
<html lang="el" lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="header_public.css">
  </head>

  <body>

    <div>
      <nav>
        <ul>
          <li><a class=home_link href="index.php">Αρχική</a></li>
          <li><a class=register_link
href="register.php">Εγγραφή</a> </li>
          <li><a class=login_link href="login.php">Συνδεθείτε</a>
</li>
        </ul>
      </nav>
    </div>

  </body>

</html>

```

/****** index_private.css *****/

```

.container{
  display: grid;
  grid-template-columns: 0.5fr 2fr 0.5fr;
  grid-template-rows: 0.3fr 2.7fr 0.3fr;
}

```

```

gap: 0px 0px;
grid-template-areas:
    ". . ."
    ". image ."
    ". . .";
}

.image{
    grid-area: image;
    width: 80%;
    height: 90%;
    margin: auto;
    min-width: 700px;
}

```

/****** index_private.php *****/

```

<?php
    session_start();
    include 'server_connection.php'; //sindesi me tin vasi dedomenwn
?>

<!DOCTYPE html>
<html lang="el" lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="icon" type="image/x-icon" href="images/favicon.ico">
        <title>My Greenhouse</title>
        <link rel="stylesheet" href="index_private.css">
    </head>

    <body>

        <!-- Header_private -->
        <div>
            <header>
                <?php include 'header_private.php'; ?>
            </header>
        </div>

        <div class="container">
            
        </div>

```



```

        <!-- Footer -->
        <div>
            <footer>
                <?php include 'footer.php'; ?>
            </footer>
        </div>

    </body>

</html>

```

/****** index.css *****/

```

.container{
    display: grid;
    grid-template-columns: 0.5fr 2fr 0.5fr;
    grid-template-rows: 0.3fr 2.7fr 0.3fr;
    gap: 0px 0px;
    grid-template-areas:
        ". . ."
        ". image ."
        ". . .";
}

.home_link{
    text-decoration: none;
    color: azure;
}

.image{
    grid-area: image;
    width: 80%;
    height: 90%;
    margin: auto;
    min-width: 700px;
}

```

/****** index.php *****/

```

<!DOCTYPE html>
<html lang="el" lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

    <link rel="icon" type="image/x-icon" href="images/favicon.ico">
    <title>My Greenhouse</title>
    <link rel="stylesheet" href="index.css">
</head>

<body>

    <!-- Header -->
    <div>
        <header>
            <?php include 'header_public.php'; ?>
        </header>
    </div>

    <div class="container">
        
    </div>

    <!-- Footer -->
    <div>
        <footer>
            <?php include 'footer.php'; ?>
        </footer>
    </div>

</body>

</html>

```

```

/***** login_check.css *****/

```

```

.alert_message{
    text-align: center;
    color: red;
    margin-top: 300px;
    margin-bottom: 460px;
}

a{
    text-decoration: none;
    color: #0275d8;
}

```

```
/****** login_check.php *****/
```

```
<?php
    session_start();
    include 'server_connection.php'; //sindesi me tin vasi dedomenwn
?>

<!DOCTYPE html>

<html lang="el" lang="en">

    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="icon" type="image/x-icon" href="images/favicon.ico">
        <title>Check login</title>
        <link rel="stylesheet" href="login_check.css">
    </head>

    <body>

        <!-- Header -->
        <div>
            <header class=header_public>
                <?php include 'header_public.php'; ?>
            </header>
        </div>

        <!-- check login -->
        <?php
            if(isset($_SESSION['Email']) && isset($_SESSION['Password'])){
                header("Location: index_private.php");
            }
            else{
                ?>
                <div class="alert_message">
                    <p><h2>Δεν υπάρχει πρόσβαση</h2></p>
                    <p><h2><a href="login.php" target="_self">Πατήστε εδώ
για να συνδεθείτε</a></h2></p>
                </div>
                <?php
            }
        ?>

        <!-- Footer -->
    </div>
```

```

        <footer>
            <?php include 'footer.php'; ?>
        </footer>
    </div>

</body>

</html>

```

```

/***** login.css *****/

```

```

.container{
    display: grid;
    grid-template-columns: 1fr 1.3fr 0.7fr;
    grid-template-rows: 0.5fr 1.5fr 1.2fr;
    gap: 0px 0px;
    grid-template-areas:
        ". . ."
        ". login-form ."
        ". . .";
}

body{
    font-family: Arial,Helvetica,sans-serif;
    font-size: 1em;
    background-color: rgb(247, 247, 247);
}

.alert_message{
    /* grid-area: alert_message; */
    text-align: center;
    color: red;
}

.login-form{
    grid-area: login-form;
    border: 2px solid rgb(223, 223, 223);
    border-radius: 10px;
    box-shadow: 0 0 5px 1px rgb(223, 223, 223);
    height: 350px;
    width: 450px;
    margin: 30px 0px 5px 60px;
    /* text-align: justify; */
    background-color: white;
}

input::placeholder{

```

```

opacity: 0.4;
font-family: Arial,Helvetica,sans-serif;
font-size: 1em;
font-style: italic;
}

/***** form-label *****/
.form-label{
display: inline-block;
margin-top: 1em;
background-color: white;
}

/***** form-control *****/
.form-control{
font-family: Arial,Helvetica,sans-serif;
font-size: 1em;
border: 1px solid rgb(223, 223, 223);
border-radius: 5px;
width: 350px;
margin: 20px 5px 5px 50px;
padding:7px 5px 5px 5px;
}

.form-control:focus{
outline: none;
border-color: #0275d8;
}

.form-control:hover{
box-shadow: 0 0 5px 1px rgb(223, 223, 223);
}

/***** submit *****/
.submit-control{
display: inline;
font-family: Arial,Helvetica,sans-serif;
font-size: 14px;
border: 2px solid #0275d8;
border-radius: 5px;
margin: 25px 0px 5px 55px;
padding: 8px 25px 8px 25px;
color: white;
}

```

```

background-color: #0275d8;
}

.submit-control:focus{
border: 2px solid #178df5;
background-color: #178df5;
}

.submit-control:hover{
box-shadow: 0px 0px 3px 2px rgb(223, 223, 223);
}

/***** Forgot password *****/
.forgot-password{
display: inline;
text-decoration: none;
color: #0275d8;
margin: 0px 0px 5px 30px;
}

/***** Create account *****/
.create-account{
display: inline-block;
text-decoration: none;
color: #0275d8;
margin: 25px 0px 5px 120px;
padding: 8px 25px 8px 25px;
}

```

```

/***** login.php *****/

```

```

<?php
session_start();
include 'server_connection.php'; //sindesi me tin vasi dedomenwn
?>

<!DOCTYPE html>

<html lang="e1" lang="en">

<head>
<meta charset="UTF-8">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="icon" type="image/x-icon" href="images/favicon.ico">
<title>My Greenhouse login</title>
<link rel="stylesheet" href="login.css">
<script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
</head>

<body>

<!-- Login form -->
<?php
function login_form(){
?>

<!-- Header -->
<div>
<header class=header_public>
<?php include 'header_public.php'; ?>
</header>
</div>

<div class="container">
<form class="login-form" method="POST" autocomplete="off">
<!-- Email input -->
<div>
<label class="form-label" for="email"></label><br>
<input class="form-control" type="email" id="email"
name="email" placeholder="E-mail" autofocus required/><br>
</div>

<!-- Password input -->
<div>
<label class="form-label" for="password"></label><br>
<input class="form-control" type="password"
id="password" name="password" placeholder="Κωδικός πρόσβασης"
autocomplete="off" required/><br><br>
</div>

<!-- Submit button & Forgot password -->
<div>
<button class="submit-control" type="submit"
name="Login" value="Συνδεθείτε">Συνδεθείτε</button>
<a class="forgot-password"
href="forget_password.php">Ξεχάσατε τον κωδικό σας;</a><br><br>
</div>

```

```

        <!-- Create account -->
        <div >
            <a class="create-account"
href="register.php">Δημιουργία λογαριασμού</a>
            </div>
        </form>
    </div>

    <!-- Footer -->
    <div>
        <footer>
            <?php include 'footer.php'; ?>
        </footer>
    </div>

<?php
}
?>

<?php
    if(isset($_SESSION['email'])){
        header("Location: login_check.php");
        exit();
    }

    if(isset($_POST['Login'])){
        // elegxos stoxeiwn kai prosvasi
        if(isset($_POST['email']) && isset($_POST['password'])){
            $stmt=$pdo->prepare("SELECT * FROM personal_data WHERE
email=?");

            $stmt->execute([$_POST['email']]);
            if($stmt->rowCount()>0){
                $user = $stmt->fetch(); //fernei mono mia eggrafi kai
o deiktis paei stin epomeni grammi, etsi otan tin kalesoume ksana na mas ferei
tin epomeni eggrafi

                if (password_verify($_POST['password'],
$user['Password']) && $_POST['email']==$user['Email']){
                    if($user['Must_change_password']=='Yes'){
                        header("Location: change_the_password.php");
                    }
                }
            }
            else{
                $_SESSION['Email']=$user['Email'];
                $_SESSION['Password']=$user['Password'];
                $_SESSION['Name']=$user['Name'];
            }
        }
    }
}
?>

```



```

        $_SESSION['Surname']=$user['Surname'];
        header("Location: login_check.php");
    }
}
else{
    login_form();
?>
    <script>
        swal({
            title: "Σφάλμα!",
            text: "Λανθασμένα στοιχεία!",
            icon: "error",
        });
    </script>
    <?php
    // header("Location: login_check.php");
    }
}
else{
?>
    <script>
        swal({
            title: "Σφάλμα!",
            text: "Λανθασμένα στοιχεία!",
            icon: "error",
        });
    </script>
    <?php
    login_form();
    exit();
    }
    exit();
}
}
else{
    login_form();
}
?>
</body>
</html>

```

```

/***** logout.css *****/

```

```

body{
    font-family: Arial,Helvetica,sans-serif;
    font-size: 1em;
}

```

```

background-color: rgb(247, 247, 247);
}

.container{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 0.5fr 1fr 0.8fr;
  gap: 0px 0px;
  grid-template-areas:
    ". . ."
    ". logout-form ."
    ". . .";
}

.logout-form{
  grid-area: logout-form;
  border: 2px solid rgb(223, 223, 223);
  border-radius: 10px;
  box-shadow: 0 0 5px 1px rgb(223, 223, 223);
  height: 300px;
  width: 450px;
  /* margin: 25px 0px 50px 180px; */
  margin: auto;
  margin: 25px;
  background-color: white;
}

.log-out{
  text-align: center;
  color: red;
}

.submit-control{
  font-family: Arial,Helvetica,sans-serif;
  font-size: 17px;
  border: 2px solid #0275d8;
  border-radius: 5px;
  text-align: center;
  background-color: #0275d8;
  color: white;
}

.alert_message{
  text-align: center;
}

```

```
color: red;
margin-top: 300px;
}
```

```
/****** logout.php *****/
```

```
<?php
    session_start();
?>

<!DOCTYPE html>
<html lang="el" lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="icon" type="image/x-icon" href="images/favicon.ico">
        <title>Logout</title>
        <link rel="stylesheet" href="logout.css">
    </head>

    <body>

        <?php
            function logout_form(){
        ?>

        <!-- Header -->
        <div>
            <header class=header_private>
                <?php include 'header_private.php'; ?>
            </header>
        </div>

        <div class="container">
            <form class="logout-form" method="POST">
                <div class="log-out">
                    <br><br><br><h2>Θέλετε να αποσυνδεθείτε; </h2><br><br>
                    <input class="submit-control"
type="submit" name="Log_out" value="Αποσύνδεση"/>
                </div>
            </form>
        </div>
```

```

        <!-- Footer -->
        <div>
            <footer>
                <?php include 'footer.php'; ?>
            </footer>
        </div>

    <?php
    }
    ?>

    <?php
        if(isset($_POST['Log_out'])){
            session_destroy();
            header("Location: login.php");
        }
        else{
            logout_form();
        }
    ?>

</body>

</html>

```

/****** register.css *****/

```

.container{
    display: grid;
    grid-template-columns: 1fr 1.3fr 0.7fr;
    grid-template-rows: 0.3fr 1.5fr 0.8fr;
    gap: 0px 0px;
    grid-template-areas:
        ". . ."
        ". register-form ."
        ". . .";
}

body{
    font-family: Arial,Helvetica,sans-serif;
    font-size: 1em;
    background-color: rgb(247, 247, 247);
}

```

```

.container_after_registration{
  display: grid;
  grid-template-columns: 1fr 1.3fr 0.7fr;
  grid-template-rows: 2fr 1.5fr 5fr;
  gap: 0px 0px;
  grid-template-areas:
    ". . ."
    ". after_registration ."
    ". . .";
}

.after_registration{
  grid-area: after_registration;
}

h2{
  padding: 0px 0px 0px 60px;
}

.login_after_registration{
  text-decoration: none;
  color: #0275d8;
  padding: 0px 0px 0px 120px;
}

.register-form{
  grid-area: register-form;
  border: 2px solid rgb(223, 223, 223);
  border-radius: 10px;
  box-shadow: 0 0 5px 1px rgb(223, 223, 223);
  height: 465px;
  width: 450px;
  margin: 5px 0px 5px 60px;
  background-color: white;
}

input::placeholder{
  opacity: 0.4;
  font-family: Arial,Helvetica,sans-serif;
  font-size: 1em;
  font-style: italic;
}

```

```

/***** form-label *****/
.form-label{
  display: inline-block;
  margin-top: 1em;
  background-color: white;
}

/***** form-control *****/
.form-control{
  font-family: Arial,Helvetica,sans-serif;
  font-size: 1em;
  border: 1px solid rgb(223, 223, 223);
  border-radius: 5px;
  width: 350px;
  margin: 5px 5px 5px 50px;
  padding:7px 5px 5px 5px;
}

.form-control:focus{
  outline: none;
  border-color: #0275d8;
}

.form-control:hover{
  box-shadow: 0 0 5px 1px rgb(223, 223, 223);
}

/***** submit *****/
.submit-control{
  font-family: Arial,Helvetica,sans-serif;
  font-size: 14px;
  border: 2px solid #0275d8;
  border-radius: 5px;
  margin: 20px 50px 10px 170px;
  padding: 8px 25px 8px 25px;
  background-color: #0275d8;
  color: white;
}

.submit-control:focus{
  border: 2px solid #178df5;
  background-color: #178df5;
}

```

```

}

.submit-control:hover{
    box-shadow: 0px 0px 3px 2px rgb(223, 223, 223);
}

```

/****** register.php *****/

```

<!DOCTYPE html>
<html lang="el" lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="icon" type="image/x-icon" href="images/favicon.ico">
        <title>My Greenhouse registration</title>
        <link rel="stylesheet" href="register.css">
        <script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
        </head>

    <body>

        <!-- Registration form -->
        <?php
            function registration_form(){
        ?>

        <!-- Header -->
        <div>
            <header class=header_public>
                <?php include 'header_public.php'; ?>
            </header>
        </div>

        <div class="container">
            <form class="register-form" method="POST" autocomplete="off">

                <!-- Name input -->
                <br>
                <div>
                    <label class="form-label" for="name"></label><br>
                    <input class="form-control" type="text" id="name"
name="name" placeholder="Όνομα" autofocus required/><br>
                </div>

```

```

        <!-- Surname input -->
        <div>
            <label class="form-label" for="surname"></label><br>
            <input class="form-control" type="text" id="surname"
name="surname" placeholder="Επώνυμο" required/><br>
        </div>

        <!-- Email input -->
        <div>
            <label class="form-label" for="email"></label><br>
            <input class="form-control" type="email" id="email"
name="email" placeholder="E-mail" required/><br>
        </div>

        <!-- Password input -->
        <div>
            <label class="form-label" for="password"></label><br>
            <input class="form-control" type="password"
id="password" name="password" placeholder="Κωδικός πρόσβασης" required/><br>
        </div>

        <!-- Password input confirmation -->
        <div>
            <label class="form-label"
for="password_confirm"></label><br>
            <input class="form-control" type="password"
id="password_confirm" name="password_confirm" placeholder="Επιβεβαίωση κωδικού
πρόσβασης" required/><br>
        </div>

        <!-- Submit button -->
        <div>
            <br>
            <button class="submit-control"
type="submit" name="Register" value="Εγγραφή"> Εγγραφή</button>
        </div>

    </form>
</div>

<!-- Footer -->
<div>
    <footer>
        <?php include 'footer.php'; ?>
    </footer>
</div>

```



```

        </div>

<?php
    }
?>

<!-- Eisagwgi neou xristi -->

<?php
    include 'server_connection.php'; //sindesi me tin vasi dedomenwn
    if(isset($_POST['Register'])){
        //prepei na dothei kanoniko email
        if(!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
            ?>
                <script>
                    swal({
                        title: "Σφάλμα!",
                        text: "Παρακαλώ συμπληρώστε σωστά το email
σας!",
                        icon: "error",
                    });
                </script>
                <?php
                    registration_form();
                    exit();
                }
            //elegxos an xrisimopieitai auto to email sti vasi dedomenwn
            $stmt=$pdo->prepare("SELECT Email FROM personal_data WHERE
email=:email");
            $criteria = [
                'email' => $_POST['email'],
            ];
            $stmt->execute($criteria);
            if($stmt->rowCount()>0){
                ?>
                    <script>
                        swal({
                            title: "Σφάλμα!",
                            text: "Αυτό το email χρησιμοποιείται,
δοκιμάστε άλλο!",
                            icon: "error",
                        });
                    </script>
                <?php
                    registration_form();
                    exit();
            }
        }
    }
}
?>

```

```

    }
    else{
        try{
            //edw ginete i kataxwrisi tou neou xristi
            if($_POST['password']===$_POST['password_confirm']){
                $hashed_password =
password_hash($_POST['password'], PASSWORD_DEFAULT);
                $stmt=$pdo->prepare("INSERT INTO personal_data
(name, surname, email, password) VALUES (?, ?, ?, ?)");
                $stmt->execute([\$_POST['name'], \$_POST['surname'],
\$_POST['email'], $hashed_password]);
                ?>
                <div class="container_after_registration">
                    <div class="after_registration">
                        <p><h2>Σας ευχαριστούμε για την
εγγραφή</h2></p><br>
                        <p><h3><a
class="login_after_registration" href="login.php">Συνδεθείτε στο λογαριασμό
σας</a></h3></p>
                    </div>
                </div>
                <?php
            }
            else{
                ?>
                <script>
                    swal({
                        title: "Σφάλμα!",
                        text: "Οι κωδικοί που δώσατε είναι
διαφορετικοί!\nΠαρακαλούμε εισάγετέ τους ξανά!",
                        icon: "error",
                    });
                </script>
                <?php
                    registration_form();
            }
        }
        catch(PDOException $e){
            ?>
            <script>
                swal({
                    title: "Σφάλμα!",
                    text: "Παρουσιάστηκε πρόβλημα,\n
δοκιμάστε ξανά!",
                    icon: "error",
                });
            </script>
            <?php
                registration_form();

```

```

        }
    }
}
else{
    registration_form();
}
?>

</body>

</html>

```

/****** server_connection.php *****/

```

<?php
    //auto to arxeio voithaei stin sindesi me tin vasi dedomenwn
    $server = "localhost";
    $username = "ntamos";
    $password = "ntamos";
    $schema = "thermokipio";
    $pdo = new PDO('mysql:dbname=' . $schema . ';host=' . $server, $username,
    $password, [PDO::ATTR_ERRMODE=>PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC, PDO::MYSQL_ATTR_INIT_COMMAND
    => "SET NAMES utf8"]);
    $pdo->exec("SET CHARACTER SET 'utf8'");
?>

```