

Κατασκευή θερμοκηπίου για την ανάπτυξη σπόρων

Δημήτριος Ζήνας

Διεθνές Πανεπιστήμιο της Ελλάδος

Σέρρες, 10-02-2022

Εργασία που υποβλήθηκε στο

Πρόγραμμα Μεταπτυχιακών Σπουδών στη Ρομποτική,

του Διεθνούς Πανεπιστημίου της Ελλάδος,

για τη μερική εκπλήρωση υποχρεώσεων για το Δίπλωμα Ειδίκευσης στη

Ρομποτική

Επιβλέπων Καθηγητής: Ιωάννης Βουρβουλάκης

Πίνακας περιεχομένων

Πίνακας περιεχομένων	2
1. Περίληψη.....	5
2. Summary	5
3. Ευχαριστίες	6
4. Το πρόβλημα	6
5. Ο όρος θερμοκήπιο.....	7
5.1. Ιστορική εξέλιξη του θερμοκηπίου	8
5.2. Μικροκλίμα Θερμοκηπίου	9
5.3. Η θερμοκρασία.....	9
5.4. Η υγρασία.....	11
5.5. Ο φωτισμός.....	11
6. Κατάσταση της Τέχνης (State of the Art)	13
7. Μεθοδολογία Επίλυσης.....	14
7.1. Εισαγωγή στους ελεγκτές.....	15
7.2. Αλγόριθμος ελέγχου	15
7.3. Αναλογική δράση P.....	16
7.4. Ολοκληρωτική δράση I.....	16
7.5. Διαφορική δράση D.....	16
7.6. P-control για την ρύθμιση της ταχύτητας του ανεμιστήρα	17
8. Υλικά κατασκευής.....	18
8.1. Raspberry pi 4B.....	18
8.1.1. Ακροδέκτες του Raspberry.....	21
8.1.2. Εγκατάσταση VNC και σύνδεση με το Raspberry Pi	22
8.2. Arduino Uno.....	23
8.2.1. Μικροελεγκτής ATmega328	24
8.2.2. Τροφοδοσία	25
8.2.3. Είσοδοι – Έξοδοι.....	25
8.3. Step down converter 5A dc	27
8.4. DC-DC Boost Converter 10-32V to 12-35V 6A Step Up (150W).....	27
8.5. Κάρτα μνήμης για το Raspberry.....	28
8.6. Smart Drive Duo-10 MDDS10.....	28
8.6.1. Δυνατότητες	29

8.6.2.	Σχεδιασμός της πλακέτας	30
8.6.3.	Παροχή ρεύματος	31
8.6.4.	Χαρακτηριστικά ασφαλείας	31
8.7.	Κάμερα USB	34
8.8.	MCP3008 - 10bit 8 channel ADC SPI	35
8.9.	Λάμπα.....	36
8.10.	1602 16X2 Character LCD Display Module with IIC/I2C/TWI/SPI Interface – Blue	36
8.11.	Soil Moisture Sensor Module	37
8.12.	Ηλεκτροβάννα 1’’ ίντζας	38
8.13.	Αισθητήρας φωτεινότητας	40
8.14.	Αντλία 12V.....	40
8.15.	Κινητήρας 12v dc	41
8.16.	Κινητήρας mg996R.....	41
8.17.	Κινητήρας 12V dc	42
8.18.	Παρελκόμενα στήριξης	43
8.19.	Φτερωτή αναρρόφησης	44
8.20.	Δεξαμενή νερού.....	45
8.21.	Ασφάλειες & Ασφαλειοθήκες	45
8.22.	Τροφοδοτικό ρεύματος.....	47
8.23.	Humidity Sensor DHT11 Module	47
8.24.	HC-SR04 αισθητήρας υπερήχων.....	48
8.25.	Τροφοδοτικό ρεύματος.....	49
8.26.	Καλώδια	49
8.27.	Ρελέ	50
9.	Το λογισμικό ελέγχου.....	50
9.2.	Το πρόγραμμα καταγραφής δειγματοληψίας των καταστάσεων του θερμοκηπίου	51
9.3.	Το πρόγραμμα εμφάνισης της κατάστασης του θερμοκηπίου σε lcd οθόνη.....	57
9.4.	Το κύριο πρόγραμμα αλληλεπίδρασης με τον χρήστη.....	61
9.5.	Το πρόγραμμα λήψης φωτογραφιών	83
9.6.	Το πρόγραμμα ενεργοποίησης του μηχανολογικού εξοπλισμού.....	85
10.	Ηλεκτρονικές συνδέσεις.....	89
11.	Κοστολόγηση	91
12.	Αξιολόγηση των αποτελεσμάτων.....	92

13.	Μελλοντική επέκταση.....	107
14.	Βιβλιογραφία.....	108

1. Περίληψη

Σκοπός της παρούσας εργασίας είναι η μελέτη, η σχεδίαση και η κατασκευή καθώς και ο προγραμματισμός μέσω της πλατφόρμας arduino και raspberry των βασικών λειτουργιών ενός θερμοκηπίου και η εφαρμογή προηγμένων μεθόδων ελέγχου των συνθηκών του.

Αρχικά γίνεται εισαγωγή στον όρο θερμοκήπιο καθώς και μια ιστορική αναδρομή στην εξέλιξη του και την αποδοτικότητα του. Έπειτα ακολουθεί η περιγραφή του μικροκλίματος του θερμοκηπίου και τις συνθήκες που επικρατούν στο εσωτερικό του.

Στην συνέχεια γίνεται αναλυτική περιγραφή στα κατασκευαστικά χαρακτηριστικά της πλατφόρμας arduino και raspberry καθώς και του υπόλοιπου υλικού που θα χρησιμοποιηθεί στην κατασκευή του συστήματος.

Ακολούθως πραγματοποιείται ο προγραμματισμός των λειτουργιών με χρήση της πλατφόρμας arduino και raspberry και η παρουσίαση της λειτουργίας του συστήματος.

Τέλος, δίνονται τα τελικά συμπεράσματα και τα σχόλια.

2. Summary

The purpose of this thesis is to study the design and construction as well as the implementation of the basic functions using the arduino and the raspberry pi platforms of a greenhouse and use advanced control methods for closed loop control such as p-control.

Initially, the term greenhouse as well as a historical review of its development and efficiency is introduced. Subsequently the description of the microclimate of the greenhouse and the conditions that prevail inside it follow.

In the next sections a detailed description of the construction features of the arduino and raspberry platform as well as the rest of the components that will be used in the construction of the system is provided.

Then the functions are programmed using the arduino and raspberry platform and the system in operation is presented.

Finally, the final conclusions and comments are discussed.

3. Ευχαριστίες

Θα ήθελα να εκφράσω ευχαριστίες τον επιβλέποντα Καθηγητή κ. Ιωάννη Βουρβουλάκη Επίκουρο Καθηγητή του Τμήματος Μηχανικών Πληροφορικής και Τηλεπικοινωνιών του Διεθνούς Πανεπιστημίου που μου έδωσε την δυνατότητα να εκπονήσω την πτυχιακή μου εργασία στο επιστημονικό τομέα που επιθυμούσα. Επίσης θα ήθελα να τον ευχαριστήσω για την καθοδήγηση και την βοήθεια του οποιαδήποτε στιγμή το χρειαζόμουν.

Επίσης θα ήθελα να ευχαριστήσω τον Καθηγητή κ. Καλόμοιρο Ιωάννη και τον Καθηγητή κ. Βολογιαννίδη Σταύρο για την συμμετοχή τους στην εξεταστική επιτροπή, τον χρόνο που αφιέρωσαν στην διόρθωση της παρούσας πτυχιακής και την βοήθεια που μου παρείχαν κατά τις σπουδές μου.

Τέλος θέλω να ευχαριστήσω τους γονείς και τον αδερφό μου για την αγάπη τους και την συνεχή ηθική και οικονομική συμπαράσταση τους όχι μόνο κατά την διάρκεια εκπόνησης της πτυχιακής μου εργασίας αλλά και κατά την διάρκεια της φοίτησης μου στο ΠΜΣ Ρομποτική.

4. Το πρόβλημα

Η ψηφιακή επανάσταση η οποία άρχισε στα τέλη του 20^{ου} αιώνα έφερε πολύ μεγάλες αλλαγές στους τομείς της βιομηχανίας και των επιστημών και σε όλες τις κοινωνικές και οικονομικές πτυχές των ανθρώπων και των ζώων, μετασχηματίζοντας τον τρόπο που αυτοί σκέφτονται, επικοινωνούν, συμπεριφέρονται και εργάζονται.

Η νέα εποχή της πληροφορίας έχει ως επίκεντρό της τα ψηφιακά συστήματα και τις απορρέουσες από αυτά τεχνολογίες όπως είναι τα κινητά τηλέφωνα, οι ηλεκτρονικοί υπολογιστές, το διαδίκτυο και οι μικροηλεκτρονικές συσκευές. Μετά από χρόνια και την εξέλιξη των τεχνολογιών βελτιώθηκαν οι δυνατότητες των υπολογιστικών συσκευών, μειώθηκε το μέγεθος τους και επιπροσθέτως αυξήθηκε η δυνατότητα συνδεσιμότητας τους, αυξάνοντας έτσι και τις δυνατότητες για εφαρμογές που μέχρι πρόσφατα ήταν ανέφικτες. Φυσική εξέλιξη όλων αυτών είναι η δικτύωση οποιουδήποτε φυσικού αντικειμένου ή ηλεκτρονικής συσκευής με αποτέλεσμα την δυνατότητα παρακολούθησης της κατάστασης καθώς και την δυνατότητα αλληλεπίδρασης με αυτά.

Ο γεωργικός τομέας δεν θα μπορούσε να μείνει ανεπηρέαστος από αυτές τις εξελίξεις. Οι νέες τεχνολογίες έδωσαν την δυνατότητα να αναπτυχθούν να σχεδιαστούν και να εφαρμοστούν νέες και καινούριες μέθοδοι στις αγροτικές καλλιέργειες βοηθώντας έτσι στην περαιτέρω ανάπτυξη του τομέα. Ενδεικτικά μερικές από τις τεχνολογίες πληροφορικής που εφαρμόζονται έως τώρα στην γεωργία είναι: συστήματα αυτοματισμού, τεχνολογίες GPS, συστήματα διαχείρισης των προϊόντων, γεωγραφικά πληροφοριακά συστήματα, ασύρματες τεχνολογίες, συστήματα παρακολούθησης, βάσεις

διαχείρισης γνώσεων κτλ. Αυτές οι τεχνολογίες βοηθούν τους καλλιεργητές να κάνουν καιρικές προβλέψεις που εξαρτώνται από τις παρατηρήσεις του κλίματος όπου βρίσκεται η καλλιέργεια, να τηρήσουν αρχείο το οποίο αφορά τις συγκομιδές και την συμπεριφορά των καλλιεργειών τους, να αναπτύξουν σχέσεις μεταξύ τους αλλά και με την αγορά, γρηγορότερα και αμεσότερα να προβλέψουν πιθανές απειλές για τους καρπούς τους αλλά και να αυτοματοποιήσουν ορισμένες καλλιεργητικές τεχνικές όπου μπορούν.

Έτσι στο πλαίσιο αυτής της εργασίας θα κατασκευαστεί και θα αναλυθεί ένα αυτοματοποιημένο θερμοκήπιο ικανό να καταγράφει και να παρακολουθήσει όλες της φάσεις ανάπτυξης ενός φυτού καθώς η ανάπτυξη των σπόρων εξαρτάται από τις κλιματολογικές συνθήκες καθώς επίσης και από τις συνθήκες του εδάφους και θα μπορούσαν να αποτελέσουν σημαντική βοήθεια για όσους δεν διαθέτουν πρότερη εμπειρία. Στο θερμοκήπιο θα είναι προσαρμοσμένοι αισθητήρες θερμοκρασίας, υγρασίας αέρα και εδάφους, φωτός, κάμερα για σύλληψη φωτογραφιών καθώς και αισθητήρα υπερήχων για την μέτρηση της στάθμης της δεξαμενής. Οι συνθήκες του θερμοκηπίου θα ελέγχονται με χρήση ανεμιστήρων. Ο έλεγχος των ανεμιστήρων θα πραγματοποιείται αυτόματα χρησιμοποιώντας προηγμένες τεχνικές ελέγχου. Η κεντρική συσκευή ελέγχου θα αποτελείται από τον κύριο ελεγκτή που θα περιλαμβάνει ένα raspberry pi, ενώ θα αξιοποιηθεί κατάλληλα ο μικροελεγκτής arduino για λειτουργίες χαμηλότερου επιπέδου. Θα πραγματοποιείται καταγραφή των μετρήσεων σε csv αρχεία και θα μπορεί να γίνει ανάκληση και παρουσίαση γραφημάτων και των αποθηκευμένων φωτογραφιών ανάλογα με τις επιθυμίες του χρήστη. Το πότισμα θα γίνεται όσο χρονικό διάστημα απαιτείται ανάλογα με την επιθυμία του χρήστη.

5. Ο όρος θερμοκήπιο

Θερμοκήπιο είναι ένας χώρος που καλύπτεται με διαπερατό υλικό από την ηλιακή ακτινοβολία που έχει σχεδιαστεί για να μεγαλώνουν τα φυτά στο εσωτερικό ανεξάρτητα από τις κλιματικές συνθήκες καθ'όλη την διάρκεια του έτους. Τα θερμοκήπια έχουν ως στόχο να δημιουργήσουν ευνοϊκό περιβάλλον για την ανάπτυξη των φυτών όπου οι κυριότερες είναι η υγρασία, η θερμοκρασία και η ένταση του φωτός.

Με το θερμοκήπιο αποφεύγονται ζημιές που τις προκαλούν οι απρόσμενες καιρικές συνθήκες. Σπουδαίο ρόλο για το επιθυμητό αποτέλεσμα και την λειτουργικότητα του θερμοκηπίου έχει η επιλογή του τόπου εγκατάστασης, του προσανατολισμού αλλά και της γεωμετρίας.

Οι κλιματολογικές συνθήκες που βοηθούν την εγκατάσταση των θερμοκηπίων σε μια περιοχή είναι η μεγάλη ένταση και διάρκεια του φωτός και οι σχετικά υψηλές τιμές υγρασίας. Η Ελλάδα ευνοείται από την μεγάλη ηλιοφάνεια, το κυριότερο πρόβλημα παραμένει η εξασφάλιση

ικανοποιητικών θερμοκρασιών κατά την διάρκεια της καλοκαιρινής περιόδου λόγω των υψηλών θερμοκρασιών που υφίστανται.

Στην χώρα μας υπάρχουν 40.000 στρέμματα με θερμοκήπια. Σε αυτά καλλιεργούνται κυρίως κηπευτικά και ανθοκομικά φυτά και λαχανικά, ενώ υπάρχουν και θερμοκήπια που χρησιμοποιούνται για ερευνητικούς σκοπούς.

Τα θερμοκήπια χρησιμοποιούνται για εμπορικούς σκοπούς. Ιδιαίτερα στα υγρά κλίματα τα θερμοκήπια είναι χρήσιμα για την ανάπτυξη και τον πολλαπλασιασμό των φυτών επειδή επιτρέπουν στην ηλιακή ακτινοβολία να εισχωρήσει και επίσης δεν επιτρέπουν την θερμότητα να διαφύγει. Η κάλυψη του θερμοκηπίου επιτρέπει στο ορατό φως να εισέλθει με αποτέλεσμα να θερμανθεί το εσωτερικό χωρίς όμως να επιτρέπει την θερμότητα να διαφύγει ανακλώντας την ενέργεια στο εσωτερικό και εμποδίζοντας το εξωτερικό περιβάλλον να την αποβάλει.

Η ορατή ακτινοβολία από τον ήλιο διέρχεται από το γυαλί και απορροφάται από το έδαφος του θερμοκηπίου και από τα φυτά. Ένα μέρος της θερμικής ακτινοβολίας των φυτών και του εδάφους απορροφάται από το γυαλί και ένα μέρος της ακτινοβολίας επανεκπέμπεται στο έδαφος. Έτσι το γυαλί δρα ως ένα κάλυμμα ακτινοβολίας που διατηρεί την επιθυμητή θερμοκρασία με κατάλληλους ηλεκτρομηχανολογικούς μηχανισμούς.

5.1. Ιστορική εξέλιξη του θερμοκηπίου

Οι άνθρωποι από την αρχαιότητα προσπαθούσαν να καλλιεργήσουν φυτά σε κλειστούς χώρους προκειμένου να αυξήσουν τον ρυθμό επιτάχυνσης τους. Το χαρακτηριστικό παράδειγμα σύμφωνα με τον Πλάτωνα είναι οι φημισμένοι << Κήποι του Άδωνη >> κατά τον 5^ο αιώνα π.Χ.

Παρόμοιες προσπάθειες έγιναν και κατά την Ρωμαϊκή περίοδο όπου στόχος ήταν να δημιουργηθεί ένα ελεγχόμενο περιβάλλον για να ευδοκιμήσουν φυτά εκτός εποχής. Έτσι οι κηπουροί του Τιβέριου Καίσαρα κατάφεραν τον χειμώνα να παράγουν καρπούς αγγουριάς προκειμένου να του προσφέρουν στην σαλάτα του.

Στην Ευρώπη κατά τον 17^ο αιώνα εμφανίζονται οι πρώτες μορφές θερμοκηπίων όπου καλλιεργούνταν φυτά για φαρμακευτικούς λόγους αρχικά και μετέπειτα για βοτανικούς και καλλωπιστικούς λόγους από την αριστοκρατία της Βόρειας Ευρώπης. Σημαντικές βελτιώσεις σημειώθηκαν το 18^ο αιώνα όσο αναφορά τον φωτισμό και την κλίση του γυαλιού καθώς είχε αναγνωριστεί η αξία του φωτισμού για την υγιή ανάπτυξη των φυτών. Μεγάλα έργα που κατασκευάστηκαν εκείνη την εποχή ήταν οι γνωστοί Βοτανικοί Κήποι του Gliford στην Αγγλία και του Πανεπιστημίου του Leyden στην Ολλανδία.

Τον 19^ο αιώνα το θερμοκήπιο εξελίχθηκε αρκετά έτσι το 1842 κατασκευάζεται το Βικτωριανό θερμοκήπιο KewGarden της Αγγλίας. Θεαματικές βελτιώσεις στις κατασκευές αλλά και στον ηλεκτρομηχανολογικό εξοπλισμό τους εμφανίζονται κατά τον 20^ο αιώνα.

Σήμερα τα θερμοκήπια είναι κατασκευασμένα από υλικά ελαφριά και ανθεκτικά με τέτοιο τρόπο ώστε να δημιουργείται ιδανικό περιβάλλον για την ανάπτυξη των φυτών. Το υλικό κάλυψης είναι από γυαλί ή από πλαστικό τέτοιας επεξεργασίας έτσι ώστε να διαπερνά η επιθυμητή ακτινοβολία. Ο ηλεκτρομηχανολογικός εξοπλισμός τους ειδικής τεχνολογίας τέτοιας ώστε να δημιουργεί ιδανικό περιβάλλον. Τα θερμοκήπια σήμερα θεωρούνται από τις μεγαλύτερες μονάδες εκμετάλλευσης στον πρωτογενή τομέα της φυτικής παραγωγής. [1]

5.2.Μικροκλίμα Θερμοκηπίου

Το περιβάλλον του θερμοκηπίου αποτελείται από τις συνθήκες της ακτινοβολίας, την θερμοκρασία, την υγρασία και το διοξείδιο του άνθρακα. Αυτές είναι οι συνθήκες που επιδρούν στην ανάπτυξη των φυτών και όταν ρυθμίζονται αποτελούν το μικροκλίμα του θερμοκηπίου.

Η διαφορά μεταξύ μικροκλίματος θερμοκηπίου και του εξωτερικού περιβάλλοντος οφείλεται σε δύο παράγοντες. Ο πρώτος είναι η ηλιακή ακτινοβολία η οποία μικρού μήκους εμποδίζεται από τα διάφανα μέρη του θερμοκηπίου ενώ η ανταλλαγή της μεγάλου μήκους ακτινοβολίας εσωτερικού και εξωτερικού περιβάλλοντος είναι αποτέλεσμα των ιδιοτήτων των υλικών κάλυψης.

Ο δεύτερος είναι ο αέρας του θερμοκηπίου ο οποίος είναι εγκλωβισμένος. Αυτό έχει ως αποτέλεσμα η ταχύτητα του αέρα στο εσωτερικό του θερμοκήπιο να είναι μικρή σε σχέση με αυτή του εξωτερικού αέρα και οι εναλλαγές του αέρα με τον εξωτερικό να είναι μειωμένες.

Το μικροκλίμα επηρεάζεται από τον προσανατολισμό, το γεωμετρικό πλάτος την περιοχή του θερμοκηπίου, τους αυτοματισμούς, τα ηλεκτρομηχανολογικά συστήματα και τις ιδιότητες των υλικών που χρησιμοποιήθηκαν στην κατασκευή του θερμοκηπίου. [2]

5.3.Η θερμοκρασία

Η ρύθμιση της θερμοκρασίας είναι πολύ σημαντική για την ανάπτυξη των φυτών του θερμοκηπίου. Η θερμοκρασία του θερμοκηπίου την νύχτα εξαρτάται από την θερμοκρασία του περιβάλλοντος και το μέγεθος του καλύμματος.

Κατά την διάρκεια της ημέρας είναι σημαντικό να μειωθεί η θερμοκρασία. Αυτό επιτυγχάνεται μέσω του εξαερισμού που εκτός από την μείωση της θερμοκρασίας συμβάλει και στον

έλεγχο της σχετικής υγρασίας του χώρου και στην αναπλήρωση του διοξειδίου του άνθρακα. Αξίζει να σημειωθεί ότι πριν το τέλος της άνοιξης οι θερμοκρασίες ημέρας γίνονται πάρα πολύ υψηλές με αποτέλεσμα να χρειαστεί ακόμα και σκίαση μέχρι τις αρχές του φθινοπώρου.

Σημαντικό ρόλο για τις τιμές της θερμοκρασίας έχει και το είδος του καλλιεργούμενου φυτού.

Τα συστήματα εξαερισμού των θερμοκηπίων επιτυγχάνονται με δύο τρόπους. Με τον δυναμικό εξαερισμό και τον φυσικό εξαερισμό.

Ο φυσικός εξαερισμός κατορθώνεται με την είσοδο του αέρα εντός του θερμοκηπίου μέσω παραθύρων. Όσο μεγαλύτερο είναι το πλάτος του θερμοκηπίου τόσο περισσότερο ανεπαρκής είναι ο αέρας. Στο άνοιγμα των παραθύρων καλό είναι να υπάρχει αυτοματισμός ρυθμισμένος με τέτοιο τρόπο ώστε να συμβαδίζει με τις εποχές και τις αλλαγές της θερμοκρασίας. Μάλιστα η χρήση του αυτοματισμού κάνει ευκολότερη την λειτουργία του φυσικού εξαερισμού συγκριτικά με τα παλαιότερα χρόνια.

Ο δυναμικός εξαερισμός επιτυγχάνεται με την χρήση ανεμιστήρων. Σκοπός των ανεμιστήρων είναι να εξάγουν τον εσωτερικό αέρα. Το πλήθος των ανεμιστήρων και η ένταση που χρησιμοποιείται είναι ανάλογος με τον ρυθμό εξαερισμού. Ο χρόνος εξαερισμού εξαρτάται φυσικά από την επιθυμητή θερμοκρασία, από την ένταση της ηλιακής ακτινοβολίας και από την θερμοκρασία του εξωτερικού αέρα

Όταν η θερμοκρασία είναι χαμηλή όπως την χειμερινή περίοδο πρέπει να βρεθούν τρόποι τέτοιοι ώστε η θερμοκρασία να ανέβει. Η θέρμανση του θερμοκηπίου επιτυγχάνεται μέσω τοπικών και κεντρικών συστημάτων θέρμανσης.

Η τοπική θέρμανση επιτυγχάνεται μέσω ηλεκτρικών αερόθερμων πετρελαίου καθώς και με θερμάστρες. Το μεγαλύτερο πλεονέκτημα της τοπικής θέρμανσης είναι η εξοικονόμηση ενέργειας και οι ελάχιστες απώλειες θερμότητας ενώ το μειονέκτημα της είναι η ανομοιομορφία θέρμανσης στα σκιασμένα μέρη.

Στο σύστημα κεντρικής θέρμανσης η θερμότητα παράγεται στον καυστήρα και μεταφέρεται με νερό ή με ατμό σε ολόκληρο το χώρο του θερμοκηπίου. Το μεγαλύτερο πλεονέκτημα είναι η ομοιόμορφη θέρμανση του θερμοκηπίου ενώ το μειονέκτημα είναι η καθυστέρηση θέρμανσης μετά την έναρξη λειτουργίας του συστήματος. Τα συστήματα αυτά είναι κατάλληλα για μεγάλης έκτασης θερμοκήπια διότι η συντήρηση και η λειτουργία τους είναι φθηνότερη συγκριτικά με ένα μεγάλο αριθμό αερόθερμων.

5.4. Η υγρασία

Η υγρασία είναι ένας από τους κυριότερους παράγοντες στο περιβάλλον ενός θερμοκηπίου και συσχετίζεται απόλυτα με την θερμοκρασία.

Στον χώρο του θερμοκηπίου όταν έχει επέλθει ισορροπία η απόλυτη υγρασία είναι ίδια σε όλο περίπου το χώρο του θερμοκηπίου όμως η θερμοκρασία είναι διαφορετική σε κάποια σημεία του χώρου, συνεπώς η σχετική υγρασία του αέρα δεν έχει ίδια τιμή σε όλο τον χώρο του θερμοκηπίου. Η ανομοιόμορφη κατανομή της θερμοκρασίας σε σχέση με την υγρασία είναι ένα από τα βασικά προβλήματα. Σε περιπτώσεις που η θερμοκρασία έχει χαμηλή τιμή, όπως την νύχτα, η σχετική υγρασία είναι πολύ υψηλή.



Εικόνα 1: Υγρασία θερμοκηπίου

Η σχετική υγρασία επηρεάζει την ικανότητα φωτοσύνθεσης, την παραγωγή και την ανάπτυξη των φυτών. Η μείωση της σχετικής υγρασίας είναι απαραίτητη για να είναι ευνοϊκές οι συνθήκες σε ένα θερμοκήπιο.

Όταν υπάρχει υψηλή υγρασία σε ένα θερμοκήπιο, ένας συνήθης τρόπος για να επιτευχθεί η μείωση της υγρασίας είναι ο εξαερισμός. Ανοίγοντας τα παράθυρα μπαίνει αέρας στο θερμοκήπιο και οι απώλειες θέρμανσης που δημιουργούνται καλύπτονται από το σύστημα θέρμανσης του θερμοκηπίου.[3]

5.5. Ο φωτισμός

Η ηλιακή ακτινοβολία βοηθά τα φυτά να φωτοσυνθέσουν αλλά αποτελεί και πηγή θερμότητας στο χώρο του θερμοκηπίου. Στις ηλιόλουστες μέρες οι θερμοκρασίες στο θερμοκήπιο είναι πολύ υψηλές και την νύχτα η θερμοκρασία μειώνεται. Η θερμοκρασία και η ακτινοβολία είναι άρρηκτα συνδεδεμένοι μεταξύ τους παράγοντες.

Στόχος σε μια καλλιέργεια είναι να δοθεί η απαραίτητη ποσότητα φωτισμού για την βέλτιστη δυνατή απόδοση. Η διάρκεια και η ποιότητα του φωτισμού παίζουν σημαντικό ρόλο ώστε να έχουμε την διαθέσιμη ενέργεια για την φωτοσύνθεση των φυτών. Ο φωτισμός του χώρου θερμοκηπίου επηρεάζεται από την ηλιακή ακτινοβολία αλλά και τις ιδιότητες του καλύμματος.

Οι φωτιστικές ανάγκες του θερμοκηπίου εξαρτώνται από την εποχή και το είδος των καλλιεργούμενων φυτών. Προκειμένου να καλυφθούν αυτές οι ανάγκες γίνεται χρήση τεχνητού φωτισμού αλλά και μεθόδων μείωσης του φωτισμού.

Όταν όλοι οι παράγοντες που επηρεάζουν την φωτοσύνθεση όπως η φωτεινή ακτινοβολία το διοξείδιο του άνθρακα, το νερό βρίσκονται στο ανώτερο επίπεδο τότε η διαδικασία της φωτοσύνθεσης βρίσκεται στο καλύτερο δυνατό σημείο.



Εικόνα 2: Φωτισμός θερμοκηπίου

Ο φυσικός φωτισμός ενός θερμοκηπίου εξαρτάται από τις καιρικές συνθήκες της περιοχής και από τα χαρακτηριστικά της κατασκευής. Πιο συγκεκριμένα ιδιαίτερη σημασία έχουν τα υλικά κάλυψης που μειώνουν τον φωτισμό αλλά και η πυκνότητα των φυτών (πόσο απόσταση θα έχει η μια ρίζα από την επόμενη) έτσι ώστε να μπορούν να καλυφθούν οι ανάγκες όλων των φυτών ως προς την φωτοσυνθετική τους λειτουργία.

Όσο αφορά τον τεχνητό φωτισμό χρησιμοποιείται κυρίως για να καλυφθούν οι ανάγκες επαρκούς φωτισμού κυρίως κατά την διάρκεια της νύχτας. Ο τεχνητός φωτισμός προέρχεται κυρίως από τους λαμπτήρες που μπορεί να είναι είτε λαμπτήρες πυρακτώσεως είτε λαμπτήρες αλογόνου. Η αποτελεσματικότητα ενός λαμπτήρα εξαρτάται κυρίως από δυο παράγοντες:

- από το ποσοστό της ηλεκτρικής ενέργειας που μετατρέπεται σε φωτεινή ακτινοβολία
- από την κατανομή εκπεμπόμενου φάσματος

Σε περίπτωση που θέλουμε να μειώσουμε τον φωτισμό ένας τρόπος σκίασης είναι οι κουρτίνες ή το δίκτυο σκίασης. Αυτές μπορούν να κλείνουν σε περίπτωση που υπάρχει ανάγκη για μείωση του φωτισμού.[4]

6. Κατάσταση της Τέχνης (State of the Art)

Η Π. Πολυκάρπου Γεωπόνος από το Ινστιτούτο Γεωργικών Ερευνών στην δημοσίευση της <<Optimization of Nocturnal Climate Management in PE-greenhouses in Cyprus>> μελέτησε την διατήρηση της εσωτερικής υγρασίας RH του θερμοκηπίου μεταξύ 70% και 85% κατά την διάρκεια της νύχτας την χειμερινή περίοδο έτσι ώστε να αποφευχθούν οι μυκητιάσεις. Η μελέτη πραγματοποιήθηκε σε 18 m² θερμοκήπιο και 3,5 m ύψους σε φυτεία αγγουριών. Ο ηλεκτρομηχανολογικός εξοπλισμός του θερμοκηπίου αποτελούνταν από έναν ανεμιστήρα εξάτμισης, δυο κυκλοφορητές αέρα και ένα αερόθερμο. Έπειτα από τα πειράματα που πραγματοποιήθηκαν παρατηρήθηκε ότι το σύστημα ανταποκρίνεται καλύτερα σε περίπτωση όπου η εξωτερική RH (υγρασία) είναι μικρότερη από 70%. [5]

Ο Α. Rahali κ.α. στην δημοσίευσή τους <<Development of a data acquisition and greenhouse control system based on GSM>> το 2011 δημιούργησαν ένα αυτόματο σύστημα ελέγχου θερμοκηπίου μέσω GSM. Κατέγραφαν την θερμοκρασία, την σχετική υγρασία στο εσωτερικό του θερμοκηπίου και έλεγχαν την θερμοκρασία μέσω των ανεμιστήρων. Επίσης ο χρήστης μπορούσε να ελέγξει την αντλία έτσι ώστε να επιτευχθεί η κατάλληλη υγρασία στα φυτά. Το πρόγραμμα ήταν εγκατεστημένο σε έναν κεντρικό υπολογιστή και η διαχείριση γινόταν μέσω του προγράμματος labview. Ο χρήστης μπορεί να παρέμβει στην λειτουργία του προγράμματος δίνοντας εντολή για την έναρξη και λήξη της άρδευσης καθώς υπάρχει και η δυνατότητα να ενημερωθεί μέσω γραπτού μηνύματος για την κατάσταση λειτουργίας του θερμοκηπίου. [6]

Ο Υ.ΕΙ Αφου, κ.α. στην δημοσίευση τους <<Feedback Techniques Using PID and PI Intelligent For Greenhouse Temperature Control>> το 2014 δημιούργησαν ένα σύστημα το οποίο αποτελούνταν από αισθητήρα θερμοκρασίας, υγρασίας και διοξειδίου του άνθρακα τόσο για την απεικόνιση της κατάστασης του εξωτερικού περιβάλλοντος όσο και του εσωτερικού περιβάλλοντος του θερμοκηπίου. Οι τιμές των μεταβλητών αυτών καταχωρούνταν σε έναν κεντρικό υπολογιστή ο οποίος ήταν υπεύθυνος για την ρύθμιση της ταχύτητας του ανεμιστήρα μέσω ελεγκτή. Οι συγγραφείς της δημοσίευσης αυτής χρησιμοποίησαν δυο ελεγκτές έναν PID και έναν PI. Ο PI ελεγκτής είχε καθυστέρηση στην απόκριση στις μεγάλες διαταραχές για αυτό τον λόγο χρησιμοποιήθηκε ο PID ελεγκτής ο οποίος μπορεί να προσαρμοστεί όταν το φυτό αλλάζει. [7]

Ο Shahad Alyousif κ.α. το Νοέμβριο του 2017 στην δημοσίευση τους <<Intelligent Temperature Control System at Greenhouse>> προσπάθησαν να προσομοιάσουν την λειτουργία ενός συστήματος διατήρησης της θερμοκρασίας στο εσωτερικό του θερμοκηπίου. Το σύστημα αυτό περιείχε έναν μικροελεγκτή, μια lcd οθόνη, έναν αισθητήρα θερμοκρασίας DS18B20, έναν ρυθμιστή τάσης, δυο ενδεικτικές λυχνίες και ανεμιστήρα για την ψύξη του χώρου. Ο μικροελεγκτής καταχωρεί την επιθυμητή θερμοκρασία του χρήστη και προσπαθεί να την συγκρίνει με την υφιστάμενη. Αν η υφιστάμενη θερμοκρασία είναι μικρότερη από την επιθυμητή θερμοκρασία του χρήστη τότε ανοίγει η λάμπα για την θέρμανση του θερμοκηπίου αλλιώς ενεργοποιείται ο ανεμιστήρας για την ψύξη του χώρου.[8]

Ο Karthikeyan, κ.α. στην δημοσίευση τους το 2020 με τίτλο <<IOT Based Moisture Control and Temperature Monitoring in Smart Farming>> δημιούργησαν μια συσκευή η οποία μπορούσε να εμφανίσει σε μια lcd οθόνη δεδομένα όπως η θερμοκρασία και η υγρασία εδάφους έτσι ώστε να αυξηθεί η παραγωγή. Δημιούργησαν ένα ασύρματο δίκτυο χρησιμοποιώντας αισθητήρες θερμοκρασίας και υγρασίας εδάφους έτσι ώστε να προγραμματίζονται οι χρονικές στιγμές ποτίσματος. Ο μικροελεγκτής ήταν ένα arduino. Στις μετρήσεις υπήρχε πρόσβαση μέσα από διάφορες πλακέτες καθώς και διαδικτυακά διαμέσου της πλατφόρμας Thingspeak.com.[9]

Ο Feifei κ.α. στην δημοσίευση τους <<Data Collection of Digital Monitoring System for Agricultural Facilities Environment>> το 2021 δημιούργησαν μια εφαρμογή αλληλεπίδρασης με τον χρήστη έτσι ώστε να υπάρχει απομακρυσμένη διαχείριση του θερμοκηπίου. Οι ενδείξεις που σύλλεγαν ήταν η θερμοκρασία και η υγρασία του αέρα, η υγρασία και η θερμοκρασία του εδάφους και η ένταση του φωτός. Το κύριο πρόγραμμα ήταν φορτωμένο σε ένα κεντρικό υπολογιστή. Η μεταφορά των αρχείων γινόταν μέσω της NRF24L01 πλακέτας η οποία συλλέγει τα δεδομένα και τα μεταδίδει στον server διαμέσου του διαδικτύου. Ο server επεξεργάζεται τα δεδομένα που λαμβάνει και ενεργοποιεί τον pid ελεγκτή ο οποίος ρυθμίζει την ταχύτητα του ανεμιστήρα για την ρύθμιση της θερμοκρασίας στο εσωτερικό του θερμοκηπίου.[10]

7. Μεθοδολογία Επίλυσης

Στο αυτοματοποιημένο θερμοκήπιο θα εγκατασταθούν αισθητήρες θερμοκρασίας, υγρασίας αέρα και εδάφους, φωτός, κάμερα για σύλληψη φωτογραφιών και αισθητήρας υπερήχων για την μέτρηση της στάθμης της δεξαμενής. Οι ενδείξεις των αισθητήρων θα καταγράφονται σε csv αρχεία και οι φωτογραφίες θα αποθηκεύονται στην μνήμη. Θα μπορεί να γίνει ανάκληση και παρουσίαση γραφημάτων και των αποθηκευμένων φωτογραφιών ανάλογα με τις επιθυμίες του χρήστη. Επίσης ανάλογα με την επιθυμία του χρήστη θα καθορίζεται η λειτουργία του ανεμιστήρα και το πότισμα. Ο

χρήστης καθορίζει την επιθυμητή υγρασία και το σύστημα θα ενεργοποιεί την απαιτούμενη διάταξη όσο χρονικό διάστημα χρειάζεται για να ξεπεραστεί η τιμή αυτή. Τέλος για την σταθεροποίηση της θερμοκρασίας στα επιθυμητά επίπεδα η λειτουργία του ανεμιστήρα θα ρυθμίζεται με την χρήση p ελεγκτή.

7.1.Εισαγωγή στους ελεγκτές

Όλα τα ηλεκτρομηχανολογικά συστήματα χρησιμοποιούν ελεγκτές για την ρύθμιση των επιθυμητών συνθηκών λειτουργίας. Έτσι θα γίνει ανάλυση και περιγραφή της θεωρίας συστημάτων αυτοματισμού για έλεγχο αναλογικής δράσης γνωστό ως P-control. Ένας ελεγκτής βρίσκει μεγάλη εφαρμογή λόγω της αξιοπιστίας, της αποτελεσματικότητας και της αποδοτικότητας του. Κύρια πλεονεκτήματα του είναι η απλότητα στην δομή του, η ευκολία στην κατανόηση και η μεγάλη ευελιξία στην εφαρμογή και υλοποίηση.

7.2.Αλγόριθμος ελέγχου

Σκοπός του ελέγχου είναι η ανάλυση των αρχών σχεδιασμού ενός μηχανισμού τέτοιου ώστε να επιτρέπει την αλλαγή και τον έλεγχο της ταχύτητας του ανεμιστήρα εξασφαλίζοντας έτσι την ασφάλεια και αποδοτικότητα του συστήματος για την διατήρηση της θερμοκρασίας στα επιθυμητά επίπεδα. Στόχος του ελεγκτή είναι:

- Η εγγύηση της σταθεροποίησης της θερμοκρασίας στο επιθυμητό επίπεδο που θέτει ο χρήστης.
- Η απόσβεση των διαταραχών και ομαλή επαναφορά.
- Παρακολούθηση δυναμικών αλλαγών της ταχύτητας του ανεμιστήρα και κατ' επέκταση της θερμοκρασίας στο εσωτερικό του θερμοκηπίου.

Η ανάδραση των ελεγκτών βασίζεται στο σφάλμα (απόκλιση) της ελεγχόμενης μεταβλητής εξόδου, από μια επιθυμητή τιμή αναφοράς.

Μετρώντας το σφάλμα ο ελεγκτής δίνει σήμα για αλλαγή της ταχύτητας του ανεμιστήρα σύμφωνα με τον αλγόριθμο ελέγχου ο οποίος αναλύεται σε επόμενες παραγράφους ώστε να μηδενιστεί το σφάλμα και η ελεγχόμενη μεταβλητή που είναι θερμοκρασία στο εσωτερικό του θερμοκηπίου να εξισωθεί με την επιθυμητή θερμοκρασία που θέτει ο χρήστης.[11]

7.3. Αναλογική δράση P

Η εσωτερική θερμοκρασία του θερμοκηπίου αποκλίνει σταθερά από την επιθυμητή θερμοκρασία το οποίο αποτελεί ένα από τα βασικότερα μειονεκτήματα της αναλογικής δράσης. Με την αύξηση της τιμής της μεταβλητής K_p το σφάλμα μεταξύ της επιθυμητής και της υφιστάμενης τιμής της θερμοκρασίας στο εσωτερικό του θερμοκηπίου θα ξεκινήσει να μικραίνει. Το σύστημα ενώ θα έχει πιο γρήγορη απόκριση διακινδυνεύεται η μεγάλη συνεισφορά της μεταβλητής K_p προκαλώντας την εμφάνιση ταλαντώσεων.

Ένα μεγάλο αναλογικό κέρδος προκαλεί μια μεγάλη αλλαγή στην έξοδο για μια δεδομένη αλλαγή του σφάλματος. Αυτό μπορεί να οδηγήσει το σύστημα σε αστάθεια. Από την άλλη ένα μικρό κέρδος έχει σαν αποτέλεσμα μια μικρή απόκριση εξόδου για ένα μεγάλο σφάλμα. Επομένως προκύπτει ένας ελεγκτής που δεν ανταποκρίνεται σωστά στις αλλαγές και είναι λιγότερο ευαίσθητος. Επίσης αν το αναλογικό κέρδος είναι πάρα πολύ χαμηλό η δράση του ελέγχου μπορεί να είναι πάρα πολύ μικρή όταν αυτός προσπαθεί να ανταποκριθεί σε διαταραχές του συστήματος.[11]

7.4. Ολοκληρωτική δράση I

Ο όρος I χρησιμοποιείται στα συστήματα ελεγκτών για την αφαίρεση του σφάλματος μόνιμης κατάστασης του συστήματος ελέγχου και την βελτίωση της απόκρισης της μόνιμης κατάστασης. Γενικά αυξάνει την ταχύτητα απόκρισης του συστήματος όχι μόνο αναλογικά με το σφάλμα αλλά και με τον χρόνο για το οποίο επιμένει το σφάλμα. Ένας ελεγκτής I θα μπορούσε να φέρει σφάλμα στο μηδέν μεταξύ της επιθυμητής θερμοκρασίας σε σχέση με την υφιστάμενη θερμοκρασία αλλά η αντίδραση του θα είναι αργή. Επειδή η επίδραση είναι μικρή χρειάζεται χρόνος. Η δράση του ελεγκτή θα αυξάνεται όσο το σφάλμα είναι θετικό ακόμη και αν το σφάλμα έχει αρχίσει να πλησιάζει στο μηδέν.[11]

7.5. Διαφορική δράση D

Ένας διαφορικός ελεγκτής δρα για να βελτιώσει την παροδική απόκριση του συστήματος. Στόχος του ελεγκτή είναι να μειώσει το σφάλμα, μεταξύ της επιθυμητής θερμοκρασίας του χρήστη και της υφιστάμενης θερμοκρασίας, και την υπέρβαση. Όταν η μείωση του σφάλματος είναι πολύ μικρή θα οδηγήσει σε υπέρβαση. Μετά την υπέρβαση εάν ο ελεγκτής έπρεπε να εφαρμόσει μια μεγάλη διόρθωση στην αντίθετη κατεύθυνση και να ξεπεράσει επανειλημμένα την επιθυμητή

θερμοκρασία του χρήστη η έξοδος θα ταλαντευόταν γύρω από την επιθυμητή θερμοκρασία είτε σε μια σταθερή αναπτυσσόμενη ή αποσυντεθειμένη ημιτονοειδής. Εάν το πλάτος των ταλαντώσεων αυξάνεται με το χρόνο το σύστημα είναι ασταθές. Εάν μειωθούν το σύστημα είναι σταθερό. Εάν οι ταλαντώσεις παραμένουν σε σταθερό μέγεθος το σύστημα είναι οριακά σταθερό.

Οι επιδράσεις του κάθε ελεγκτή φαίνονται στο παρακάτω πίνακα.

Πίνακας 1: Η επίδραση του κάθε ελεγκτή

Response	Χρόνος ανύψωσης (Rise Time)	Υπερύψωση (Overshoots)	Χρόνος αποκατάστασης (Setting Time)	Μόνιμο σφάλμα (Steady- State Error)
k_p	Μειώνεται	Αυξάνεται	Μικρή αλλαγή	Μειώνεται
k_i	Μειώνεται	Αυξάνεται	Αυξάνεται	Εξαλείφεται
k_d	Μικρή αλλαγή	Μειώνεται	Μειώνεται	Μικρή αλλαγή

Παρακάτω φαίνονται οι ορισμοί των πιο πάνω μεγεθών.

Ο χρόνος ανύψωσης είναι ο χρόνος που απαιτείται ώστε να φτάσει η θερμοκρασία από το 10% στο 90% της τελικής κατάστασης.

Υπερύψωση είναι ο ποσοστιαίος λόγος της διαφοράς μεταξύ της πρώτης κορυφής και της μόνιμης τιμής της εξόδου του συστήματος μετά από βηματική διέγερση.

Ο χρόνος αποκατάστασης είναι ο χρόνος που απαιτείται για την βηματική απόκριση να φθάσει στο 2% της τελικής της τιμής.

Μόνιμο σφάλμα είναι η διαφορά της επιθυμητής τιμής της θερμοκρασίας και της τελικής θερμοκρασίας του συστήματος. [11]

7.6.P-control για την ρύθμιση της ταχύτητας του ανεμιστήρα

Στο σύστημα μας θα χρησιμοποιηθεί μόνο η αναλογική δράση του ελεγκτή.

Η συνάρτηση μεταφοράς του P ελεγκτή που χρησιμοποιήθηκε για την ρύθμιση της ταχύτητας του ανεμιστήρα είναι:

$$fanspeed = 520 \frac{k_p E}{s^2 + 1 + k_p E}$$

όπου:

- k_p είναι ένας συντελεστής
- E είναι το σφάλμα δηλαδή η διαφορά μεταξύ της επιθυμητής θερμοκρασίας που δίνει ο χρήστης και της υφιστάμενης θερμοκρασίας στο εσωτερικού του θερμοκηπίου
- S είναι η επιθυμητή θερμοκρασία του χρήστη
- fanspeed είναι η ταχύτητα του ανεμιστήρα σε rpm (min:0 - max: 128)

Η παραπάνω συνάρτηση προέκυψε έπειτα από πειράματα που πραγματοποιήθηκαν και επιλέχθηκε λόγω της καλύτερης απόκρισης στην διατήρηση της θερμοκρασίας στο επιθυμητό επίπεδο που θέτει ο χρήστης. [12]

8. Υλικά κατασκευής

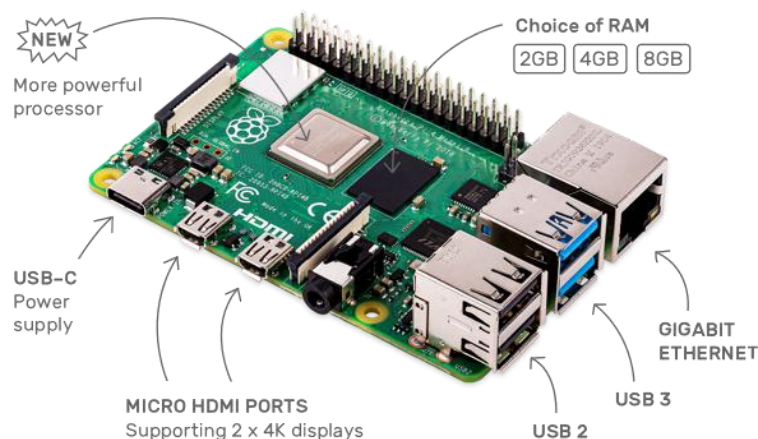
Τα βήματα που ακολουθήθηκαν για την κατασκευή του θερμοκηπίου είναι:

1. Κατασκευή σκελετού διαστάσεως 48x43x45cm ύψους με ηλεκτροσυγκόλληση.
2. Στερέωση των plexiglass πάχους 4mm με βίδες 12x3mm.
3. Κατασκευή βάσης στερέωσης κινητήρα 12000 rpm.
4. Στερέωση των κινητήρων για την μείωση της θερμοκρασίας καθώς και της κάμερας και της λάμπας.
5. Διάνοιξη οπής Φ20 στην δεξαμενή ποτίσματος για την έξοδο του νερού από την δεξαμενή και στερέωση του λάστιχου με σιλικόνη.
6. Διάνοιξη οπών Φ16 και Φ2 στο καπάκι της δεξαμενής για την στερέωση του αποστασιόμετρου που είναι υπεύθυνο για την λήψη του ύψους της στάθμης του νερού στο εσωτερικό της δεξαμενής.
7. Τοποθέτηση της αντλίας στο εσωτερικό της δεξαμενής και διάνοιξη οπής για την έξοδο των καλωδίων για την τροφοδοσία ρεύματος της αντλίας
8. Στερέωση του βραχίονα συγκράτησης του λάστιχου ποτίσματος του θερμοκηπίου.
9. Συνδεσμολογία όλου του ηλεκτρολογικού εξοπλισμού στο εσωτερικό του κουτιού συναρμολόγησης.

Τα υλικά που χρησιμοποιήθηκαν για την υλοποίηση της κατασκευής θα αναλυθούν στις επόμενες παραγράφους.

8.1.Raspberry pi 4B

Το Raspberry pi 4B αποτελεί τον εγκέφαλο του συστήματος καθώς αλληλεπιδρά με τον χρήστη και δίνει όλες τις εντολές για την λειτουργία του θερμοκηπίου.

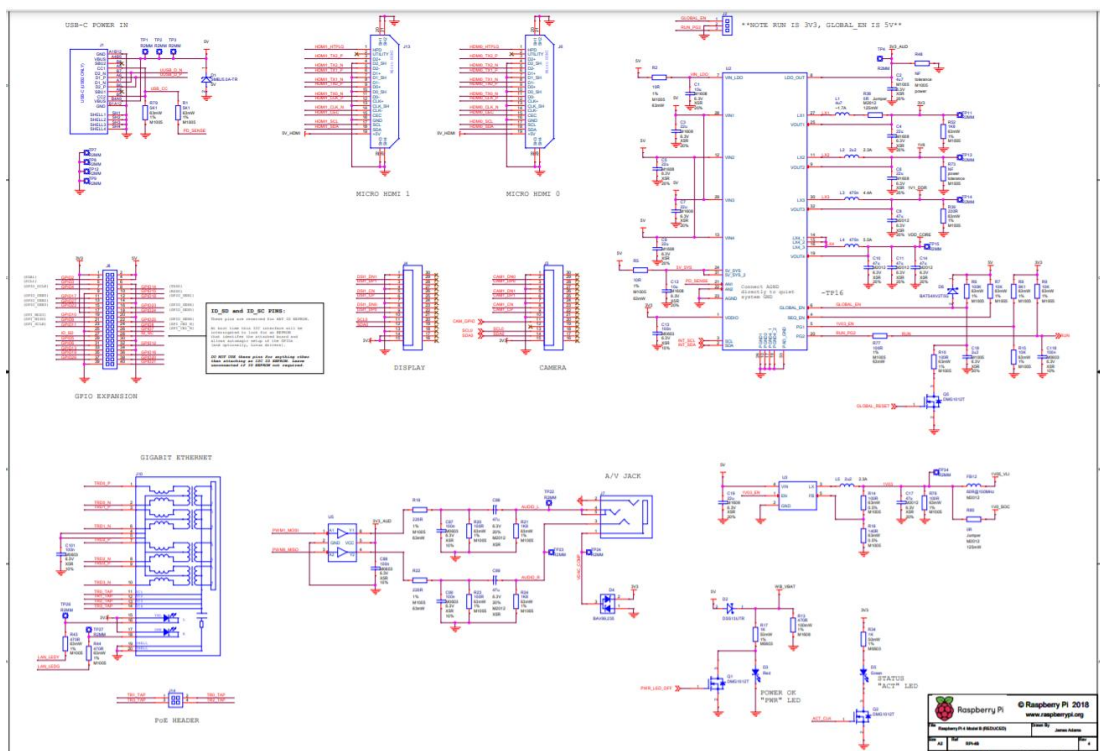


Εικόνα 3: RaspberryPi 4B

Τα τεχνικά χαρακτηριστικά του RaspberryPi 4B φαίνονται στον παρακάτω πίνακα:

Πίνακας 2: Τεχνικά χαρακτηριστικά RaspberryPi 4B [5]

Raspberry pi 4B	
Επεξεργαστής	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Μνήμη	4GB
Συνδεσιμότητα	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE, Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports.
GPIO	40-pin GPIO header
Βίντεο και ήχος	2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port
Πολυμέσα	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics
SD κάρτα	MicroSDcardslot για το λειτουργικό σύστημα και αποθηκευτικό χώρο
Τροφοδοσία	5V DC via USB-C connector (ελάχιστο 3A*) 5V DC via GPIO header (ελάχιστο 3A*)
Περιβάλλον εργασίας	Θερμοκρασία 0-50 °C



Εικόνα 4: Σχηματικό διάγραμμα Raspberry 4B [5]

Το λειτουργικό σύστημα του RaspberryPi βρίσκεται μέσα στην MicroSD κάρτα από την οποία γίνεται η εκκίνηση. Κάποια από τα λειτουργικά συστήματα που μπορεί να επιλέξει ο χρήστης είναι:

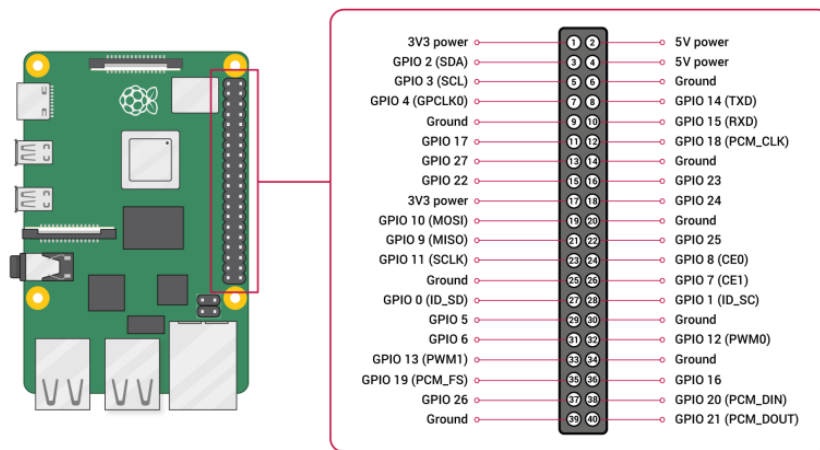
Πίνακας 3: Λειτουργικά συστήματα Raspberry

Λογισμικά
Raspbian
Ubuntu
Arch Linux
Pidora
Linutop
SARPi

Στην προκειμένη εργασία έχει επιλεγθεί το λειτουργικό σύστημα Raspbian. Είναι ένα δωρεάν λειτουργικό σύστημα. Οι παροχές που διαθέτει είναι πολλές καθώς συμπεριλαμβάνονται εργαλεία ανάπτυξης λογισμικού, εκμάθησης προγραμματισμού κ.α. Η προκαθορισμένη συμπεριφορά του είναι να κάνει boot κατευθείαν στο γραφικό περιβάλλον Desktop. Το περιβάλλον είναι φιλικό προς τον χρήστη και περιέχει δυνατότητα ενεργοποίησης / απενεργοποίησης διεπαφών (interfaces), αλλαγή ζώνης ώρας και γλώσσας πληκτρολογίου.

8.1.1. Ακροδέκτες του Raspberry

Ένα από τα πιο σημαντικά χαρακτηριστικά του RaspberryPi είναι οι ακίδες GPIO. Ο συνδετήρας του RPI αποτελείται από 40 pins. Τα 26 είναι χρήσης εισόδου/εξόδου ενώ τα υπόλοιπα έχουν την ιδιότητα τροφοδοσίας γείωσης. Μπορούν να καταναλώσουν μέχρι 50mA ρεύματος με ασφάλεια ενώ κάθε ακίδα ξεχωριστά μπορεί μόνο 16mA. Τα pins φαίνονται αναλυτικά στην παρακάτω εικόνα.



Εικόνα 5: Pinout του Raspberry

Πιο συγκεκριμένα κάθετα από τον 1^ο ακροδέκτη υπάρχουν οι μονοί αριθμοί ακροδεκτών ενώ κάθετα από τον 2ο ακροδέκτη υπάρχουν αντίστοιχα οι ζυγοί αριθμοί ακροδεκτών. Συνολικά υπάρχουν 9 διαφορετικά είδη.

- 4 ακροδέκτες για τάση
- 8 ακροδέκτες για γείωση
- 12 ακροδέκτες για γενική χρήση
- 5 ακροδέκτες SPI
- 4 ακροδέκτες PWM
- 2 ακροδέκτες I2C
- 2 ακροδέκτες UART
- 1 ακροδέκτης CLK

Αναλυτικότερα:

- I2C: Χρησιμοποιείται για την επικοινωνία μεταξύ πολλών απλών συσκευών και αισθητήρων μέσω μόνο δύο καλωδίων.
- UART: Μετάδοση δεδομένων σειριακά

- SPI: Χρησιμοποιείται για την ανάγνωση πιο περίπλοκων αισθητήρων, οθονών υγρών κρυστάλλων ή και για επικοινωνία μεταξύ συσκευών. Η επικοινωνία μεταξύ συσκευών πραγματοποιείται ανάμεσα σε μια συσκευή host και μια η περισσότερες συσκευές client
- PWM: Κύκλωμα που παράγει κυματομορφές διαμορφωμένες κατά εύρος και μπορεί να χρησιμοποιηθεί στον έλεγχο ταχύτητας περιστροφής κινητήρων.
- CLK: Χρονικά σήματα που χρησιμοποιούνται για να παρέχουν παλμούς με μέγιστη συχνότητα περίπου 75MHz.

Τα χαρακτηριστικά που ανήκει ο κάθε ακροδέκτης είναι:

- Οι ακροδέκτες Pin1, 17 δίνουν τάση 3,3V
- Οι ακροδέκτες Pin2, 4 δίνουν τάση 5V
- Οι ακροδέκτες Pin6,9, 14, 20, 25, 30, 34, 39 δίνουν τάση 0V
- Οι ακροδέκτες Pin11, 13, 15, 16, 18, 22, 29, 31, 36, 37, 38, 40. Αποτελούν ακροδέκτες γενικής χρήσης
- Οι ακροδέκτες Pin 19, 21 ,23,24,26 είναι για SPI επικοινωνία
- Οι ακροδέκτες Pin 12, 32, 33, 35 ανήκουν στην κατηγορία PWM
- Οι ακροδέκτες Pin 3, 5 ανήκουν στην κατηγορία I2C
- Οι ακροδέκτες Pin 8, 10 ανήκουν στην κατηγορία UART
- Ο ακροδέκτης Pin 7 ανήκει στην κατηγορία CLK. [13][14]

8.1.2. Εγκατάσταση VNC και σύνδεση με το Raspberry Pi

Το VNC Connect αποτελείται από το VNC Viewer. Το VNC Viewer επιτρέπει την σύνδεση και τον έλεγχο ενός υπολογιστή ή ενός άλλου Raspberry. Παρακάτω παρουσιάζονται τα βήματα που χρειάζονται να ακολουθήσει ο χρήστης ούτως ώστε να υπάρχει επικοινωνία μεταξύ υπολογιστή και Raspberry.

Ο χρήστης εγκαθιστά το λογισμικό του VNC Viewer στον υπολογιστή που θέλει να ελέγχει το RaspberryPi.

Στο τερματικό του RaspberryPi εκτελεί τις ακόλουθες εντολές για να βεβαιωθεί πως έχει την νεότερη έκδοση του λογισμικού.

```
$sudo apt-get update
```

```
$sudo apt-get install realvnc-vnc-viewer
```

Πλοηγείται στο Menu>Preferences>Raspberry Pi Configuration> Interfaces και επιλέγεται VNC να είναι Enabled.

Ο χρήστης ανοίγει το τερματικό του RaspberryPi και εκτελεί την εντολή <<hostname -I>> για να μάθει την IP διεύθυνση του δεδομένου ότι βρίσκεται στο ίδιο δίκτυο. Στην συνέχεια ανοίγει το

VNCViewer που εγκατέστησε προηγουμένως στον υπολογιστή του και γράφει την διεύθυνση IP του Raspberry.

Στην συνέχεια θα ερωτηθεί για τα στοιχεία εισόδου username και password τα οποία είναι pi και raspberry αντίστοιχα. Τέλος καλό θα ήταν να γίνει αλλαγή των στοιχείων αυτών για περαιτέρω προστασία και ασφάλεια της ιδιωτικότητας του χρήστη.

8.2.Arduino Uno

Πρόκειται για μια πλατφόρμα ανάπτυξης ανοικτού hardware και software με ενσωματωμένο μικροελεγκτή και εισόδους / εξόδους. Προγραμματίζεται με μια απλοποιημένη μορφή της γλώσσας προγραμματισμού C++ και ένα σύνολο βιβλιοθηκών βασισμένων στην C++. Η χρήση της συγκεκριμένης γλώσσας προσφέρει ευκολία και ταχύτητα ανάπτυξης εφαρμογών. Τα βασικότερα πλεονεκτήματα της πλατφόρμας σε σχέση με άλλες πλατφόρμες είναι:

- **Το χαμηλό κόστος:** Οι πλακέτες arduino δεν είναι ακριβές σε σύγκριση με άλλες πλατφόρμες μικροελεγκτών. Μπορούμε επίσης να συναρμολογήσουμε μια πλακέτα arduino αφού το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα μειώνοντας έτσι περισσότερο το κόστος.
- **Το απλό περιβάλλον προγραμματισμού:** Το λογισμικό arduino (IDE) λειτουργεί σε λειτουργικά συστήματα Mac , Linux και Windows. Το περιβάλλον προγραμματισμού του είναι ιδιαίτερα φιλικό και εύκολο στη χρήση για αρχάριους
- **Η πλατφόρμα είναι επεκτάσιμη:** Ο κάθε άνθρωπος μπορεί να βρει τον πηγαίο κώδικα στο ίντερνετ και να τον προσαρμόσει ανάλογα με τις επιθυμίες του, να δημιουργήσει δικές του βιβλιοθήκες ακόμα και να επεξεργαστεί το περιβάλλον ανάπτυξης.



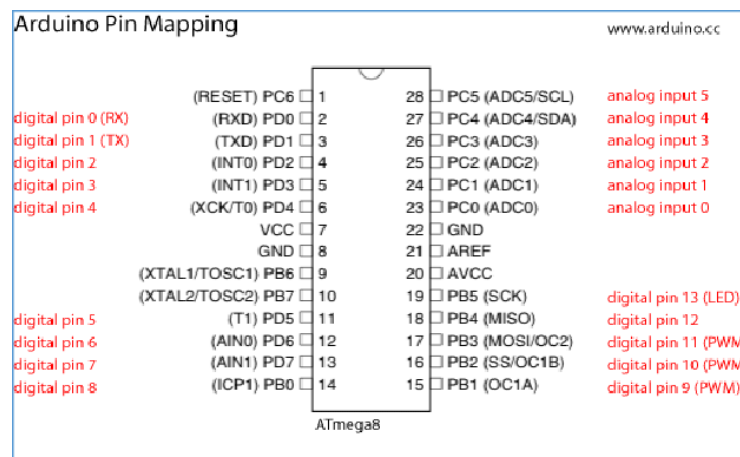
Εικόνα 6: ArduinoUno

Τα τεχνικά χαρακτηριστικά της πλατφόρμας ArduinoUno είναι:

- Μικροελεγκτής: ATmega328
- Τάση λειτουργίας: 5V
- Τάση εισόδου: 7-12V
- Τάση εισόδου(όριο): 6-20V
- Digital output pins I/O: 14, 6 PWM
- Analog input pins: 6
- DC ρεύμα για κάθε I/O pin: 20mA .[15]

8.2.1. Μικροελεγκτής ATmega328

Ο μικροελεγκτής που περιέχεται στην πλακέτα ArduinoUNO είναι ο ATmega328. Στο παρακάτω σχήμα φαίνονται που τοποθετούνται οι είσοδοι και οι έξοδοι καθώς και η τάση τροφοδοσίας και η γείωση.



Εικόνα 7: Το διάγραμμα ακροδεκτών του ATmega328P

Ο Atmega328 είναι ένας 8-bit μικροελεγκτής ο οποίος χρονίζεται έως τα 16MHz ενώ διαθέτει ενσωματωμένη μνήμη τριών τύπων.

- 2kb μνήμης SRAM που είναι μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματα για να αποθηκεύσουν δεδομένα κατά την λειτουργία. Αν η παροχή του ρεύματος στο Arduino διακοπεί ή αν γίνει επανεκκίνηση η μνήμη χάνει τα δεδομένα.
- 1kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για εγγραφή / ανάγνωση δεδομένων ανά byte από τα προγράμματα κατά την εκτέλεση. Σε αντίθεση με την προηγούμενη μνήμη που αναφέρθηκε η μνήμη αυτή δεν χάνει τα περιεχόμενα της με απώλεια τροφοδοσίας ή reset
- 32Kb μνήμης Flash από τα οποία τα 2Kb χρησιμοποιούνται για τον bootloader. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση των προγραμμάτων αφού

πρώτα μεταγλωττιστούν στον υπολογιστή. Η μνήμη Flash όπως και η EEPROM δεν χάνει τα περιεχόμενα της με την απώλεια ρεύματος ή reset. [15]

8.2.2. Τροφοδοσία

Το arduino μπορεί να τροφοδοτηθεί με ρεύμα από τον υπολογιστή μέσω της σύνδεσης USB ή από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φισ 2.1mm. Για να μην υπάρχουν προβλήματα η εξωτερική τροφοδοσία πρέπει να είναι από 7 έως 12V και μπορεί να προέρχεται από ένα μετασχηματιστή εναλλασσόμενου ρεύματος σε συνεχές , από μπαταρίες ή οποιαδήποτε άλλη πηγή DC.

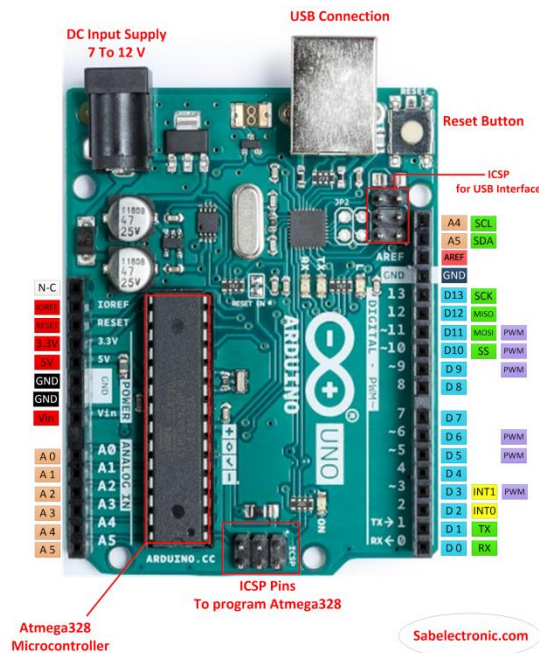
Η λειτουργία του κάθε ακροδέκτη της πλακέτας Arduino είναι η εξής:

- Το pin με την ένδειξη RESET όταν πατηθεί έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- Το pin με την ένδειξη 3.3V μπορεί να τροφοδοτήσει τα εξαρτήματα με τάση 3.3V.
- Τα pin με την ένδειξη GND είναι οι γειώσεις.
- Το pin με την ένδειξη Vin σε συνδυασμό με το pin της γείωσης δίπλα του μπορεί να χρησιμοποιηθεί για τροφοδοσία του Arduino από εξωτερική πηγή στην περίπτωση που δεν είναι δυνατόν να χρησιμοποιηθεί η υποδοχή του φισ 2.1 mm. [15]

8.2.3. Είσοδοι – Έξοδοι

Ο μικροελεγκτής ATmega υποστηρίζει σειριακή επικοινωνία με την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των προγραμμάτων από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή. Επιπλέον στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin αριθμημένα από το 0 έως το 13 που μπορούν να λειτουργήσουν ως ψηφιακές είσοδοι και έξοδοι.

Ως ψηφιακή έξοδος ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμα σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Επίσης αν ένα από αυτά τα pin καθοριστεί ως ψηφιακή είσοδος το πρόγραμμα μπορεί με την κατάλληλη εντολή να διαβάσει την κατάσταση του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχει συνδεθεί σε αυτό το pin καθορίζει χαμηλό ή υψηλό δυναμικό



Εικόνα 8: ArduinoUno pinout

Μερικά από αυτά τα 14 pin εκτός από ψηφιακές εισοδοί / έξοδοι έχουν και δεύτερη λειτουργία.

Συγκεκριμένα:

- Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμα ενεργοποιεί την σειριακή θύρα. Έτσι όταν το πρόγραμμα στέλνει δεδομένα στην σειριακή αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να διαβαστούν ενδεχομένως από μια άλλη συσκευή. Αυτό σημαίνει ότι αν στο πρόγραμμα ενεργοποιηθεί το σειριακό interface χάνονται 2 ψηφιακές εισοδοί / έξοδοι. Επίσης υπάρχουν δυο LED με τις σημάνσεις TX και RX τα οποία ανάβουν όταν το Arduino στέλνει ή λαμβάνει δεδομένα μέσω USB.
- Τα pin 2 και 3 ρυθμίζονται μέσα από το πρόγραμμα με τέτοιο τρόπο ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοί
- Οι ακροδέκτες 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM. Έτσι μπορεί να συνδεθεί ένα LED σε κάποιο από αυτά τα pin και να ελεγχθεί πλήρως η φωτεινότητα του με ανάλυση 8bit αντί να υπάρχει απλά η δυνατότητα αναμμένο-σβηστό που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι. Αυτό πραγματοποιείται με χρήση της συνάρτησης analogWrite().
- 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) Οι ακροδέκτες αυτοί παρέχουν σειριακή επικοινωνία με χρήση της βιβλιοθήκης SPI.

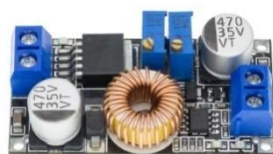
Υπάρχει μια ακόμη σειρά από 6 pin αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC του μικροελεγκτή. Η τάση αναφοράς είναι 5V. Είναι εφικτό επίσης να χρησιμοποιηθεί ένας από τους ακροδέκτες ως εξωτερική τάση αναφοράς [16]

8.3.Step down converter 5A dc

Χρησιμοποιήθηκε ο παρακάτω μετατροπές τάσης από 12V dc σε 5V dc για την τροφοδοσία ρεύματος των ρελέ. Τα τεχνικά χαρακτηριστικά του φαίνονται στον παρακάτω πίνακα. [17]

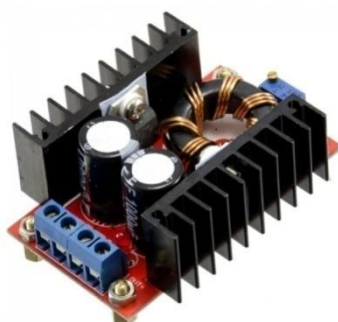
Πίνακας 4: Τεχνικά χαρακτηριστικά

Διάσταση	5.2x2.65x1.4cm
Τάση εισόδου	8-36V
Τάση εξόδου	1.25V-32V
Μέγιστο ρεύμα εξόδου	5A (ρυθμιζόμενο)
Συχνότητα	180KHz
Θερμοκρασία λειτουργίας	-40 °C έως +85 °C



Εικόνα 9: Step down converter

8.4.DC-DC Boost Converter 10-32V to 12-35V 6A Step Up (150W)



Εικόνα 10: Step up converter

Χρησιμοποιήθηκε ο μετατροπές τάσης από 12V σε 24Vdc διότι η τάση λειτουργίας της ηλεκτροβάνας είναι στα 24V. Τα χαρακτηριστικά του φαίνονται στον παρακάτω πίνακα. [18]

Πίνακας 5: Χαρακτηριστικά Step up converter

Τάση εισόδου	DC 10-32V
Τάση εξόδου	DC 12V-35V
Μέγιστο Ρεύμα εξόδου	6A
Μέγιστο ρεύμα εισόδου	16A
Ισχύς εξόδου	100W
Θερμοκρασία λειτουργίας	- 40 °C έως +85 ° C
Ρεύμα χωρίς φορτίο	25mA

8.5.Κάρτα μνήμης για το Raspberry



Εικόνα 11: Micro Sd with adapter

Χρησιμοποιήθηκε κάρτα μνήμης τύπου microSD χωρητικότητας 32gb για το Raspberry. Στην κάρτα αυτή εγκαταστάθηκε το λειτουργικό σύστημα και λειτουργεί ως σκληρός δίσκος. Επιλέγουμε αυτή την χωρητικότητα για να υπάρχει αρκετός ελεύθερο χώρος στο σύστημα. [19]

8.6.Smart Drive Duo-10 MDDS10

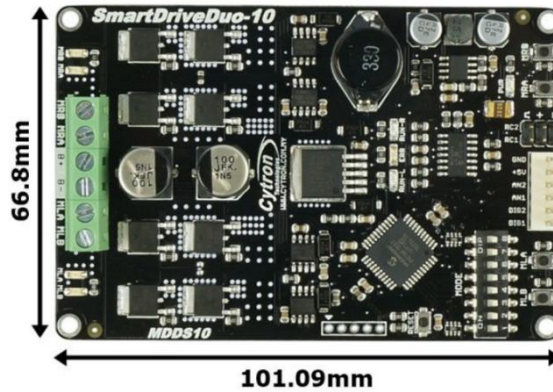
Η SmartDriveDuo-10 είναι μια πλακέτα η οποία σχεδιάστηκε για οδήγηση DC κινητήρων με ρεύμα λειτουργίας 30A για λίγα δευτερόλεπτα και 10A συνεχόμενης λειτουργίας. Μπορεί να ελεγχθεί με χρήση μικροελεγκτή. Τα MOSFETS ενεργοποιούνται στα 16 KHz για να διασφαλίσουν την αθόρυβη λειτουργία. Είναι εξοπλισμένη με microcontroller ώστε να έχει τη δυνατότητα να παρέχει ορισμένα “έξυπνα” χαρακτηριστικά.

Ορισμένες από τις λειτουργίες της φαίνονται παρακάτω:

- Ελεγχόμενη εναλλαγή κατεύθυνσης του dc κινητήρα.
- Υποστηριζόμενη τάση λειτουργίας του κινητήρα 7V έως 35V
- Μέγιστο ρεύμα λειτουργίας έως 30 A για λίγα δευτερόλεπτα και 10A συνεχόμενα
- Συχνότητα λειτουργίας 16KHz για αθόρυβη λειτουργία
- Ένδειξη χαμηλής στάθμης τροφοδοσίας
- Ένδειξη υψηλής στάθμης τροφοδοσίας
- Προστασίας από υπερθέρμανση
- Διάφορες καταστάσεις λειτουργίας: RC, Analog, PWM, Serial

- Κουμπί για γρήγορο έλεγχο και χειροκίνητη λειτουργία.
- Δεν παρέχει προστασία σύνδεσης πολικότητας της μπαταρίας [20]

8.6.1. Δυνατότητες



Εικόνα 12: Διαστάσεις SmartDriveDuo-10

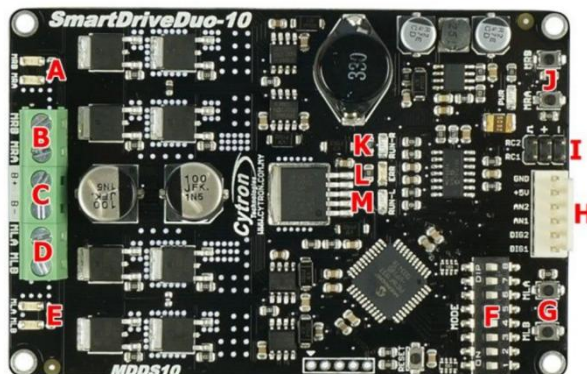
Πίνακας 6: Χαρακτηριστικά λειτουργίας

No	Παράμετροι	Min	Τυπική	Μέγιστη	Μονάδες
1	Τάση λειτουργίας κινητήρα	7	-	35	V
2	Μέγιστη συνεχόμενη τάση λειτουργίας κινητήρα*	-	-	10	A
3	Μέγιστο ρεύμα λειτουργίας**	-	-	30	A
4	Λογική είσοδος για HighLevel	1.3	-	5	V
5	Λογική είσοδο για LowLevel	0	-	0.7	V
6	5V τάση λειτουργίας	-	-	500	mA

*Εξαρτάται από την θερμοκρασία του χώρου

**Δεν πρέπει να υπερβαίνει το 1 second [20]

8.6.2. Σχεδιασμός της πλακέτας



Εικόνα 13: Λειτουργίες

Πίνακας 7: Ενδείξεις - Λειτουργίες

Ένδειξη	Λειτουργία
A	Ένδειξη led δεξιού κινητήρα
B	Σημείο σύνδεσης δεξιού κινητήρα
C	Τροφοδοσία
D	Σημείο σύνδεσης αριστερού κινητήρα
F	Ένδειξη led αριστερού κινητήρα
G	DIP Switch επιλογής λειτουργίας
H	Αναλογικό/ PWM / Σειριακό pin εισόδου
I	RC Pin
J	Κουμπί χειροκίνητου ελέγχου δεξιού κινητήρα
K	Ένδειξη λειτουργίας δεξιού κινητήρα
L	Ένδειξη σφάλματος
M	Ένδειξη λειτουργίας αριστερού κινητήρα

Πίνακας 8: Ονομασία - Περιγραφή PIN

No	Ονομασία Pin	Περιγραφή
1	DIG1	Ψηφιακό σήμα διεύθυνσης για αριστερό κινητήρα
2	DIG2	Ψηφιακό σήμα διεύθυνσης για δεξί κινητήρα
3	AN1	Αναλογικό / PWM σήμα για αριστερό κινητήρα
4	AN2	Αναλογικό / PWM σήμα για δεξί κινητήρα
5	+5V	+5V έξοδο. Να μην συνδεθεί σε άλλη 5V πηγή
6	GND	Γείωση

*DiG1 μπορεί να δεχθεί 2 τύπους εισόδων

Πίνακας 9: Σφάλματα - Περιγραφή

Πλήθος από αναμοσβήσιμα	Σφάλμα	Περιγραφή
Σβηστό	Κανένα σφάλμα	Κανένα σφάλμα δεν έχει εμφανιστεί
2	Σφάλμα εισαγωγής	Λάθος ή καμία είσοδο δεν εμφανίστηκε
3	Χαμηλή τάση	Η τάση της μπαταρίας είναι χαμηλή

4	Υψηλή τάση	Η παρεχόμενη τάση είναι πάνω από το όριο. MDDS10 δεν λειτουργεί
5	Υψηλή θερμοκρασία	Η θερμοκρασία της πλακέτας είναι υψηλή
6	Τερματισμός λειτουργίας	Η θερμοκρασία της πλακέτας είναι πολύ υψηλή! Ο κινητήρας σταματάει αυτόματα

[20]

8.6.3. Παροχή ρεύματος

Η SmartDriveDuo-10 υποστηρίζει τάση εισαγωγής από 7V έως 35V. Οι συνιστώμενες πηγές τάσης είναι:

- Μπαταρία με NiMH ή NiCd 6-18 κελιά
- Μπαταρία LiPo ή Li-Ion 2-6 κελιά
- 7V -35V μπαταρία μολύβδου
- 7V-35V παροχή τάσης (Πρέπει να είναι παράλληλα συνδεδεμένη με μπαταρία ίδιας τάσης

Η τάση μπορεί να συνδεθεί στο SmartDriveDuo-10 είτε από το μπλοκ σύνδεσης ή μπορεί να κολληθεί από το κάτω μέρος. Δεν υπάρχει προστασία πολικότητας κατά την σύνδεση. [20]

8.6.4. Χαρακτηριστικά ασφαλείας

Σφάλμα εισαγωγής (Η ένδειξη LED αναβοσβήνει 2 φορές)

Κάθε φορά που η SmartDriveDuo-10 δέχεται ρεύμα τα δεδομένα που εισάγονται πρέπει να είναι σταματημένα. Αυτό το χαρακτηριστικό παρεμποδίζει τον κινητήρα από ξαφνική κίνηση, ειδικά όταν η πλακέτα κάνει επανεκκίνηση.

Ένδειξη χαμηλής τάσης (Η ένδειξη LED αναβοσβήνει 3 φορές)

Όταν δέχεται ρεύμα η SmartDriveDuo-10 μπορεί αυτόματα να αναγνωρίσει το πλήθος των κελιών της μπαταρίας. Εάν η μπαταρία είναι LiPo αν η εισαγόμενη τάση πέσει κάτω από 3V ανά κελί κατά την διάρκεια της λειτουργίας η ένδειξη LED θα αναβοσβήνει για να προειδοποιήσει τον χρήστη. Παρόλα αυτά η τάση στον κινητήρα θα διατηρηθεί και δεν θα διακοπεί. Έτσι ο χρήστης μπορεί να αποφασίσει αν θα αντικαταστήσει την μπαταρία και να αποφύγει περισσότερη φθορά της μπαταρίας. Εάν χρησιμοποιούνται άλλα είδη μπαταριών η ένδειξη χαμηλής τάσης δεν θα υπάρχει. Σε αυτή την περίπτωση ο χρήστης μπορεί να αγνοήσει το σφάλμα και να υπολογίσει το χρονικό σημείο που η μπαταρία θα χρειαστεί αλλαγή.

Ένδειξη υψηλής τάσης (Η ένδειξη LED αναβοσβήνει 4 φορές)

Εάν η τάση είναι μεγαλύτερη από 35V , η SmartDriveDuo-10 δεν θα λειτουργεί

Ένδειξη υπερθέρμανσης (Η ένδειξη LED αναβοσβήνει 5 φορές)

SmartDriveDuo-10 είναι εξοπλισμένη με αισθητήρα θερμοκρασίας για να γνωρίζει της θερμοκρασία λειτουργίας. Σταδιακά θα μειώνεται ο κύκλος ζωής ανάλογα με την θερμοκρασία όπως φαίνεται στον παρακάτω πίνακα.

Πίνακας 10: Θερμοκρασία - Κύκλος

Διάστημα θερμοκρασίας	Κύκλος ζωής (%)
<50	100
50-55	99
55-60	80
60-65	70
65-70	60
70-80	50
80-85	40
85-90	30
90-95	20
95-100	10
100-105	5
>105	0(Τερματισμός)

Τερματισμός (Η ένδειξη LED αναβοσβήνει 6 φορές)

Ο τερματισμός μπορεί να εμφανιστεί όταν η θερμοκρασία της SmartDriveDuo-10 είναι πολύ υψηλή (temperature > 105 °C) Ως αποτέλεσμα ο κινητήρας μπορεί να σταματήσει. Ο κινητήρας του κάθε καναλιού μπορεί να εκκινηθεί όταν η θερμοκρασία πέσει κάτω από τους 70 °C.

Κατάσταση εισαγωγής

Όταν η SmartDriveDuo-10 τροφοδοτηθεί με ρεύμα ή ένδειξη RUN-L, ένδειξη σφάλματος και RUN-R ένδειξη θα αναβοσβήσουν μια φορά. Έπειτα από αυτό η κατάσταση εισαγωγής μπορεί να διασφαλιστεί από το DIP switch και να παραμείνει όσο η πλακέτα έχει ρεύμα. Εάν χρειαστεί να αλλαχθεί η κατάσταση λειτουργίας θα πρέπει να γίνει αλλαγή στην DIPswitch και να πατηθεί το κουμπί επανεκκίνησης.

Αναλογική / PWM κατάσταση λειτουργίας

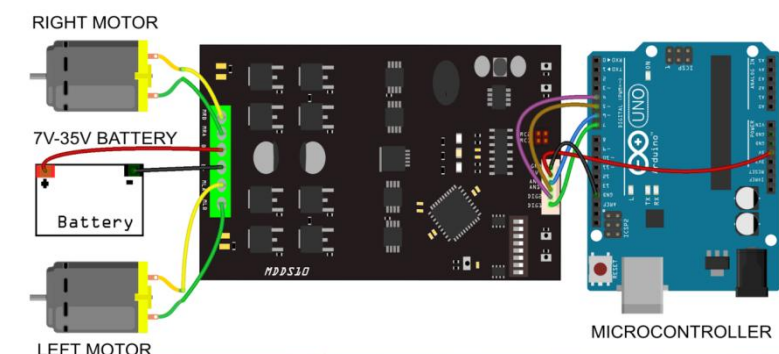
Στην αναλογική / PWM κατάσταση λειτουργίας η ταχύτητα και η φορά κίνησης του κινητήρα ελέγχεται από την αναλογική τάση και το PWM σήμα. Η κανονική τάση εισόδου είναι από 0 έως 5V.

Αναλογική είσοδος επιλέγεται όταν το SW1 είναι 0 (κάτω) και SW2 στο 1 (πάνω). PWM είσοδος επιλέγεται όταν SW1 στο 1 (πάνω) και SW2 στο 0 (κάτω). SW3-SW6 μπορούν να διαφοροποιηθούν ανάλογα με τις απαιτήσεις του χρήστη.

Πίνακας 11: Ρυθμίσεις της DIPswitch για PWM λειτουργία

Είσοδος	SW1-SW2 01	Αναλογική Είσοδος. Το εισαγόμενο σήμα είναι αναλογική είσοδος. Ο πυκνωτής δεν είναι συνδεδεμένος και το σήμα εισαγωγής δεν φιλτράρεται. SW1 είναι για την είσοδο 1 ενώ το SW2 είναι για την είσοδο 2.
	10	PWM είσοδος. Η είσοδος είναι PWM σήμα. Ο πυκνωτής είναι συνδεδεμένος και το εισαγόμενο σήμα είναι χαμηλοπερατό σήμα και αναλογικό. SW1 είναι για την είσοδο 1 ενώ SW2 είναι για την είσοδο 2. SW2 πρέπει να είναι 0 αν κατά την έναρξη
Απλή / Μικτή λειτουργία	SW3-SW4 00	Μικτή. Η ταχύτητα του κινητήρα ελέγχεται από το κανάλι RC1 και η κατεύθυνση ελέγχεται από το RC2. Αυτή η λειτουργία ενδείκνυται εάν χρησιμοποιείται Joystick
	01	Ανεξάρτητη Δεξιά. Ενεργοποιείται μόνο ο δεξιός κινητήρας. Ο αριστερός κινητήρας παραμένει ανενεργός. Ο δεξιός κινητήρας ελέγχεται από το κανάλι RC2.
	10	Ανεξάρτητη αριστερή. Ενεργοποιείται μόνο ο αριστερός κινητήρας. Ο δεξιός κινητήρας παραμένει ανενεργός. Ο δεξιός κινητήρας ελέγχεται από το κανάλι RC1.
	11	Ανεξάρτητη διπλή. Και οι δυο κινητήρες είναι ενεργοί. Μπορούν να ελεγχθούν ξεχωριστά από τα κανάλια RC1 και RC2. Αυτή η λειτουργία ενδείκνυται όταν χρησιμοποιείται κάποιος μικροκοντρόλερ
Εκθετική λειτουργία	SW5 0	Off. Η ταχύτητα είναι γραμμική με το εισαγόμενο σήμα. Αυτή η λειτουργία ενδείκνυται για κινητήρες με χαμηλής και μέσης ταχύτητας
	1	On. Η είσοδος είναι εκθετική και συνήθως ελέγχεται η ταχύτητα γύρω στο 0. Αυτή η λειτουργία είναι κατάλληλη για κινητήρες υψηλής ταχύτητας και πολύ ευαίσθητους κινητήρες
Λειτουργία	SW6 0	Ο κινητήρας σταματάει όταν το εισαγόμενο σήμα είναι 2.5V. Ο κινητήρας κινείται προς την μια κατεύθυνση όταν η είσοδος είναι μικρότερη από 2.5V και προς την άλλη κατεύθυνση όταν η είσοδος είναι μεγαλύτερη από 2.5V. Αυτή η λειτουργία ενδείκνυται όταν χρησιμοποιείται Joystick.
	1	AN1 και AN2 ελέγχουν την ταχύτητα του

	<p>κινητήρα. Ο κινητήρας σταματάει όταν η είσοδος είναι 0V και κινείται στην μέγιστη ταχύτητα όταν η είσοδος είναι 5V. DIG1 και DIG2 είναι ψηφιακή είσοδος και ελέγχουν την φορά του κάθε κινητήρα ξεχωριστά</p>
--	--



Εικόνα 14: PWM είσοδος με μικροελεγκτή, DIPswitch. DIPSWITCH: 10110100

[20]

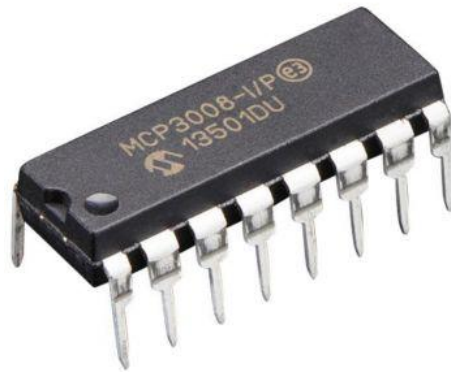
8.7. Κάμερα USB



Εικόνα 15: Κάμερα USB

Χρησιμοποιήθηκε η παραπάνω webcamera με ανάλυση 2MP για την σύλληψη φωτογραφιών σε συγκεκριμένη χρονική στιγμή της ημέρας.

8.8.MCP3008 - 10bit 8 channel ADC SPI

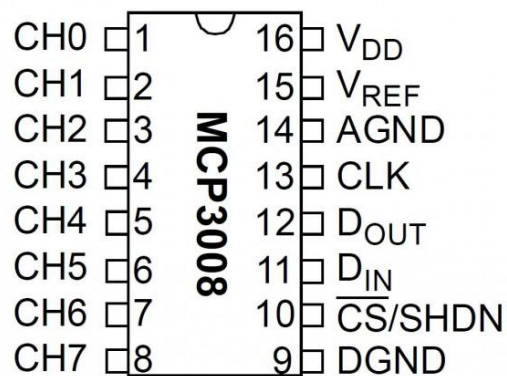


Εικόνα 16: MCP3008

Αυτό το chip έχει 8 κανάλια από 10-bit αναλογικών εισόδων για το raspberry. Είναι πολύ χρήσιμο για το raspberry καθώς αυτό δεν έχει αναλογικές εισόδους. Χρησιμοποιεί SPI πρωτόκολλο επικοινωνίας.[21]

Πίνακας 12: Χαρακτηριστικά MCP3008

Τάση λειτουργίας	(2.7V - 5.5V)
Κατανάλωση ρεύματος σε κατάσταση ηρεμίας	5 nA
Κατανάλωση ρεύματος σε λειτουργία	320μΑ
Ρεύμα τροφοδοσίας	425μΑ
Πρωτόκολλο επικοινωνίας	SPI
Πλήθος καναλιών	8



Εικόνα 17: MCP3008 pins

Το παραπάνω chip λαμβάνει αναλογικές τιμές από τους αισθητήρες υγρασίας εδάφους και φωτεινότητας.

8.9. Λάμπα

Χρησιμοποιήθηκε η παρακάτω λάμπα 12V dc, 5W για την αύξηση της φωτεινότητας στο εσωτερικό του θερμοκηπίου



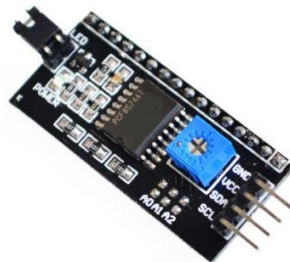
Εικόνα 18: Λάμπα 12v dc

8.10. 1602 16X2 Character LCD Display Module with IIC/I2C/TWI/SPI Interface – Blue

Χρησιμοποιήθηκε η παρακάτω οθόνη lcd 16 χαρακτήρων και 2 γραμμών για την ένδειξη της κατάστασης του θερμοκηπίου.



Εικόνα 19: LCD οθόνη



Εικόνα 20: I2C για LCD οθόνη

Πίνακας 13: Τεχνικά χαρακτηριστικά LCD οθόνης [2]

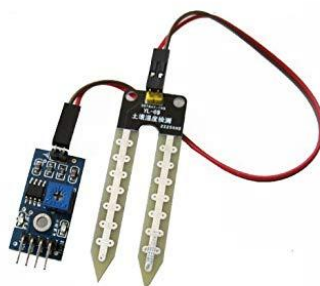
Τροφοδοσία	5V
Διάσταση	82x35x18 mm
Επικοινωνία	I2C
Εμφάνιση	Άσπρα γράμματα σε μπλε οθόνη

Ο διάυλος επικοινωνίας I2C είναι ένας σειριακός διάυλος που δημιουργήθηκε και χρησιμοποιείται για την σύνδεση περιφερειακών μικρής ταχύτητας σε μητρικές πλακέτες (motherboards), ενσωματωμένα συστήματα (embedded systems), κινητά τηλέφωνα ή άλλες ηλεκτρονικές συσκευές. Στην παρούσα εργασία χρησιμοποιείται για την σύνδεση του Raspberry με την οθόνη lcd.

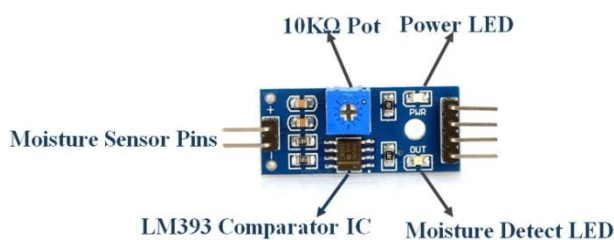
Τα καλώδια που χρειάζονται για την επικοινωνία είναι:

- Η γραμμή SCL είναι η γραμμή ρολογιού
- η SDA είναι η γραμμή δεδομένων
- η γείωση (GND) ή 0 V.
- η γραμμή τροφοδοσίας (VCC ή VDD), με την οποία τροφοδοτούνται με ισχύ οι διάφορες συσκευές που συνδέονται στο διάυλο. Τυπικές τάσεις που χρησιμοποιούνται στο διάυλο είναι τα +5V ή +3,3V. [22][23]

8.11. Soil Moisture Sensor Module



Εικόνα 21: Moisture sensor



Εικόνα 22: LM393 συγκριτής

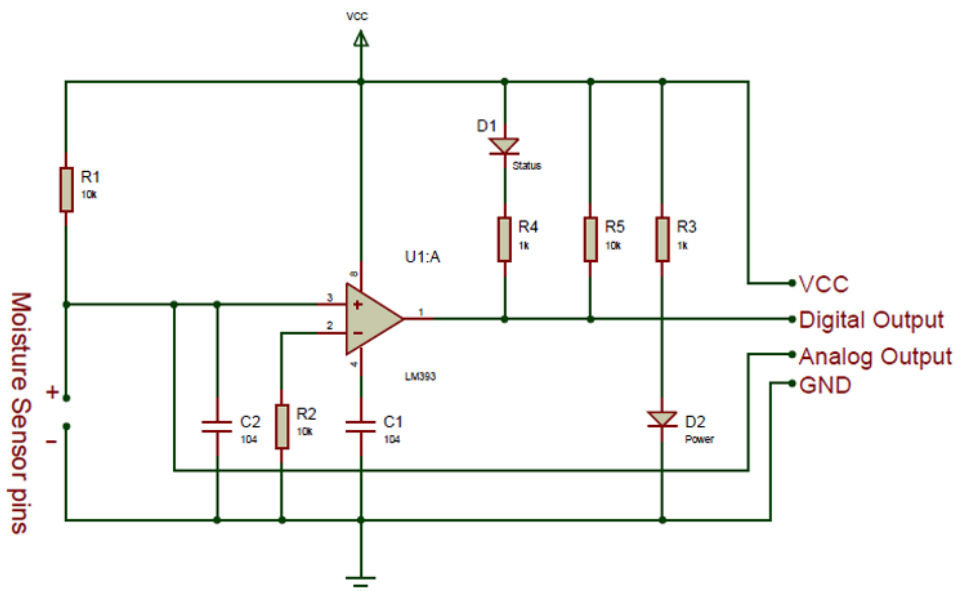
Πίνακας 14: Τεχνικά χαρακτηριστικά υγρασίας εδάφους

Τάση λειτουργίας	3.3V to 5V DC
Ρεύμα λειτουργίας	15mA
Ψηφιακή τάση εξόδου	0V to 5V
Αναλογική τάση εξόδου	0V to 5V
Διαστάσεις	3.2cm x 1.4cm
Ένδειξη	Τροφοδοσίας και εξόδου σήματος

Ο συγκριτής τάσης LM393 IC χρησιμοποιείται για τον αισθητήρα ανίχνευσης υγρασίας εδάφους. Pin 2 του LM393 είναι συνδεδεμένο σε 10KΩ αντίσταση ενώ το pin 3 είναι συνδεδεμένο στο pin του αισθητήρα υγρασίας εδάφους. Ο συγκριτής IC μπορεί να συγκρίνει την τάση που εμφανίζει το pin2 με την τάση του pin 3 που αποτελεί τον αισθητήρα της ανίχνευσης υγρασίας.

Ο αισθητήρας υγρασίας εδάφους αποτελείται από δυο αισθητήρια που χρησιμοποιούνται για την ανίχνευση της υγρασίας του εδάφους. Οι αισθητήρες είναι επικαλυμμένοι με γαλβάνη για να προστατευτούν από την οξείδωση. Αυτά τα δύο αισθητήρια χρησιμοποιούνται για να περνάει το ρεύμα μέσα από το έδαφος ώστε ο αισθητήρας να μπορεί να μετρήσει την υγρασία του εδάφους

Ο αισθητήρας περιλαμβάνει τέσσερα pin VCC, GND, DO, AO. Το ψηφιακό pin είναι συνδεδεμένο στο pin εξόδου του συγκριτή Digital LM393 IC όταν το αναλογικό pin είναι συνδεδεμένο στον αισθητήρα. Το διάγραμμα του αισθητήρα φαίνεται παρακάτω.[24]



Εικόνα 23: Ηλεκτρονικό σχέδιο αισθητήρα υγρασίας εδάφους

8.12. Ηλεκτροβάννα 1'' ίντζας

Χρησιμοποιήθηκε η παρακάτω ηλεκτροβάννα για τον έλεγχο της παροχής ποτίσματος του θερμοκηπίου. Τα χαρακτηριστικά της φαίνονται στον παρακάτω πίνακα. [25]

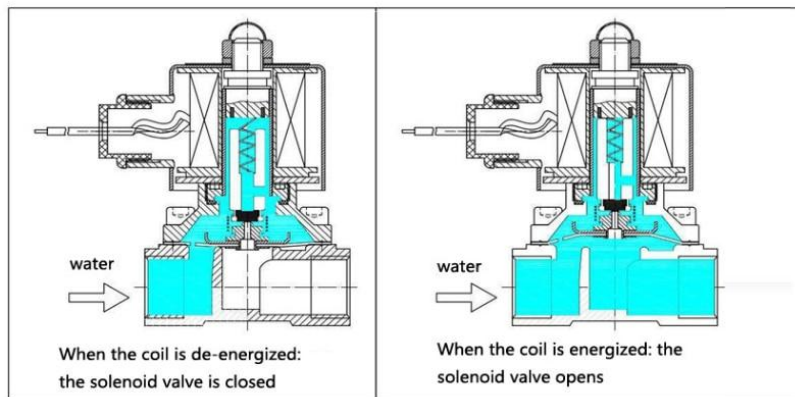
Πίνακας 15: Τεχνικά χαρακτηριστικά ηλεκτροβάνας

Τάση λειτουργίας	24V DC
Διάσταση σωλήνα	1''
Κάτασταση	Normally Closed

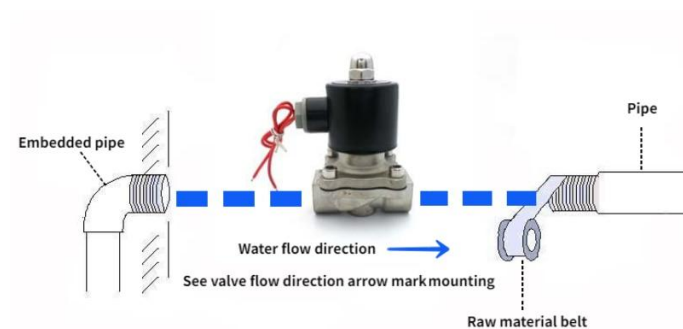


Εικόνα 24: Ηλεκτροβάνα

Working principle diagram



Εικόνα 25: Κατάσταση λειτουργίας ηλεκτροβάνας



Εικόνα 26: Σύδεση ηλεκτροβάνας

8.13. Αισθητήρας φωτεινότητας



Εικόνα 27: Αισθητήρας φωτεινότητας

Τα χαρακτηριστικά του αισθητήρα είναι:

- Διαστάσεις: 33x115x8mm
- Βάρος: 4gm
- Κύριο chip: LM393
- Τάση λειτουργίας: 3-5 VDC
- Ενδεικτική λυχνία λειτουργίας
- LDR λειτουργία με 4 PIN
- Τάση λειτουργίας 3.3V-5V
- Ικανό να διακρίνει περιβάλλουσα φωτεινότητα και ελαφριά ένταση.
- Ρυθμιζόμενη ευαισθησία μέσω μπλέ ποτενσιόμετρου
- Έξοδος: DO: ψηφιακή έξοδος (0 και 1). AO: αναλογική τάση εξόδου. [26]

8.14. Αντλία 12V

Χρησιμοποιήθηκε η παρακάτω αντλία σεντίνας για την τροφοδοσία του θερμοκηπίου με νερό.



Εικόνα 28: Αντλία σεντίνας

Τα τεχνικά χαρακτηριστικά της αντλίας φαίνονται στον παρακάτω πίνακα. [27]

Πίνακας 16: Τεχνικά χαρακτηριστικά Αντλίας

Τάση τροφοδοσίας	12V
Ρεύμα	3A
Διατομή	3/4''

8.15. Κινητήρας 12v dc



Εικόνα 29: Κινητήρας 12Vdc

Χρησιμοποιήθηκε ο παραπάνω κινητήρας χαμηλής στάθμης θορύβου για την μείωση της εσωτερικής θερμοκρασίας του θερμοκηπίου. Τα τεχνικά χαρακτηριστικά του κινητήρα φαίνονται στο παρακάτω πίνακα.

Πίνακας 17: Τεχνικά χαρακτηριστικά ανεμιστήρα

Διάσταση	140mm
Επίπεδο θορύβου (max)	19,8 db
Τάση τροφοδοσίας	12 V
Δυναμικότητα	0,36 W
Ροή αέρα (max)	61,2 cfm
Στροφές ανεμιστήρα (max)	1000 rpm
Σύνδεση	4-Pin PWM

[28]

8.16. Κινητήρας mg996R

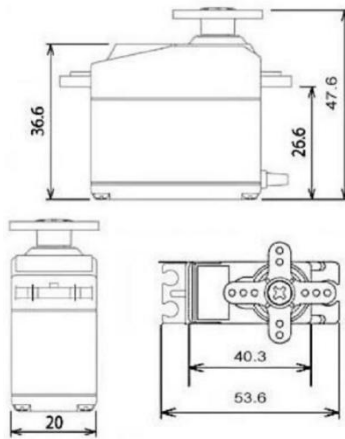
Ο κινητήρας mg996R είναι υπεύθυνος στην κατασκευή για το άνοιγμα και το κλείσιμο του παραθύρου. Τα τεχνικά χαρακτηριστικά του κινητήρα φαίνονται στον παρακάτω πίνακα.

Πίνακας 18: Τεχνικά χαρακτηριστικά κινητήρα

Τάση λειτουργίας	5 – 6V
Κατανάλωση ρεύματος	2.5 A (6 V)
Μέγιστη ροπή	11 kg/cm (6V)
Ταχύτητα λειτουργίας	0.17 s / 60°
Περιστροφή	0° – 180°
Βάρος	55gm



Εικόνα 30: Κινητήρας mg996



Εικόνα 31: Διαστάσεις κινητήρα

[29]

8.17. Κινητήρας 12V dc



Εικόνα 32: Κινητήρας dc 12V

Χρησιμοποιήθηκε ο παραπάνω κινητήρας χαμηλής στάθμης θορύβου και υψηλής ροπής για την μείωση της εσωτερικής θερμοκρασίας του θερμοκηπίου. Τα τεχνικά χαρακτηριστικά του κινητήρα φαίνονται στο παρακάτω πίνακα. [30]

Πίνακας 19: Τεχνικά χαρακτηριστικά κινητήρα

Τάση τροφοδοσίας	12V dc
Ταχύτητα	12000 rpm



Εικόνα 33: Διαστάσεις κινητήρα

8.18. Παρελκόμενα στήριξης

Χρησιμοποιήθηκαν γωνίες σιδήρου 2,5x2,5x3mm για τον σκελετό του θερμοκηπίου, για τον κινητήρα και για την βάση της δεξαμενής οι οποίοι κολλήθηκαν με ηλεκτροκόλληση. Τα plexiglass είναι πάχους 4mm. Στηρίχθηκαν με βίδες διάφορων διαμετρημάτων.



Εικόνα 34: Γωνίες 2.5x2.5x3mm



Εικόνα 35: Βίδες και περικόγλια



Εικόνα 36: Plexiglass 4mm

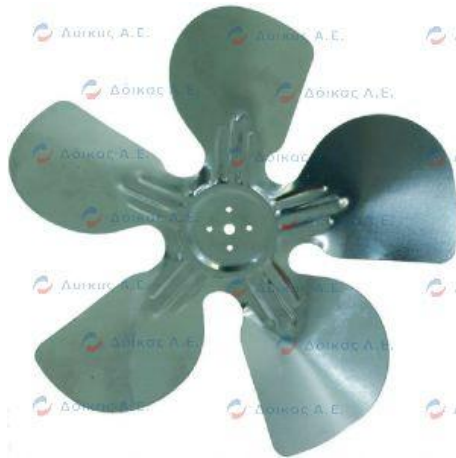
Για την στήριξη του εύκαμπτου λάστιχου χρησιμοποιήθηκε κοιλοδοκός 16x16x2mm.



Εικόνα 37: Κοιλοδοκός

8.19. Φτερωτή αναρρόφησης

Η αναρρόφηση του ζεστού αέρα πραγματοποιείται με φτερωτή $\Phi 170\text{mm}$.



Εικόνα 38: Φτερωτή

8.20. Δεξαμενή νερού

Το νερό που θα χρησιμοποιηθεί για το πότισμα του θερμοκηπίου προέρχεται από δεξαμενή. Την δεξαμενή αποτελεί ένα δοχείο χωρητικότητας 30L.



Εικόνα 39: Δοχείο 30L

8.21. Ασφάλειες & Ασφαλειοθήκες

Χρησιμοποιήθηκαν ασφάλειες 10 A , 5 A, 4 A, 2 A για την διασφάλιση της ομαλής λειτουργίας του ηλεκτρολογικού εξοπλισμού και την αποφυγή καταστροφής τους. Πιο αναλυτικά:

- 10A ασφάλεια χρησιμοποιήθηκε στην τροφοδοσία της πλακέτας mdds10 SmartDriveDuo-10 καθώς και στην είσοδο του κινητήρα 12Vdc 12000 rpm.

- 5 A ασφάλεια χρησιμοποιήθηκε στην τροφοδοσία του μετατροπέα τάσης dc-dc 12- 5V
- 5 A ασφάλεια χρησιμοποιήθηκε στην τροφοδοσία του μετατροπέα τάσης dc-dc 12-24V



Εικόνα 40: Ηλεκτρολογικές ασφάλειες



Εικόνα 41: Ασφαλειοθήκη



Εικόνα 42: Ασφαλειοθήκη

8.22. Τροφοδοτικό ρεύματος



Εικόνα 43: Τροφοδοτικό ρεύματος

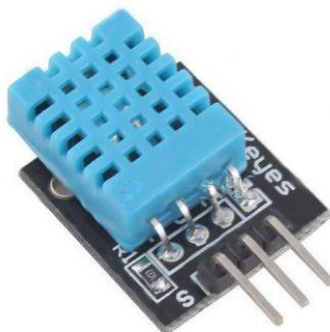
Για την τροφοδοσία ρεύματος όλης της εγκατάστασης χρησιμοποιήθηκε ο παραπάνω μετατροπέας τάσης. Τα χαρακτηριστικά του φαίνονται στον παρακάτω πίνακα. [31]

Πίνακας 20: Τεχνικά χαρακτηριστικά τροφοδοτικού

Τάση εισόδου	100-250V AC 50/60 Hz
Τάση εξόδου	12V DC
Ρεύμα εξόδου	41A
Ισχύς εξόδου	500W max
Βαθμός προστασίας	IP 20
Διαστάσεις	245 x 120 x 70 mm

8.23. Humidity Sensor DHT11 Module

Χρησιμοποιήθηκαν δυο αισθητήρες dht11 για την μέτρηση της θερμοκρασίας του εξωτερικού περιβάλλοντος και του εσωτερικού του θερμοκηπίου.[32]



Εικόνα 44: Αισθητήρας υγρασίας και θερμοκρασίας DHT11

Πίνακας 21: Τεχνικά χαρακτηριστικά αισθητήρα υγρασίας και θερμοκρασίας

Τύπος αισθητήρας	Humidity, Temperature
Τάση λειτουργίας	3VDC – 3.3VDC – 5VDC
Τάση τροφοδοσίας	5V
Ένταση λειτουργίας	2.5 mA
Πρωτόκολλο λειτουργίας	Single Wire
Κατανάλωση σε κατάσταση λειτουργίας	0.5mA typ. (2.5mA max.)
Κατανάλωση (stand-by)	100uA typ. (150uA max.)
Εύρος ένδειξης θερμοκρασίας	0 / +50°C ±2°C
Εύρος ένδειξης υγρασίας	20-90%RH ±5%RH
Διαστάσεις	21.9 x 16.6 x 7mm

8.24. HC-SR04 αισθητήρας υπερήχων



Εικόνα 45: Αισθητήρας υπερήχων

Ο αισθητήρας υπερήχων χρησιμοποιείται για την εύρεση της στάθμης της δεξαμενής. Είναι τοποθετημένος στο πάνω μέρος, στο καπάκι της δεξαμενής. Τα χαρακτηριστικά του φαίνονται στον παρακάτω πίνακα. [33]

Πίνακας 22: Χαρακτηριστικά αισθητήρα υπερήχων

Τάση λειτουργίας	5V DC
Κατανάλωση ρεύματος	15mA
Συχνότητα λειτουργίας	40Hz
Μέγιστη απόσταση	4 m
Ελάχιστη απόσταση	2 cm
Διάσταση	45*20*15mm

8.25. Τροφοδοτικό ρεύματος



Εικόνα 46: Τροφοδοτικό ρεύματος

Χρησιμοποιήθηκαν 2 τεμάχια του παραπάνω τροφοδοτικού ρεύματος για την παροχή ρεύματος στο arduino και στο raspberry. Τα χαρακτηριστικά του φαίνονται στον παρακάτω πίνακα.[34]

Πίνακας 23: Τεχνικά χαρακτηριστικά τροφοδοτικού

Ισχύς	15Watt
Τάση εξόδου	5V DC
Ρεύμα εξόδου	3A
Τάση εισόδου	240V

8.26. Καλώδια



Εικόνα 47: Καλώδια

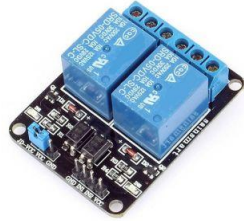
Χρησιμοποιήθηκαν τα παρακάτω είδη καλωδίου διαφόρων διαμετρημάτων :

- Αρσενικό – θηλυκό
- Αρσενικό – Αρσενικό
- θηλυκό – θηλυκό

[35]

8.27. Ρελέ

Χρησιμοποιήθηκαν τα παρακάτω ρελέ για τον έλεγχο της λάμπας, της αντλίας και της ηλεκτροβάνας.



Εικόνα 48: Ρελέ

Πίνακας 24: Τεχνικά χαρακτηριστικά ρελέ

Τάση πηνίου	5V DC
Μέγιστο φορτίο	10A
Τάση ελέγχου	0-250V
Διαστάσεις	51x38.5x18mm

[36]

9. Το λογισμικό ελέγχου

Σε αυτό το κεφάλαιο περιγράφεται ο τρόπος που έχουν γραφτεί τα προγράμματα στην γλώσσα προγραμματισμού Python. Όταν μπει σε λειτουργία το Raspberry Pi αναμένει από τον χρήστη να πάρει εντολές για την έναρξη των προγραμμάτων που χρησιμοποιούνται για την λειτουργία του θερμοκηπίου .

Στην γλώσσα προγραμματισμού Python δεν υπάρχει μια μόνο ροή προγράμματος. Έχουν δημιουργηθεί πέντε διαφορετικά προγράμματα που το καθένα από αυτά πραγματοποιεί μια λειτουργία. Αναλυτικότερα υπάρχει:

1. Το πρόγραμμα δειγματοληψίας (teliko4.py) το οποίο καταγράφει σε csv την κατάσταση του θερμοκηπίου, δηλαδή
 - τις τιμές της υγρασίας, θερμοκρασίας του εσωτερικού και εξωτερικού αέρα,
 - την τιμή της υγρασίας του εδάφους
 - την φωτεινότητα
 - το ύψος του νερού στο εσωτερικό της δεξαμενής.
2. Το πρόγραμμα εμφάνισης της κατάστασης του θερμοκηπίου σε lcd οθόνη(teliko-lcd.py) για την άμεση ενημέρωση του χρήστη όταν αυτός δεν βρίσκεται μπροστά από την οθόνη του υπολογιστή

3. Το κύριο πρόγραμμα αλληλεπίδρασης του συστήματος με τον χρήστη από το οποίο ελέγχεται το θερμοκήπιο (teliko-user8.py)
4. Το πρόγραμμα λήψης φωτογραφιών (teliko-camera.py) το οποίο λαμβάνει φωτογραφίες σε συγκεκριμένα χρονικά διαστήματα.
5. Το πρόγραμμα ενεργοποίησης του μηχανολογικού εξοπλισμού (teliko-arduino.ino)

Όλα αυτά τα προγράμματα θα αναλυθούν σε επόμενα κεφάλαια.

9.2. Το πρόγραμμα καταγραφής δειγματοληψίας των καταστάσεων του θερμοκηπίου

Το παρακάτω πρόγραμμα χρησιμοποιείται για την λήψη των μετρήσεων και την καταγραφή τους στα csv αρχεία. Η κάθε μέτρηση που λαμβάνει το Raspberry από τους αισθητήρες καταχωρείται κάθε φορά στην αντίστοιχη στήλη του data.csv αρχείου ενώ ανά 5 λεπτά καταχωρούνται και στο data1.csv. Αναλυτικότερα στην:

- 1^η στήλη καταχωρείται το διαφορετικό id της κάθε χρονικής στιγμής που λαμβάνεται η μέτρηση στην ουσία αποτελεί έναν αύξοντα αριθμό
- 2^η και 3^η στήλη καταχωρείται η ημερομηνία και η ώρα της δειγματοληψίας.
- 4^η και 5^η στήλη καταχωρείται η εσωτερική θερμοκρασία και υγρασία του θερμοκηπίου
- 6^η και 7^η καταχωρείται η θερμοκρασία και υγρασία του περιβάλλοντος
- 8^η καταχωρείται η φωτεινότητα
- 9^η καταχωρείται η υγρασία του εδάφους
- 10^η καταχωρείται η στάθμη της δεξαμενής νερού

Πιο συγκεκριμένα

Κατά την εκκίνηση του προγράμματος εισάγονται οι βιβλιοθήκες

```
import RPi.GPIO as GPIO

import dht11

import time

import os

from time import sleep

from datetime import date

from datetime import datetime

import busio

import digitalio

import board
```

```

import adafruit_mcp3xxx.mcp3008 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn
import adafruit_dht
import smbus
import serial
from scipy.interpolate import interp1d
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.cleanup()

```

Στο pin D17 λαμβάνεται η εσωτερική θερμοκρασία και υγρασία του αέρα του θερμοκηπίου

```
dhtDevicees = adafruit_dht.DHT11(board.D17)
```

Στο pin D27 καταχωρείται η εξωτερική θερμοκρασία και υγρασία του αέρα

```
dhtDevice = adafruit_dht.DHT11(board.D27)
```

Δημιουργεί τον spi δίαυλο επικοινωνίας

```
spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)
```

Δημιουργεί το cs (chip select)

```
cs = digitalio.DigitalInOut(board.CE0)
```

Δημιουργεί το mcp αντικείμενο

```
mcp = MCP.MCP3008(spi, cs)
```

Δημιουργεί την αναλογική είσοδο στο pin 0, pin 7 του MCP3008 όπου παίρνει τιμές για την υγρασία του εδάφους και την φωτεινότητα του θερμοκηπίου αντίστοιχα.

```
chan = AnalogIn(mcp, MCP.P0)
```

```
chan7 = AnalogIn(mcp, MCP.P7)
```

Εισαγωγή βιβλιοθήκης

```
import pandas as pd
```

Ανοίγει το έγγραφο data.csv για εισαγωγή δεδομένων στο τέλος του αρχείου

```
file = open("/home/pi/data.csv", "a")
```

Εάν το έγγραφο είναι καινούργιο δημιουργεί τους τίτλους των στηλών

```
if os.stat("/home/pi/data.csv").st_size == 0:
```

```
    file.write("id,Date,Time,Inside Temperature (C),Inside Air Humidity(%),Outside Temperature (C),Outside Humidity(%),Brightness (%),Soil moisture(%),Tank (lt)\n")
```

Ανοίγει το αρχείο και βρίσκει το μέγιστο id και το αυξάνει κατά 1

```
df=pd.read_csv("/home/pi/data.csv")
```

```
p=df['id'].max()
```

```
id=p+1
```

Δημιουργεί το έγγραφο data1.csv όπου καταχωρούνται οι τιμές των αισθητήρων ανά 5 λεπτά

```
file1 = open("/home/pi/data1.csv", "a")
```

Αν είναι κενό δημιουργεί τους τίτλους των στηλών

```
if os.stat("/home/pi/data1.csv").st_size == 0:
```

```
    file1.write("id,Date,Time,Inside Temperature (C),Inside Air Humidity(%),Outside Temperature (C),Outside Humidity(%),Brightness (%),Soil moisture(%), Tank (lt)\n")
```

Ανοίγει το αρχείο και βρίσκει το μέγιστο id και το αυξάνει κατά 1

```
df=pd.read_csv("/home/pi/data1.csv")
```

```
pn=df['id'].max()
```

```
idn=pn+1
```

Καταχωρεί στην μεταβλητή start1_time την υφιστάμενη ώρα.

```
start1_time=time.time()
```

Καταχωρεί στο gpio 23, 25 το trigger pin καθώς και το echo pin του αποστασιομέτρου για την μέτρηση της στάθμης της δεξαμενής

```
GPIO_TRIGGER = 23
```

```
GPIO_ECHO = 25
```

```
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
```

```
GPIO.setup(GPIO_ECHO, GPIO.IN)
```

Η συνάρτηση litra είναι υπεύθυνη για τον υπολογισμό του λίτρων που υπάρχουν στο εσωτερικό της δεξαμενής με την βοήθεια του αποστασιομέτρου. Η τιμή των λίτρων που υπάρχουν στο εσωτερικό της δεξαμενής καταχωρούνται στην μεταβλητή litra.

```

def litra():
    Καταχώρηση του Trigger ως HIGH
    GPIO.output(GPIO_TRIGGER, True)
    Καταχώρηση του trigger μετά από 0.01ms σε LOW
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)
    StartTime = time.time()
    StopTime = time.time()
    while GPIO.input(GPIO_ECHO) == 0:
        StartTime = time.time()
    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()
    Χρονική διαφορά μεταξύ της έναρξης και της επιστροφής του παλμού
    TimeElapsed = StopTime - StartTime
    Μέτρηση της απόστασης και κατ' επέκταση των λίτρων που υπάρχουν στο εσωτερικό της
    δεξαμενής
    dist = (TimeElapsed * 34300) / 2
    litra= ((48-dist)*30)/48
    litra= int((litra*2)/2)
    return litra

```

Το τμήμα που υπάρχει στο εσωτερικό της συνάρτησης while εκτελείται για πάντα.

```

while True:
    try:
        Λαμβάνει τιμές από τους αισθητήρες
        chan = AnalogIn(mcp, MCP.P0)
        chan7 = AnalogIn(mcp, MCP.P7)
        br=((chan7.voltage)/3.29)*100

```

```

br=(100-br)
br=int((br*2)/2)
time.sleep(1)
m=(chan.voltage)
g = interp1d([2.2,3.3],[100,0])
time.sleep(0.5)
m_map=float(g(m))
m_map=int((m_map*2)/2)
m=m_map
time.sleep(0.5)
today=date.today()
d1 = today.strftime("%d-%m-%Y")
now = datetime.now()
t_string = now.strftime("%H:%M:%S")
tempes = dhtDevicees.temperature
humes = dhtDevicees.humidity
time.sleep(0.3)
lit = litra()
temperature_c = dhtDevice.temperature
humidity = dhtDevice.humidity
time.sleep(0.3)
    Εκτυπώνει τις τιμές στο terminal
print('.....')
print('id: ', id)
print ("Measured Litres: %.1f It" % lit)
print('Soil ADC Value: ', chan.value)

```

```

print('Soil ADC Voltage: ' + str(chan.voltage) + 'V')

print( 'Soil moisture ' +str(m) + ' %')

print( 'Brightness ' +str(br) + ' %')

print('Brightness Raw ADC Value: ', chan7.value)

print('Brightness ADC Voltage: ' + str(chan7.voltage) + 'V')

print("Inside Air Temperature: %-3.1f C" % tempes)

print("Inside Air Humidity: %-3.1f %% " % humes)

print("Out Air Temp: {:.1f} C   Out Air Humidity: {}% ".format(temperature_c, humidity))

time.sleep(0.5)

```

Καταχωρεί τις τιμές αυτές στο αρχείο data.csv

```

file.write(str(id)+" "+str(d1)+" "+str(t_string)+" "+str(tempes)+" "+str(humes)+" "+str(temperature_c)
+" "+str(humidity)+" "+str(br)+" "+str(m)+" "+str(lit)+"\n")

```

```

file.flush()

```

```

time.sleep(0.5)

```

```

id=id+1

```

Κάθε 5 λεπτά καταχωρεί τις τιμές που λαμβάνει από τους αισθητήρες στο αρχείο data1.csv

```

if time.time() - start1_time >= 300:

```

```

    start1_time = time.time()

```

```

file1.write(str(idn)+" "+str(d1)+" "+str(t_string)+" "+str(tempes)+" "+str(humes)+" "+str(temperature_c)
+" "+str(humidity)+" "+str(br)+" "+str(m)+" "+str(lit)+"\n")

```

```

file1.flush()

```

```

time.sleep(1)

```

```

print('Written. id : ', idn)

```

```

idn=idn + 1

```

```

time.sleep(1)

```

Επειδή οι αισθητήρες dht11 συχνά δημιουργούν σφάλματα κατά την δειγματοληψία οι παρακάτω εντολές εξασφαλίζουν την διαγραφή των σφαλμάτων και την ομαλή λειτουργία του προγράμματος


```

except OverflowError as error:

    pass

    time.sleep(1.0)

    continue

except RuntimeError as error:

    pass

    time.sleep(2.0)

    continue

except Exception as error:

    dhtDevice.exit()

    raise error

    time.sleep(2.0)

if __name__ == '__main__':

    try:

        main()

    except KeyboardInterrupt:

        pass

```

9.3. Το πρόγραμμα εμφάνισης της κατάστασης του θερμοκηπίου σε lcd οθόνη

Το παρακάτω πρόγραμμα είναι υπεύθυνο για την εμφάνιση της κατάστασης του θερμοκηπίου στην lcd οθόνη, δηλαδή ανακαλεί κάθε φορά από το data.csv αρχείο τις τιμές της θερμοκρασίας και υγρασίας του αέρα στο εσωτερικό του θερμοκηπίου, τις τιμές θερμοκρασίας και υγρασίας στο περιβάλλον του θερμοκηπίου, της φωτεινότητας καθώς και την στάθμη της δεξαμενής.

Πιο συγκεκριμένα:

Εισάγει τις βιβλιοθήκες

```

import smbus

import time

```

```
from datetime import date
```

```
from datetime import datetime
```

```
import csv
```

```
    Καταχωρεί παραμέτρους της συσκευής
```

```
Καταχωρεί την I2C διεύθυνση της συσκευής
```

```
I2C_ADDR = 0x27
```

```
    Καταχωρεί το μέγιστο πλήθος χαρακτήρων ανά γραμμή
```

```
LCD_WIDTH = 16
```

```
LCD_CHR = 1
```

```
LCD_CMD = 0
```

```
    Ram διεύθυνση πρώτης γραμμής
```

```
LCD_LINE_1 = 0x80
```

```
    Ram διεύθυνση δεύτερης γραμμής
```

```
LCD_LINE_2 = 0xC0
```

```
    Πίσω φως ανοιχτό
```

```
LCD_BACKLIGHT = 0x08 # On
```

```
    Ενεργοποίηση bit
```

```
ENABLE = 0b00000100
```

```
E_PULSE = 0.0005
```

```
E_DELAY = 0.0005
```

```
    Ανοίγει την I2C διάλο επικοινωνίας
```

```
bus = smbus.SMBus(1)
```

```
def lcd_init():
```

```
    Αρχικοποίηση οθόνης
```

```
    lcd_byte(0x33,LCD_CMD)
```

```
    lcd_byte(0x32,LCD_CMD)
```

```
    lcd_byte(0x06,LCD_CMD)
```

```

lcd_byte(0x0C,LCD_CMD)
lcd_byte(0x28,LCD_CMD)
lcd_byte(0x01,LCD_CMD)
time.sleep(E_DELAY)
def lcd_byte(bits, mode):
    bits_high = mode | (bits & 0xF0) | LCD_BACKLIGHT
    bits_low = mode | ((bits<<4) & 0xF0) | LCD_BACKLIGHT
    bus.write_byte(I2C_ADDR, bits_high)
    lcd_toggle_enable(bits_high)
    bus.write_byte(I2C_ADDR, bits_low)
    lcd_toggle_enable(bits_low)
def lcd_toggle_enable(bits):
    time.sleep(E_DELAY)
    bus.write_byte(I2C_ADDR, (bits | ENABLE))
    time.sleep(E_PULSE)
    bus.write_byte(I2C_ADDR,(bits & ~ENABLE))
    time.sleep(E_DELAY)
def lcd_string(message,line):
    message = message.ljust(LCD_WIDTH," ")
    lcd_byte(line, LCD_CMD)
    for i in range(LCD_WIDTH):
        lcd_byte(ord(message[i]),LCD_CHR)
def main():
    lcd_init()
    Το πρόγραμμα που περιλαμβάνεται στην while εκτελείται κάθε φορά
    while True:

```

```

today=date.today()
d1 = today.strftime("%d-%m-%Y")
now = datetime.now()
t_string = now.strftime("%H:%M:%S")
    Εξάγει τις τιμές της κατάστασης του θερμοκηπίου και τις καταχωρεί σε μεταβλητές
file = open("/home/pi/data.csv")
csv_file = list(csv.reader(file))
reader = csv.reader(file)
lines= len(list(reader))
lines=lines-1
lines=int(lines)
humes = int(csv_file[lines][4])
tempes= int(csv_file[lines][3])
humidity = int(csv_file[lines][6])
temperature_c = int(csv_file[lines][5])
br = int(csv_file[lines][7])
m = int(csv_file[lines][8])
    Εμφανίζει τις μεταβλητές στην lcd οθόνη
lcd_string('Date: ' +str(d1) + ',LCD_LINE_1)
lcd_string('Time: '+str(t_string) +',LCD_LINE_2)
time.sleep(8)
lcd_string('In Hum: '+str(humes)+'%',LCD_LINE_1)
lcd_string('In Temp: ' +str(tempes)+'C',LCD_LINE_2)
time.sleep(8)
lcd_string('Out Hum:'+str(humidity)+'%',LCD_LINE_1)
lcd_string('Out Temp:'+str(temperature_c)+'C',LCD_LINE_2)

```

```

time.sleep(8)

lcd_string('Soil moist: '+str(m)+'%',LCD_LINE_1)

lcd_string('Brightness: '+str(br)+'%' , LCD_LINE_2)

time.sleep(8)

if __name__ == '__main__':

    try:

        main()

    except KeyboardInterrupt:

        pass

    finally:

        lcd_byte(0x01, LCD_CMD)

```

9.4. Το κύριο πρόγραμμα αλληλεπίδρασης με τον χρήστη

Παρακάτω φαίνεται το πρόγραμμα με το οποίο ο χρήστης αλληλεπιδρά με το θερμοκήπιο. Το πρόγραμμα αυτό είναι γραμμένο σε γλώσσα προγραμματισμού Python. Αναλυτικότερα:

Κατά την εκκίνηση πραγματοποιείται η εισαγωγή βιβλιοθηκών.

```

from tkinter import *

import math

import serial

import time

import matplotlib

import csv

import _csv

import numpy as np

import random

import os

import sys

import time

```

```

import datetime

from itertools import count

import pandas as pd

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

from matplotlib import style

from PIL import ImageTk, Image

from tkinter import filedialog

import numpy as np

from scipy.interpolate import interp1d

from time import sleep

from datetime import date

from datetime import datetime

from datetime import timedelta

```

Ανοίγει συριακό διάλογο επικοινωνίας μεταξύ του raspberry και του arduino.

```

ser = serial.Serial('/dev/ttyACM0',9600, timeout=1)

ser.flush()

def focusthermokrasia(event):

    thermokrasia_field.focus_set()

def focusygrasia(event):

    ygrasia_field.focus_set()

```

Όταν καλεί το πρόγραμμα την συνάρτηση onanemisthras τότε το raspberry στέλνει την τιμή 128 στο arduino αφού πρώτα την έχει μετατρέψει σε string για να μπορεί να διαβαστεί από το arduino και η 7 σειρά και 4 στήλη της επιφάνειας εργασίας του χρήστη παίρνει την τιμή on. Αυτή η συνάρτηση ενεργοποιείται μόνο κατά την χειροκίνητη λειτουργία του ανεμιστήρα.

```

def onanemisthras():

    anemisthras_katastash = Label(root, text="On", bg="grey").grid(row=7, column=4)

    senv4=128

    angle_value_list = [str(senv4)]

```

```

send_string = ','.join(angle_value_list)

send_string += "\n"

ser.write(send_string.encode('utf-8'))

```

Όταν καλεί το πρόγραμμα την συνάρτηση `offanemisthras` τότε το raspberry στέλνει την τιμή 262 στο arduino αφού πρώτα την έχει μετατρέψει σε string για να μπορεί να διαβαστεί από το arduino και η 7 σειρά και 4 στήλη της επιφάνειας εργασίας του χρήστη παίρνει την τιμή off. Αυτή η συνάρτηση ενεργοποιείται κατά την χειροκίνητη και αυτόματη λειτουργία.

`def offanemisthras():`

```

anemisthras_katastash = Label(root, text="Off", bg="grey").grid(row=7, column=4)

senv4=262

angle_value_list = [str(senv4)]

send_string = ','.join(angle_value_list)

send_string += "\n"

ser.write(send_string.encode('utf-8'))

time.sleep(1)

```

Όταν καλεί το πρόγραμμα την συνάρτηση `onnero` τότε το raspberry στέλνει την τιμή 259 στο arduino αφού πρώτα την έχει μετατρέψει σε string για να μπορεί να διαβαστεί από το arduino και η 8 σειρά και 4 στήλη της επιφάνειας εργασίας του χρήστη παίρνει την τιμή on. Αυτή η συνάρτηση ενεργοποιείται κατά την χειροκίνητη και αυτόματη λειτουργία.

`def onnero():`

```

nero_katastash = Label(root, text="On", bg="grey").grid(row=8, column=4)

senv1=259

angle_value_list = [str(senv1)]

send_string = ','.join(angle_value_list)

send_string += "\n"

ser.write(send_string.encode('utf-8'))

time.sleep(1)

```

Όταν καλεί το πρόγραμμα την συνάρτηση `offnero` τότε το raspberry στέλνει την τιμή 258 στο arduino αφού πρώτα την έχει μετατρέψει σε string για να μπορεί να διαβαστεί από το arduino και η 8 σειρά και 4 στήλη της επιφάνειας εργασίας του χρήστη παίρνει την τιμή off. Αυτή η συνάρτηση ενεργοποιείται κατά την χειροκίνητη και αυτόματη λειτουργία.

```
def offnero():
```

```
    nero_katastash = Label(root, text="Off", bg="grey").grid(row=8, column=4)
```

```
    senv1=258
```

```
    angle_value_list = [str(senv1)]
```

```
    send_string = ','.join(angle_value_list)
```

```
    send_string += "\n"
```

```
    ser.write(send_string.encode('utf-8'))
```

```
    time.sleep(1)
```

Όταν καλεί το πρόγραμμα την συνάρτηση onfws τότε το raspberry στέλνει την τιμή 256 στο arduino αφού πρώτα την έχει μετατρέψει σε string για να μπορεί να διαβαστεί από το arduino και η 9 σειρά και 4 στήλη της επιφάνειας εργασίας του χρήστη παίρνει την τιμή on. Αυτή η συνάρτηση ενεργοποιείται κατά την χειροκίνητη και αυτόματη λειτουργία.

```
def onfws():
```

```
    fwteinothta_katastash = Label(root, text="On", bg="grey").grid(row=9, column=4)
```

```
    senv2=256
```

```
    angle_value_list = [str(senv2)]
```

```
    send_string = ','.join(angle_value_list)
```

```
    send_string += "\n"
```

```
    ser.write(send_string.encode('utf-8'))
```

```
    time.sleep(1)
```

Όταν καλεί το πρόγραμμα την συνάρτηση offfws τότε το raspberry στέλνει την τιμή 257 στο arduino αφού πρώτα την έχει μετατρέψει σε string για να μπορεί να διαβαστεί από το arduino και η 9 σειρά και 4 στήλη της επιφάνειας εργασίας του χρήστη παίρνει την τιμή off. Αυτή η συνάρτηση ενεργοποιείται κατά την χειροκίνητη και αυτόματη λειτουργία.

```
def offfws():
```

```
    fwteinothta_katastash = Label(root, text="Off", bg="grey").grid(row=9, column=4)
```

```
    senv2=257
```

```
    angle_value_list = [str(senv2)]
```

```
    send_string = ','.join(angle_value_list)
```

```
    send_string += "\n"
```



```
ser.write(send_string.encode('utf-8'))
```

```
time.sleep(1)
```

Όταν καλεί το πρόγραμμα την συνάρτηση parathyroon τότε το raspberry στέλνει την τιμή 261 στο arduino αφού πρώτα την έχει μετατρέψει σε string για να μπορεί να διαβαστεί από το arduino και η 10 σειρά και 4 στήλη της επιφάνειας εργασίας του χρήστη παίρνει την τιμή on. Αυτή η συνάρτηση ενεργοποιείται κατά την χειροκίνητη και αυτόματη λειτουργία.

```
def parathyroon():
```

```
    parathyro_katastash = Label(root, text="On", bg="grey").grid(row=10, column=4)
```

```
    senv3=261
```

```
    angle_value_list = [str(senv3)]
```

```
    send_string = ','.join(angle_value_list)
```

```
    send_string += "\n"
```

```
        ser.write(send_string.encode('utf-8'))
```

```
    time.sleep(1)
```

Όταν καλεί το πρόγραμμα την συνάρτηση parathyrooff τότε το raspberry στέλνει την τιμή 260 στο arduino αφού πρώτα την έχει μετατρέψει σε string για να μπορεί να διαβαστεί από το arduino και η 10 σειρά και 4 στήλη της επιφάνειας εργασίας του χρήστη παίρνει την τιμή off. Αυτή η συνάρτηση ενεργοποιείται κατά την χειροκίνητη και αυτόματη λειτουργία.

```
def parathyrooff():
```

```
    parathyro_katastash = Label(root, text="Off", bg="grey").grid(row=10, column=4)
```

```
    senv3=260
```

```
    angle_value_list = [str(senv3)]
```

```
    send_string = ','.join(angle_value_list)
```

```
    send_string += "\n"
```

```
        ser.write(send_string.encode('utf-8'))
```

```
    time.sleep(1)
```

Όταν ο χρήστης θέλει να εξάγει διαγράμματα με κριτήριο την ημερομηνία εισάγει στο πεδίο που εμφανίζεται στην 13^η γραμμή και 4^η στήλη του παράθυρου διαχείρισης την ημερομηνία από την οποία θέλουμε να εμφανίσουμε την αντίστοιχη κατάσταση του θερμοκηπίου. Το πρόγραμμα επικαλείται την συνάρτηση hmer και λαμβάνει την τιμή της ημερομηνίας.

```
def hmer():
```

```
hmapo_set=m.get()
```

```
hmer_setpoint = Label(root, text= hmapo_set, bg="grey").grid(row=13, column=4)
```

Όταν ο χρήστης επιθυμεί να εξάγει διαγράμματα με κριτήριο την ημερομηνία εισάγει στο πεδίο που εμφανίζεται στην 14^η γραμμή και 4^η στήλη του παράθυρου διαχείρισης την ημερομηνία έως την οποία θέλουμε να εμφανίσουμε την αντίστοιχη κατάσταση του θερμοκηπίου. Το πρόγραμμα επικαλείται την συνάρτηση `getvalue1` και λαμβάνει την τιμή της ημερομηνίας.

```
def getvalue1():
```

```
    hmews_set=z.get()
```

```
    getvaluetherm = Label(root, text=hmews_set, bg="grey").grid(row=14, column=4)
```

Όταν ο χρήστης επιθυμεί να εξάγει διάγραμμα θερμοκρασιών το πρόγραμμα επικαλείται την συνάρτηση `diagramma_therm`

```
def diagramma_therm():
```

```
    Διαβάζει το αρχείο data1.csv
```

```
    seira = []
```

```
    csv_file = csv.reader(open('/home/pi/data1.csv', "r"), delimiter=",")
```

```
    number = m.get()
```

```
    for row in csv_file:
```

Εάν η γραμμή που διαβάζει το πρόγραμμα είναι ίση με την μεταβλητή `number` στην οποία έχει καταχωρηθεί η τιμή της ημερομηνίας τότε το πρόγραμμα δημιουργεί μια λίστα και κάθε φορά προσθέτει στο τέλος της λίστας τον αριθμό του `id`. Έπειτα εκτυπώνει την λίστα.

```
        if number == row[1]:
```

```
            seira.append(row[0])
```

```
        print(seira)
```

```
        time.sleep(0.2)
```

Εάν η γραμμή που διαβάζει το πρόγραμμα είναι ίση με την μεταβλητή `number` στην οποία έχει καταχωρηθεί η τιμή της ημερομηνίας έως ότου θέλουμε να εμφανίσουμε την τιμή της εσωτερικής θερμοκρασίας τότε το πρόγραμμα κάθε φορά προσθέτει στο τέλος της λίστας τον αριθμό του `id`. Έπειτα εκτυπώνει την λίστα

```
    csv_file = csv.reader(open('/home/pi/data1.csv', "r"), delimiter=",")
```

```
    number = z.get()
```

```
    for row in csv_file:
```

```

if number == row[1]:
    seira.append(row[0])
print(seira)
print ("Original list is : " + str(seira))
for i in range(0, len(seira)):
    seira[i] = int(seira[i])
print ("Modified list is : " + str(seira))

    Εκτυπώνει το μέγιστο και το ελάχιστο στοιχείο της λίστας
min_ele,max_ele=seira[0],seira[0]
for i in range(1,len(seira)):
    if seira[i]<min_ele:
        min_ele=seira[i]
    if seira[i]>max_ele:
        max_ele=seira[i]
print('Minimum Element in the list',seira,'is',min_ele)
print('Maximum Element in the list',seira,'is',max_ele)
newdata = []
data = pd.read_csv("/home/pi/data1.csv", index_col="id")
for i in range(min_ele,max_ele+1,1):
    rows = data.loc[i]
    newdata.append(rows)
df = pd.DataFrame(data=newdata)
df.to_csv (r'diagrammatherm.csv', index = True, header=False)
s = df.to_csv()
print(s)

```

Δημιουργεί ένα καινούργιο αρχείο το οποίο περιέχει την ημερομηνία και τον χρόνο σε μια στήλη και στην άλλη στήλη την θερμοκρασία στο εσωτερικό του θερμοκηπίου.

```
with open("diagrammatherm.csv", "r") as source, open("result1.csv", "w") as result:
```

```

rdr = csv.reader(source)

wtr = csv.writer(result)

writer = csv.DictWriter(result, fieldnames=["time", "temperature"])

writer.writeheader()

for r in rdr:

    wtr.writerow([r[1] + ' ' + r[2]] + r[3:])

wtr.writerow([r[1] + ' ' + r[2]] + r[3:])

time.sleep(0.4)

plt.style.use('seaborn')

x_vals = []

y_vals = []

index = count()

```

Η συνάρτηση `animate` εμφανίζει το διάγραμμα της θερμοκρασίας στο εσωτερικό του θερμοκηπίου σε σχέση με τον χρόνο

```

def animate(i):

    data = pd.read_csv('result1.csv')

    x= data ['time']

    y1 = data ['temperature']

    plt.cla()

    plt.plot(x, y1, label='Θερμοκρασία (oC)',marker='o', linestyle='--', color='r')

    plt.xlabel('Χρόνος')

    plt.xticks(rotation=90)

    plt.legend(loc='upper left')

    plt.tight_layout()

ani = FuncAnimation(plt.gcf(), animate, interval=1000)

plt.tight_layout()

plt.show()

```

Όταν ο χρήστης επιθυμεί να εξάγει διάγραμμα εσωτερικής υγρασίας αέρα το πρόγραμμα επικαλείται την συνάρτηση `diagramma_ygr`

```
def diagramma_ygr():
```

```
    seira = []
```

```
    csv_file = csv.reader(open('/home/pi/data1.csv', "r"), delimiter=",")
```

```
    number = m.get()
```

```
    for row in csv_file:
```

Εάν η γραμμή που διαβάζει το πρόγραμμα είναι ίση με την μεταβλητή `number` στην οποία έχει καταχωρηθεί η τιμή της ημερομηνίας τότε το πρόγραμμα δημιουργεί μια λίστα και κάθε φορά προσθέτει στο τέλος της λίστας τον αριθμό του `id`. Έπειτα εκτυπώνει την λίστα.

```
        if number == row[1]:
```

```
            seira.append(row[0])
```

```
    print(seira)
```

```
    time.sleep(0.3)
```

Εάν η γραμμή που διαβάζει το πρόγραμμα είναι ίση με την μεταβλητή `number` στην οποία έχει καταχωρηθεί η τιμή της ημερομηνίας έως ότου θέλουμε να εμφανίσουμε την τιμή της εσωτερικής θερμοκρασίας τότε το πρόγραμμα κάθε φορά προσθέτει στο τέλος της λίστας τον αριθμό του `id`. Έπειτα εκτυπώνει την λίστα

```
    csv_file = csv.reader(open('/home/pi/data1.csv', "r"), delimiter=",")
```

```
    number = z.get()
```

```
    for row in csv_file:
```

```
        if number == row[1]:
```

```
            seira.append(row[0])
```

```
    print(seira)
```

```
    print ("Original list is : " + str(seira))
```

```
    for i in range(0, len(seira)):
```

```
        seira[i] = int(seira[i])
```

```
    print ("Modified list is : " + str(seira))
```

```
    min_ele,max_ele=seira[0],seira[0]
```

```
    for i in range(1,len(seira)):
```

```

if seira[i]<min_ele:
    min_ele=seira[i]
if seira[i]>max_ele:
    max_ele=seira[i]
print('Minimum Element in the list',seira,'is',min_ele)
print('Maximum Element in the list',seira,'is',max_ele)
newdata = []
data = pd.read_csv("/home/pi/data1.csv", index_col="id")
for i in range(min_ele,max_ele+1,1):
    rows = data.loc[i]
    newdata.append(rows)
df = pd.DataFrame(data=newdata)
df.to_csv('diagrammatherm.csv', index = True, header=False)
s = df.to_csv()
print(s)

```

Δημιουργεί ένα καινούργιο αρχείο το οποίο περιέχει την ημερομηνία και τον χρόνο σε μια στήλη και στην άλλη στήλη την υγρασία του αέρα που επικρατεί στο εσωτερικό του θερμοκηπίου.

with open("diagrammatherm.csv", "r") as source, open("result_ygr1.csv", "w") as result:

```

rdr = csv.reader(source)
wtr = csv.writer(result)
writer = csv.DictWriter(result, fieldnames=["time", "humidity"])
writer.writeheader()
for r in rdr:
    wtr.writerow([r[1] + ' ' + r[2]] + [r[4]])
plt.style.use('seaborn-whitegrid')
x_vals = []
y_vals = []
index = count()

```

Η συνάρτηση `animate` δημιουργεί την γραφική παράσταση της εσωτερικής υγρασίας αέρα σε συνάρτηση με τον χρόνο

```
def animate(i):  
    data = pd.read_csv('result_ygr1.csv')  
    x= data ['time']  
    y1 = data ['humidity']  
    plt.cla()  
    plt.plot(x, y1, label='Υγρασία αέρα (%)',linewidth=3,marker='o', linestyle='--', color='b')  
    plt.xlabel('Χρόνος')  
    plt.xticks(rotation=90)  
    plt.legend(loc='upper left')  
    plt.tight_layout()  
    ani = FuncAnimation(plt.gcf(), animate, interval=1000)  
    plt.tight_layout()  
    plt.show()
```

Όταν ο χρήστης θέλει να εμφανίσει μια φωτογραφία η συνάρτηση `openfn` ανοίγει ένα παράθυρο διαλόγου με τον χρήστη.

```
def openfn():
```

Άνοιγμα παραθύρου επικοινωνίας με τον χρήστη για την επιλογή της κατάλληλης φωτογραφίας που θα επιλέξει ο χρήστης να ανοίξει

```
    filename = filedialog.askopenfilename(title='open')
```

```
    return filename
```

```
def open_img():
```

Επιλογή φωτογραφίας από τον φάκελο

```
    x = openfn()
```

Άνοιγμα φωτογραφίας

```
    img = Image.open(x)
```

Μετατροπή του μεγέθους της φωτογραφίας

```
    img = img.resize((250, 250), Image.ANTIALIAS)
```

```
img = ImageTk.PhotoImage(img)
```

```
print(img)
```

Δημιουργία τίτλου φωτογραφίας

```
panel = Label(root, image=img)
```

```
panel.image = img
```

```
panel.grid(row=15)
```

Η συνάρτηση `user_interface` περιλαμβάνει όλο το περιβάλλον εργασίας του χρήστη

```
def user_interface():
```

```
    today=date.today()
```

Στην μεταβλητή `d1` καταχωρείται η υφιστάμενη ημερομηνία

```
d1 = today.strftime("%d-%m-%Y")
```

```
    now = datetime.now()
```

Στην μεταβλητή `t_string` καταχωρείται η υφιστάμενη ώρα

```
t_string = now.strftime("%H:%M:%S")
```

```
hmeromhnia_katastash = Label(root, text=d1, bg="grey").grid(row=1, column=6)
```

```
wra_katastash = Label(root, text=t_string, bg="grey").grid(row=2, column=6)
```

Ανοίγει το αρχείο `data.csv` για να καταχωρήσει στις αντίστοιχες μεταβλητές τις τιμές της κατάστασης του θερμοκηπίου και να τις εμφανίσει στο αντίστοιχο πεδίο.

```
file = open("/home/pi/data.csv")
```

```
csv_file = list(csv.reader(file))
```

```
reader = csv.reader(file)
```

```
lines= len(list(reader))
```

```
lines=lines-1
```

```
lines=int(lines)
```

```
finalvalue = int(csv_file[lines][3])
```

```
thermokrasia_katastash = Label(root, text=finalvalue, bg="grey").grid(row=1, column=4)
```

```
finalvalue1 = int(csv_file[lines][4])
```

```
ygrasia_katastash = Label(root, text=finalvalue1, bg="grey").grid(row=2, column=4)
```



```

finalvalue2 = int(csv_file[lines][7])

fwteinothta_katastash = Label(root, text=finalvalue2, bg="grey").grid(row=4, column=4)

finalvalue3 = int(csv_file[lines][5])

eks_therm_katastash = Label(root, text=finalvalue3, bg="grey").grid(row=5, column=4)

finalvalue4 = int(csv_file[lines][6])

eks_ygr_katastash = Label(root, text=finalvalue4, bg="grey").grid(row=6, column=4)

finalvalue5 = int(csv_file[lines][8])

soil_katastash = Label(root, text=finalvalue5, bg="grey").grid(row=3, column=4)

finalvalue6 = int(csv_file[lines][9])

tank_katastash = Label(root, text=finalvalue6, bg="grey").grid(row=12, column=4)

```

Η συνάρτηση `allagh_katastashs` είναι υπεύθυνη για την αποστολή στο arduino της αντίστοιχης εντολής ώστε να πραγματοποιηθούν οι αλλαγές στον ηλεκτρομηχανολογικό εξοπλισμό στο εσωτερικό του θερμοκηπίου.

```
def allagh_katastashs():
```

```

    file = open("/home/pi/data.csv")

    csv_file = list(csv.reader(file))

    reader = csv.reader(file)

    lines= len(list(reader))

    lines=lines-1

    lines=int(lines)

    finalvalue = int(csv_file[lines][3])

    finalvalue5 = int(csv_file[lines][8])

    finalvalue7 = csv_file[lines-1][2]

    finalvalue8 = csv_file[lines][2]

    finalvalue2 = int(csv_file[lines][7])

```

Το πρόγραμμα ελέγχει αν έχει επιλεγθεί η χειροκίνητη λειτουργία. Αν ο χρήστης έχει επιλέξει την αυτόματη λειτουργία τότε ελέγχει την τιμή που έχει καταχωρηθεί από τον χρήστη στο πεδίο της υγρασίας. Αν η τιμή που έχει καταχωρήσει ο χρήσης είναι μεγαλύτερη από την τιμή της υφιστάμενης υγρασίας τότε δίνει εντολή για την έναρξη του ποτίσματος έως ότου η υφιστάμενη τιμή ξεπεράσει την τιμή που έδωσε ο χρήστης.

```

if cb1.get()==0:
    ygr_set1=h.get()
    if len(ygr_set1)==0:
        offnero()
        print('potisma off')
        ygrasia_setpoint = Label(root, text='    ', bg="grey").grid(row=3, column=5)
    else:
        ygr_set=float(h.get())
        if ygr_set>=0:
            ygrasia_setpoint = Label(root, text=ygr_set, bg="grey").grid(row=3, column=5)
        if finalvalue5<ygr_set:
            onnero()
            print('potisma on')
        else:
            offnero()
            print('potisma off')
    fws_set=f.get()

```

Αν ο χρήστης έχει επιλέξει την αυτόματη λειτουργία τότε ελέγχει την τιμή που έχει καταχωρηθεί από τον χρήστη στο πεδίο της φωτεινότητας. Αν η τιμή που έχει καταχωρήσει ο χρήστης είναι μεγαλύτερη από την τιμή της υφιστάμενης φωτεινότητας τότε καλεί την συνάρτηση για να δώσει εντολή για το άναμμα της λάμπας.

```

if len(fws_set)==0:
    fwteinothta_setpoint = Label(root, text='    ', bg="grey").grid(row=4, column=5)
    print('fws off')
    offfws()
else:
    fws_set=float(f.get())
    if fws_set>=0:
        fwteinothta_setpoint = Label(root, text=fws_set, bg="grey").grid(row=4, column=5)

```

```
if finalvalue2<fws_set:
```

```
    onfws()
```

```
    print('fws on')
```

```
    th_set1=t.get()
```

Αν ο χρήστης έχει καταχωρήσει κάποια τιμή στο κενό της εισαγωγής θερμοκρασίας τότε αν αυτή είναι μικρότερη από την υφιστάμενη θερμοκρασία την εμφανίζει στην στήλη setpoint δίνει εντολή στο arduino για να ανοίξει το παράθυρο και να ξεκινήσει η λειτουργία του ανεμιστήρα. Η ταχύτητα του ανεμιστήρα όπως έχουμε πει υπολογίζεται από p ελεγκτή.

```
if len(th_set1)==0:
```

```
    anemisthras_katastash = Label(root, text="Off", bg="grey").grid(row=7, column=4)
```

```
    thermokrasia_setpoint = Label(root, text='    ', bg="grey").grid(row=1, column=5)
```

```
    print('parathyro off')
```

```
    parathyrooff()
```

```
    fanspeed=0
```

```
    print(" O anemisthras einai kleistos ", fanspeed)
```

```
    offanemisthras()
```

```
else:
```

```
    th_set1=float(t.get())
```

```
    if th_set1>=0:
```

```
        thermokrasia_setpoint = Label(root, text=th_set1, bg="grey").grid(row=1, column=5)
```

```
    if finalvalue<=th_set1-2:
```

```
        print('parathyro off')
```

```
        parathyrooff()
```

```
    else:
```

```
        print ('parathyro on')
```

```
        parathyroon()
```

```
    if finalvalue<=th_set1-2:
```

```
        fanspeed=0
```

```
        print(" O anemisthras einai kleistos ", fanspeed)
```

```
offanemisthras()
```

```
else:
```

```
anemisthras_katastash = Label(root, text="On", bg="grey").grid(row=7, column=4)
```

Υπολογίζει το σφάλμα μεταξύ της επιθυμητής και της υφιστάμενης θερμοκρασίας του θερμοκηπίου. Υπολογίζει την ταχύτητα του ανεμιστήρα στην μεταβλητή fanspeed.

```
s= th_set1 = th_set1 - 3
```

```
E= finalvalue - th_set1
```

```
kp=20
```

```
fanspeed=520*((kp*E)/((s*s)+1+(kp*E)))
```

```
print('P: ', fanspeed)
```

```
if fanspeed<=25:
```

```
    fanspeed=25
```

```
    print('gia asfaleia ', fanspeed)
```

```
if(fanspeed < 0):
```

```
    fanspeed = 0
```

```
    print('gia asfaleia ', fanspeed)
```

```
if(fanspeed > 128) :
```

```
    fanspeed = 128
```

```
    print('gia asfaleia ', fanspeed)
```

```
print(" Η taxytha toy anemisthra einai: ", fanspeed)
```

```
senv4=fanspeed
```

Μετατρέπει τον ακέραιο αριθμό σε string και στέλνει τον αριθμό αυτό στο arduino.

```
angle_value_list = [str(senv4)]
```

```
send_string = ','.join(angle_value_list)
```

```
send_string += "\n"
```

```
ser.write(send_string.encode('utf-8'))
```

```
time.sleep(1)
```

```
root.after(2000,user_interface)
```

```
root.update()
```

```
root.after(4,allagh_katastashs)
```

Η συνάρτηση `switchState` ενεργοποιείται όταν ο χρήστης κάνει εναλλαγή μεταξύ χειροκίνητης και αυτόματης λειτουργίας.

```
def switchState():
```

```
    if cb1.get() == 1:
```

```
        cb=cb1.get()
```

```
        disp_Lb.config(text='ON')
```

```
        print(cb1.get())
```

```
    elif cb1.get() == 0:
```

```
        disp_Lb.config(text='OFF')
```

```
        cb=cb1.get()
```

```
        print(cb1.get())
```

```
    else:
```

```
        disp_Lb.config(text='error!')
```

Η παρακάτω συνάρτηση ενεργοποιείται όταν ο χρήστης επιλέξει να εξάγει το διάγραμμα υγρασίας εδάφους.

```
def soildiagram():
```

```
    Διαβάζει το αρχείο data1.csv
```

```
    seira = []
```

```
    csv_file = csv.reader(open('/home/pi/data1.csv', "r"), delimiter=",")
```

Εάν η γραμμή που διαβάζει το πρόγραμμα είναι ίση με την μεταβλητή `number` στην οποία έχει καταχωρηθεί η τιμή της ημερομηνίας τότε το πρόγραμμα δημιουργεί μια λίστα και κάθε φορά προσθέτει στο τέλος της λίστας τον αριθμό του `id`. Έπειτα εκτυπώνει την λίστα.

```
number = m.get()
```

```
for row in csv_file:
```

```
    if number == row[1]:
```

```
        seira.append(row[0])
```

```
print(seira)
```

```
time.sleep(0.3)
```

Εάν η γραμμή που διαβάζει το πρόγραμμα είναι ίση με την μεταβλητή number στην οποία έχει καταχωρηθεί η τιμή της ημερομηνίας έως ότου θέλουμε να εμφανίσουμε την τιμή της υγρασίας εδάφους τότε το πρόγραμμα κάθε φορά προσθέτει στο τέλος της λίστας τον αριθμό του id. Έπειτα εκτυπώνει την λίστα

```
csv_file = csv.reader(open('/home/pi/data1.csv', "r"), delimiter=",")
```

```
number = z.get()
```

```
for row in csv_file:
```

```
    if number == row[1]:
```

```
        seira.append(row[0])
```

```
print(seira)
```

```
print ("Original list is : " + str(seira))
```

```
for i in range(0, len(seira)):
```

```
    seira[i] = int(seira[i])
```

```
print ("Modified list is : " + str(seira))
```

Εκτυπώνει το μέγιστο και το ελάχιστο στοιχείο της λίστας

```
min_ele,max_ele=seira[0],seira[0]
```

```
for i in range(1,len(seira)):
```

```
    if seira[i]<min_ele:
```

```
        min_ele=seira[i]
```

```
    if seira[i]>max_ele:
```

```
        max_ele=seira[i]
```

```
print('Minimum Element in the list',seira,'is',min_ele)
```

```
print('Maximum Element in the list',seira,'is',max_ele)
```

```
newdata = []
```

```
data = pd.read_csv("/home/pi/data1.csv", index_col="id")
```

```
for i in range(min_ele,max_ele+1,1):
```

```
    rows = data.loc[i]
```

```
    newdata.append(rows)
```

```
df = pd.DataFrame(data=newdata)
df.to_csv (r'diagrammasoil.csv', index = True, header=False)
s = df.to_csv()
print(s)
```

Δημιουργεί ένα καινούργιο αρχείο το οποίο περιέχει την ημερομηνία και τον χρόνο σε μια στήλη και στην άλλη στήλη την υγρασία του εδάφους που επικρατεί στο εσωτερικό του θερμοκηπίου.

with open("diagrammasoil.csv", "r") as source, open("result_soil1.csv", "w") as result:

```
rdr = csv.reader(source)
wtr = csv.writer(result)
writer = csv.DictWriter(result, fieldnames=["time", "soilmoisture"])
writer.writeheader()
for r in rdr:
    wtr.writerow([r[1] + ' ' + r[2]] + [r[8]])
plt.style.use('seaborn-whitegrid')
x_vals = []
y_vals = []
index = count()
```

Η συνάρτηση animate εμφανίζει το διάγραμμα της υγρασίας στο εσωτερικό του θερμοκηπίου σε σχέση με τον χρόνο

```
def animate(i):
    data = pd.read_csv('result_soil1.csv')
    x= data ['time']
    y1 = data ['soilmoisture']
    plt.cla()
    plt.plot(x, y1, label='Υγρασία εδάφους (%)',linewidth=3,marker='o', linestyle='--', color='b')
    plt.xlabel('Χρόνος')
    plt.xticks(rotation=90)
    plt.legend(loc='upper left')
```

```

plt.tight_layout()

ani = FuncAnimation(plt.gcf(), animate, interval=1000)

plt.tight_layout()

plt.show()

if __name__ == '__main__':

    root = Tk()

    th_set=0

    t=StringVar()

    h=StringVar()

    m=StringVar()

    z=StringVar()

    x=StringVar()

    f=StringVar()

    var1 = StringVar()

    var1.set('OFF')

    root.configure(background='grey')

    root.title("Θερμοκρήπιο")

    root.geometry("1000x1000")

    root.resizable(width = True, height = True)

    Label(root, text="Στοιχεία", width=12, bg="grey", font = "Times 10 bold").grid(row=0, column=0,
sticky=W,)

    Label(root, text="Εισαγωγή",width=8, bg="grey", font = "Times 10 bold") .grid(row=0, column=1,
sticky=W,)

    Label(root, text="Αποστολή",width=12, bg="grey", font = "Times 10 bold") .grid(row=0,
column=3, sticky=W,)

    Label(root, text=" Setpoint",width=12, bg="grey", font = "Times 10 bold") .grid(row=0, column=5,
sticky=W,)

    Label(root, text=" Κατάσταση",width=12, bg="grey", font = "Times 10 bold") .grid(row=0,
column=4, sticky=W,)

```



```

Label(root, text=" M. Μέτρησης",width=12, bg="grey", font = "Times 10 bold") .grid(row=0,
column=2, sticky=W,)

button_quit = Button(root, text="Exit", command=root.destroy, bd=5).grid(row=0, column=6)

thermokrasia = Label(root, text=" Εσ. Θερμοκρασία", bg="grey", font = "Times 10
bold").grid(row=1, column=0)

thermokrasia_field = Entry(root, text=t, bd=5, width=8 ).grid(row=1, column=1, ipadx="6",
sticky=W)

thermokrasia_monada = Label(root, text="οC", bg="grey").grid(row=1, column=2)

deksamenh_monada = Label(root, text="lt", bg="grey").grid(row=12, column=2)

fwteinothta_monada = Label(root, text="%", bg="grey").grid(row=4, column=2)

fwteinothta_field = Entry(root, text=f, bd=5, width=8 ).grid(row=4, column=1, ipadx="6",
sticky=W)

es_ygrasia_aera = Label(root, text="Εσ. Υγρασία αέρα", bg="grey", font = "Times 10
bold").grid(row=2, column=0)

es_ygrasia_aera_monades = Label(root, text="%", bg="grey").grid(row=2, column=2)

ygrasia = Label(root, text="Υγρασία εδάφους", bg="grey", font = "Times 10 bold").grid(row=3,
column=0)

ygrasia_field = Entry(root, text=h, bd=5, width=8).grid(row=3, column=1, ipadx="6", sticky=W)

ygrasia_monada = Label(root, text="%", bg="grey").grid(row=3, column=2)

fwteinothta = Label(root, text="Φωτεινότητα", bg="grey", font = "Times 10 bold").grid(row=4,
column=0)

eks_thermokrasia = Label(root, text=" Εξ θερμοκρασία", bg="grey", font = "Times 10
bold").grid(row=5, column=0)

eks_thermokrasia_monades = Label(root, text="οC", bg="grey").grid(row=5, column=2)

eks_ygrasia = Label(root, text="Εξ. Υγρασία αέρα", bg="grey", font = "Times 10
bold").grid(row=6, column=0)

eks_ygrasia_monades = Label(root, text="%", bg="grey").grid(row=6, column=2)

anemisthras = Label(root, text="Ανεμιστήρας", bg="grey", font = "Times 10 bold").grid(row=7,
column=0)

Button(root, text="On", width=6,command=onanemisthras, bd=5 ).grid(row=7,column=1,
sticky=W)

```

```

Button(root,text="Off", width=6, command=offanemisthras, bd=5 ).grid(row=7, column=2,
sticky=W)

nero = Label(root, text="Νερό", bg="grey", font = "Times 10 bold").grid(row=8, column=0)

Button(root, text="On", width=6, command=onnero,bd=5 ).grid(row=8,column=1, sticky=W)

Button(root,text="Off", width=6, command=offnero,bd=5 ).grid(row=8, column=2, sticky=W)

fwteinothta = Label(root, text="Φώς", bg="grey", font = "Times 10 bold").grid(row=9, column=0)

Button(root, text="On", width=6,command=onfws,bd=5 ).grid(row=9,column=1, sticky=W)

Button(root,text="Off", width=6, command=offfws,bd=5 ).grid(row=9, column=2, sticky=W)

parathyro=Label(root,text="Παράθυρο",bg="grey",font = "Times 10 bold").grid(row=10,
column=0)

Button(root, text="On", width=6,command=parathyroon,bd=5 ).grid(row=10,column=1, sticky=W)

Button(root,text="Off", width=6, command=parathyrooff,bd=5 ).grid(row=10, column=2,
sticky=W)

deksameni=Label(root,text="Δεξαμενή",bg="grey",font = "Times 10 bold").grid(row=12,
column=0)

Button(root, text = 'Διάγ Εσ Θερμοκρασιών', width = 20, command = diagramma_therm, bd=5
).grid(row=12, column=6)

Button(root, text = 'Διάγ Εσ Υγρασιών Αέρα', width = 20, command = diagramma_ygr, bd=5
).grid(row=13, column=6)

Button(root, text ='Εύρεση φωτογραφίας',width = 20, bd=5, command=open_img).grid(row = 14,
column = 6)

Label(root, text="Ημ/νία από", bg="grey", font = "Times 10 bold").grid(row=13, column=0)

hmeromhniaapo_field = Entry(root, text=m, bd=5, width=8 ).grid(row=13, column=1, ipadx="6",
sticky=W)

Button(root, text="Submit", fg="Black", bg="Red" , command=hmer, bd=5).grid(row=13,
column=3)

Label(root, text="Ημ/νία έως", bg="grey" , font = "Times 10 bold").grid(row=14, column=0)

hmeromhniaews_field = Entry(root, text=z, bd=5 , width=8).grid(row=14, column=1, ipadx="6",
sticky=W)

Button(root, text="Submit", fg="Black", bg="Red" , command=getvalue1, bd=5).grid(row=14,
column=3)

```

Label(root, text="1) Στο πεδίο εισαγωγής θερμοκρασίας, υγρασίας \n & χρόνος θα πρέπει να εισάγονται με την μορφή πχ 8.0 \n 2) Στο πεδίο εισαγωγής ημερομηνίας θα πρέπει \n να εισάγονται οι ημερομηνίες με την μορφή πχ 8-4-2015 ",width=120, bg="grey") .grid(row=0, column=7, sticky=W,)

```
switch_on = PhotoImage(width=50, height=50)
```

```
switch_off = PhotoImage(width=50, height=50)
```

```
switch_on.put(("green",), to=(0, 0, 23,23))
```

```
switch_off.put(("red",), to=(24, 0, 47, 23))
```

```
cb1 = IntVar()
```

```
Label(root, text="Manual", bg="grey" , font = "Times 10 bold").grid(row=8, column=6)
```

```
Checkbutton(root, image=switch_off, selectimage=switch_on, onvalue=1, offvalue=0,  
variable=cb1, indicatoron=False, command=switchState).grid(row = 9, column = 6)
```

```
disp_Lb = Label(root)
```

```
disp_Lb.grid(row = 10, column = 6)
```

```
Button(root, text ='Διάγ υγρασίας εδάφους', command=soildiagram, width = 20, bd=5).grid(row =  
11, column = 6)
```

```
root.after(1, allagh_katastashs)
```

```
root.mainloop()
```

9.5. Το πρόγραμμα λήψης φωτογραφιών

Το παρακάτω πρόγραμμα είναι υπεύθυνο για την λήψη της φωτογραφίας. Η λήψη της φωτογραφίας πραγματοποιείται κάθε μέρα στις 14:00 μμ.

Πιο συγκεκριμένα:

Κατά την εκκίνηση εισάγονται οι βιβλιοθήκες

```
import cv2
```

```
import schedule
```

```
import time
```

```
from datetime import date
```

```
from datetime import datetime
```

```
vid = cv2.VideoCapture(0)
```

```
today=date.today()
```

Στην μεταβλητή d1 εισάγεται η υφιστάμενη ημερομηνία

```
d1 = today.strftime("%d-%m-%Y")
```

```
now = datetime.now()
```

Στην μεταβλητή t_string εισάγεται η υφιστάμενη ώρα

```
t_string = now.strftime("%H:%M:%S")
```

Όταν καλείται η συνάρτηση take_picture η κάμερα βγάζει φωτογραφία. Η φωτογραφία αποθηκεύεται με την μορφή png και το όνομα της είναι photo και ακολουθεί η ημερομηνία και η ώρα που λήφθηκε η φωτογραφία

```
def take_picture():
```

Άνοιγμα της κάμερας

```
ret, frame = vid.read()
```

Εμφάνιση σε παράθυρο με το όνομα Zinas Greenhouse την εικόνα του θερμοκηπίου

```
cv2.imshow('Zinas Greenhouse',frame)
```

```
img_name = "photo_{}_{}.png".format(d1,t_string)
```

```
cv2.imwrite(img_name, frame)
```

```
print("photo_{}_{} written!".format(d1,t_string))
```

```
vid.release()
```

```
schedule.every().day.at("14:00").do(take_picture)
```

Το τμήμα του κώδικα που περικλείεται μέσα στην while εκτελείται κάθε φορά

```
while(True):
```

Μόλις πατηθεί το q πλήκτρο κλείνει το παράθυρο και σταματάει το πρόγραμμα

```
if cv2.waitKey(1) & 0xFF == ord('q'):
```

```
break
```

```
schedule.run_pending()
```

```
cv2.destroyAllWindows()
```

9.6. Το πρόγραμμα ενεργοποίησης του μηχανολογικού εξοπλισμού

Το πρόγραμμα αυτό φορτώνεται στο arduino το οποίο παίρνει εντολές από το Raspberry Pi για την ενεργοποίηση του ποτίσματος, του ανεμιστήρα για την μείωση της θερμοκρασία στο εσωτερικό του θερμοκηπίου και την έναρξη της κίνησης του κινητήρα του παραθύρου .

Αναλυτικότερα καταχωρεί:

- το relay της λάμπας στο pin 8
- το relay της αντλίας στο pin 7
- το relay της ηλεκτροβάνας στο pin 12 του arduino.
- το AN2 της MDDS10 στο pin 5 του arduino
- το DiG2 της MDDS10 στο pin 11 του arduino

```
#define relayfws 8
```

```
#define relayantlia 7
```

```
#define relaybana 12
```

```
int pwm_motor_right=5;
```

```
int dir_motor_right=11 ;
```

Συμπεριλαμβάνει την βιβλιοθήκη Servo.h στο πρόγραμμα

```
#include <Servo.h>
```

Θέτει στο servo το ψευδώνυμο myservo και λαμβάνει την θέση 0.

```
Servo myservo;
```

```
int servo_position = 0;
```

Θέτει στην μεταβλητή ser την τιμή 259 μόνο για την πρώτη φορά που θα τροφοδοτηθεί με ρεύμα το arduino

```
int ser=259;
```

Το τμήμα του κώδικα που βρίσκεται στο τμήμα void setup ενεργοποιείται μια μόνο φορά

```
void setup() {
```

Έναρξη της serial επικοινωνίας στα 9600 bit/second

```
Serial.begin(9600);
```

Καταχωρεί τα ρελέ και την πλακέτα MDDS10 ως έξοδοι.

```
pinMode(5,OUTPUT);
```

```
pinMode(11,OUTPUT);
```

```
pinMode(relayfws, OUTPUT);
```

```
pinMode(relayantlia, OUTPUT);
```

```
pinMode(relaybana, OUTPUT);
```

Επειδή τα ρελέ ενεργοποιούνται στην κατάσταση LOW τα απενεργοποιούμε κατά την εκκίνηση του προγράμματος

```
digitalWrite(relayfws, HIGH);
```

```
digitalWrite(relayantlia, HIGH);
```

```
digitalWrite(relaybana, HIGH);
```

Καταχωρεί το servo στο pin 10

```
myservo.attach (10,600,2300);
```

Μηδενίζει την ταχύτητα του ανεμιστήρα στην διεύθυνση High

```
analogWrite(pwm_motor_right,0);
```

```
digitalWrite(dir_motor_right,HIGH);}
```

Το τμήμα του κώδικα που βρίσκεται στο τμήμα void loop εκτελείται άπειρες φορές και είναι υπεύθυνο για την λειτουργία του θερμοκηπίου ανάλογα με τις εντολές που λαμβάνει από το Raspberry pi στην serial επικοινωνία του

```
void loop() {
```

```
if (Serial.available(>0) {
```

```
ser=Serial.parseInt();
```

Αν κατά την επικοινωνία του arduino με το Raspberry διαβάσει την τιμή 259 εκτυπώνει την τιμή αυτή και δίνει εντολή στη βάνα να ανοίξει. Περιμένει για 500 msec μέχρι να ολοκληρωθεί η εντολή. Έπειτα ενεργοποιεί την αντλία και περιμένει 500 msec μέχρι να ενεργοποιηθεί.

```
if (ser==259) {
```

```
Serial.print(ser);
```

```
Serial.println(" nero on");
```

```
digitalWrite(relaybana, LOW);
```

```
delay(500);
```

```
digitalWrite(relayantlia, LOW);
```

```
delay(500); }
```

Αν η μεταβλητή ser λάβει τιμή από 0 έως 128 ενεργοποιεί τον ανεμιστήρα. Ουσιαστικά αυτή αποτελεί και την ταχύτητα του ανεμιστήρα. Εκτυπώνει το μήνυμα fan on και την ταχύτητα του ανεμιστήρα. Το πρόγραμμα καθυστερεί 500msec έως ότου αλλάξει ταχύτητα ο ανεμιστήρας

```
else if ( ser>20 && ser<=128){  
  
Serial.print(ser);  
  
Serial.println(" fan on")  
  
analogWrite(pwm_motor_right,ser);  
  
digitalWrite(dir_motor_right,HIGH);  
  
delay(500);  
  
Serial.print(ser);  
  
Serial.println(" Speed"); }
```

Αν η μεταβλητή ser λάβει την τιμή 256 δίνει εντολή να ανάψει το φως και εκτυπώνει την τιμή της μεταβλητής ser

```
else if (ser==256) {  
  
digitalWrite(relayfws, LOW);  
  
Serial.print(ser);  
  
Serial.println(" fws on ");  
  
delay(500);}
```

Αν η μεταβλητή ser λάβει την τιμή 260 δίνει εντολή στο παράθυρο να κλείσει και εκτυπώνει parathyro off. Το πρόγραμμα περιμένει 500 msec μέχρι να ολοκληρωθεί η κίνηση από τον κινητήρα mg996

```
else if (ser==260) {  
  
Serial.print(ser);  
  
Serial.println(" parathyro off ");  
  
myservo.write(90);  
  
delay(500);}
```

Αν η μεταβλητή ser πάρει την τιμή 257 δίνει την εντολή να σβήσει το φως. Το πρόγραμμα περιμένει 500msec μέχρι να ολοκληρωθεί η εντολή

```
else if (ser==257) {
```

```
  Serial.print(ser);
```

```
  Serial.println(" fws off ");
```

```
  digitalWrite(relayfws, HIGH);
```

```
  delay(500); }
```

Αν η μεταβλητή ser πάρει την τιμή 261 δίνει την εντολή να ανοίξει το παράθυρο. Το πρόγραμμα περιμένει 500msec μέχρι να ολοκληρωθεί η εντολή

```
else if (ser==261) {
```

```
  Serial.print(ser);
```

```
  Serial.println(" parathyro on ");
```

```
  myservo.write(-90);
```

```
  delay(500); }
```

Αν η μεταβλητή ser πάρει την τιμή 258 δίνει την εντολή να σταματήσει το πότισμα. Το πρόγραμμα περιμένει 500msec μέχρι να ολοκληρωθεί η εντολή

```
else if (ser==258) {
```

```
  Serial.print(ser);
```

```
  Serial.println(" potisma off");
```

```
  digitalWrite(relayantlia, HIGH);
```

```
  delay(500);
```

```
  digitalWrite(relaybana, HIGH);
```

```
  delay(500); }
```

Αν η μεταβλητή ser πάρει την τιμή 262 δίνει την εντολή να σταματήσει η λειτουργία του ανεμιστήρα. Το πρόγραμμα περιμένει 500msec μέχρι να ολοκληρωθεί η εντολή

```
else if (ser==262) {
```

```
  Serial.print(ser);
```

```
  Serial.println(" fan off");
```

```
  analogWrite(pwm_motor_right,0);
```

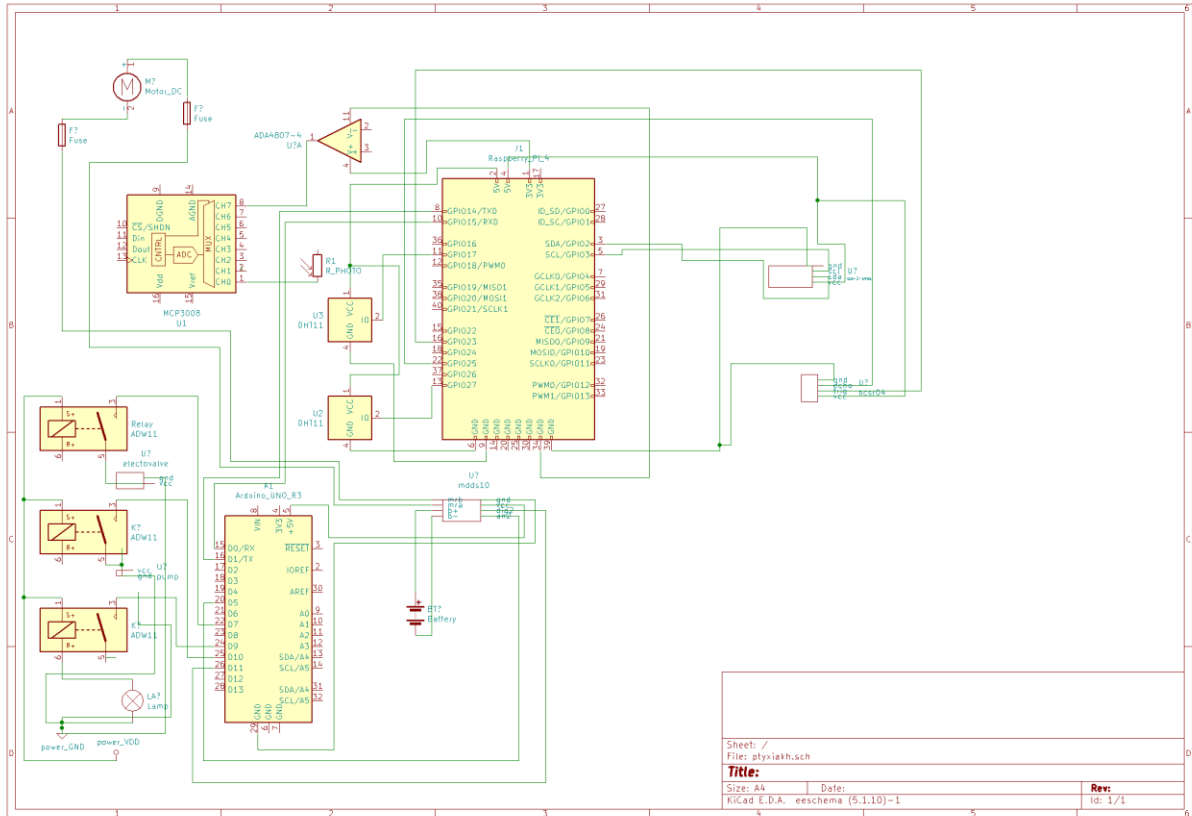
```
  digitalWrite(dir_motor_right,HIGH);
```



```
delay(500); } }
```

10. Ηλεκτρονικές συνδέσεις

Παρακάτω φαίνεται το ηλεκτρονικό σχέδιο:



Εικόνα 49: Ηλεκτρονικό σχέδιο

Παρακάτω φαίνονται οι συνδέσεις των καλωδίων

Πίνακας 25: Συνδέσεις καλωδίων

SOIL MOISTURE SENSOR	
AO	CH7 του MCP3008
VCC	3,3V του Breadboard
GND	GND του Breadboard
LCD 16X2	
GND	Pin 6 Raspberry (Ground)
VCC	Pin 4 Raspberry (VCC)
SDA	GPIO 2 Raspberry (SDA)
SCL	GPIO 3 Raspberry (SCL)
DHT11	
VCC	5 V του Breadboard

GND	GND του Breadboard
SIGNAL	GPIO 17 του Raspberry
DHT11	
VCC	5V του Breadboard
GND	GND του Breadboard
SIGNAL	GPIO 27 του Raspberry
Relay 2 channel	
VCC	5V του Breadboard
IN2	DIGITAL PIN 7 του Arduino
GND	GND του Breadboard
Relay 2 channel	
VCC	5V του Breadboard
IN1	DIGITAL PIN 10 του Arduino
IN2	DIGITAL PIN 9 του Arduino
GND	GND του Breadboard
MCP3008	
VDD	3.3 V του Breadboard
VREF	3.3 V του Breadboard
AGND	GND του Breadboard
CLK	Pin 23/SCLK του Raspberry
DOUT	Pin 21/MISO του Raspberry
DIN	Pin 19/MOSI του Raspberry
CS	Pin 24/CE0 του Raspberry
DGND	GND του Breadboard
RASPBERRY	
Pin 2 (5V)	5V του Breadboard
Pin 2 (3,3V)	3,3V του Breadboard
Arduino	
+5V	5V του Breadboard
GND	GND του Breadboard
HCSR04	
VCC	5V του Breadboard
TRIG	PIN 16 του Raspberry
GND	GND του breadboard
ECHO	PIN 22 του Raspberry
MDSS10 Smart Drive Duo -10	
GND	GND του Breadboard
+5V	+5V του Breadboard
AN2	DIGITAL PIN 5 του Arduino
DIG2	DIGITAL PIN 11 του Arduino
Ηλεκτροβάνια	
GND	GND 12V μπαταρίας
POWER	NO του Relay
Αντλία	
GND	GND 12V μπαταρίας
POWER	NO του Relay
Λάμπα 12V	
GND	GND 12V μπαταρίας
POWER	NO του Relay
Photoresistor ldr module	
AO	CN0 του MCP3008

GND	GND του Breadboard
VCC	+5V του Breadboard

11.Κοστολόγηση

Το κόστος υλοποίησης φαίνεται στον παρακάτω πίνακα.

Πίνακας 26: Το κόστος

Εξοπλισμός	Τιμή
Δεξαμενή 30 L	15,00
Ηλεκτρόδια	5,00
Γωνίες 25x25x3mm	20,00
Κουτί συναρμολόγησης	19,00
Raspberry pi 4B – 4GB	65,00
Τροφοδοτικό ρεύματος arduino	10,00
Τροφοδοτικό ρεύματος raspberry	6,00
Arduino Uno	11,00
Καλώδια	15,00
Αισθητήρας υπερήχων HC-SR04	8,00
DHT11 Αισθητήρας θερμοκρασίας & υγρασίας (2 τεμ)	6,00
Αισθητήρας υγρασίας εδάφους	4,00
LCD οθόνη με I2C 16x2	14,00
Αισθητήρας φωτεινότητας	5,00
Βίδες, ροδέλες, παξιμάδια διαφόρων διατομών και διαμετρημάτων	15,00
Αντλία 12V	14,00
Σφικτήρες (4 τεμ)	4,00
Βάνα ½ αρσενικό θηλυκό	3,00
Συστολή 1' x ½ (2 τεμ)	2,00
Τάπα ½	1,00
Servo motor MG996R	12,00
Εύκαμπτο λάστιχο Φ20mm	6,00
Plexi glass πάχους 4mm	60,00
Ηλεκτροβάνα 1'	18,00
Κάμερα USB	25,00
Ντίζα M8	3,00
Ντίζα M4	2,00
MDDS10 Smart Drive Duo-10	150,00
Φορτιστής μπαταριών	20,00
Ψηφιακό οπτικό στροφόμετρο leizer	22,00
Σιλκόνη διάφανη	7,00
Αποστάτης πολυαμιδίου	2,00
Δεματικά καλωδίων	1,00
Μονωτική ταινία teflon	1,00
Φτερωτή ανεμιστήρα	6,00
Κινητήρας ανεμιστήρα 15000 rpm	15,00
Κινητήρας αναρρόφησης αέρα 140 mm	16,00
Θήκη & ασφάλεια 10 A	5,00
Αντιστάσεις 1 & 2 & 10 k ohm	1,00

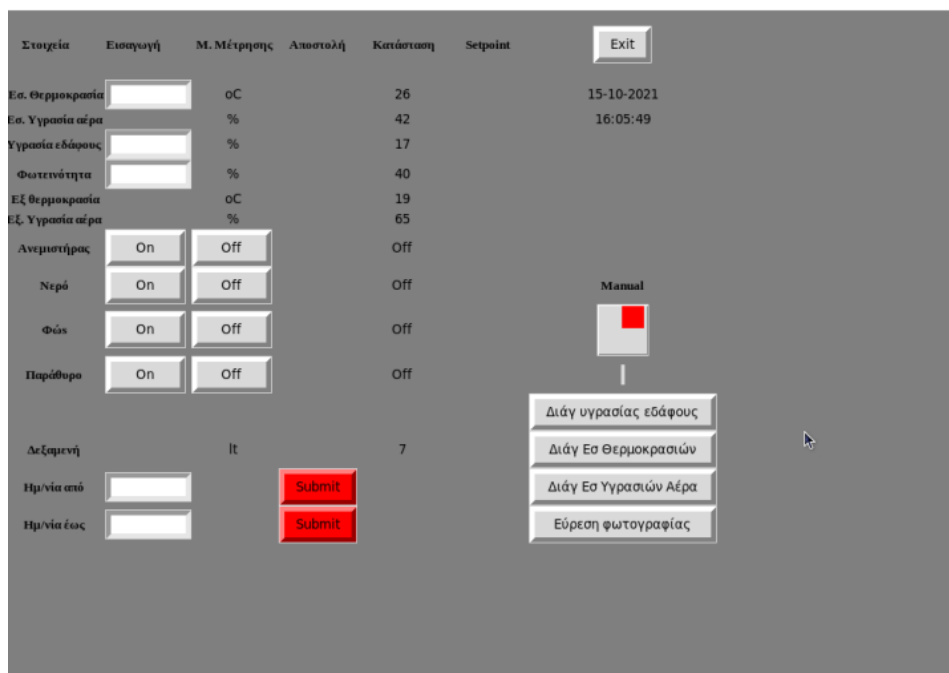
Λάμπα 12V	5,00
Αντάπτορας λάμπας	1,00
Κάρτα μνήμης 32 gb	10,00
Ασφάλειες 10 ^A	2,00
MCP3008	5,00
Ασφαλειοθήκες	8,00
DC-DC Boost Converter 10-32V to 12-35V 6A Step Up (150W)	12,00
DC-DC STEP-DOWN	5,00
Ηλεκτροβάνα	35,00
Κάρτα μνήμης 32gb	10,00
Τροφοδοτικό ρεύματος 40A	40,00
Ρελέ	10,00
Σύνολο	757,00€

12.Αξιολόγηση των αποτελεσμάτων

Παρακάτω φαίνεται η κατασκευή



Εικόνα 50: Η κατασκευή

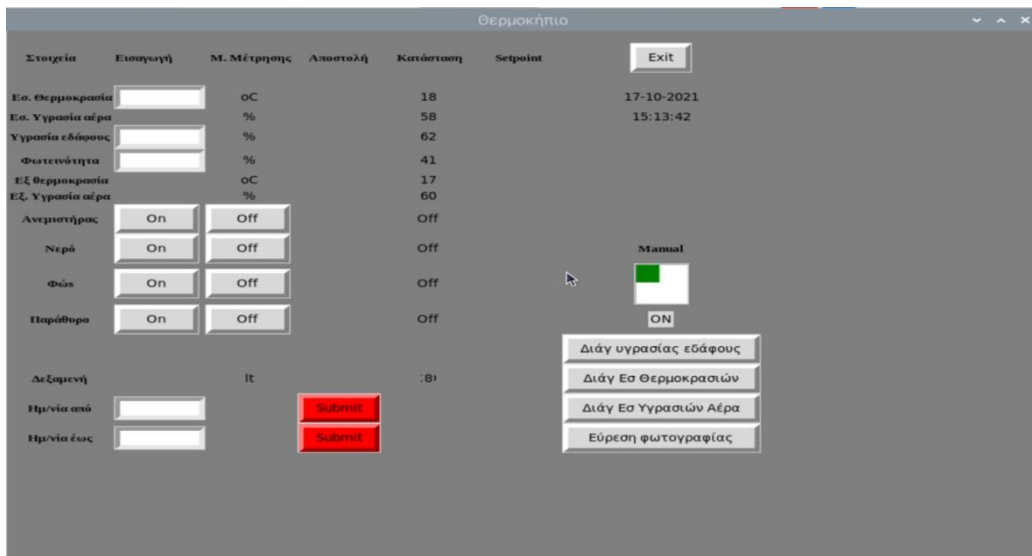


Εικόνα 51: Περιβάλλον χειρισμού του χρήστη

Στην παραπάνω εικόνα φαίνεται το περιβάλλον χειρισμού του χρήστη. Περιλαμβάνει στην στήλη εισαγωγή τα κενά όπου ο χρήστης αλληλεπιδρά με το θερμοκήπιο βάζοντας την απαραίτητη υγρασία εδάφους, φωτεινότητα και εσωτερική θερμοκρασία στην οποία θα πρέπει να φτάσει το εσωτερικό του θερμοκηπίου. Στην στήλη κατάσταση φαίνεται η υφιστάμενη κατάσταση του θερμοκηπίου δηλαδή οι τιμές της εσωτερικής θερμοκρασίας και υγρασίας αέρα, υγρασίας εδάφους, φωτεινότητας, εξωτερικής υγρασίας αέρα και εδάφους, την στάθμη της δεξαμενής καθώς και την κατάσταση λειτουργίας δηλαδή αν είναι ανοιχτό ή σβηστό το φως, αν το πότισμα και ο ανεμιστήρας είναι σε λειτουργία και αν το παράθυρο είναι ανοιχτό ή όχι. Στην 3η στήλη φαίνονται οι μονάδες μέτρησης των μεγεθών που εμφανίζονται τόσο στην στήλη της εισαγωγής όσο και στην στήλη κατάσταση.

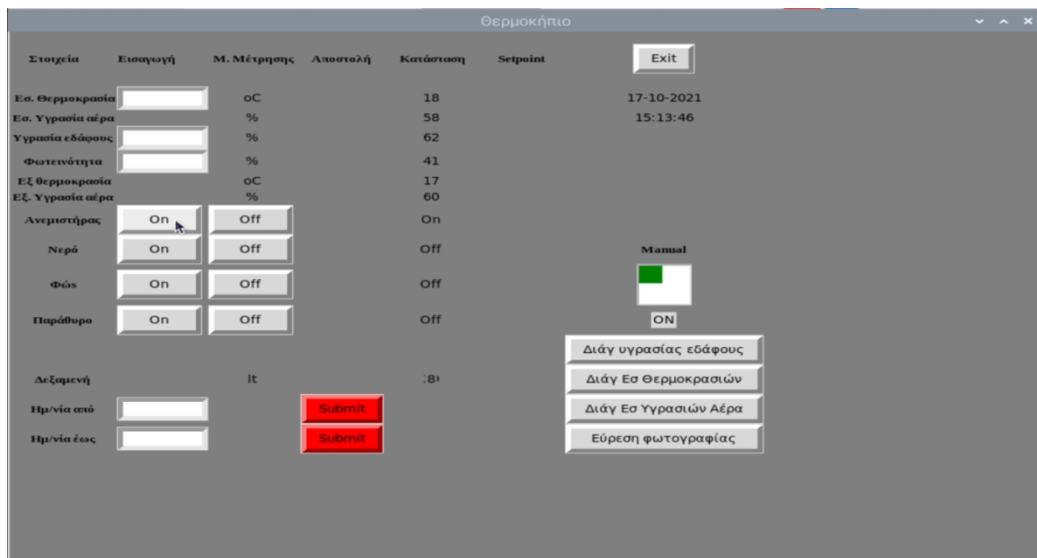
Αναλυτικότερα ο χρήστης επιλέγει αν θα ενεργοποιήσει τον ανεμιστήρα, το πότισμα, το φως και το παράθυρο χειροκίνητα ή αυτόματα.

Σε περίπτωση χειροκίνητης λειτουργίας ο χρήστης πατάει το κουμπί manual και αυτό γίνεται πράσινο όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 52: Χειροκίνητη λειτουργία

Έπειτα στα αριστερά υπάρχουν τα κουμπιά on / off για την χειροκίνητη λειτουργία. Όταν π.χ. ο χρήστης πατήσει το κουμπί on του ανεμιστήρα ο ανεμιστήρας ενεργοποιείται με πλήρη ταχύτητα και η ένδειξη on φαίνεται στην στήλη της κατάστασης όπως φαίνεται στην παρακάτω εικόνα. Το ίδιο συμβαίνει και με τις υπόλοιπες διατάξεις.

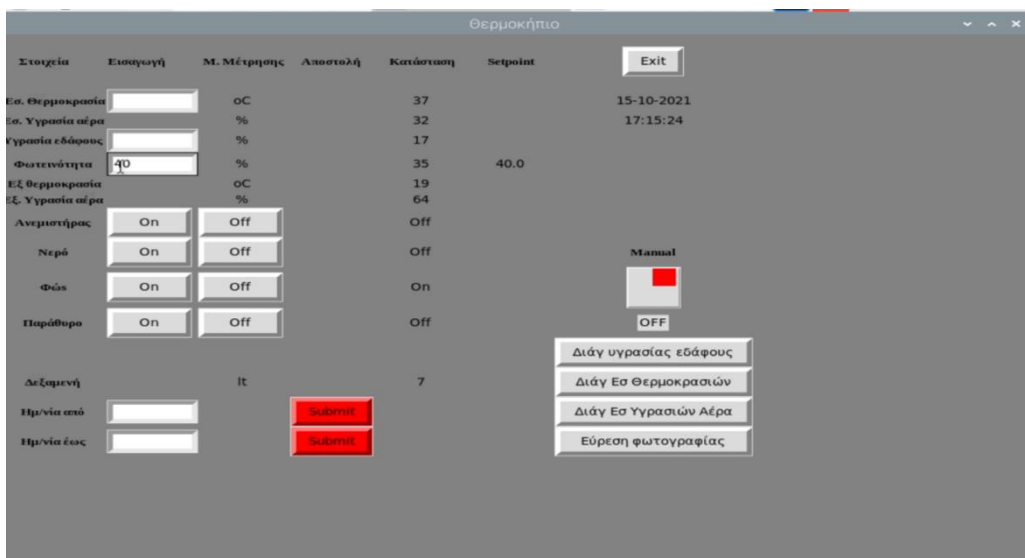


Εικόνα 53: Χειροκίνητη ενεργοποίηση ανεμιστήρα

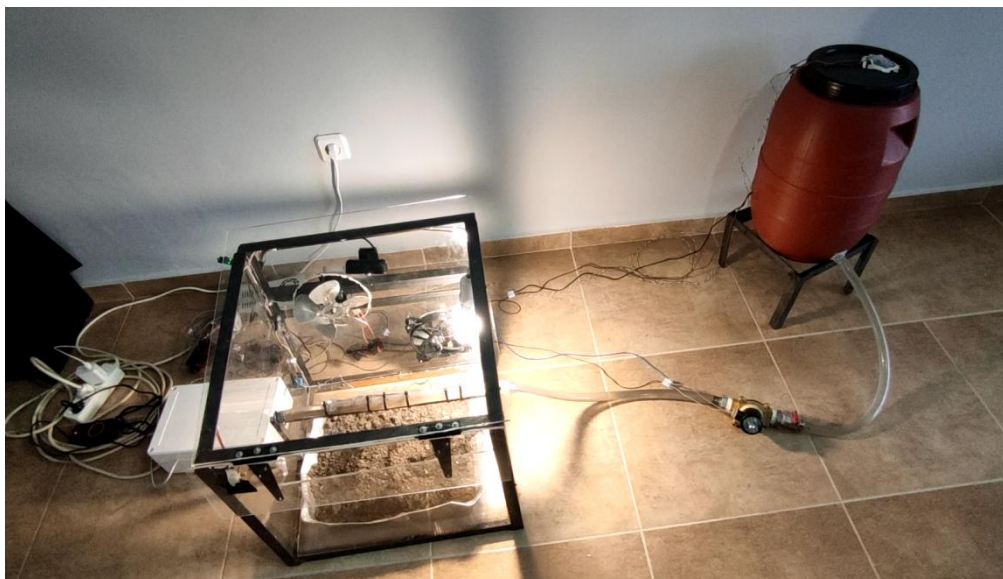


Εικόνα 54: Χειροκίνητη ενεργοποίηση του ανεμιστήρα

Η αυτόματη λειτουργία του θερμοκηπίου ενεργοποιείται μόνο όταν το κουμπί manual είναι στο off και ο χρήστης έχει εισάγει κάποια τιμή στην στήλη εισαγωγής στο πεδίο εσωτερική θερμοκρασία ή υγρασία εδάφους ή φωτεινότητα. Πιο συγκεκριμένα όπως φαίνεται στην παρακάτω εικόνα αν ο χρήστης εισάγει μια τιμή στην στήλη εισαγωγή στο πεδίο της φωτεινότητας και αυτή είναι μεγαλύτερη από την υφιστάμενη τιμή της φωτεινότητας στην στήλη κατάσταση η λάμπα στο εσωτερικό του θερμοκηπίου ανάβει και η φωτεινότητα αυξάνεται. Η τιμή αυτή καταχωρείται στην στήλη Setpoint στο ύψος της φωτεινότητας. Η λάμπα θα παραμείνει σβηστέ αν η τιμή που καταχωρεί ο χρήστης είναι μικρότερη από την υφιστάμενη τιμή της φωτεινότητας. Στο σημείο αυτό θα πρέπει να σημειωθεί ότι επειδή δεν υπάρχει κάποιος ελεγκτής στην διάταξη οι τιμές της φωτεινότητας συνήθως είναι πολύ μεγαλύτερες μετά την ενεργοποίηση της λάμπας.



Εικόνα 55: Αυτόματη λειτουργία αύξησης της φωτεινότητας



Εικόνα 56: Αύξηση της φωτεινότητας

Επίσης αν ο χρήστης εισάγει μια τιμή στο πεδίο της υγρασίας εδάφους ενεργοποιείται το πότισμα μέχρι η υφιστάμενη τιμή της υγρασίας να ξεπεράσει την τιμή που έθεσε ο χρήστης.

Στοιχεία	Εισαγωγή	Μ. Μέτρησης	Αποστολή	Κατάσταση	Setpoint
Εσ. Θερμοκρασία	<input type="text"/>	οC		20	22-10-2021
Εσ. Υγρασία αέρα	<input type="text"/>	%		71	16:24:12
Υγρασία εδάφους	<input type="text" value="25"/>	%		15	
Φωτεινότητα	<input type="text"/>	%		35	
Εξ θερμοκρασία	<input type="text"/>	οC		19	
Εξ Υγρασία αέρα	<input type="text"/>	%		84	
Ανεμιστήρας	<input type="button" value="On"/>	<input type="button" value="Off"/>		Off	
Νερό	<input type="button" value="On"/>	<input type="button" value="Off"/>		Off	
Φως	<input type="button" value="On"/>	<input type="button" value="Off"/>		Off	
Παράθυρο	<input type="button" value="On"/>	<input type="button" value="Off"/>		Off	
Δεξαμενή		lt		:7)	
Ημ/νία από	<input type="text" value="15-10-2021"/>	<input type="button" value="Submit"/>		15-10-2021	
Ημ/νία έως	<input type="text" value="15-10-2021"/>	<input type="button" value="Submit"/>		15-10-2021	

Manual

Εικόνα 57: Αυτόματη ενεργοποίηση πότισματος



Εικόνα 58: Αυτόματη ενεργοποίηση ποτίσματος



Εικόνα 59: Αυτόματη ενεργοποίηση του ποτίσματος

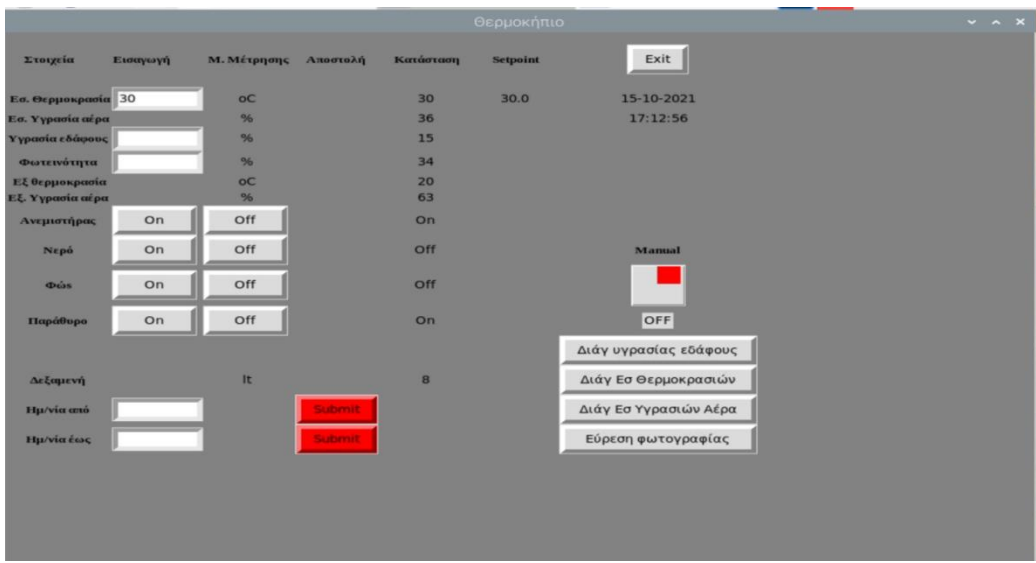
Ακόμη αν ο χρήστης κατά την αυτόματη λειτουργία βάλει κάποια τιμή στο πεδίο εισαγωγής της εσωτερικής θερμοκρασίας η οποία είναι μικρότερη από την υφιστάμενη θερμοκρασία τότε ενεργοποιούνται οι ανεμιστήρες και ανοίγει το παράθυρο έως ότου επιτευχθεί η επιθυμητή θερμοκρασία στο εσωτερικό του. Στην στήλη κατάσταση η κατάσταση του ανεμιστήρα και του παραθύρου γίνεται on όπως φαίνεται και στην παρακάτω εικόνα. Αν η θερμοκρασία πέσει περισσότερο ή ίσο με δυο βαθμούς από την τιμή που έχει θέσει ο χρήστης τότε ο ανεμιστήρας αυτόματα απενεργοποιείται και κλείνει το παράθυρο.



Εικόνα 60: Αυτόματη λειτουργία ανεμιστήρα



Εικόνα 61: Άνοιγμα παραθύρου

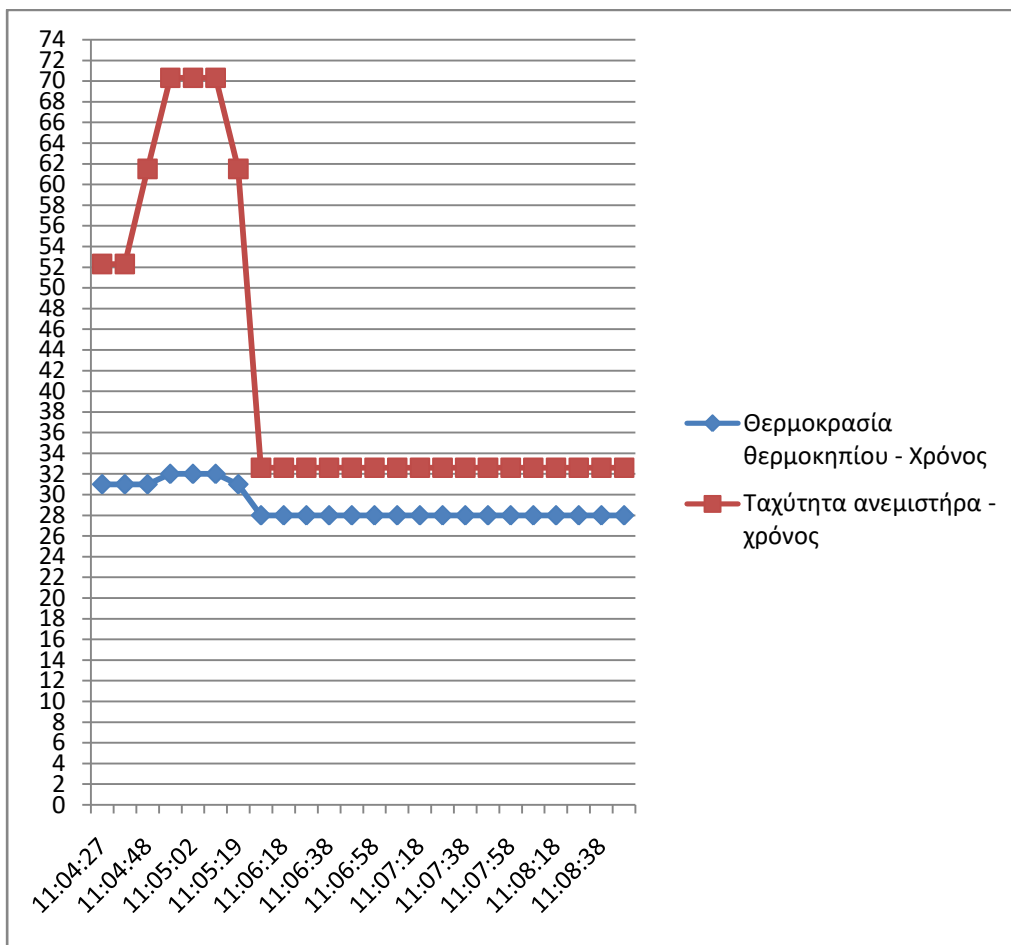


Εικόνα 62: Αυτόματη λειτουργία ανεμιστήρα και παραθύρου

Η ταχύτητα του ανεμιστήρα ελέγχεται από p ελεγκτή και μειώνεται μέχρι ένα όριο έτσι ώστε να υπάρχει στο εσωτερικό η επιθυμητή θερμοκρασία του χρήστη. Η μείωση της θερμοκρασίας σε σχέση με τον χρόνο φαίνεται στο παρακάτω διάγραμμα.



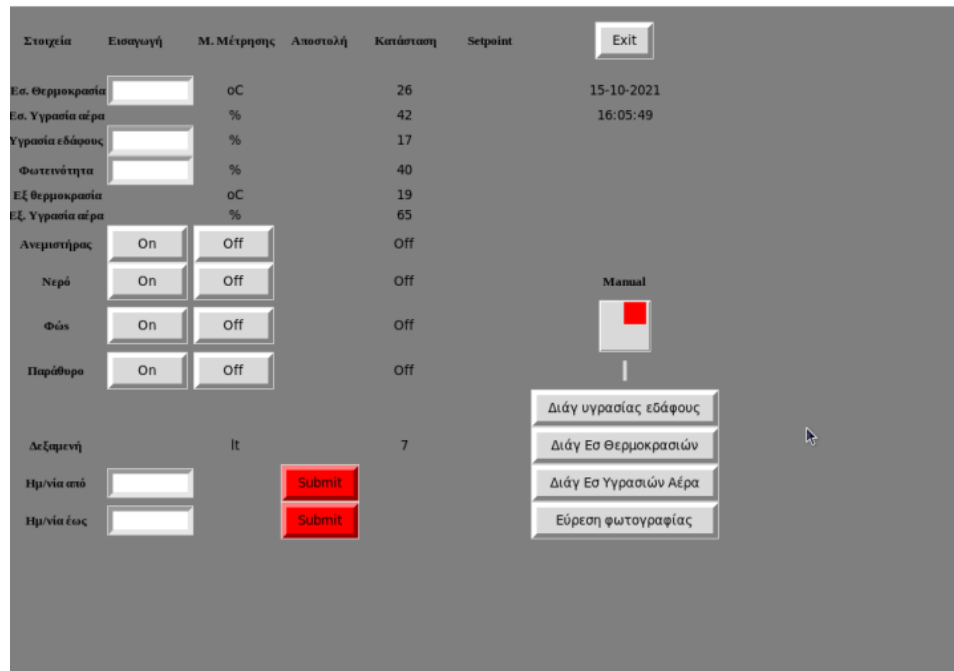
Η μεταβολή της ταχύτητας του ανεμιστήρα σε σχέση με τον χρόνο και την εσωτερική θερμοκρασία του θερμοκηπίου φαίνεται στο παρακάτω διάγραμμα



Η κάμερα λαμβάνει φωτογραφία συγκεκριμένη χρονική στιγμή της ημέρας.



Εικόνα 63: Λήψη φωτογραφίας από την κάμερα



Εικόνα 64: Επιφάνεια χειρισμού του θερμοκηπίου

Στην έβδομη στήλη της επιφάνειας χειρισμού του χρήστη υπάρχει:

- η υφιστάμενη ώρα και ημερομηνία,
- το κουμπί exit με το οποίο αν το πατήσει ο χρήστης κλείνει το παράθυρο αλληλεπίδρασης με τον χρήστη.
- τα κουμπιά διάγραμμα υγρασίας εδάφους, διάγραμμα εσωτερικής υγρασίας και θερμοκρασίας αέρα και
- το κουμπί εύρεση φωτογραφίας.

Αναλυτικότερα τα κουμπιά των διαγραμμάτων λειτουργούν σε συνεργασία με τα πεδία ημερομηνία από - έως. Δηλαδή ο χρήστης εισάγει την επιθυμητό εύρος των ημερομηνιών από το οποίο θέλει να εμφανίσει τα διαγράμματα πατάει τα κουμπιά submit που φαίνονται στην στήλη της αποστολής και έπειτα το πρόγραμμα εμφανίζει τα διαγράμματα όπως φαίνεται στις παρακάτω εικόνες. Στα διαγράμματα αυτά οι τιμές φαίνονται ανά 5 λεπτά και εξάγονται από το αρχείο data1.csv.

data.csv - LibreOffice Calc

File Edit View Insert Format Styles Sheet Data Tools Window Help

Liberation Sans 10

A10525 Σ = 10524

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	id	Date	Time	Inside Temperature (°C)	Inside Air Humidity(%)	Outside Temperature (°C)	Outside Humidity(%)	Brightness (%)	Soil moisture (%)	Tank (l)			
10525	10524	17-10-2021	15:34:04	29	36	20	70	47	48	9			
10526	10525	17-10-2021	15:34:46	29	37	20	72	47	48	9			
10527	10526	17-10-2021	15:34:53	29	37	20	74	47	47	9			
10528	10527	17-10-2021	15:35:48	30	35	20	57	47	50	9			
10529	10528	17-10-2021	15:36:20	30	34	20	56	47	50	9			
10530	10529	17-10-2021	15:36:34	29	35	20	56	47	50	9			
10531	10530	17-10-2021	15:36:52	29	35	20	56	47	44	9			
10532	10531	17-10-2021	15:37:29	29	36	20	57	48	49	9			
10533	10532	17-10-2021	15:37:59	27	37	20	57	48	50	9			
10534	10533	17-10-2021	15:38:06	27	37	20	57	48	47	9			
10535	10534	17-10-2021	15:38:16	26	38	20	57	48	48	9			
10536	10535	17-10-2021	15:38:39	26	39	20	57	48	48	9			
10537	10536	17-10-2021	15:38:49	26	39	20	57	48	48	9			
10538	10537	17-10-2021	15:38:59	26	39	20	57	48	49	9			
10539	10538	22-10-2021	14:59:39	19	68	20	89	50	14	9			
10540	10539	22-10-2021	15:00:05	19	68	21	72	63	15	9			
10541	10540	22-10-2021	15:01:03	19	69	21	68	90	14	8			
10542	10541	22-10-2021	15:01:21	19	69	21	68	25	16	8			
10543	10542	22-10-2021	15:01:27	19	69	20	68	90	14	8			
10544	10543	22-10-2021	15:02:28	19	69	20	71	37	15	8			
10545	10544	22-10-2021	15:03:08	19	69	20	72	25	14	8			
10546	10545	22-10-2021	15:03:34	19	69	19	73	90	14	8			
10547	10546	22-10-2021	15:07:22	22	72	19	79	90	14	8			
10548	10547	22-10-2021	15:07:52	22	69	19	79	90	15	8			
10549	10548	22-10-2021	15:08:06	22	67	19	80	24	14	8			
10550	10549	22-10-2021	15:08:29	21	64	20	79	26	14	8			
10551	10550	22-10-2021	15:09:42	21	62	20	75	59	15	8			
10552	10551	22-10-2021	15:30:27	21	64	20	93	52	14	8			
10553	10552	22-10-2021	15:30:54	21	65	19	81	54	14	8			
10554	10553	22-10-2021	15:31:44	21	69	19	84	51	14	8			
10555	10554	22-10-2021	15:32:10	21	68	19	83	54	15	8			
10556	10555	22-10-2021	15:32:28	21	68	19	83	54	14	8			
10557	10556	22-10-2021	15:32:42	21	67	19	83	52	14	8			
10558	10557	22-10-2021	15:33:00	21	67	19	83	56	14	8			
10559	10558	22-10-2021	15:34:06	21	67	19	84	53	15	8			
10560	10559	22-10-2021	15:34:52	20	70	19	84	52	15	8			
10561	10560	22-10-2021	15:37:01	20	70	19	85	50	15	8			
10562	10561	22-10-2021	15:39:26	20	70	19	85	47	15	8			

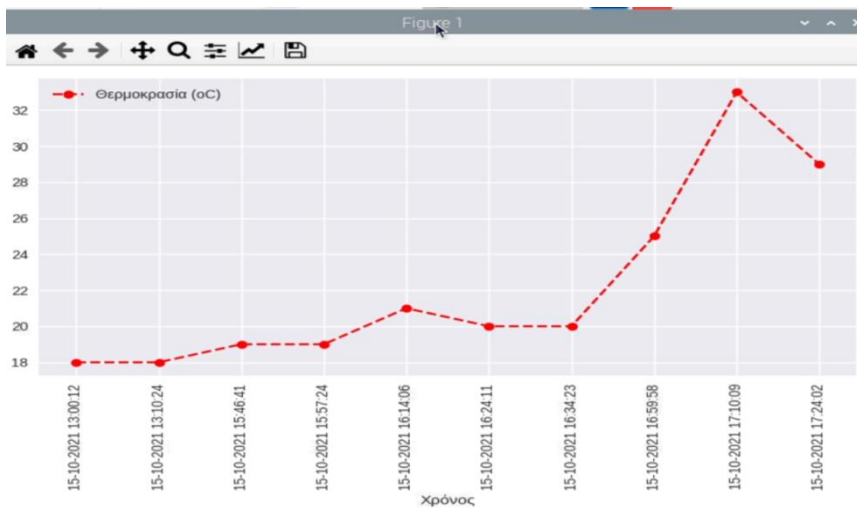
Sheet 1 of 1 | Default | English (USA) | Average: 10524; Sum: 10524 | 73%

Εικόνα 65: Αρχείο καταγραφής κατάστασης θερμοκηπίου

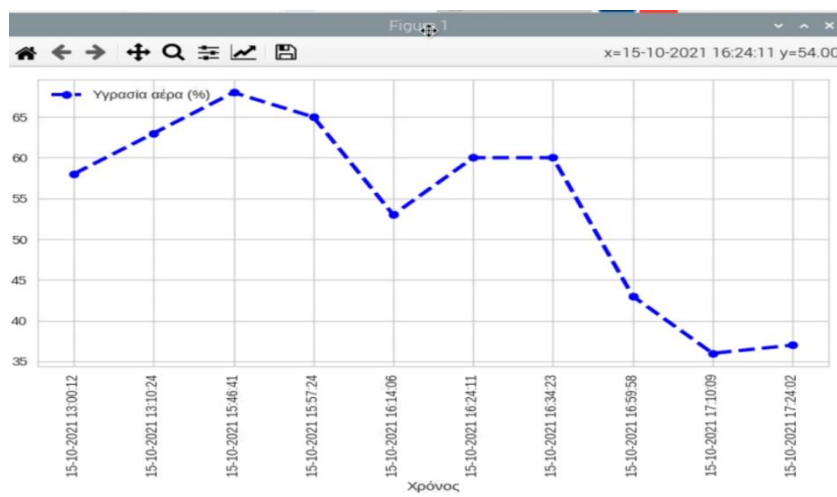
Θερμοκήπιο

Στοιχεία	Εισαγωγή	Μ. Μέτρησης	Αποστολή	Κατάσταση	Setpoint
Εσ. Θερμοκρασία	<input type="text"/>	°C	21	15-10-2021	
Εσ. Υγρασία αέρα	<input type="text"/>	%	55	18:42:44	
Υγρασία εδάφους	<input type="text"/>	%	53		
Φωτεινότητα	<input type="text"/>	%	39		
Εξ. Θερμοκρασία	<input type="text"/>	°C	19		
Εξ. Υγρασία αέρα	<input type="text"/>	%	70		
Ανεμιστήρας	<input type="button" value="On"/> <input type="button" value="Off"/>		Off		
Νερό	<input type="button" value="On"/> <input type="button" value="Off"/>		Off		
Φως	<input type="button" value="On"/> <input type="button" value="Off"/>		Off		<input type="button" value="Manual"/>
Παράθυρο	<input type="button" value="On"/> <input type="button" value="Off"/>		Off		<input type="button" value="Διάγ υγρασίας εδάφους"/>
Δεξαμενή	<input type="text" value="l"/>		29		<input type="button" value="Διάγ Εξ. Θερμοκρασιών"/>
Ημ/νία από	<input type="text" value="15-10-2021"/>		15-10-2021		<input type="button" value="Διάγ Εσ. Υγρασιών Αέρα"/>
Ημ/νία έως	<input type="text" value="15-10-2021"/>		15-10-2021		<input type="button" value="Εύρεση φωτογραφίας"/>

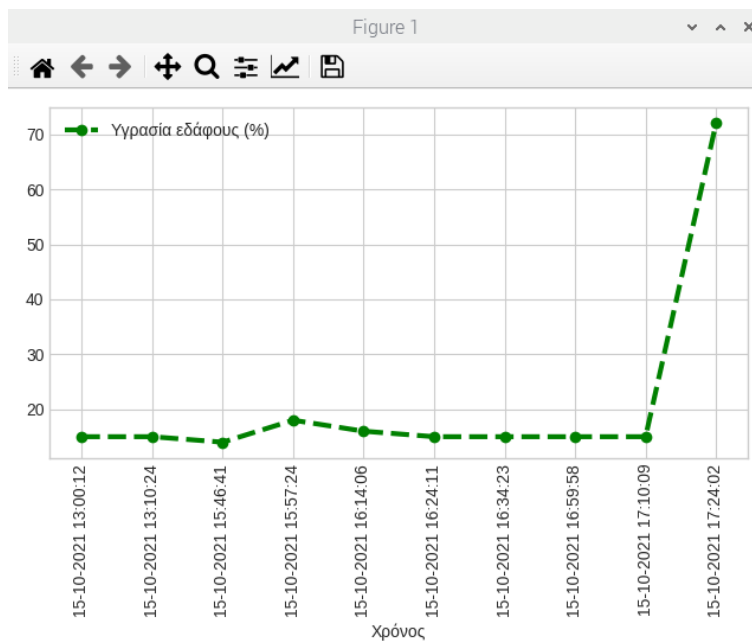
Εικόνα 66: Διαδικασία δημιουργίας διαγράμματος



Εικόνα 67: Διάγραμμα εσωτερικής θερμοκρασίας αέρα – χρόνος

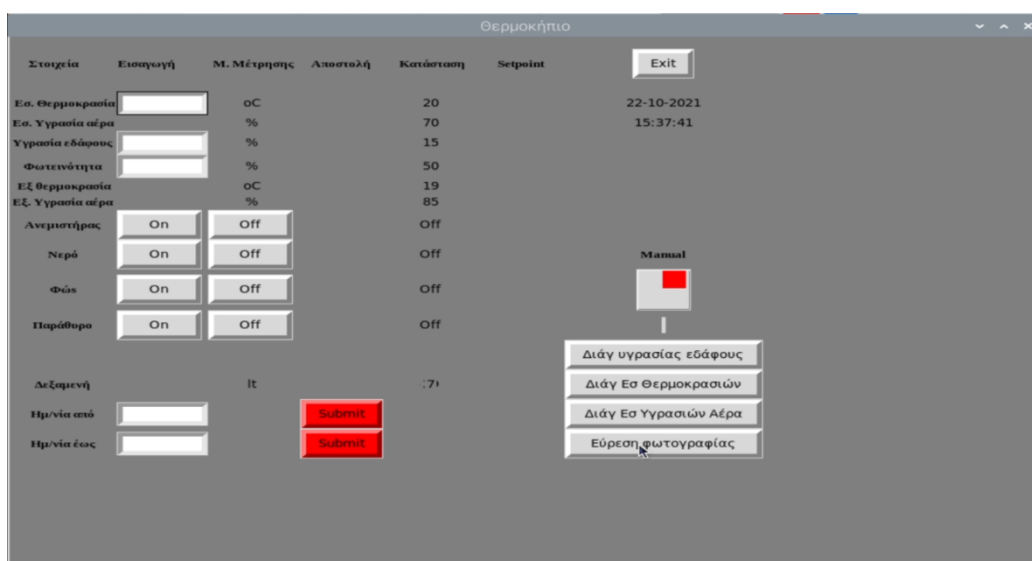


Εικόνα 68: Διάγραμμα υγρασίας αέρα – χρόνος

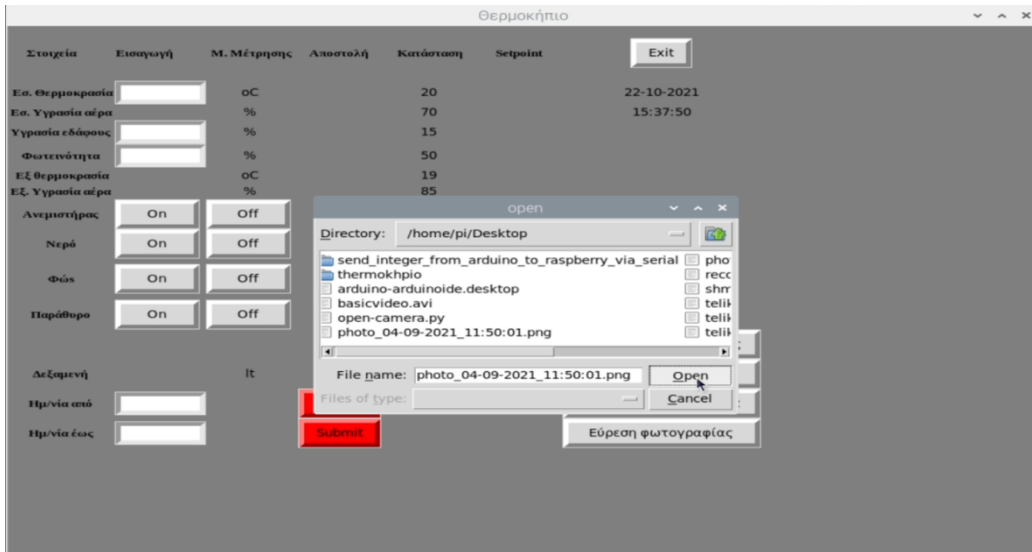


Εικόνα 69: Διάγραμμα υγρασίας εδάφους- χρόνος

Όταν ο χρήστης πατήσει το κουμπί εύρεση φωτογραφίας τότε ανοίγει ένα περιβάλλον από το οποίο δίνεται η δυνατότητα στον χρήστη να επιλέξει μια φωτογραφία και να εμφανιστεί στο περιβάλλον εργασίας του χρήστη.



Εικόνα 70: Κουμπί εύρεση φωτογραφίας



Εικόνα 71: Εύρεση φωτογραφίας



Εικόνα 72: Εμφάνιση φωτογραφίας στο περιβάλλον του χρήστη

Η εμφάνιση της κατάστασης του θερμοκηπίου εκτός από το περιβάλλον χειρισμού του χρήστη εμφανίζεται και στην lcd οθόνη όπως φαίνεται στις παρακάτω φωτογραφίες.



Εικόνα 73: Υφιστάμενη ώρα - ημερομηνία



Εικόνα 74: Υφιστάμενη υγρασία εδάφους και φωτεινότητα



Εικόνα 75: Εμφάνιση εξωτερικής υγρασίας και θερμοκρασίας



Εικόνα 76: Εμφάνιση εσωτερικής υγρασίας και θερμοκρασίας

13.Μελλοντική επέκταση

Σκέψεις για μελλοντική επέκταση είναι:

- Τοποθέτηση μετασχηματιστή 12V dc σε 5V dc και 7V dc για την τροφοδοσία του raspberry και του arduino αντίστοιχα κατευθείαν από το τροφοδοτικό
- Ανάπτυξη λογισμικού για μέτρηση του ύψους του φυτού κατευθείαν από την κάμερα.
- Τοποθέτηση σωλήνων νερού στο έδαφος του θερμοκηπίου και αντίσταση στο εσωτερικό της δεξαμενής ζεστού νερού για την γρήγορη ανάπτυξη των φυτών.
- Χρήση PID ελεγκτών για το πότισμα και για την φωτεινότητα του θερμοκηπίου
- Αναγνώριση ασθένειας από την κάμερα
- Απομακρυσμένος έλεγχος θερμοκηπίου
- Δυνατότητα δημιουργίας διαγραμμάτων όχι μόνο με κριτήριο την ημερομηνία αλλά παίρνοντας υπόψη και το εύρος συγκεκριμένων ωρών.
- Προσθήκη επιπλέον ανεμιστήρα για μεγαλύτερη δυνατότητα μείωσης της θερμοκρασίας

14.Βιβλιογραφία

Οι πηγές που χρησιμοποιήθηκαν για την παρούσα εργασία φαίνονται παρακάτω:

1. Ζαρμπούτης Γ, & Γκάνη Α., 1992, <<Καλλιέργεια σε θερμοκήπιο>>, εκδόσεις ΙΩΝ
2. Γεωργία Ακριβείας, Σπύρος Φούντας, Θεοφάνης Γέμτος, 2015
3. <http://www.gaiapedia.gr/gaiapedia/index.php/%CE%A5%CE%B3%CF%81%CE%B1%CF%83%CE%AF%CE%B1%CF%83%CF%84%CE%BF%CE%B8%CE%B5%CF%81%CE%BC%CE%BF%CE%BA%CE%B7%CF%80%CE%AF%CE%BF>
4. Μαυρογιαννόπουλος Γ., Ν., 2001, 'Θερμοκήπια, Έκδοση Γ', Εκδόσεις Σταμούλη, Αθήνα
5. P. Polycarpou, Agricultural Research Institute Soils and Water use Department / Agricultural Engineering , Optimization of Nocturnal Climate Management in PE-greenhouses in Cyprus
6. A. Rahali , M. Guerbaoui , A. Ed-dahhak, Y. El Afou, A. Tannouche, A. Lachhab, B. Bouchikhi, Development of a data acquisition and greenhouse control system based on GSM, 2011
7. Y. El Afou, L. Belkoura, M. Outanoute, M. Gurbaoui, A. Rahali, A. Ed-Dahhak, A. Lachhab, C. Joi, B. Bouchikhi, Feedback Techniques Using PID and PI Intelligent For Greenhouse Temperature Control, 2014
8. Shahad Alyousif, Nurul Fadzilawati Zainuddin, Baharudin Bin Hamzah, Intelligent Temperature Control System at Greenhouse, International Journal of Applied Engineering Research, 2017
9. Karthikeyan, Gokul Chandrasekaran, Neelan Sanjeev Kumar, Elango Sengottaiyant, Prabu Mani, D T Kalavathi, V Gowrishankar, IOT Based Moisture Control and Temperature Monitoring in Smart Farming, 2020
10. Feifei Li, Hongling Wang, Lu Sun, Zilin Wang, Yuting Ran, Yingjun Xia, Data Collection of Digital Monitoring System for Agricultural Facilities Environment, 2021
11. Έλεγχος διεργασιών, Νταουτίδης, Π., Μαστρογεωργόπουλος Σπ, Παπαδοπούλου Σημ., Έκδοση: 1 η Έκδοση/2012, ISBN: 978-960-418-390-6, Διαθέτης (Εκδότης): ΕΚΔΟΣΕΙΣ Α. ΤΖΙΟΛΑ & ΥΙΟΙ Α.Ε»
12. G.F. Franklin et al. "Feedback Control of Dynamic Systems", 5th Edition. Upper Saddle River, NJ, 2006, Pearson Education, Inc, <<Block Diagrams, Feedback and Transient Response Specifications module>>
13. <https://www.etechnophiles.com/raspberry-pi-4-gpio-pinout-specifications-and-schematic/>
14. <https://grobotronics.com/raspberry-pi-4-model-b-4gb.html?sl=en>
15. <https://arduinolaboratory.weebly.com/piepsilononrhoiota-arduino.html>
16. <https://www.informatique-mania.com/el/materiel/arduino-uno/>
17. https://www.aliexpress.com/item/1005001622038553.html?algo_pvid=1a375786-2578-427f-bb5f-7aa85787726c&aem_p4p_detail=202109221146086696285878695140001925597&algo_exp_id=1a375786-2578-427f-bb5f-7aa85787726c-4
18. https://www.hellasdigital.gr/electronics/boost-and-buck-converters/dc-dc-boost-converter-10-32v-to-12-35v-6a-step-up/?skr_prm=WyJiYzE5ZTdjMCIhMzE3LTQyMjEtYWl1Zi1kY2I5YTU0NWY0NjUiLDE2MzIzMjA4NzkyMTAseyJhcHBfdHlwZSI6IndlYiIsImNwIjojZiIsInRhZ3MiOiIifV0
19. https://www.gadget4you.gr/kingston-canvas-select-plus-sdhc-32gb-read-100mb-s-class-10-sd-adaptersdcs232gb.html?skr_prm=WyJiYzE5ZTdjMCIhMzE3LTQyMjEtYWl1Zi1kY2I5YTU0NWY0NjUiLDE2MzIzMjMTI4MzE3NjIseyJhcHBfdHlwZSI6IndlYiIsImNwIjojZiIsInRhZ3MiOiIifV0
20. SmartDrive Duo-10 MDDS10, User's Manual V1.0, June 2015

21. <https://grobotronics.com/mcp3008-10bit-8-channel-adc-spi.html>
22. <https://grobotronics.com/basic-16x2-character-lcd-white-on-blue-5v-i2c-protocol.html>
23. <https://el.wikipedia.org/wiki/I%C2%B2C>
24. <https://components101.com/modules/soil-moisture-sensor-module>
25. https://www.aliexpress.com/item/1005001501328269.html?algo_pvid=2118b1ce-d455-4c92-978c-f77954112e8b&algo_exp_id=2118b1ce-d455-4c92-978c-f77954112e8b-0&pdp_ext_f=%7B%22sku_id%22%3A%2212000016604771921%22%7D
26. <https://sparkfruit.ph/product/photosensitive-light-sensor-analog/>
27. <https://www.skafatos.gr/antlia-sentinas/antlia-sentinas-seaflo-1900-litra-ora-500-gph-18-525->
28. <https://grobotronics.com/high-torque-motor-12v-dc-775-12000rpm.html>
29. <https://components101.com/motors/mg996r-servo-motor-datasheet>
30. https://www.dotpower.gr/index.php?route=product/product&path=82_83_94_123&product_id=18117&skr_prm=WyJiYzE5ZTdjMC1hMzE3LTQyMjEtYWU1Zi1kY2I5YTU0NWY0NjUiLDE2MzI0MTIzNDQxNTkseyJhcHBfdHlwZSI6ImNwIjoiZiIsInRhZ3MiOiIifV0
31. https://www.ergo-tel.gr/product_info.php?products_id=27208
32. <https://grobotronics.com/humidity-sensor-dht11-module.html>
33. Elec Freaks, Datasheet, Ultrasonic Ranging Module HC-SR04
34. <https://grobotronics.com/power-supply-5v-3a-usb-plug.html>
35. <https://grobotronics.com/jumper-wires-30cm-male-to-male-pack-of-10.html>
36. <https://grobotronics.com/relay-module-2-channel.html>