



DEPARTMENT OF INDUSTRIAL ENGINEERING AND MANAGEMENT

Ασύρματος Πολύ-Αισθητήρας Ρευστών (Wireless Fluid Multi Sensor)

Διπλωματική Εργασία

του

Διαμαντή Ζαφείρη 2016/105

και του

Δεληγιάννη Στέφανου 2016/096

Επιβλέποντες:

Κιζήρογλου Μιχαήλ, Μπεχτσής Δημήτριος

Θεσσαλονίκη, Ιούλιος 2021

Ευχαριστίες

Στο σημείο αυτό, θα ήθελα να ευχαριστήσω όλους εκείνους τους ανθρώπους που συνέβαλαν, με διαφορετικό τρόπο ο καθένας, στην εκπόνηση αυτής της διπλωματικής εργασίας.

Αρχικά, θα θέλαμε να εκφράσουμε τις θερμές ευχαριστίες στους Επιβλέποντες Καθηγητές μας κ. Κιζήρογλου Μιχαήλ και κ. Μπεχτσή Δημήτριο για την εμπιστοσύνη που μας έδειξαν και μας ανέθεσαν το παρόν θέμα. Με την εκπόνηση της παρούσας Διπλωματικής Εργασίας μας δόθηκε η ευκαιρία να διευρύνουμε τις γνώσεις μας σε βάθος πάνω στο κλάδο που θα θέλαμε μελλοντικά να ασχοληθούμε περνώντας από την θεωρία στην πράξη.

Έπειτα, θα θέλαμε να ευχαριστήσουμε τον Υποψήφιο Διδάκτορα Σούλα Γκέρσι, αλλά και τους υπόλοιπους Υποψήφιους Διπλωματούχους Μηχανικούς Αυτοματισμού Νίκο Παπάζογλου και Νίκο Μπαξεβάνο που εκπόνησαν παράλληλα τις Διπλωματικές τους Εργασίες, συμβάλλοντας όλοι μαζί στην επίτευξη του συνολικού Συστήματος με τον καλύτερο δυνατό τρόπο.

Από το πλαίσιο των ευχαριστιών αυτών, δε θα μπορούσαν να λείπουν οι οικογένειές μας, οι οποίες αποτελούν τον αφανή ήρωα πίσω από κάθε μας προσπάθεια, στηρίζοντάς μας κατά τη διάρκεια των πέντε αυτών ετών με κάθε δυνατό τρόπο. Τέλος, ένα μεγάλο ευχαριστώ αξίζει να δοθεί στην αμερόληπτη συνεργασία μας με τον Στέφανο όλα αυτά τα χρόνια, με την οποία καταφέραμε να βγάλουμε εις πέρας τις υποχρεώσεις του Τμήματος, αλλά και στην φιλία μας εκτός της Σχολής που μας καθιστά ικανούς να αντιμετωπίσουμε κάθε πρόκληση.

Δήλωση Λογοκλοπής

Η παρούσα Πτυχιακή Εργασία και τα συμπεράσματά της, σε οποιαδήποτε μορφή, αποτελούν συνιδιοκτησία του Τμήματος Μηχανικών Παραγωγής και Διοίκησης του Διεθνούς Πανεπιστημίου Ελλάδος και του φοιτητή. Οι προαναφερόμενοι διατηρούν το δικαίωμα ανεξάρτητης χρήσης και αναπαραγωγής (τμηματικά ή συνολικά) για διδακτικούς και ερευνητικούς σκοπούς. Σε κάθε περίπτωση πρέπει να αναφέρεται ο τίτλος, ο συγγραφέας, ο επιβλέπων και το τμήμα του ΔιΠαΕ.

Η έγκριση της παρούσας Διπλωματικής Εργασίας από το Τμήμα Μηχανικών Παραγωγής και Διοίκησης δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Οι υπογεγραμμένοι δηλώνουμε υπεύθυνα ότι η παρούσα Πτυχιακή Εργασία είναι εξ' ολοκλήρου δικό μας έργο και συγγράφηκε ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Παραγωγής και Διοίκησης.

Δηλώνουμε υπεύθυνα ότι κατά τη συγγραφή ακολουθήσαμε την πρότυπα ακαδημαϊκή δεοντολογία αποφυγής λογοκλοπής και έχουμε αποφύγει οποιαδήποτε ενέργεια που συνιστά παράπτωμα λογοκλοπής.

31/08/2021

Διαμαντής Ζαφείρης, Δεληγιάννης Στέφανος

Περίληψη

Ένα σύστημα βιομηχανικού ελέγχου αποτελεί το γενικό πλαίσιο, στο οποίο εντάσσονται οι διάφοροι τύποι συστημάτων ελέγχου, καθώς και ο απαραίτητος εξοπλισμός που χρησιμοποιείται για τον έλεγχο βιομηχανικών διεργασιών. Τα συστήματα ελέγχου είναι αρμόδια για τον έλεγχο μίας διεργασίας, αξιοποιώντας δεδομένα απομακρυσμένων αισθητηρίων, έτσι ώστε οι μεταβλητές της διεργασίας αυτής να λαμβάνουν επιθυμητές τιμές. Τη βάση για την ανάπτυξη ενός συστήματος ελέγχου μικρής κλίμακας, αποτελεί ο προγραμματιζόμενος λογικός ελεγκτής (PLC), ο οποίος μέσω του κατάλληλου διαύλου αλληλοεπιδρά με τα αισθητήρια και τους ενεργοποιητές της ελεγχόμενης διεργασίας. Η παρούσα διπλωματική εργασία στοχεύει στην ανάπτυξη ενός ανάλογου συστήματος βιομηχανικού ελέγχου, το οποίο εκκινείται και λειτουργεί κάνοντας χρήση ενός Συστήματος Αισθητήρων Ρευστών και συγκεκριμένα αισθητήρες μέτρησης ύδατος. Τα αποτελέσματα των τιμών που λαμβάνουμε από αυτό το Σύστημα αποθηκεύονται σε ένα αρχείο CSV(Excel) και τυπώνονται σε μια πλατφόρμα αποθήκευσης αρχείων(Dropbox). Καθώς το Σύστημά μας είναι ασύρματο, για να καταγραφούν τα αποτελέσματα στο Αρχείο Excel γίνεται χρήση του πρωτοκόλλου ασύρματης επικοινωνίας WIFI. Πιο συγκεκριμένα, τη θέση του προγραμματιζόμενου λογικού ελεγκτή αναλαμβάνει ο μικροϋπολογιστής ESP32 που διαθέτει το πρωτόκολλο WIFI και καθιστά την επικοινωνία ευκολότερη. Ο ESP32 καλείται να ελέγξει το Σύστημα Αισθητήρων, το οποίο αποτελείται από ένα αισθητήριο Ph και ένα αισθητήριο θερμοκρασίας τύπου PT100. Τα αισθητήρια είναι προγραμματισμένα στην πλατφόρμα Arduino IDE σε γλώσσα C wiring. Ο μικροελεγκτής συνδεδεμένος σε τοπικό δίκτυο WIFI και με την βοήθεια ειδικού script σε γλώσσα Python μεταφέρει τις πληροφορίες από την σειριακή οθόνη του Arduino IDE στο αρχείο Excel σε στήλες. Έχοντας δημιουργήσει ένα πλήρως λειτουργικό και αποδοτικό σύστημα ελέγχου, ο χρήστης μπορεί να ελέγξει απομακρυσμένα τις καταμετρήσεις των αισθητηρίων, μέσω ενός λογαριασμού αποθήκευσης αρχείων(Dropbox), όπου αποθηκεύεται το αρχείο με τα αποτελέσματα.

Λέξεις Κλειδιά:

Automation, PLC, ESP32, Arduino IDE, Python, CSV, Dropbox.

Abstract

The European Union and all the developed countries have enacted mandatory legislation to ensure acceptable water quality parameters, which is why countries are conducting sampling inspections at specific locations, such as water dams. However, the traditional sampling method is both time consuming and an expensive process. That is why modern techniques have been developed for systematic monitoring of water quality.

The purpose of this dissertation is to design and manufacture a sensor system that will examine the quality of a water sample. To achieve the quality measurement, four sub-sensors are used, which measure turbidity, pH and temperature. Four suitable commercial sensors were selected, which were programmed for parallel use, in conjunction with the Arduino platform. The results of the measurements are sent and stored in a Csv file on a Dropbox database that is connected to the Esp32.

Keywords: Water Quality, Sensors, Turbidity, pH, Esp32

Diamantis Zafeiris, Deligiannis Stefanos, September 2021

Department of Industrial Engineering and Managment

International Hellenic University

aut2016105@autom.teithe.gr, aut2016096@autom.teithe.gr

zafirisdiamantis@gmail.com, stevedel98@gmail.com

Περιεχόμενα

1	Εισαγωγή	14
1.1	Περιγραφή Προβλήματος.....	15
1.2	Αντικείμενο Διπλωματικής Εργασίας	16
1.3	Διάρθρωση Διπλωματικής Εργασίας	17
2	Hardware	19
2.1	Αισθητήρια.....	19
2.1.1	Αισθητήριο Θερμοκρασίας.....	19
2.1.2	Αισθητήριο PH.....	20
2.2	Επιπρόσθετα Αισθητήρια	22
2.2.1	Αισθητήριο TDS	22
2.2.2	Αισθητήριο Θολότητας.....	24
2.2.3	Αισθητήριο Πίεσης	24
2.3	Μικρό-Ελεγκτής.....	27
2.3.1	Σύγκριση ελεγκτών όσον αφορά την αρχιτεκτονική τους.....	27
2.3.2	Σύγκριση ελεγκτών όσον αφορά τον ADC	28
2.3.3	Σύγκριση ελεγκτών όσον αφορά την μνήμη τους	29
2.3.4	Σύγκριση όσον αφορά την μνήμη Flash Memory	30
2.3.5	Σύγκριση όσον αφορά την SRAM	31
2.4	Επιπρόσθετα Υλικά για την Υλοποίηση του Hardware.....	34
3	Εργαλεία Λογισμικού	36
3.1	ARDUINO IDE.....	36
3.2	IDLE PYTHON.....	37
3.3	Διάγραμμα Ροής Arduino IDE	39
3.3.1	Εξήγηση Κώδικα Arduino IDE	40
3.4	Εξήγηση Κώδικα Python	44

3.4.1	Διάγραμμα Ροής.....	45
.....		45
3.4.2	Πινακάκι Εξήγησης Εντολών Python.....	46
3.5	Μία Αναφορά στο Internet of Things(IOT).....	51
3.6	Πρωτόκολλα Ασύρματης Επικοινωνίας.....	52
3.6.1	Bluetooth-Le	52
3.6.2	LoRa.....	53
3.6.3	Wi-Fi.....	53
3.6.4	Wi-Fi HaLow	54
4	Υλοποίηση Κατασκευής.....	58
4.1	Σκοπός της Κατασκευής	58
4.2	Σχεδίαση κελύφους ηλεκτρονικών εξαρτημάτων	61
4.3	Προσαρμογή Αισθητήρων	64
4.3.1	Σχεδίαση αισθητήρων στο SOLIDWORKS	64
4.3.2	Εργαλεία και εξαρτήματα που χρησιμοποιήθηκαν για την υλοποίηση .66	
4.4	Υλοποίηση Κατασκευής	69
5	Ενδεικτικά Πειράματα.....	72
5.1	Πειράματα με τα αισθητήρια	72
5.2	Συνδεσμολογίες αισθητήρων μικροελεγκτή	73
5.3	Λήψη μετρήσεων	74
6	Παράρτημα	76
7	Βιβλιογραφία	85

Λίστα Εικόνων

Σχήμα 2.1 : Αισθητήριο Θερμοκρασίας	16
Σχήμα 2.2 : Αισθητήριο Ph.....	17
Σχήμα 2.3 : Αισθητήριο TDS.....	18
Σχήμα 2.4 : Αισθητήριο Θολότητας (Turbidity Sensor).....	19
Σχήμα 2.5 : Αισθητήριο Πίεσης.....	20
Σχήμα 2.6 : Πλατφόρμα Esp32 NodeMCU-32S.....	21
Σχήμα 2.7 : Πλατφόρμα Nordic Nrf52832	22
Σχήμα 2.8 : Πλατφόρμα Msp430 LaunchPad.....	23
Σχήμα 2.9 : Τροφοδοτικό μονάδας Esp32.	25
Σχήμα 2.10 : Επαναφορτιζόμενη μπαταρία λιθίου Li-ion	26
Σχήμα 3.1 : Περιβάλλον προγράμματος Arduino IDE.	37
Σχήμα 3.2 : Περιβάλλον προγραμματισμού IDLE Python.....	38
Σχήμα 4.1 : Κατασκευαστική Διπλωματική του συμφοιτητή μας Νίκου Μπαξεβάνου, πάνω στην οποία θα εγκατασταθεί η δική μας εργασία, έτος 2021.....	59
Σχήμα 4.2 : Ο κυκλοφορητής που χρησιμοποιεί το παραπάνω σύστημα (Διπλωματική Εργασία του Νίκου Μπαξεβάνου, 2021)	60
Σχήμα 4.3 : Το σημείο στο οποίο θα προσαρμοστεί το σύστημα αισθητήρων μας, (Διπλωματική Εργασία του Νίκου Μπαξεβάνου, 2021)..	61
Σχήμα 4.4 : Το κέλυφος προσαρμογής του συστήματος αισθητήρων σχεδιασμένο στο πρόγραμμα SOLIDWOKRS, (Διπλωματική Εργασία του Νίκου Παπάζογλου, 2021, με θέμα την σχεδίαση και την κατασκευή ενός κελύφους με χρήση CNC).....	62
Σχήμα 4.5 : Παλαιότερη προσέγγιση στήριξης του κελύφους επάνω στην σωλήνα, (Διπλωματική Εργασία του Νίκου Παπάζογλου, 2021, με θέμα την σχεδίαση και την κατασκευή ενός κελύφους με χρήση CNC).....	63
Σχήμα 4.6 : Μακρινή οπτική παλαιότερης προσέγγισης, (Διπλωματική Εργασία του Νίκου Παπάζογλου, 2021, με θέμα την σχεδίαση και την κατασκευή ενός κελύφους με χρήση CNC).....	63
Σχήμα 5.1 : Πειράματα με το αισθητήριο pH.....	72
Σχήμα 5.2 : Σύνδεση αισθητήρα pH με τον Esp32.....	73
Σχήμα 5.3 : Σύνδεση αισθητήρα θερμοκρασίας με τον Esp32.....	74
Σχήμα 5.4 : Λήψη μετρήσεων σε πραγματικό χρόνο.	75

1 Εισαγωγή

Με τον όρο *Ασύρματος Πολύ-αισθητήρας Ρευστών* νοείται το αυτοματοποιημένο σύστημα το οποίο τοποθετείται σε ένα δίκτυο ύδρευσης, και έχει ικανότητα να μας εμφανίζει διάφορα δεδομένα στοιχεία σχετικά με την περιεκτικότητα του εν προκειμένω κυκλώματος, και όλα αυτά με την χρήση ασύρματης επικοινωνίας. Η ανάγκη ύπαρξης ενός τέτοιου συστήματος είναι επιτακτική σε πολλές περιπτώσεις, από την επιστημονική έρευνα όπου θέλουμε να έχουμε μια απλή εποπτεία ενός συστήματος με κάποιο ρευστό, μέχρι την παρακολούθηση συστημάτων όπου είναι απαραίτητη μια ενδεχόμενη άμεση επέμβαση στην περιεκτικότητα ενός ρευστού.

Προκειμένου ο κύριος ελεγκτής του συστήματος μας να φέρει εις πέρας την λειτουργία του εποπτικού ελέγχου, απαιτείται επικοινωνία με ένα σύνολο αισθητηρίων τα οποία με την σειρά τους δίνουνε το κάθε ένα ξεχωριστά πληροφορίες για διαφορετικά στοιχεία. Έπειτα ο ελεγκτής συλλέγει τα δεδομένα αυτά και τα στέλνει ασύρματα με χρήση κάποιου άλλου πρωτοκόλλου επικοινωνίας (π.χ LoRa, Wi-fi κ.α) σε κάποια ενδεχόμενη βάση δεδομένων ενός διακομιστή όπου θα έχουμε πρόσβαση, ή σε μία άλλη συσκευή έτσι ώστε να μπορούμε να αντλήσουμε τα απαραίτητα δεδομένα. Η μετάδοση πληροφοριών μέσω δικτύου παρέχει σημαντικά οφέλη όσον αφορά την αξιοπιστία, την ευκολότερη διάγνωση και συντήρηση, καθώς και τη μειωμένη καλωδίωση. Ωστόσο υπάρχουνε διάφορες λύσεις για ασύρματη επικοινωνία και διασύνδεση συσκευών, όπου ανάλογα με τις απαιτήσεις χρησιμοποιείται και η κατάλληλη. Για παράδειγμα το Wifi είναι μια προφανής λύση για κάποιον προγραμματιστή, για την γρήγορη μεταφορά και διαχείριση δεδομένων, παρόλα αυτά καταναλώνει μεγάλα ποσά ενέργειας τα οποία κάποιος θέλει να αποφύγει όταν λειτουργεί ένα σύστημα το οποίο δεν είναι άμεσα συνδεδεμένο στο δίκτυο ηλεκτρικής ενέργειας. Μία άλλη γνωστή λύση είναι το πρωτόκολλο Zigbee το οποίο ενώ έχει χαμηλή κατανάλωση ενέργειας και σχετικά γρήγορη μετάδοση δεδομένων ,δεν δίνει μεγάλη ευελιξία στην απόσταση των δύο διασυνδεδεμένων συσκευών καθώς επιτρέπει μέγιστη απόσταση μέχρι 100m. Μία κατάλληλη λύση για την ασύρματη επικοινωνία συστημάτων ,φάνηκε πως είναι το πρωτόκολλο LoRa το οποίο ενώ διατηρεί την επιθυμητή χρήση χαμηλής κατανάλωσης ενέργειας, προσφέρει μεγάλες δυνατότητες και στην εμβέλεια που θα έχουνε οι συσκευές μεταξύ τους , η οποία σε κατάλληλες

καιρικές συνθήκες μπορεί να φτάσει τα 10+ χιλιόμετρα. Αν επιλέξει ωστόσο κάποιος να χρησιμοποιήσει αυτό το πρωτόκολλο επικοινωνίας, για την δική μας εργασία θα χρειαζόταν να χρησιμοποιήσει έναν μικρό-ελεγκτή ο οποίος να συλλέγει τα δεδομένα από τα αισθητήρια, και να τα στέλνει σε έναν άλλο μικρό-ελεγκτή μέσω αυτού το πρωτοκόλλου προκειμένου εκείνος να τα στείλει τελικά σε μια βάση δεδομένων από την οποία θα βλέπουμε τα αποτελέσματα. Σε αυτή όμως την περίπτωση θα χρειάζεται και οι δύο μικρό-ελεγκτές να είναι συνδεδεμένοι με μία μονάδα LoRa ο καθένας και έτσι να είναι επιτεύξιμη αυτού του είδους η επικοινωνία. Περισσότερα για το κομμάτι της ασύρματης επικοινωνίας όμως θα αναφέρουμε στο 2^ο μέρος της εργασίας.

1.1 Περιγραφή Προβλήματος

Στο προηγούμενο απόσπασμα αναφέραμε την γενική ιδέα της εργασίας μας αναλύοντας λίγα περισσότερα πράγματα για την ασύρματη επικοινωνία, τα οποία θα φανούν χρήσιμα αφού βρούμε κατάλληλα αισθητήρια τα οποία θα υπερκαλύπτουν τις ανάγκες της επιθυμητής διακριτικής ικανότητας και αφού βρούμε έναν κατάλληλο ελεγκτή ο οποίος θα είναι συμβατός με αυτά και θα μπορεί να τα τροφοδοτήσει, και να τα διαχειριστεί με τον τρόπο που θέλουμε.

Για το λόγο ότι υφίσταται ένα μεγάλο εύρος αισθητηρίων οργάνων και μικρό-ελεγκτών, η επιλογή του καταλληλότερου από αυτά στα πλαίσια μίας συγκεκριμένης εφαρμογής, αποτελεί επιτακτική ανάγκη. Κάθε εφαρμογή αυτοματισμού διαφοροποιείται από κάθε άλλη, εξαιτίας του υλικού που κάνει χρήση. Το υλικό σε συνδυασμό με τα λογισμικά προσθέτουν περιορισμούς, οι οποίοι απορρέουν από τις δυνατότητες που υποστηρίζουν. (Έπιπλέον, οι απαιτήσεις της εφαρμογής αυτής καθ' αυτής καθορίζουν με τη σειρά τους σε πολύ μεγάλο βαθμό τα χαρακτηριστικά και τις αποδόσεις που πρέπει να φέρει ένας δίαυλος μεταφοράς δεδομένων). Τέλος, πάντοτε υπάρχουν και παράγοντες εξωγενείς του συστήματος που πρέπει να ληφθούν υπόψιν, όπως επί παραδείγματι το κόστος της υλοποίησης.

Όλα τα παραπάνω, και πολλά ακόμη, αποτελούν προβληματισμούς, τους οποίους καλείται να επιλύσει ο μηχανικός κατά την υλοποίηση ενός ασύρματου συστήματος ποιοτικού ελέγχου ρευστών. Τα προβλήματα αυτά είναι όμοια είτε γίνεται λόγος για μία μεγάλη υποδομή ζωτικής σημασίας, είτε πρόκειται για μία μικρή εφαρμογή αυτομάτου ελέγχου. (Όπως είναι φυσικό, σε μία εποχή που κατακλύζεται από

συστήματα αυτοματισμών, το πρόβλημα της επιλογής δικτύου διασύνδεσης που πρόκειται να χρησιμοποιηθεί είναι σύνηθες. Τον προβληματισμό αυτό αποσκοπεί να διερευνήσει η παρούσα διπλωματική εργασία και τελικώς, να επιλύσει, στα πλαίσια πάντοτε μίας συγκεκριμένης εφαρμογής).

1.2 Αντικείμενο Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασία αποσκοπεί στην ανάπτυξη ενός ασύρματου συστήματος ελέγχου ρευστών, με κύρια συστατικά στοιχεία αυτά των αισθητηρίων οργάνων και του προγραμματιζόμενου μικρό-ελεγκτή, ο οποίος θα επεξεργάζεται τα δεδομένα που δέχεται από τα αισθητήρια, και θα εμφανίζει τα αποτελέσματα στην πλατφόρμα του Drogbox που έχει δημιουργηθεί για τον συγκεκριμένο σκοπό. Το σύστημα που περιγράψαμε, θα βρίσκεται σε ένα πλαστικό περίβλημα το οποίο θα είναι προσαρμοσμένο σε μια σωλήνα θερμοκόλλησης τύπου PPR από την οποία θα διέρχεται το υγρό, και από όπου θα παίρνουμε τις απαραίτητες μετρήσεις που θέλουμε.

Πρωταρχική απαίτηση του παρόντος θέματος είναι ο καθορισμός των αισθητηρίων που θα χρησιμοποιηθούν στο σύστημα ελέγχου για εύρεση των επιθυμητών ιδιοτήτων του ρευστού, του ελεγκτή που θα τα διαχειρίζεται, και του λογισμικού που θα χρησιμοποιηθεί για προγραμματισμό των αισθητηρίων. Ως προγραμματιζόμενος ελεγκτής, (συμβατός με το διεθνές βιομηχανικό πρότυπο IEC 61131-3), επιλέχθηκε ο μικροϋπολογιστής ESP32 Development Board - DEVKIT V1, λόγω της μεγάλης ευελιξίας που έχει σε εφαρμογές με αισθητήρια όργανα, της χαμηλής κατανάλωσης του, του μικρού μεγέθους του, και άλλων παραμέτρων που θα επισυναφθούν πιο λεπτομερώς στο κεφάλαιο του Hardware. Από την άλλη πλευρά, για τα 2 αισθητήρια όργανα επιλέχθηκαν: για αισθητήριο θερμοκρασίας το Gravity: Analog High Temperature Sensor (<https://www.hellasdigital.gr/electronics/sensors/temperature-sensors/waterproof-ds18b20-sensor-kit-kit0021/>) και για το αισθητήριο Ph το Gravity: Analog Ph Sensor (<https://grobotronics.com/gravity-analog-ph-sensor-meter-kit-v2.html?sl=en>). Όσον αφορά το λογισμικό προγραμματισμού το ESP32 επιλέξαμε να χρησιμοποιήσουμε το γνωστό λογισμικό Arduino IDE με χρήση της κατάλληλης βιβλιοθήκης για να είναι συμβατό με τον ελεγκτή μας.

Καθώς το υλικό και το λογισμικό είναι πλέον καθορισμένα, στο σημείο αυτό λαμβάνει χώρα το πρόβλημα τροφοδότησης του ελεγκτή μας, το οποίο θα καθιστά και

την ασύρματη λειτουργία του, που είναι και το κατεξοχήν ζητούμενο της εργασίας μας. Κάνοντας μια έρευνα βρήκαμε πως η κατάλληλη λύση είναι η χρήση του φορτιστή μπαταρίας Battery Charger 18650 with Step-Up 5V – USB, όπου δίνεται η δυνατότητα υποδοχής επαναφορτιζόμενης μπαταρίας τύπου Li-ion 3.7 V η οποία μέσω καλωδίου Usb to micro Usb τροφοδοτεί τον ελεγκτή μας με την επιθυμητή τάση των 3.7 V.

Για την επίτευξη του καθολικού στόχου, η εργασία εκκινεί με τον καθορισμό των απαιτήσεων του συστήματος που καλείται να ελεγχθεί, δηλαδή αυτό της ποιότητας του ρευστού στις διάφορες ιδιότητες , που μας ενδιαφέρουν. Έχοντας κατανοήσει τη συμπεριφορά και τις ανάγκες ελέγχου του συγκεκριμένου κυκλώματος, η εργασία συνεχίζει με τη δημιουργία ενός πλήρως λειτουργικού συστήματος ελέγχου, αξιολογώντας την εφαρμογή τριών(μπορεί και παραπάνω) διαφορετικών αισθητηρίων μεταξύ των κόμβων του συστήματος. Τέλος, το σύστημα αυτό του ελέγχου, τίθεται σε λειτουργία μέσω του μικρό- ελεγκτή μας που αναφέραμε πιο πριν , επιτρέποντας την επεξεργασία των δεδομένων που αντλούν τα αισθητήρια και αναλαμβάνοντας την αποστολή τους σε ένα απομακρυσμένο σύστημα ενός δέκτη, δηλαδή ενός Server.

1.3 Διάρθρωση Διπλωματικής Εργασίας

Η διάρθρωση της παρούσας διπλωματικής εργασίας είναι η ακόλουθη:

- **Κεφάλαιο 2^ο** : Στο κεφάλαιο αυτό παρατίθενται το υλικό(Hardware) με τα οποία έγινε η υλοποίηση του ασύρματου πολύ-αισθητήρα ρευστών. Σε αυτό το κεφάλαιο αναλύεται και εξηγείται γιατί αγοράστηκαν τα προαναφερθέντα αισθητήρια όργανα και προτείνονται και άλλα πιθανά που θα μπορούσαν να χρησιμοποιηθούν προκειμένου να εξειδικευτεί η διπλωματική εργασία και σε περαιτέρω μετρήσεις. Επίσης, γίνεται μια λεπτομερής ανάλυση και σύγκριση βασικών χαρακτηριστικών μεταξύ των μικρό-ελεγκτών. Ο μικρό-ελεγκτής αποτελεί την κεντρική μονάδα του συστήματος μας όπως είπαμε, και είναι υπεύθυνος για την εμφάνιση των αποτελεσμάτων των αισθητηρίων σε μια βάση δεδομένων ή οπουδήποτε αλλού θέλουμε εμείς. Τέλος, περιλαμβάνεται και μια μικρή λίστα αναγκαίων επιπρόσθετων υλικών.

- **Κεφάλαιο 3^ο:** Στο Κεφάλαιο 3 παρουσιάζεται το λογισμικό (software) και τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος μας. Γίνεται αναφορά στα προγράμματα που χρησιμοποιήθηκαν, στην επικοινωνία μεταξύ τους, καθώς επίσης γίνεται μια ανάλυση για τους κώδικες που γράφτηκαν στο κάθε ένα από αυτά προκειμένου να γίνει πιο κατανοητή η λειτουργία τους. Επίσης, γίνεται μια αναφορά στον όρο Internet of Things ,που είναι ένας πολύ διαδεδομένος όρος τα τελευταία χρόνια, και η εργασία μας σχετίζεται άμεσα με αυτόν. Τέλος, αναφέρονται και τα πρωτόκολλα που υποστηρίζουν τα αισθητήρια που επιλέξαμε , όπως επίσης και τα πρωτόκολλα ασύρματης επικοινωνίας που είχαμε στην διάθεση μας και τα οποία καθιστούν εφικτή την ασύρματη μετάδοση δεδομένων.

- **Κεφάλαιο 4^ο :** Στο Κεφάλαιο 4 καταγράφεται βήμα προς βήμα η υλοποίηση του κατασκευαστικού μέρους της εργασίας. Αρχικά, παρατίθεται ο σχεδιασμός που έγινε μέσω του προγράμματος SolidWorks για το περίβλημα του κελύφους που θα περιέχει τα ηλεκτρονικά μέρη αλλά και ο σχεδιασμός της προσαρμογής των αισθητηρίων στο σωλήνα. Στην συνέχεια, ακολουθεί η υλοποίηση της προσαρμογής των αισθητηρίων στον σωλήνα δείχνοντας μέσω φωτογραφιών τα κατάλληλα εργαλεία που χρησιμοποιήθηκαν για την προσαρμογή, παρουσιάζοντας το τελικό αποτέλεσμα. Τέλος, παραθέτουμε την τελική κατασκευή για την οποία δημιουργήθηκε η εργασία μας.

- **Κεφάλαιο 5^ο :** Στο Κεφάλαιο 5 διακρίνουμε πειράματα που έγιναν για τον έλεγχο της σωστής λειτουργίας των αισθητήρων που χρησιμοποιήσαμε.

- **Κεφάλαιο 6^ο :** Στο Κεφάλαιο 6 παραθέτουμε μελλοντικές επεκτάσεις που θα ήταν πολύ ενδιαφέρουσες να προστεθούν στην εργασία μας.

2 Hardware

2.1 Αισθητήρια

Σε αυτό το κεφάλαιο θα αναλύσουμε και θα εξηγήσουμε γιατί αγοράσαμε τα προαναφερθέντα αισθητήρια όργανα και θα προτείνουμε και άλλα πιθανά που θα μπορούσαν να χρησιμοποιηθούν προκειμένου να εξειδικευτεί η διπλωματική εργασία και σε περαιτέρω μετρήσεις.

2.1.1 Αισθητήριο Θερμοκρασίας

Το αισθητήριο θερμοκρασίας σαν γενική έννοια είναι μια ηλεκτρονική συσκευή που μας δίνει τη δυνατότητα να μετράμε τις θερμοκρασίες περιβάλλοντος

Εμείς έχοντας ορισμένες επιλογές στην διάθεση μας, αγοράσαμε το αισθητήριο με ονομασία Waterproof DS18B20 Sensor Kit(KIT0021), το οποίο έχει τάση λειτουργίας 3-5.5V και ακρίβεια ± 0.5 όταν τοποθετείται σε υγρά θερμοκρασίας $-10-85^{\circ}\text{C}$. Επίσης, οι θερμοκρασίες στις οποίες μπορεί να μετρήσει είναι από $-55-125^{\circ}\text{C}$ και έχει δυνατότητα επιλογής από 9 έως και 12 bits ανάλυσης, δηλαδή διακριτικής ικανότητας, πράγμα που μπορεί να μας δώσει ακριβέστερες μετρήσεις. Παρακάτω απεικονίζεται το αισθητήριο θερμοκρασίας που αγοράσαμε.



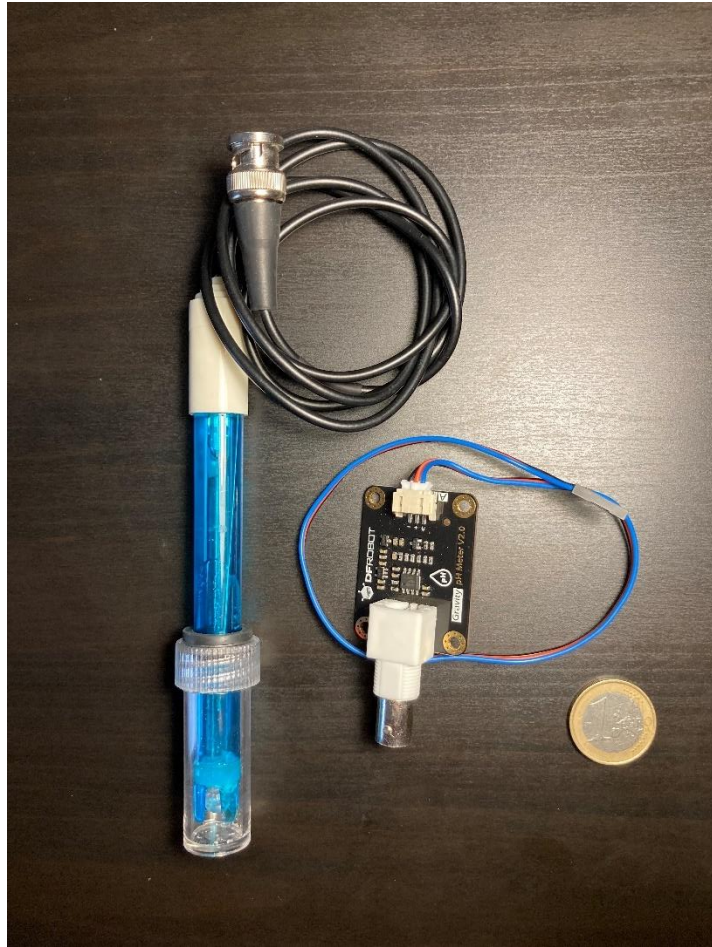
Σχήμα 2.1 : Αισθητήριο Θερμοκρασίας μαζί με την πλακέτα διασύνδεσής της με τον Esp32.

2.1.2 Αισθητήριο PH

Το αισθητήριο PH είναι μια ηλεκτρική συσκευή που χρησιμοποιείται για να μετρήσει τα ιόντα υδρογόνου ή αλλιώς την οξύτητα και την αλκαλικότητα στα διάφορα διαλύματα. Ουσιαστικά ένα τέτοιο αισθητήριο αποτελείται από ένα βολτόμετρο το οποίο είναι προσδεμένο σε ένα ηλεκτρόδιο απόκρισης (responsive electrode) και σε ένα ηλεκτρόδιο αναφοράς (reference electrode). Το ηλεκτρόδιο απόκρισης είναι συνήθως γυαλί και το ηλεκτρόδιο αναφοράς είναι συνήθως από άργυρο-άργυρο, αν και μερικές φορές χρησιμοποιούνται και ηλεκτρόδια αναφοράς από υδράργυρο-χλωριούχο υδράργυρο. Όταν τα δύο προαναφερθέντα ηλεκτρόδια βυθιστούν σε ένα διάλυμα λειτουργούν όπως μία μπαταρία. Το γυάλινο ηλεκτρόδιο απόκρισης αναπτύσσει ένα ηλεκτρικό δυναμικό το οποίο άμεσα σχετίζεται με την δραστηριότητα των ιόντων υδρογόνου. Το βολτόμετρο μετρά την διαφορά δυναμικού μεταξύ του ηλεκτροδίου

απόκρισης και του ηλεκτροδίου αναφοράς. Το ηλεκτρόδιο αναφοράς δεν ποικίλλει ανάλογα με την δραστηριότητα των ιόντων υδρογόνου. Το ηλεκτρόδιο pH εξαιτίας της φύσης της κατασκευής του πρέπει προκειμένου να λειτουργεί σωστά, να είναι διαρκώς υγρό. Η ενυδάτωση είναι απαραίτητη προκειμένου να καθίσταται δυνατή η ανταλλαγή των ιόντων. Αν το αισθητήριο pH στεγνώσει, είναι καλό να τοποθετηθεί σε νερό βρύσης έτσι ώστε να προετοιμαστεί το γυαλί. Τέλος για το συγκεκριμένο αισθητήριο δεν είναι σωστό να παραληφθεί ότι είναι απαραίτητο να γίνεται σε τακτική βάση μια διαδικασία ισοστάθμισης(calibration) του αισθητηρίου, καθώς με το πέρασμα του χρόνου το ηλεκτρόδιο παλιώνει, και το γυαλί του αλλάζει αντίσταση. Αισθητήρια Ph χρησιμοποιούνται σε ποιοτικές μετρήσεις ιδιοτήτων του εδάφους στον αγροτικό τομέα, σε δημοτικές παροχές νερού, σε πισίνες ,σε προσπάθειες περιβαλλοντικής αποκατάστασης, για την παρασκευή κρασιού ή μύρας, για την φροντίδα υγείας, για κλινικές εφαρμογές στα μικροβιολογικά εργαστήρια και σε πολλές άλλες εφαρμογές.

Εμείς για την εργασία μας επιλέξαμε να χρησιμοποιήσουμε τον αισθητήρα με ονομασία Gravity: Analog pH Sensor/Meter Kit V2 (SEN0161-V2) ο οποίος έχει ονομαστική τάση λειτουργίας 3.5-5.5V και μπορεί να μετράει από pH=0-14. Επίσης, έχει ανοχή σε θερμοκρασίες από 5-60°C, και έχει χρόνο απόκρισης μικρότερο των 2 ms ,με ακρίβεια ± 0.1 βαθμό κελσίου. [1] [2]



Σχήμα 2.2 : Αισθητήριο Ph, μαζί με την πλακέτα διασύνδεσης με τον Esp32.

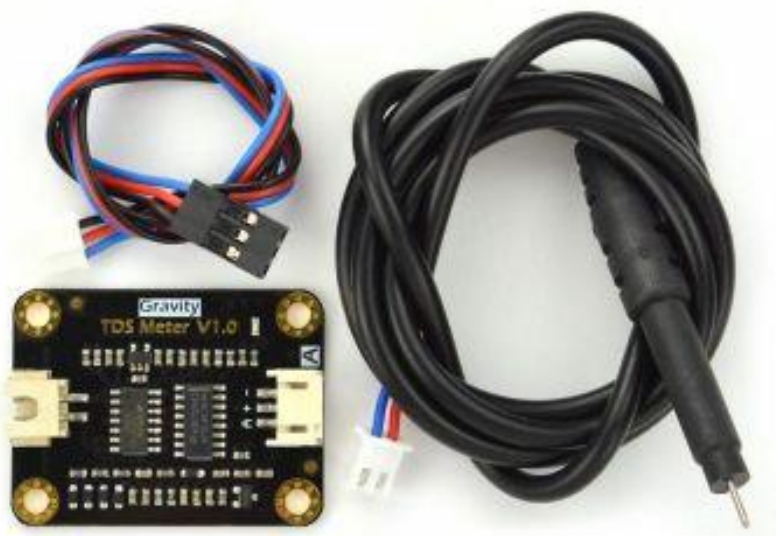
2.2 Επιπρόσθετα Αισθητήρια

Φυσικά στην εργασία μας θα ήταν δυνατό να χρησιμοποιηθούν και άλλα επιπλέον αισθητήρια μέτρησης τα οποία θα μπορούσαν να κάνουν την εφαρμογή μας ακόμα πιο εξειδικευμένη και για επαγγελματικούς σκοπούς. Παρακάτω παραθέτουμε μερικά από αυτά και επισυνάπτουμε κάποιες γενικές πληροφορίες για το κάθε ένα από αυτά.

2.2.1 Αισθητήριο TDS

Ένα αισθητήριο TDS είναι μια συσκευή που χρησιμοποιείται για να υποδεικνύει τα συνολικά διαλυμένα στερεά (Total Dissolved Solids) σε ένα διάλυμα το οποίο συνήθως είναι νερό. Εφόσον τα διαλυμένα ιονισμένα στερεά, όπως τα άλατα και τα μέταλλα αυξάνουν την αγωγιμότητα του διαλύματος, ένα αισθητήριο TDS μετράει την αγωγιμότητα του διαλύματος και υπολογίζει το TDS από αυτή την μέτρηση. Η κύρια εφαρμογή των TDS είναι στη μελέτη της ποιότητας του νερού για ρυάκια, ποταμούς και λίμνες, αν και τα TDS δεν θεωρούνται γενικά ως βασικοί ρύποι (π.χ. δεν

θεωρούνται ότι συσχετίζονται με επιπτώσεις υγείας) ,χρησιμοποιούνται ως μια ένδειξη των αισθητικών χαρακτηριστικών του πόσιμου νερού και ως ένας συνολικός δείκτης της παρουσίας ενός ευρέος φάσματος χημικών ρύπων. Τα πιο συνηθισμένα χημικά συστατικά είναι ασβέστιο, φωσφορικά, νιτρικά άλατα, νάτριο, κάλιο και χλώριο, που βρίσκεται σε απορροή θρεπτικών συστατικών, γενική απορροή όμβριων υδάτων, και σε κλίματα με χιόνια όπου ρίχνεται αλάτι για ξεπάγωμα των δρόμων. Οι δύο βασικές μέθοδοι μέτρησης των ολικών διαλυμένων στερεών είναι η σταθμική ανάλυση και η αγωγιμότητα. Οι σταθμικές μέθοδοι είναι οι πιο ακριβείς αν και είναι χρονοβόρες. Εμείς θα ασχοληθούμε με την μέτρηση του TDS με την βοήθεια της αγωγιμότητας, η οποία όταν συσχετιστεί με εργαστηριακές μετρήσεις TDS, παρέχει μια προσεγγιστική τιμή για τη συγκέντρωση των χημικών συστατικών, συνήθως με απόκλιση της ακρίβειας έως 10%. [3]



Σχήμα 2.3 : Αισθητήριο TDS (https://grobotronics.com/gravity-analog-tds-sensor-meter.html?sl=en&fbclid=IwAR2vVhMFtS595Lni-kprIjO5l-axOS_1QvfbNyr01PtvngtXgiiDHxhH4xU, Grobotronics , εταιρία με εμπόριο ηλεκτρονικών ειδών).

2.2.2 Αισθητήριο Θολότητας

Το αισθητήριο θολότητας αξιολογεί την ποιότητα του νερού μετρώντας το επίπεδο θολότητας ή αδιαφάνειας που βρίσκεται σε αυτό. Χρησιμοποιεί το φως για να ανιχνεύσει αιωρούμενα σωματίδια που βρίσκονται στο νερό, μετρώντας την μετάδοση του φωτός και του ρυθμού σκέδασης, ο οποίος αλλάζει με το ποσοστό των συνολικών αιωρούμενων στερεών (TTS) μέσα στο νερό. Όσο αυτή η παράμετρος αυξάνεται, το επίπεδο θολότητας του νερού αυξάνεται. Αισθητήρια θολότητας χρησιμοποιούνται σε ποτάμια, ρυάκια (streams), λύματα, σε μετρήσεις εκροών, σαν όργανα ελέγχου σε καθιζήσεις λιμνών, σε έρευνες μεταφοράς ιζημάτων, και σε εργαστηριακές μετρήσεις. Μπορούμε να επιλέξουμε την λειτουργία και με βάση τον μικρό-ελεγκτή που χρησιμοποιούμε. [4]



Σχήμα 2.4 : Αισθητήριο Θολότητας (Turbidity Sensor), (https://www.hellasdigital.gr/electronics/sensors/humidity-sensor/gravity-analog-turbidity-sensor-for-arduino-sen0189/?fbclid=IwAR3TEG9sx_jG0I7X8dvxq2kgqPAvIAuy-chkN9ismDMym2ehhi2TesVH5g, Hellas Digital, εταιρία με εμπόριο ηλεκτρονικών ειδών).

2.2.3 Αισθητήριο Πίεσης

Ένας αισθητήρας πίεσης είναι μια συσκευή για μέτρηση πίεσης αερίων και ρευστών. Η Πίεση είναι έκφραση της απαιτούμενης δύναμης που χρειάζεται για να σταματήσει ένα υγρό να επεκτείνεται, και καθορίζεται συνήθως σε όρους δύναμης ανά περιοχή μονάδας. Ένας αισθητήρας πίεσης συνήθως συμπεριφέρεται σαν μορφοτροπέας. Παράγει ένα σήμα ως συνάρτηση της επιβαλλόμενης πίεσης(ηλεκτρικό σήμα).

Αισθητήρες πίεσης χρησιμοποιούνται για έλεγχο και παρακολούθηση σε χιλιάδες καθημερινές εφαρμογές. Οι αισθητήρες πίεσης ποικίλλουν δραστικά ανάλογα την

τεχνολογία, τον σχεδιασμό, την απόδοση, την καταλληλότητα σε εφαρμογές και στο κόστος. Μια συντηρητική εκτίμηση θα ήταν ότι υπάρχουνε πάνω από 50 είδη τεχνολογίας και τουλάχιστον 300 εταιρίες που κατασκευάζουν αισθητήρες πίεσης παγκοσμίως.

Υπάρχουνε και αισθητήρες πίεσης που είναι σχεδιασμένοι να μετράνε σε μια δυναμική λειτουργία για να συλλαμβάνουνε πολύ γρήγορες αλλαγές στην πίεση. Παραδείγματα τέτοιου αισθητηρίου είναι στην μέτρηση της πίεσης καύσης σε έναν κύλινδρο μηχανής ή σε μια τουρμπίνα αερίου. Αισθητήρες τέτοιου τύπου κατασκευάζονται από πιεζό-ηλεκτρικά υλικά όπως ο χαλαζίας. [5]



Σχήμα 2.5 : Αισθητήριο Πίεσης, (<https://www.hellasdigital.gr/electronics/sensors/pressure-sensors/g1-4-inch-5v-0-1.2-mpa-pressure-transducer-sensor-oil-fuel-diesel-gas-water-air/?fbclid=IwAR3u-25IzrZTVyr5IoxD6iOKXin2jhBubA8bc-lu2qxpBW0R8DW0qjP7YF8>, Hellas Digital, εταιρία με εμπόριο ηλεκτρονικών ειδών).

Παρακάτω παραθέτουμε πινακάκια τα οποία περιλαμβάνουν βασικά χαρακτηριστικά από τα αισθητήρια που επιλέξαμε :

Waterproof DS18B20 Sensor Kit(KIT0021)	
Operating voltage	3.0~ 5.5V
±0.5°C Accuracy from	-10°C to +85°C
Usable temperature range	-55 to 125°C (-67°F to +257°F)
Resolution	9 to 12 bit

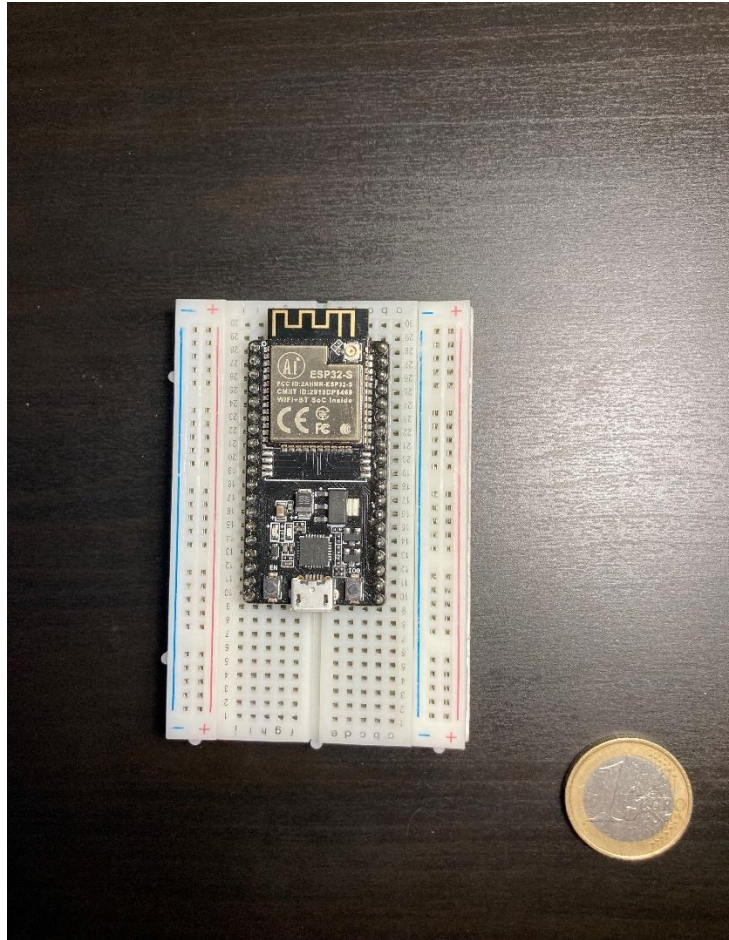
Gravity: Analog pH Sensor/Meter Kit V2 (SEN0161-V2)		
Signal Conversion Board (Transmitter) V2	Supply Voltage	3.3~5.5V
	Output Voltage	0~3.0V
	Measurement Accuracy	±0.1@25°C
pH Probe	Response Time	<2min
	Detection Range	0~14
	Temperature Range	5~60°C
	Probe Life	>0.5 year (depending on frequency of use)

2.3 Μικρό-Ελεγκτής

Όσον αφορά το κομμάτι του μικρό-ελεγκτή που είναι και το πιο σημαντικό για τον ασύρματο πολύ-αισθητήρα ρευστών, είχαμε διάφορες επιλογές, μέχρι να καταλήξουμε στην αγορά του ESP32. Αυτές οι πλατφόρμες ήτανε του Arduino, και του MSP430 κυρίως, καθώς και άλλων παρεμφερών όπως του Nordic NRF52832 , NRF52840.

2.3.1 Σύγκριση ελεγκτών όσον αφορά την αρχιτεκτονική τους

Όσον αφορά την σύγκριση των bits των ελεγκτών που αναφέραμε, κερδίζουν ο ESP32 και οι δύο Nordic καθώς είναι 32 bits ,ενώ οι ελεγκτές Arduino βγαίνουνε συνήθως σε εκδοχές των 8 bits, ενώ οι MSP430 είναι 16bits. Μας ενδιαφέρει ο μεγάλος αριθμός bits επειδή αυτοί οι ελεγκτές είναι πιο γρήγοροι σε επεξεργαστική ικανότητα, καθώς επίσης και μπορούν να ρυθμιστούν στο ρολόι τους μέχρι και αρκετά εκατοντάδες MHz συχνότητα ενώ οι 8-μπιτοι επεξεργαστές μπορούνε μόνο μέχρι 8MHz. Επίσης οι 32-μπιτοι επεξεργαστές συνήθως έχουνε οκταπλάσια ram σε σύγκριση με τους 8-μπιτους, οπότε και σε αυτή την περίπτωση μας βολεύει καθώς μπορούμε να αναπτύξουμε ένα πρόγραμμα για την κατασκευή μας, στο οποίο να μην έχουμε περιορισμούς όσον αφορά τον αριθμό των μεταβλητών που θα δηλώσουμε. [6]

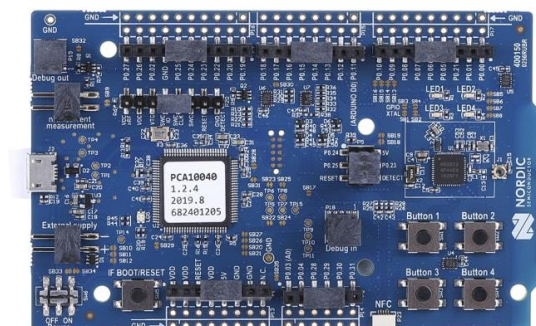


Σχήμα 2.6 : Πλατφόρμα ESP32 NODEMCU-32S.

2.3.2 Σύγκριση ελεγκτών όσον αφορά τον ADC

Όπως είναι γνωστό ένας A/D converter είναι αυτή η συσκευή η οποία στην περίπτωση του μικρο-ελεγκτή είναι και ενσωματωμένη, και μετατρέπει το αναλογικό σήμα το οποίο λαμβάνει το αισθητήριο μας από το περιβάλλον, σε κάποια ψηφιακή τιμή τάσης, την οποία στην συνέχεια θα είναι υπεύθυνος ο ελεγκτής να επεξεργαστεί. Στον ADC διακρίνουμε δύο συγκριτικά χαρακτηριστικά τα οποία μας αφορούν και αυτά είναι η συχνότητα του και η ανάλυση του. Όσον αφορά την ανάλυση, αυτή είναι η διακριτική ικανότητα που θα έχει ο ADC ώστε να αναπαραστήσει το πραγματικό σήμα μας, σε ψηφιακό, και μετριέται σε bits. Είναι επομένως φανερό πως όσο μεγαλύτερη διακριτική ικανότητα έχει ο ADC τόσο πιο ακριβής θα είναι η αναπαράσταση ενός αναλογικού σήματος με την πάροδο του χρόνου. Ο απλός Arduino Uno έχει ADC ανάλυσης 10-bits. Στην πλατφόρμα MSP430, στην σειρά MSP430x5xx υπάρχει μετατροπέας ADC με 10 & 12- bits ενώ στις υπόλοιπες σειρές της ίδιας πλατφόρμας διαφέρουν. Ο Nordic NRF52832 και ο NRF52840 ενώ και αυτοί έχουν στάνταρ ανάλυση του ADC 10-bits,

μπορούν να ρυθμιστούν μέσω χρήσης συνάρτησης να μετατρέπουν μέχρι και 14-bits δεδομένων. Ο ESP32 που επιλέξαμε έχει ανάλυση ADC 12-bits. Σε αυτή την περίπτωση βλέπουμε πως νικάει η ομάδα ελεγκτών της εταιρίας Nordic, αλλά υπάρχουνε και άλλες παράμετροι οι οποίες θα αναφερθούν παρακάτω και αιτιολογούν την επιλογή του ESP32. [7]



Σχήμα 2.7 : Πλατφόρμα Nordic Nrf52832 (https://eg.rsdelivers.com/product/nordic-semiconductor/nrf52-dk/nordic-semiconductor-nrf52810-nrf52832-bluetooth/1463538?fbclid=IwAR2vn8L9CDjaXV_Gbnb0BA9s1P3_1aOd9_iWaHOXk3HjFDHT7WKX6cZuoBg, ιστοσελίδα του εξωτερικού με εμπόριο ηλεκτρονικών ειδών).

2.3.3 Σύγκριση ελεγκτών όσον αφορά την μνήμη τους

Σε αυτό το υπό-κεφάλαιο αναφέρουμε και συγκρίνουμε την μνήμη που υπάρχει σε κάθε πλατφόρμα. Αρχικά, διακρίνουμε δύο είδη μνήμης, τα οποία είναι η Flash Memory και η μνήμη Ram, η οποία με τη σειρά της χωρίζεται στην στατική (Static Ram ή SRAM) και στην δυναμική Ram (Dynamic Ram ή DRAM). Τα δύο αυτά είδη της ram διαφέρουνε όσον αφορά την αποθήκευση δεδομένων. Η αρχιτεκτονική τους είναι τελείως διαφορετική, με την στατική να είναι πολύ πιο γρήγορη από την δυναμική καθώς αποτελείται από μια διάταξη flip-flops όπου είναι διαμορφωμένα έτσι ώστε να συγκρατούν κάθε bit μνήμης. Κάθε flip-flop το οποίο αποτελεί ένα κύτταρο μνήμης έχει 4-6 τρανζίστορες τα οποία είναι συνδεδεμένα μεταξύ τους και δεν χρειάζεται ποτέ να ανανεωθούν. Αυτό το γεγονός την καθιστά πολύ πιο γρήγορη από την δυναμική ram, αλλά επίσης και πιο δαπανηρή καθώς για να δημιουργηθεί ένα κύτταρο στατικής μνήμης καταλαμβάνεται περισσότερος χώρος σε ένα τσιπ μνήμης, και αυτό έχει σαν

αποτέλεσμα να υπάρχει λιγότερη μνήμη σε κάθε τσιπ στατικής μνήμης από ότι σε ένα τσιπ δυναμικής μνήμης, και αυτό την καθιστά αυτομάτως και πιο ακριβή. Η μνήμη flash memory με την σειρά της, είναι ένα είδος μόνιμης μνήμης το οποίο χρησιμοποιείται για μεταφορά δεδομένων μεταξύ ενός προσωπικού υπολογιστή και ψηφιακών συσκευών. Έχει την δυνατότητα να επαναπρογραμματιστεί ηλεκτρονικά και να διαγραφεί. Αποτελεί read only (EEPROM) μνήμη , αλλά μπορεί να είναι και αυτόνομη συσκευή αποθήκευσης όπως ένα USB drive. Παρακάτω θα συνοψίσουμε περίπου τη μνήμη έχει κάθε πλατφόρμα από τις υποψήφιες της λίστας μας, ώστε να έχουμε μια καλύτερη εποπτεία και για να αιτιολογήσουμε την επιλογή μας στην πλατφόρμα ESP32 NODE-MCU32s. [8] [9] [10] [11]



Σχήμα 2.8 : Πλατφόρμα Msp430 LaunchPad (<https://www.ebay.com/itm/TI-MSP430-LaunchPad-Value-Line-Development-kit-Texas-Instruments-MSP-EXP430G2/171074970306?fbclid=IwAR1dRKYKEJEvmtGMrbPIXF-TaPmz4UobF3w2MuaVI3GdDJJeqRugrOK2A80> , Ebay).

2.3.4 Σύγκριση όσον αφορά την μνήμη Flash Memory

- Για την πλατφόρμα Nordic NRF52832 υπάρχουν εκδοχές που έχουν 256 KB μνήμη ενώ υπάρχουν και εκδοχές που έχουν 512 KB. Η εταιρία Nordic έχει βγάλει και μία ακόμα πλατφόρμα με όνομα NRF52840 η οποία έχει μνήμη 1 Mb.
- Η πλατφόρμα Arduino UNO έχει μνήμη 32 KB η οποία επαληθεύει την χρήση του Arduino ως εκπαιδευτικού κυρίως εργαλείου που είναι, και χρησιμοποιείται για πιο μικρά projects τα οποία ενδεχομένως να μην χρειάζονται πολλές μεταβλητές και αισθητήρια όργανα με πολλαπλές μετρήσεις.

- Όσον αφορά τον MSP430 υπάρχουν διάφορες εκδοχές του που έχουν τις δικιές τους μνήμες flash memory. Για παράδειγμα η σειρά MSP430F2xx έχει επιλογή από 1-120KB, ενώ η σειρά MSP430x6xx SERIES φτάνει μέχρι και 512 KB flash memory.
- Τέλος ο ESP32 που ήταν εν τέλει και η επιλογή μας, έχει μνήμη flash memory 4 Mb το οποίο την καθιστά την μεγαλύτερη από όλες τις προηγούμενες πλατφόρμες, και ένα σημαντικό πλεονέκτημα με την συγκεκριμένη πλατφόρμα είναι ότι μπορεί να διατηρήσει την flash memory και μετά από την ανακύκλωση της ισχύς. Αυτό είναι πολύ χρήσιμο καθώς συγκεκριμένη ποσότητα δεδομένων μπορεί να αποθηκευτεί ακόμα και μετά το κλείσιμο της συσκευής.



Σχήμα 2.9 : Πλατφόρμα Arduino Uno (https://www.hellasdigital.gr/go-create/arduino/arduino-uno-r3-atmega328p-board-dip-version/?fbclid=IwAR0dD_YNCJf4aSzyZVzvdBViAZ20qOduhNphpkOKFuE8q0UzRcPyIbhOnRs, Hellas Digital, εταιρία με εμπόριο ηλεκτρονικών ειδών).

2.3.5 Σύγκριση όσον αφορά την SRAM

- Για την πλατφόρμα Nordic NRF52832 η μνήμη SRAM είναι 32 KB σε αυτή που έχει flash memory 256 KB , ενώ σε αυτές που έχουν 512 KB, η SRAM είναι 64 KB. Οι πλατφόρμες με τις μεγάλες μνήμες είναι οι NRF52832-QFAA και η NRF52832-CIAA, ενώ αυτή με την μικρή είναι η NRF52832-QFAB. Εδώ αξίζει να επισημάνουμε και την μνήμη στην πλατφόρμα NRF52840 την οποία αναφέραμε και πιο πριν. Αυτή έχει SRAM ίση με 256 KB η οποία είναι πολύ μεγαλύτερη από το μοντέλο NRF52832, και αυτό είναι λογικό καθώς και η flash memory του ήταν 1 Mb, που ήταν διπλάσια από την flash memory του NRF52832.

- Ο Arduino UNO έχει SRAM ίση με 2 KB η οποία είναι πολύ μικρή και είναι ανάλογη επίσης και με το μέγεθος της flash memory.
- Ο MSP430 της Texas Instrument έχει SRAM που ποικίλλει ανάλογα με το μοντέλο. Η σειρά MSP430x1xx SERIES έχει δυνατότητα SRAM από 128b-10kb ,ενώ οι σειρές MSP430x5xx SERIES και MSP430x6xx SERIES έχουν δυνατότητα μέχρι και 66 KB.
- Ο ESP32 που πήραμε είναι και πάλι ο νικητής στο θέμα της SRAM καθώς είχε και την πιο μεγάλη flash memory, που όπως είδαμε και με τους Nordic πριν, αυτά τα είδη μνήμης ενώ μεν είναι διαφορετικά, σε μέγεθος πάνε αναλογικά. Ο ESP32 επομένως με τα 4mb flash memory, συνοδεύεται από μέγεθος SRAM ίσο με 520 KB.

Παρακάτω παραθέτουμε ένα πίνακα στον οποίο φαίνονται πιο συνοπτικά τα συγκριτικά χαρακτηριστικά όλων των μικρό-ελεγκτών που λάβαμε υπόψιν μας. Στο συγκεκριμένο πίνακα μερικά χαρακτηριστικά των δεδομένων που δώσαμε προηγουμένως μπορεί να διαφέρουν, καθώς σε μερικές περιπτώσεις επισημαίνονται διαφορετικά μοντέλα:

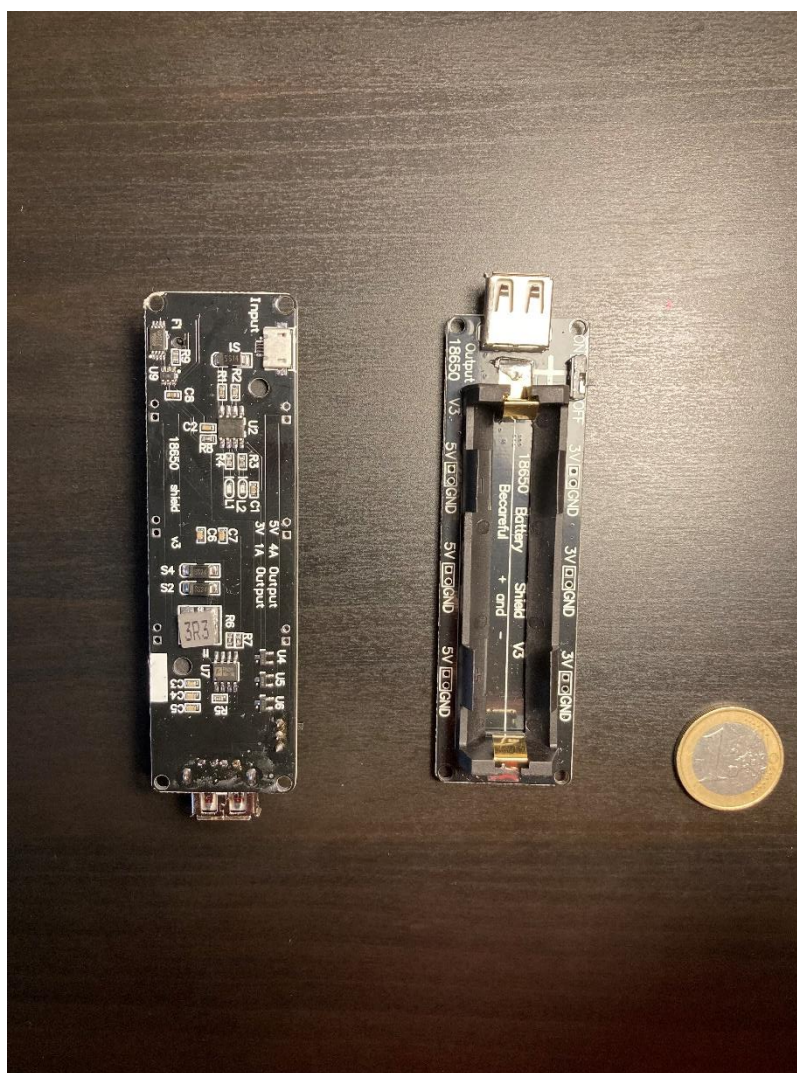
	TI MSP430	Arduino UNO	nRF52832	nRF52840	ESP32- S3 Series
Αρχιτεκτονική	<u>16-bit</u>	ATmega328/P CMOS <u>8-bit</u> microcontroller based on the AVR® enhanced RISC (reduced instruction set computer) architecture	<u>32-bit</u> ARM Cortex M4F processor	<u>32-bit</u> ARM® Cortex™-M4	Dual-core LX7 <u>32-bit</u> processor operating up to 240 MHz

ADC Μετατροπέας	MSP430G2xx series : <u>10-bit</u> MSP430x5xx series: <u>10 & 12-bit SAR</u> MSP430x6xx series: 12-bit SAR	<u>10-bit</u>	<u>10-bit</u> (it can be configured to convert data with up to 14-bit resolution)	<u>10-bit</u> (it can be configured to convert data with up to 14-bit resolution)	<u>12-bit SAR</u> (“Successive Approximation Register”) ADC
Flash Memory	MSP430F2xx series: <u>1-120 KB</u> MSP430x6xx series: <u>up to 512 KB</u>	<u>32 KB</u> (ATmega328P) of which 0.5 KB used by bootloader	<u>512/256 KB</u> Flash	<u>1 MB</u> Flash	<u>8MB</u> of internal flash memory(4MB Flash Memory έχουνε τα μοντέλα της σειράς s2).
SRAM	MSP430x1xx series: <u>128 B– 10 KB</u> MSP430x5xx series: <u>up to 66 KB</u> MSP430x6xx series: <u>up to 66 KB</u>	<u>2 KB</u> (ATmega328P)	<u>64/32 KB</u> RAM	<u>256 KB</u> RAM	<u>512 KB</u>

2.4 Επιπρόσθετα Υλικά για την Υλοποίηση του Hardware

Ακολουθεί μια λίστα επιπρόσθετων υλικών που χρειάστηκαν για να ολοκληρωθεί η υλοποίηση του ασύρματου πολύ-αισθητήρα ρευστών καθώς και μερικά λόγια για το κάθε ένα από αυτά:

- Battery Charger 18650 with Step-Up 5V – USB, το οποίο είναι μια ηλεκτρονική διάταξη δομημένη έτσι, ώστε να τροφοδοτεί με επαναφορτιζόμενη μπαταρία λιθίου το σύστημα μας. Η μπαταρία λιθίου που είναι συμβατή για αυτή την διάταξη δίνει 3 volts (στην περίπτωση μας αγοράσαμε 3.6 V μπαταρία), την οποία τάση το κύκλωμα ανορθώνει σε 5Volts τα οποία πηγαίνουνε μέσω καλωδίου USB-micro USB στην υποδοχή Vin της πλατφόρμας ESP32.



Σχήμα 2.10 : Τροφοδοτικό μονάδας ESP32.

- Rechargeable Battery 18650 3.6V 3350 MAh, η οποία είναι η επαναφορτιζόμενη μπαταρία λιθίου που προτιμήσαμε για να τοποθετήσουμε στην μπαταριοθήκη και έχει χωρητικότητα 3350 MAh.



Σχήμα 2.11 : Επαναφορτιζόμενη μπαταρία λιθίου Li-ion.

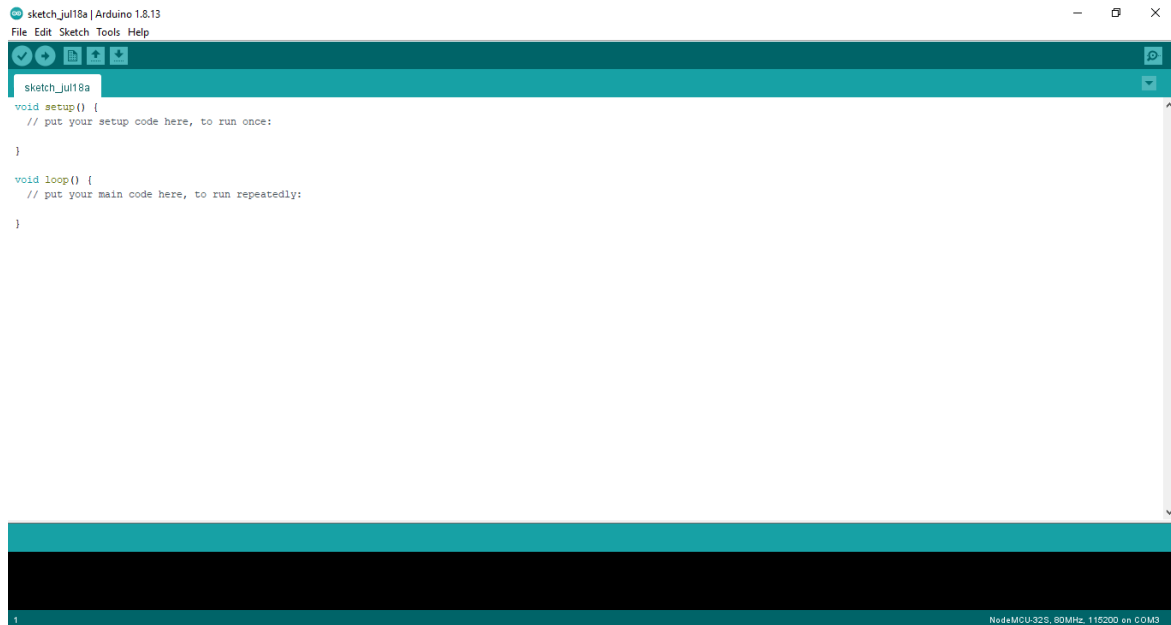
3 Εργαλεία Λογισμικού

Σε αυτό το κεφάλαιο θα αναφερθούν τα προγράμματα που χρησιμοποιήσαμε για την πραγματοποίηση του συστήματος μας, καθώς και τις προσαρμογές που χρειάστηκαν προκειμένου να χρησιμοποιηθούν. Επίσης, θα εξηγηθούν με τον καλύτερο δυνατό τρόπο τα κομμάτια κώδικα που γράφτηκαν.

3.1 ARDUINO IDE

Το πρόγραμμα το οποίο χρησιμοποιήθηκε για τον προγραμματισμό του ESP32 και των αισθητηρίων είναι το Arduino ide, το οποίο είναι το κατεξοχήν πρόγραμμα που χρησιμοποιείται για τον προγραμματισμό των μικρό-ελεγκτών της εταιρίας Arduino ,αλλά είδαμε ότι με μερικές προσαρμογές μπορεί να εξυπηρετήσει και άλλες πλατφόρμες, όπως η δική μας.

Αρχικά χρειάστηκε μέσα από την εφαρμογή Arduino ide να κατεβάσουμε ένα πακέτο με το όνομα ESPRESSIF το οποίο γενικά λειτουργεί και σαν αυτοτελές πρόγραμμα με το όνομα ESPRESSIF IDF, και το οποίο χρησιμοποιείται για ανάπτυξη εφαρμογών IOT. Το προαναφερθέν αρχείο το προσθέτουμε από τον τομέα File>Preferences και τοποθετούμε το πακέτο με όνομα https://dl.espressif.com/dl/package_esp32_index.json, http://arduino.esp8266.com/stable/package_esp8266com_index.json. Αυτό το πακέτο αφού το προσθέσουμε, μας δίνει την δυνατότητα να εγκαταστήσουμε την κατάλληλη βιβλιοθήκη του ESP32 από την ESPRESSIF ,για το Arduino IDE. Αφού την εγκαταστήσουμε στο πρόγραμμα, έχουμε πλέον την δυνατότητα μέσα από το Arduino IDE να επιλέξουμε μία από τις πολλαπλές επιλογές για πλατφόρμες ESP32 , από αυτές που μας δίνονται. Εμείς επιλέξαμε την NODEMCU-32S καθώς αυτό είναι το μοντέλο του ESP32 που αγοράσαμε. Μετά από αυτά τα βήματα, και εφόσον φυσικά επιλέξουμε πρώτα μια σειριακή θύρα επικοινωνίας της πλατφόρμας με τον υπολογιστή, μπορούμε πλέον να προγραμματίσουμε και να περάσουμε το πρόγραμμα στην συσκευή μας. [12]

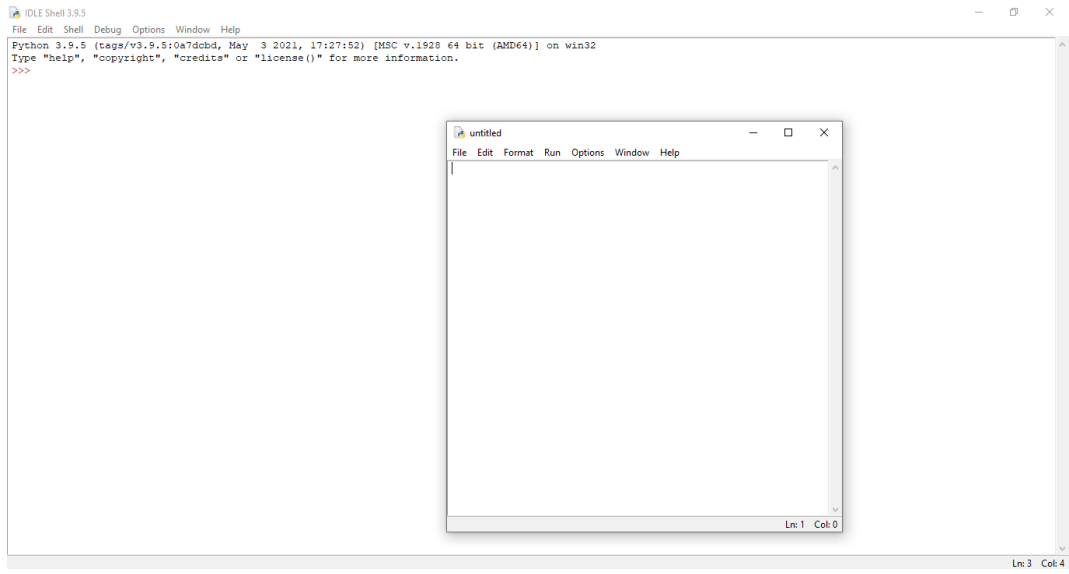


Σχήμα 3.1 : Περιβάλλον προγράμματος Arduino IDE.

3.2 IDLE PYTHON

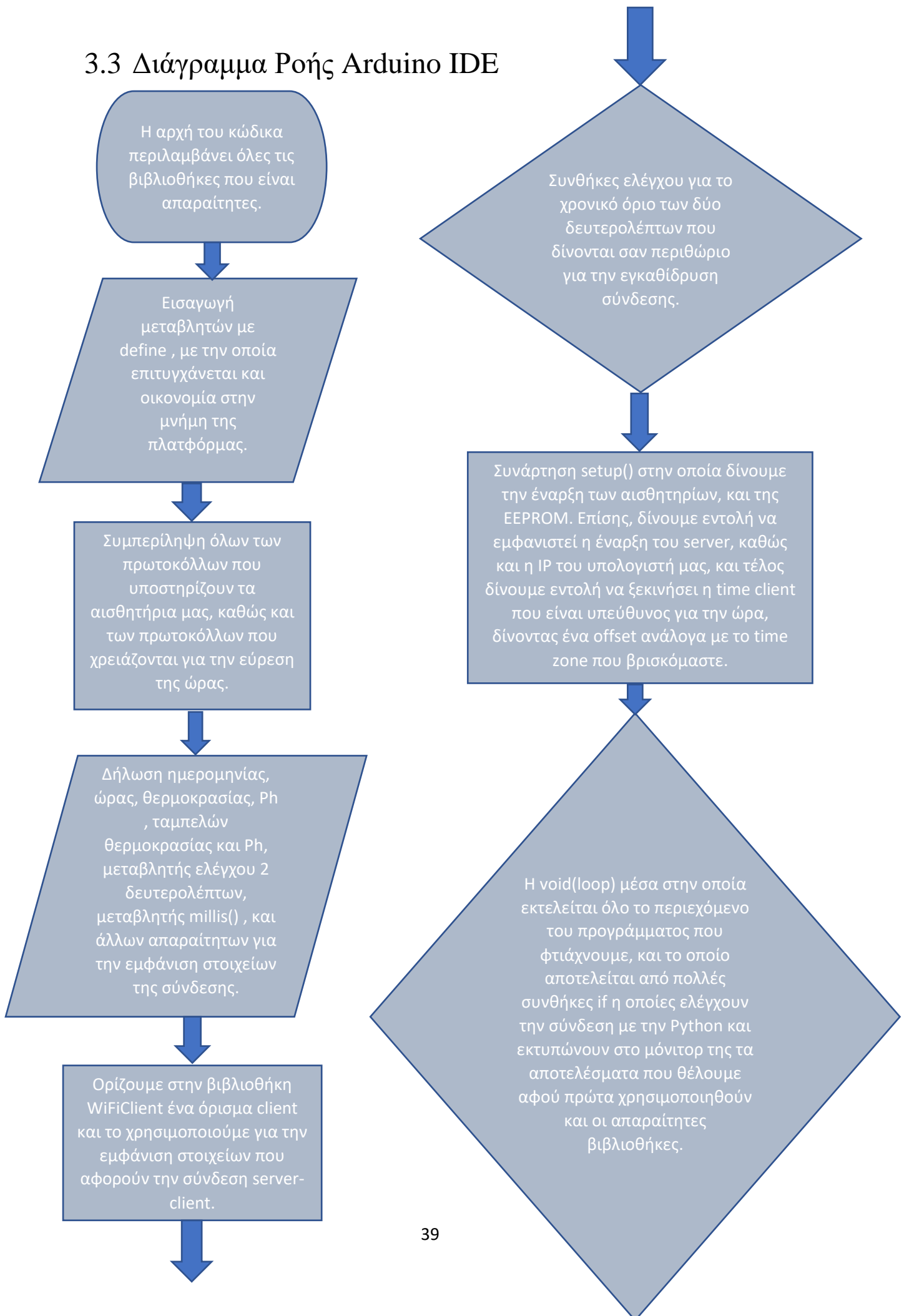
Γενικότερα η python είναι μια γλώσσα προγραμματισμού γενικής χρήσης η οποία χρησιμοποιείται σε πολλούς τομείς. Μερικά παραδείγματα χρήσης της είναι για ανάπτυξη δικτύου, για λειτουργικά συστήματα, για αριθμομηχανές, για εφαρμογές υπολογιστή/κινητού, για τεχνητή νοημοσύνη και εκμάθηση μηχανών καθώς και πολλές άλλες. Είναι εύκολη για να την μάθει κάποιος, και ακολουθεί μια απλή και οργανωμένη δομή. Επίσης έχει μεγάλη ευελιξία και απλή σύνταξη, κάτι που την καθιστά κατάλληλη γλώσσα για όλων των ειδών projects.

Η γλώσσα προγραμματισμού Python έχει και το δικό της πρόγραμμα το οποίο ονομάζεται με το ίδιο όνομα. Εγκαθιστώντας το, εγκαθίσταται και το πρόγραμμα Python IDLE. Το πρόγραμμα αυτό είναι ένα ενσωματωμένο περιβάλλον ανάπτυξης και μάθησης για Python (INTEGRATED DEVELOPMENT and LEARNING ENVIRONMENT→IDLE). [13]



Σχήμα 3.2 : Περιβάλλον προγραμματισμού IDLE Python.

3.3 Διάγραμμα Ροής Arduino IDE



3.3.1 Εξήγηση Κώδικα Arduino IDE

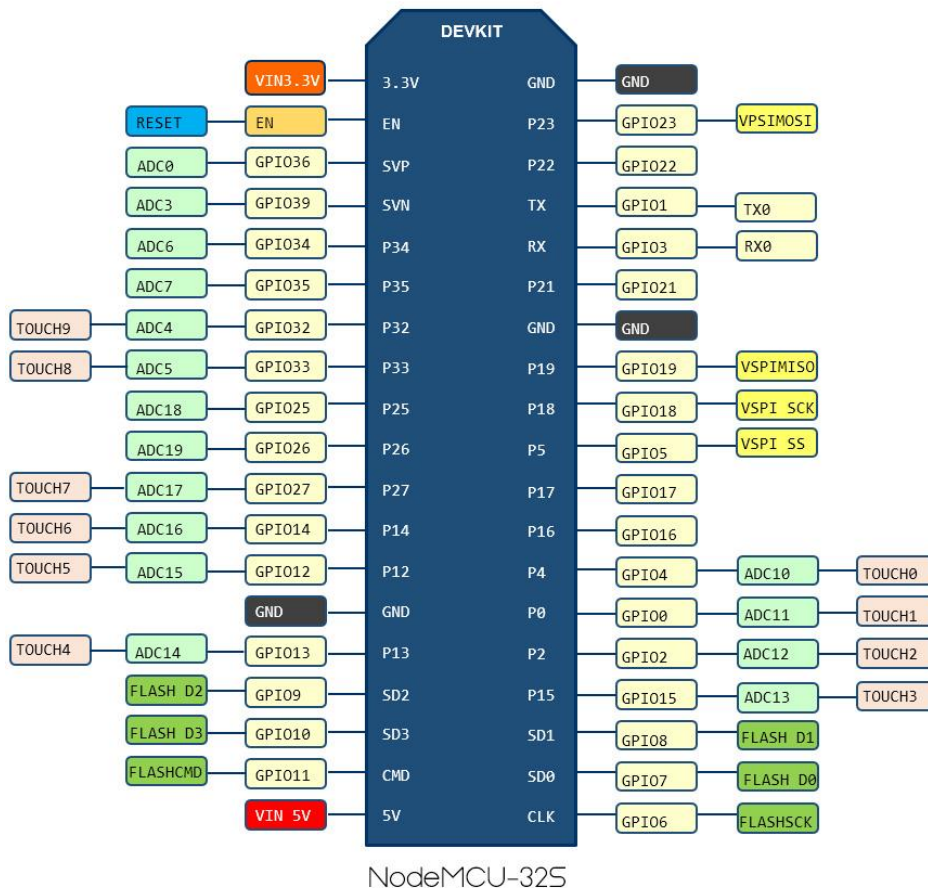
Σε αυτή την παράγραφο καθώς και στην επόμενη του κεφαλαίου 3 θα δοθούν οι κώδικες που γράψαμε και θα γίνει μια ανάλυση για το πώς λειτουργούν. Στην παράγραφο αυτή θα εξηγήσουμε τον κώδικα Arduino IDE, ενώ στην παράγραφο 3.4 θα εξηγήσουμε τον κώδικα της Python IDLE.

Στον κώδικα της Arduino IDE σαν πρώτο βήμα θέλουμε να δηλώσουμε όλες τις απαραίτητες βιβλιοθήκες που θα μας χρειαστούν για να πραγματοποιήσουμε όλες τις λειτουργίες που θέλουμε. Αρχικά δηλώνουμε μία βιβλιοθήκη η οποία είναι υπεύθυνη για την εμφάνιση της ημερομηνίας και της ώρας, και ονομάζεται Network Time Protocol(NTP) ,ενώ η βιβλιοθήκη της έχει όνομα NTPClient.h. Εμείς στην εφαρμογή μας θα χρησιμοποιήσουμε την συγκεκριμένη βιβλιοθήκη για να εμφανίζουμε μόνο την ώρα. Στην συνέχεια δηλώνουμε μία βιβλιοθήκη με ονομασία WiFiUdp.h η οποία είναι υπεύθυνη για την δημιουργία ενός διακομιστή NTP. Τέλος για την λειτουργία της ώρας και όχι μόνο, δηλώνουμε την βιβλιοθήκη του Wifi.h, η οποία καθιστά εφικτή της σύνδεση με το δίκτυο μας.

Στην συνέχεια δηλώνουμε τις βιβλιοθήκες των αισθητηρίων μας. Πρώτα δηλώνουμε του pH με ονομασία DFRobot_ESP_PH.h και ύστερα της θερμοκρασίας DallasTemperature.h , μαζί με το πρωτόκολλο που απαιτεί για την μεταφορά δεδομένων OneWire.h.

Έπειτα, δηλώνουμε την βιβλιοθήκη της EEPROM.h όπου είναι βιβλιοθήκη η οποία θα μας συγκρατεί τα δεδομένα ακόμα και όταν η πλατφόρμα του μικρό-ελεγκτή θα είναι εκτός τροφοδοσίας.

Το επόμενο βήμα είναι να καθορίσουμε με define τους αισθητήρες μας για τον αριθμό του pin του μικρό-ελεγκτή που θα βρίσκονται. Το αισθητήριο θερμοκρασίας θα το συνδέσουμε σε μια ψηφιακή θύρα του μικρό-ελεγκτή και συγκεκριμένα επιλέξαμε την GPIO4 , ενώ το αισθητήριο Ph χρειάζεται να τοποθετηθεί σε μια αναλογική θύρα και εμείς επιλέξαμε την θύρα ADC0(Analog to Digital Conversion). Το πλεονέκτημα της define σε σχέση με την κανονική δήλωση μεταβλητών, είναι ότι δεν δεσμεύουνε χώρο στην μνήμη του ελεγκτή. Τέλος δηλώνουμε με define δύο σταθερές οι οποίες θα μας βοηθήσουν να υπολογίσουμε αργότερα την τιμή του Ph, και αυτές είναι η σταθερά μετατροπής του αναλογικού σήματος σε ψηφιακό, και η τιμή τροφοδοσίας του ESP32.



Σχήμα 3.3 : Pins εισόδων-εξόδων που χρησιμοποιήθηκαν για την συνδεσμολογία (<https://grobotronics.com/esp32-development-board-nodemcu-32s.html?sl=en&fbclid=IwAR2A0iZvIAQsgaqphHYZX821LDjfQA3YI5QI9Ybn1M46N6VMIK1MK3mA5XQ>, Grobotronics, εταιρία με εμπόριο ηλεκτρονικών ειδών).

Στο σημείο αυτό θα πρέπει να καθορίσουμε τις βιβλιοθήκες και τα πρωτόκολλα με τα οποία θα συνεργάζονται τα αισθητήρια μας. Αρχικά κάνουμε ρύθμιση ένα παράδειγμα του OneWire , ώστε να επικοινωνεί με τις συσκευές που υποστηρίζουν αυτό το πρωτόκολλο, και ύστερα διαβιβάζουμε αυτή την αναφορά, στο αισθητήριο θερμοκρασίας. Όσον αφορά το αισθητήριο Ph, δηλώνουμε απλά την βιβλιοθήκη του Ph και από δίπλα την μεταβλητή Ph.

Τέλος, πριν ξεκινήσουμε την δήλωση των μεταβλητών πρέπει να καθορίσουμε τον NTP Client έτσι ώστε να μπορούμε να παίρνουμε τιμές ώρας.

Στην συνέχεια δηλώνουμε τις μεταβλητές. Σαν strings δηλώνουμε την ημερομηνία, την ώρα, μία μεταβλητή βοηθητική με βάση την οποία βρίσκουμε την ώρα η οποία έχει όνομα formatted data, την τιμή της θερμοκρασίας , την τιμή του Ph, και τις ταμπέλες “TempC:”, και “PH:” τις οποίες χρησιμοποιούμε σαν τίτλους για την εμφάνιση στο

αρχείο csv. Έπειτα έχουμε τις μεταβλητές τύπου unsigned long οι lastTime=0, η timerDelay=2000 και η timeout, οι οποίες χρησιμοποιούνται σε συνθήκες if για την εκτέλεση ή όχι κάποιων τμημάτων κώδικα ανάλογα με το πόσος χρόνος έχει περάσει. Πιο αναλυτικές πληροφορίες θα δοθούν παρακάτω. Τελειώνουμε την δήλωση των βασικών μεταβλητών μας με τιμές pH,voltage,temp οι οποίες είναι μορφής float και είναι αυτές που προκύπτουν αρχικά από τους υπολογισμούς μας, και για την εμφάνιση τους στο αρχείο χρειάζεται να μετατραπούν πρώτα σε strings , η θερμοκρασία και το Ph.

Μετά από αυτά, ακολουθούν εντολές δήλωσης μεταβλητών οι οποίες χρησιμοποιούνται για την εγκαθίδρυση επικοινωνίας μεταξύ της Python και του Arduino IDE, έτσι ώστε να ξέρουμε ότι υπάρχει σύνδεση, και να εμφανίσουμε κάποια στοιχεία που αφορούν την σύνδεση που δημιουργήθηκε, όπως είναι η IP του υπολογιστή μέσω της οποίας επικοινωνούνε . Θέτουμε ένα χρονικό όριο 2000 msec δηλαδή 2 sec, στο οποίο όριο αν ο client μας δηλαδή ο κώδικας της Python δεν τρέχει, να μην δημιουργείται σύνδεση μεταξύ των δύο, και συνεπώς να εμφανίζεται ένα μήνυμα στο serial monitor του Arduino IDE ότι το χρονικό όριο το οποίο είχε ο client για να ενεργοποιηθεί, έχει λήξει. Αν αντίθετα υπάρχει διαθέσιμος client, τότε δίνουμε εντολή να εκτυπωθούν τα στοιχεία που σχετίζονται με την σύνδεση των δύο προγραμμάτων.

Στην συνέχεια ακολουθεί η ρουτίνα ρύθμισης διαφόρων παραμέτρων, setup στην οποία γίνονται κάποιες βασικές αρχικοποιήσεις και ρυθμίσεις. Αρχικά βάζουμε σε λειτουργία τους δύο αισθητήρες μας, για να αρχίσουν να μετράνε. Επίσης ξεκινάμε την μνήμη EEPROM του ελεγκτή, έτσι ώστε να μπορεί να αποθηκεύει δεδομένα και χωρίς την ύπαρξη τροφοδοσίας, όπως ακριβώς αναφέραμε και νωρίτερα. Ακολουθούνε εντολές σχετικές με την ρύθμιση του Wi-fi, το οποίο είναι απαραίτητο για την εγκαθίδρυση σύνδεσης μεταξύ των δύο προγραμμάτων Στις εντολές αυτές τοποθετούνται τα στοιχεία του τοπικού μας δικτύου από το οποίο αποφασίζουμε να λειτουργήσουμε το ασύρματο σύστημα, δηλαδή το username και το password. Αναλυτικότερα, αν ο ESP32 βρισκότανε συνδεδεμένος προηγουμένως σε άλλο τοπικό δίκτυο, δίνουμε εντολή να αποσυνδεθεί και αφήνοντας ένα χρονικό περιθώριο τριών δευτερολέπτων, ξανά ξεκινάμε το Wi-fi και δίνουμε τα δικά μας στοιχεία. Χρησιμοποιούμε στη συνέχεια συνθήκη If η οποία κάνει έλεγχο για το ενδεχόμενο της σύνδεσης και αν η σύνδεση γίνει, εμφανίζονται τα στοιχεία της Ip μας, στο serial monitor.

Τέλος, εκκινούμε τον client που είναι υπεύθυνος για την μέτρηση της ώρας, και δίνουμε μια σταθερή παράμετρο η οποία μας διευκρινίζει την ζώνη ώρα στην οποία βρισκόμαστε. Στην δική μας περίπτωση καθώς βρισκόμαστε στην Ελλάδα τοποθετούμε την Θερινή ώρα Ανατολικής Ευρώπης, η οποία δίνεται με τον αριθμό 10.800.

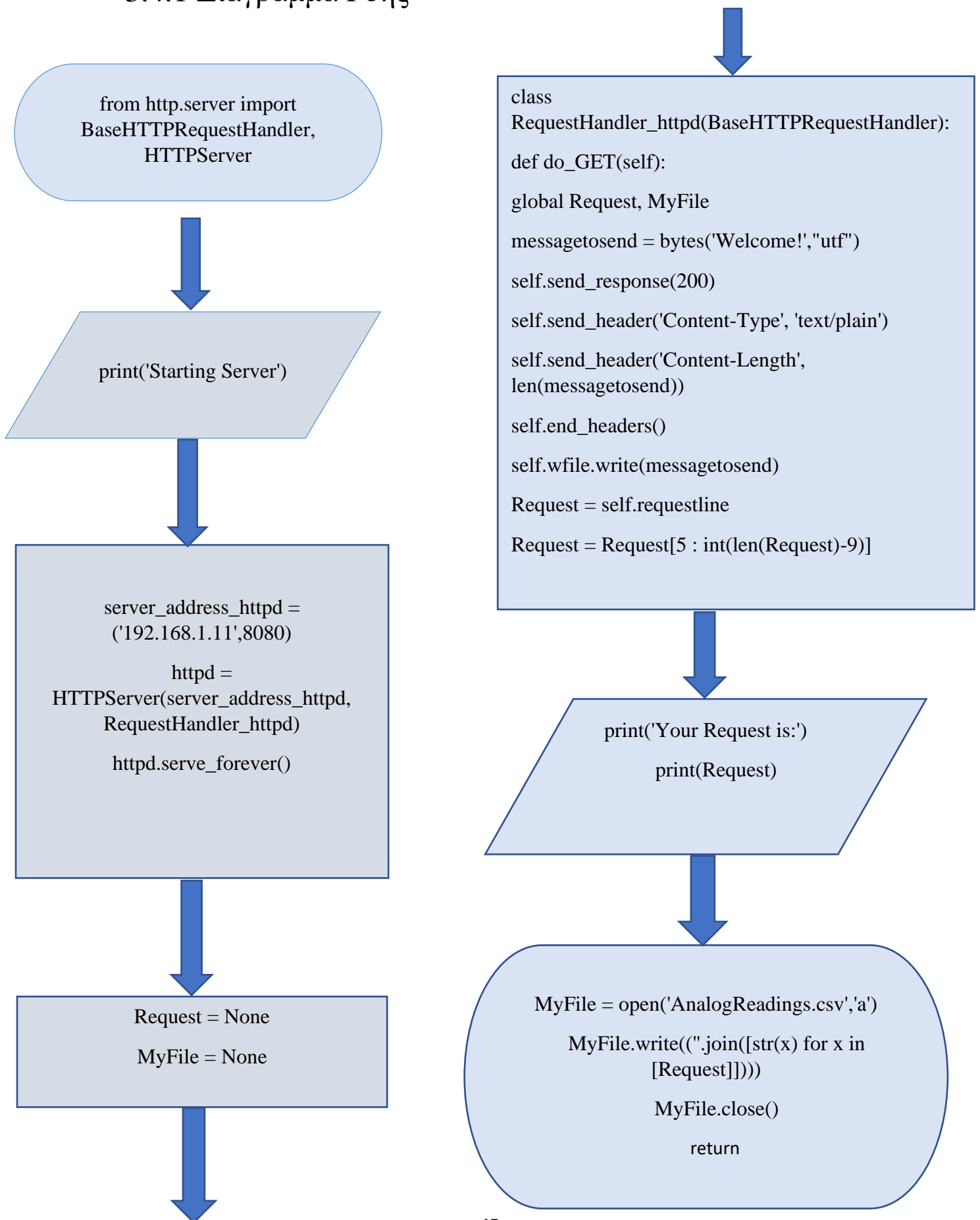
Το τελευταίο κομμάτι κώδικα που υπάρχει στο Arduino IDE, είναι η κύρια λειτουργία του προγράμματος η οποία γράφεται στην ρουτίνα void(loop). Ξεκινάμε την void(loop) με μία συνθήκη if η οποία επιβεβαιώνει με την βοήθεια της ενσωματωμένης μεταβλητής millis() η οποία μετράει τα msec, ότι θα υπάρχει ένα κενό 2 δευτερολέπτων μεταξύ των εκτελέσεων της void (loop). Στη συνέχεια τοποθετούμε συνθήκη η οποία ελέγχει αν υπάρχει διαθέσιμος ο client της python, έτσι ώστε να εκτελέσει αυτά που ακολουθούν. Αξίζει εδώ πέρα να επισημανθεί, πως κάθε φορά που θέλουμε να σταλεί και να εμφανιστεί κάτι από το Arduino στο αρχείο csv που θα δημιουργηθεί και θα έχει τις μετρήσεις μας, θα περνάει πρώτα από το πρόγραμμα της Python το οποίο δέχεται αιτήματα από το Arduino IDE μέσω της IP θύρας που έχει δοθεί, και εκτελεί τον κώδικα για να το εγγράψει στο αρχείο. Ακολουθεί συνθήκη while() η οποία αναγκάζει τον NTP Server που μας δίνει την ώρα, να την κάνει ενημέρωση, και κατόπιν χρήσης μερικών συναρτήσεων της βιβλιοθήκης βρίσκουμε και την ημερομηνία και την ώρα αλλά εκτυπώνουμε μόνο την ώρα δηλαδή το timestamp, το οποίο δείχνει την ώρα ,τα λεπτά και τα δευτερόλεπτα. Στην ίδια συνθήκη ελέγχου σύνδεσης βάζουμε και τους υπολογισμούς εύρεσης της θερμοκρασίας, η οποία βρίσκεται και αυτή μέσω χρήσης συναρτήσεων της βιβλιοθήκης της, και των μεταβλητών που δηλώσαμε. Αυτό το κάνουμε, γιατί θέλουμε να έχουμε την θερμοκρασία ακριβώς εκείνη την ώρα που θα εκτυπωθεί το timestamp και όχι να την υπολογίσουμε λίγα δευτερόλεπτα μετά, καθώς υπάρχουνε πολλές συνθήκες σύνδεσης των δύο προγραμμάτων στην void(loop) ανάλογα με τα requests που θέλουμε να στέλνουμε στο αρχείο , και από την μια συνθήκη στην άλλη μεσολαβούν μερικά δευτερόλεπτα καθώς εκτελείται ενδιάμεσα και ο κώδικας της Python. Στη συνέχεια στέλνουμε μέσω του ίδιου ελέγχου σύνδεσης ένα βελάκι → και έπειτα με τον ίδιο τρόπο τον τίτλο TemperatureC:, ενώ μετά την ταμπέλα στέλνουμε την μέτρηση που έβγαλε το αισθητήριο θερμοκρασίας. Αφήνουμε ένα κενό το οποίο το στέλνουμε και αυτό στην Python μέσω ελέγχου σύνδεσης σε μορφή “ “ , και ακολουθούμε την ίδια διαδικασία που εφαρμόσαμε και για το αισθητήριο θερμοκρασίας, με καινούργιο timestamp για το pH. Το pH, και αυτό όπως και την

θερμοκρασία, το βρίσκουμε μέσα στην ρουτίνα που εμφανίζει το timestamp του. Αρχικά για τον υπολογισμό του, βρίσκουμε την τάση που δέχεται το αισθητήριο, με βάση τον αριθμό που δέχεται από το pin στο οποίο το τοποθετήσαμε, καθώς και από την σταθερά του ADC και την τάση που δέχεται ο ESP, τις οποίες μεταβλητές τις ορίσαμε με define στην αρχή του κώδικα. Μετά τον υπολογισμό της τάσης, υπολογίζουμε με τη βοήθεια της συνάρτησης ph.readPH την τιμή του Ph. Στο τέλος, όταν έχουμε εγγράψει μια μέτρηση θερμοκρασίας και pH, το αρχείο csv είναι αυτής της μορφής: 15:03:58-->TempC:27.19 15:04:04-->PH:7.69. Ένας χαρακτήρας κενού συμπληρώνεται στο τέλος, για την επόμενη μέτρηση. [14]

3.4 Εξήγηση Κώδικα Python

Για την εξήγηση του κώδικα της Python θα δημιουργήσουμε ένα διάγραμμα ροής στο οποίο φαίνεται συνοπτικά η δομή του προγράμματος και η λειτουργία του, ενώ στη συνέχεια θα δημιουργήσουμε ένα πινακάκι στο οποίο θα εξηγείται η λειτουργία και η χρησιμότητα της κάθε εντολής της. Είναι σημαντικό να εξηγηθεί καλά ο κώδικας της καθώς μέσω της Python καθίσταται δυνατή η εγγραφή των μετρήσεων μας, στο αρχείο csv. [15] [16] [17] [18]

3.4.1 Διάγραμμα Ροής



3.4.2 Πιννακάκι Εξήγησης Εντολών Python

<pre>from http.server import BaseHTTPRequestHandler, HTTPServer</pre>	<p>Εντολή η οποία εισάγει τις απαραίτητες κλάσεις οι οποίες βρίσκονται στο directory του προγράμματος της python και είναι υπό την μορφή ενσωματωμένων βιβλιοθηκών. Αυτές είναι που μας βοηθούν να χτίσουμε τον Web Server μας.</p>
<pre>print('Starting Server')</pre>	<p>Εκτυπώνει στο IDLE SHELL της Python την φράση ότι ξεκινάει ο server.</p>
<pre>server_address_httpd = ('192.168.1.11',8080)</pre>	<p>Δημιουργεί μια μεταβλητή η οποία δέχεται σαν ορίσματα την επιθυμητή διεύθυνση Ip και την port επικοινωνίας, τα οποία στη συνέχεια θα χρησιμοποιηθούν για την δημιουργία του server.</p>
<pre>httpd = HTTPServer(server_address_httpd, RequestHandler_httpd)</pre>	<p>Εντολή εκκίνησης του server με ορίσματα την Ip Και Port που δώσαμε πριν, και την κλάση χειρισμού των αιτημάτων(RequestHandler_httpd), την οποία θα ορίσουμε και θα προγραμματίσουμε στη συνέχεια.</p>
<pre>httpd.serve_forever()</pre>	<p>Ακολουθεί εντολή με την οποία δίνουμε προσταγή ο server να τρέχει επ' άοριστον περιμένοντας αιτήματα(requests). Αν θέλουμε να τον σταματήσουμε πατάμε ctrl+c, με την οποία ενεργοποιείται η μέθοδος server.socket.close(), η οποία προκαλεί τον server να σταματήσει.</p>

<pre>Request = None MyFile = None</pre>	<p>Δήλωση δύο μεταβλητών. Η μία είναι για τα αιτήματα που δεχόμαστε από τον client δηλαδή το Arduino ide, και η άλλη είναι για το αρχείο csv που θα δημιουργηθεί και θα περιέχει τις μετρήσεις των αισθητηρίων οργάνων.</p>
<pre>class RequestHandler_httpd(BaseHTTPRequestHandler):</pre>	<p>Ορισμός της κλάσης RequestHandler_httpd η οποία είναι υπεύθυνη για τον χειρισμό των αιτημάτων, και η οποία έχει σαν χρησιμοποιεί την ενσωματωμένη βιβλιοθήκη BaseHTTPRequestHandler, καθώς περιλαμβάνει μεθόδους και μεταβλητές οι οποίες βοηθούν να υλοποιηθεί ο server.</p>
<pre>def do_GET(self):</pre>	<p>Η εντολή do_GET ανήκει στην ενσωματωμένη βιβλιοθήκη που αναφέραμε μόλις πριν, και καλείται κάθε φορά που κάνουμε ένα request στον server. Στην παρένθεση τοποθετούμε το self εννοώντας να πάρει το request από την κλάση που είναι υπεύθυνη για τα requests δηλαδή την RequestHandler_httpd.</p>
<pre>global Request, MyFile</pre>	<p>Τις μεταβλητές τις οποίες δηλώσαμε πιο πριν τις καθιστούμε γενικές.</p>
<pre>messagetosend = bytes('Welcome!',"utf")</pre>	<p>Δηλώνουμε μία μεταβλητή εκχωρώντας της ένα μήνυμα το οποίο έχει την κωδικοποίηση Unicode και συγκεκριμένα την UTF-8 η οποία έχει πολλά πλεονεκτήματα σε σχέση με άλλες, όπου ένα από αυτά είναι το συμπαγές μέγεθος τους , καθώς πιάνουνε 1 με 2 bytes χώρο.</p>

<pre>self.send_response(200)</pre>	<p>Αυτή είναι μια εντολή η οποία σε συνδυασμό με τον αριθμό 200 που βρίσκεται στην παρένθεση, επιβεβαιώνει ότι όλα πάνε καλά με το request που έφτασε στον server. Αυτή η κλάση έχει έναν αριθμό από μεταβλητές παρουσίας(instance variables), οι οποίες χαρακτηρίζουν το συγκεκριμένο αίτημα το οποίο χειρίζεται η Python.</p>
<pre>self.send_header('Content-Type', 'text/plain')</pre>	<p>Από την εντολή do_GET που έχουμε ήδη γράψει, το request που στέλνεται στον server, αντιστοιχίζεται σε ένα τοπικό αρχείο καθώς ερμηνεύεται (το request), ως ένα μονοπάτι σχετικό με το directory εργασίας. Αυτή η εντολή είναι υπεύθυνη να στείλει στο serial monitor του Arduino IDE, το υποτιθέμενο περιεχόμενο του φακέλου στον οποίο αντιστοιχίστηκε το request μας. Από δίπλα ορίζεται η μορφή που θέλουμε να εμφανιστεί που είναι σε μορφή απλού κειμένου.</p>
<pre>self.send_header('Content-Length', len(messagetosend))</pre>	<p>Αφού εμφανίσουμε στο serial monitor του Arduino IDE το περιεχόμενο του φακέλου όπως αναφέραμε στην εξήγηση της προηγούμενης εντολής, εμφανίζουμε με αυτή την εντολή το μέγεθος του προηγούμενου φακέλου.</p>
<pre>self.end_headers()</pre>	<p>Με αυτή την εντολή τερματίζεται η αποστολή μηνυμάτων στο serial monitor του Arduino IDE.</p>
<pre>self.wfile.write(messagetosend)</pre>	<p>Με αυτή την εντολή στέλνεται ένα μήνυμα στο Serial Monitor του Arduino IDE, ως απόκριση(response) στο αίτημα που στάλθηκε στον server. Αυτό το μήνυμα είναι το messagetosend που παραπάνω δηλώσαμε ότι είναι το welcome!</p>

Request = self.requestline	Με αυτή την εντολή εκχωρούμε στην μεταβλητή Request που έχουμε δηλώσει από πριν, το περιεχόμενο του Request που δέχεται κάθε φορά ο server. Αυτά είναι τα requests που στέλνουμε από τον κώδικα του Arduino που μπορεί να είναι ώρα, βελάκι, ταμπέλα και η μέτρηση των αισθητηρίων.
Request = Request[5 : int(len(Request)-9)]	Σε αυτή την εντολή καθορίζουμε το μήκος του Request να είναι μέσα σε επιτρεπόμενα όρια.
print('Your Request is:')	Εκτυπώνουμε στο IDLE Python την φράση “Your Request is:”.
print(Request)	Αμέσως μετά την φράση που γράψαμε, εμφανίζουμε το request.
MyFile = open('AnalogReadings.csv','a')	Με την εντολή αυτή ξεκινάμε την διαδικασία εγγραφής στο επιθυμητό αρχείο csv. Αρχικά ανοίγουμε με αυτή την εντολή το αρχείο το οποίο το ονομάζουμε όπως θέλουμε. Εμείς το ονομάσαμε AnalogReadings, και ακολουθείται με .csv ο τύπος του αρχείου που επιθυμούμε να δημιουργηθεί, που είναι τύπου csv. Τέλος, βάζουμε τον τρόπο με τον οποίο θέλουμε να ανοιχτεί το αρχείο(δηλαδή το Mode), και εμείς επιλέγουμε το ‘a’ καθώς θέλουμε να κάνουμε εγγραφή στο αρχείο και συγκεκριμένα το a χρησιμοποιείται για άνοιγμα του αρχείου και προσάρτηση του περιεχομένου στο τέλος του αρχείου, εάν αυτό υπάρχει. Τέλος, αφού ανοίξει το αρχείο, εκχωρεί το αντίστοιχο αντικείμενο του αρχείου στην μεταβλητή MyFile. Στη ουσία δημιουργείται σύνδεση μεταξύ της μεταβλητής MyFile της Python, με το πραγματικό μας αρχείο csv,

	<p>ώστε να πηγαίνουν όλες οι εγγραφές στην μεταβλητή MyFile και απευθείας να περνιούνται και στο αρχείο μας.</p>
<pre>MyFile.write(''.join([str(x) for x in [Request]]))</pre>	<p>Εδώ πέρα κάνουμε εγγραφή του αρχείου μας το εκάστοτε request, αφού πρώτα το μετατρέψουμε σε μορφή συμβολοσειράς.</p>
<pre>MyFile.close()</pre>	<p>Κλείνουμε το αρχείο MyFile, επομένως δεν μπορούμε πλέον να διαβάσουμε από αυτό ή να εγγράψουμε σε αυτό.</p>
<pre>return</pre>	<p>Με την εντολή return τελειώνει η εκτέλεση της κλάσης RequestHandler_httpd που ορίσαμε , και επιστρέφεται η επιθυμητή έξοδος(Output) την οποία μας επιστρέφει αυτή , που είναι το request.</p>

3.5 Μία Αναφορά στο Internet of Things(IOT)

Δεν ήτανε δυνατό να παραληφθεί στην εργασίας μας ο όρος Διαδίκτυο των Πραγμάτων(Internet of Things-IOT), όρος με μεγάλη σημασία, ιδιαίτερα στα χρόνια που ζούμε, και πάνω στον οποίο στηρίζεται και η δική μας εργασία. Τα τελευταία χρόνια ο όρος Internet of Things έχει τραβήξει σε μεγάλο βαθμό την ερευνητική προσοχή. Με τον όρο IOT εννοούμε ένα σύνολο συγγενικών ,συνδεδεμένων με το Internet, αντικειμένων, τα οποία συλλέγουν και μεταφέρουν δεδομένα σε ένα server, χωρίς την διαμεσολάβηση ανθρώπινης παρέμβασης.

Έχει υποστηριχθεί από τον Pretz, ότι είναι ένα δίκτυο συνδεδεμένο με αντικείμενα, τα οποία αντικείμενα είναι ασύρματα συνδεδεμένα μέσω “έξυπνων” αισθητήρων. Η ανάπτυξη του IOT περιλαμβάνει πολλούς τομείς όπως υποδομές, επικοινωνίες , διεπαφές , πρωτόκολλα και πρότυπα.

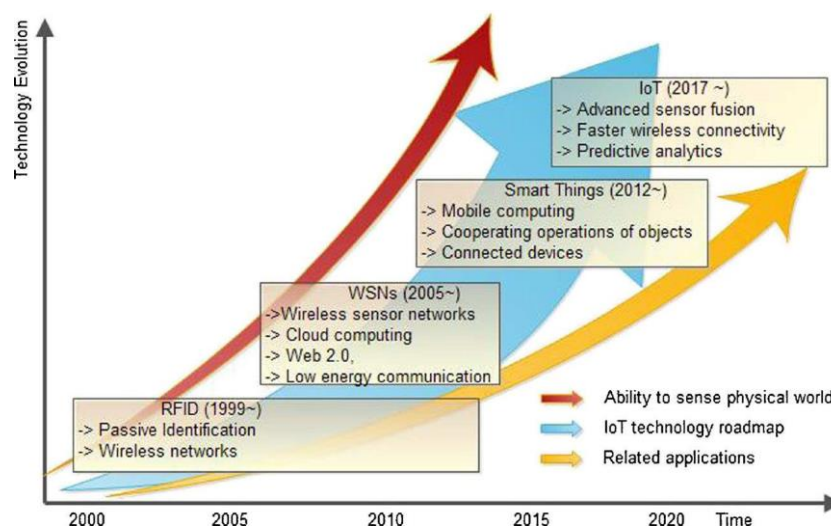
Ο Kevin Ashton ο οποίος είναι βρετανός καινοτόμος στην τεχνολογία, πρότεινε το 1999 την ιδέα του IOT και ανέφερε ως ορισμό ότι είναι “μοναδικά αναγνωρίσιμα διαλειτουργικά συνδεδεμένα αντικείμενα, με τεχνολογία αναγνώρισης μέσω ραδιοσυχνότητων(RFID)”. Γενικά, έχει χαρακτηριστεί ότι είναι μια “δυναμική παγκόσμια υποδομή, η οποία έχει δυνατότητες αυτοδιαμόρφωσης, βασισμένη σε πρότυπα και διαλειτουργικά πρωτόκολλα επικοινωνίας”. Ο ορισμός του IOT έχει ευρέως συζητηθεί και πολλές τεχνολογίες ανταποκρίσεων έχουν ταχέως αναπτυχθεί από διάφορα ινστιτούτα. Επίσης, ασύρματες τεχνολογίες και έξυπνες αισθήσεις έχουν γίνει μέρος του IOTκαι καινούργιες προκλήσεις, και ορίζοντες έρευνας έχουν αναδυθεί.

Μέχρι σήμερα ένας αριθμός από τεχνολογίες εμπλέκονται στον όρο IOT , όπως ασύρματα δίκτυα αισθητήρων(WSNs), Barcodes, έξυπνη αίσθηση, RFID, NFC, χαμηλής κατανάλωσης ασύρματες επικοινωνίες, και πολλές άλλες. Ανάλογα με τις πολλούς τεχνολογικούς κλάδους που εμπλέκονται για την εφαρμογή, ο ορισμός του IOT διαφέρει.

Τα τελευταία χρόνια, η ταυτοποίηση που βασίζεται σε ραδιοσυχνότητες(RFID), χρησιμοποιείται ευρέως στα logistics, στο λιανεμπόριο και στην φαρμακευτική.

Από το 2010 εμφανίζοντας μεγάλη πρόοδο στην κατασκευή “έξυπνων” αισθητήρων , στις χαμηλής κατανάλωσης ασύρματους αισθητήρες, και στην τεχνολογία δικτύων αισθητήρων, μεγάλος αριθμός αντικειμένων(things) μπορούν να διασυνδεθούν σε ένα

ΙΟΤ. Τα αντικείμενα που βρίσκονται σε ένα δίκτυο ΙΟΤ πρέπει να είναι ικανό να διασυνδέονται και να ανταλλάσσουν δεδομένα το ένα με το άλλο αυτόνομα. [19]



Σχήμα 0.4 : Εξέλιξη του ΙοΤ (Internet of Things). Από έρευνα Α. King, "The internet of things: a survey," Springer Science+Business Media, 26 Απριλίου 2014.

3.6 Πρωτόκολλα Ασύρματης Επικοινωνίας

Κατά την υλοποίηση της διπλωματικής, έπρεπε να επιλέξουμε το πρωτόκολλο που θα καταστήσει δυνατή την ασύρματη μεταφορά δεδομένων. Ακολουθεί μια μικρή περιγραφή και ανάλυση των επιλογών που εξετάστηκαν.

3.6.1 Bluetooth-Le

Το Bluetooth είναι μία από τις βασικές τεχνολογίες ασύρματης επικοινωνίας, η οποία είναι ο κατεξοχήν τρόπος μεταφοράς δεδομένων για συσκευές κινητών τηλεφώνων. Η τεχνολογία Bluetooth-Le είναι μια βελτιωμένη τεχνολογία Bluetooth, η οποία περιορίζει την κατανάλωση ενέργειας σε πολύ χαμηλή, παρόλαυτά όμως, διατηρεί την περιορισμένη εμβέλεια αποστολής δεδομένων.

Το Bluetooth με το χαρακτηριστικό LE έχει βγει εδώ και περίπου μία δεκαετία, και οι εκδόσεις που το υποστηρίζουν είναι από το Bluetooth 4.0 και μετά.

Ο ρυθμός μετάδοσης των δεδομένων για την έκδοση 4.0 είναι ικανοποιητική, με ταχύτητα 1Mbit/s, και εμβέλεια μέχρι 100 μέτρα, ενώ για την έκδοση 5.0 είναι μέχρι

και 2 Mit/s και εμβέλεια έως 200 μέτρα. Η έκδοση 5.0 συγκεκριμένα έχει φέρει επανάσταση στον τομέα του IOT που αναφέραμε πιο πριν, επιτρέποντας στους μηχανικούς σχεδίασης να πρωτοπορούν σε καινοτόμες λύσεις, προωθώντας παράλληλα και τον τομέα της σχεδιαστικής μηχανικής.

Η ταχύτητα των 2 Mit/s που αναφέραμε πιο πάνω στοχεύει επομένως κυρίως για αντικείμενα του κλάδου IOT. Παράλληλα όμως μας δίνεται η δυνατότητα με το Bluetooth LE 5.0 , να μειώσουμε το εύρος ζώνης των 2 Mit/s προκειμένου να κερδίσουμε μέχρι και 4 φορές μεγαλύτερο φάσμα, διατηρώντας παράλληλα παρόμοιες ενεργειακές απαιτήσεις. Με την δυνατότητα που αναφέραμε πριν, μέσω της οποίας μπορούν οι συσκευές να πολλαπλασιάζουν επί 4 την απόσταση στην οποία μπορούν να δέχονται και να εκπέμπουν δεδομένα, οι μηχανικοί μπορούν να δημιουργούν αυτοματισμούς σπιτιού και συστήματα ασφαλείας, παρέχοντας κάλυψη σε ένα ολόκληρο σπίτι ή κτίριο. [20] [21] [22]

3.6.2 LoRa

Το πρωτόκολλο LoRa είναι πρωτόκολλα ασύρματης μεταφοράς δεδομένων, το οποίο στοχεύει σε εφαρμογές με ευρείας περιοχής δίκτυα (wide-area network ή WAN). Είναι σχεδιασμένο να παρέχει χαμηλής κατανάλωσης δίκτυα ευρείας περιοχής, τα οποία είναι ειδικά σχεδιασμένα να εξασφαλίζουν ασφαλή με παράλληλα χαμηλό κόστος, αμφίδρομη επικοινωνία, η οποία μπορεί να χρησιμοποιείται σε πολλών ειδών εφαρμογές, όπως εφαρμογές έξυπνης πόλης ή ακόμα και σε βιομηχανικές εφαρμογές. Με χρήση του πρωτοκόλλου αυτού, έχουμε οφέλη όσον αφορά το εύρος της επικοινωνίας. Δύο μικρό-ελεγκτές που είναι συνδεδεμένοι με LoRa Module, έχουν την δυνατότητα να επικοινωνούν μέσα σε μια αστική περιοχή, σε ένα εύρος από 2 έως και 5 χιλιόμετρα απόσταση μεταξύ τους, ενώ σε προαστιακές περιοχές μπορούν να επικοινωνούν σε εύρος μέχρι και 15 χιλιόμετρα. Σαν μειονέκτημα του πρωτοκόλλου αυτού, μπορούμε να αναφέρουμε πως είναι η αδυναμία του στον τομέα του ρυθμού μετάδοσης δεδομένων, καθώς μπορεί να μεταφέρει από 0.3 έως 50 Kbps. [23] [24] [25]

3.6.3 Wi-Fi

Το Wi-fi είναι η πιο προφανής λύση από όσες έχουμε μέχρι στιγμής αναφέρει και δικαίως, καθώς τα τελευταία χρόνια έχει εξελιχθεί με ραγδαίους ρυθμούς και σχεδόν κάθε άνθρωπος έχει την δυνατότητα να διαθέτει από έναν δρομολογητή. Επίσης έχουν

γίνει μεγάλες καινοτομίες στο θέμα των ταχυτήτων μεταφοράς δεδομένων, καθώς με την πολύ πρόσφατη δημιουργία του 5g έχουν γίνει πλέον οι ταχύτητες ασυναγώνιστες, φτάνοντας μια μέση ταχύτητα 70 Mbps. Η συνδεσιμότητα 5g βέβαια, αξίζει να σημειωθεί ότι ακόμα στις περισσότερες χώρες δεν έχει εγκαθιδρυθεί, παρόλα αυτά το ίντερνετ είναι πλέον το καταλληλότερο μέσο για μαζική αποστολή και χειρισμό δεδομένων, με πολύ γρήγορες ταχύτητες, καθώς και η 4g σύνδεση η οποία υπάρχει τα τελευταία χρόνια, παρέχει μια ικανοποιητική μέση ταχύτητα ίση με 14 Mbps. Εδώ, αξίζει να σημειωθεί πως η ταχύτητα της σύνδεσης Wi-Fi εξαρτάται και σε μεγάλο βαθμό από την απόσταση μια συσκευής από την πηγή. Καθώς μία συσκευή πλησιάζει στην πηγή ίντερνετ που είναι ο δρομολογητής μας, η ταχύτητα αυξάνεται, ενώ η απομάκρυνση της από την πηγή ίντερνετ εξασθενεί την ταχύτητα. Το εύρος μιας απλής σύνδεσης Wi-fi φτάνει περίπου τα 100 μέτρα, ωστόσο όμως ένα πραγματικό εύρος συνδεσιμότητας φτάνει τα 10-35 μέτρα απόστασης της συσκευής από τον δρομολογητή. Αυτό το εύρος εξαρτάται και από την αντοχή της κεραίας, όπως επίσης και από την συχνότητα μετάδοσης που είναι 2.4 Ghz ή 5 Ghz. Οι προαναφερθείσες συχνότητες έχουν έναν αριθμό καναλιών μέσω των οποίων μπορούν να συνδεθούν πολλές διαφορετικές ασύρματες συσκευές, και οι οποίες βοηθάνε να κατανεμηθεί το φορτίο με τέτοιο τρόπο ,ώστε να μην υπάρχουνε διακοπές των συνδέσεων των διαφόρων συσκευών. Σαν αρνητικό του Wi-fi μπορούμε να αναφέρουμε αυτό που αναφέραμε και στην εισαγωγή, ότι δηλαδή σε σχέση με τα υπόλοιπα πρωτόκολλα ασύρματης επικοινωνίας , χρησιμοποιεί πιο μεγάλα ποσά ενέργειας. Παρόλα αυτά όμως για συσκευές που έχουν ενσωματωμένη Wi-Fi συμβατότητα όπως ο ESP32, μπορούν να χρησιμοποιηθούν διάφορα modes τα οποία εξοικονομούν σε μεγάλο ποσοστό, την ενέργεια της μπαταρίας τροφοδότησης μια τέτοιας πλατφόρμας. [26] [27]

3.6.4 Wi-Fi HaLow

Στο σημείο αυτό, κρίθηκε σημαντικό να γίνει αναφορά σε ένα πρότυπο Wi-Fi, το οποίο είναι σχεδιασμένο να πληροί τις μοναδικές απαιτήσεις για projects που σχετίζονται με το IOT. Το πρότυπο αυτό ονομάζεται Wi-Fi HaLow [28]και χρησιμοποιείται σε μεγάλο βαθμό σε βιομηχανικές και γεωργικές εφαρμογές , καθώς και σε εφαρμογές έξυπνου κτιρίου και έξυπνου σπιτιού.

Το IEEE 802.11ah είναι ένα πρωτόκολλο ασύρματης δικτύωσης το οποίο εκδόθηκε το 2017 και ονομάστηκε Wi-Fi HaLow. Το πρωτόκολλο αυτό, λειτουργεί σε φάσμα μικρότερο του 1 GHz και ως αποτέλεσμα αυτού, ενεργοποιεί την ελάχιστη απαιτούμενη ενέργεια που χρειάζεται στο να επιτευχθεί σύνδεση. Η σύνδεση που εγκαθιδρύεται χρησιμοποιείται σε εφαρμογές που περιλαμβάνουν αισθητήρια, δίκτυα και φορητές συσκευές. Το εύρος του είναι μεγαλύτερο σε σχέση με άλλες τεχνολογίες που επιλέγονται για εφαρμογές IOT και παρέχει μια πιο ισχυρή σύνδεση σε “δύσκολα” περιβάλλοντα, καθώς μπορεί λόγω της χρήσης χαμηλών συχνοτήτων, να διαπεράσει πιο εύκολα τοίχους και εμπόδια τα οποία οι στάνταρ συχνοτήτες 2.4 GHz και 5 GHz που προαναφέρθηκαν, δεν μπορούν να διαπεράσουν. [21] [22] [23] [24]

Παρακάτω παραθέτουμε ένα πίνακάκι στο οποίο φαίνονται και πιο συνοπτικά τα συγκριτικά χαρακτηριστικά των προαναφερθέντων ασύρματων πρωτοκόλλων επικοινωνίας. Το Wi-Fi θα το χωρίσουμε σε 4G και 5G [29] καθώς οι διαφορές μεταξύ τους αξίζει να επισημανθούν και πιο ξεκάθαρα. [30] [31] Τα αποτελέσματα που θα εισαχθούν θα είναι όσο το δυνατόν κοντινότερα σε πραγματικά δεδομένα: [32] [33]

	Wi-Fi	4G	5G	LoRa	Bluetooth-LE
Μέση Ταχύτητα	12-25 Mbps	30 Mbps	60-1000 Mbps	0.3-50 kbps	BLUETOOTH 4.0 (LE) : 1 Mbit/s BLUETOOTH 5 (LE): 2 Mbit/s
Εύρος	46(indoors)- 92(outdoors) m	52800 feet~16 Km	1000 feet ~ 305 m	Up to 10 miles	BLUETOOTH 4.0 (LE) :Up to 100m BLUETOOTH 5 (LE): Up to 400 m(1 Km Outdoors)
Συχνότητες	2.4 & 5 GHz	2 - 8 GHz	Frequency range 1 (sub-6 GHz range): 450 MHz-6 GHz Frequency range 2 (mmWave spectrum):	<ul style="list-style-type: none"> ●169 MHz ●433 MHz ●868 MHz(Europe) ●915 MHz (North America) 	2.402 - 2.481 GHz

			24.25-52.6 GHz		
Κατανάλωση Ενέργειας	2 to 20 watts.(6 watts being average for a wireless router).	around 6 kilowatts	11.5 Kw	2.25 Watts	≤ 100 mW

4 Υλοποίηση Κατασκευής

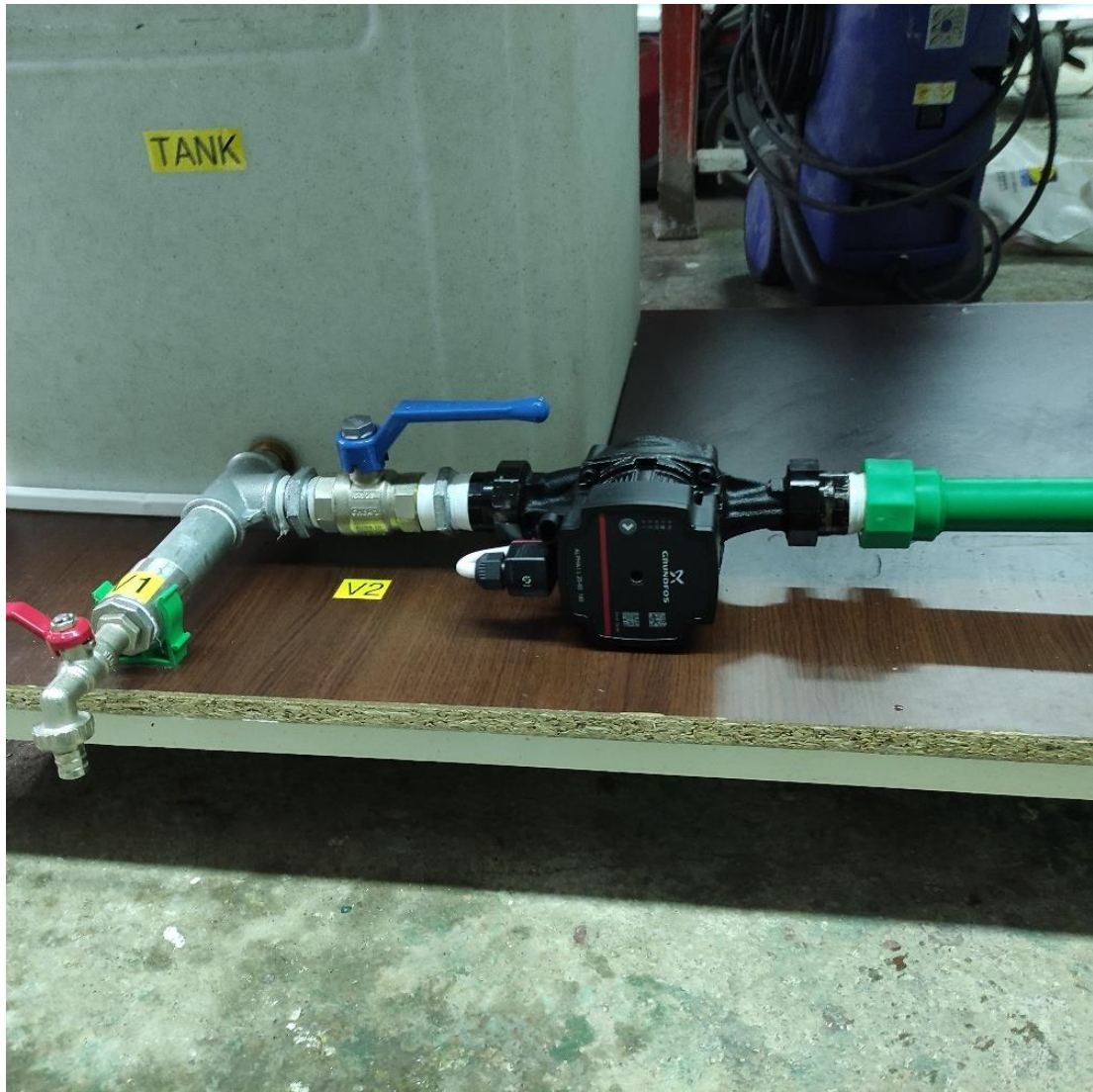
Στο συγκεκριμένο κεφάλαιο θα περιγράψουμε όλα τα στάδια στα οποία προβήκαμε για να υλοποιήσουμε τόσο την ενσωμάτωση των αισθητήρων με ειδικό τρόπο σε τμήμα σωληνώσεων όσο και την τοποθέτηση των ηλεκτρονικών εξαρτημάτων της εργασίας, για την μέγιστη προφύλαξη τους από την υγρασία και άλλους εξωγενείς παράγοντες. Η κατασκευή που ολοκληρώσαμε πραγματοποιήθηκε για τον σκοπό ενός ξεχωριστού έργου και έρευνας που θα σας αναλύσουμε παρακάτω.

4.1 Σκοπός της Κατασκευής

Όπως αναφέραμε και παραπάνω η κατασκευή μας δημιουργήθηκε έτσι ώστε να λαμβάνουμε μετρήσεις σε ένα κλειστό κύκλωμα σωληνώσεων. Στο συγκεκριμένο τμήμα σωληνώσεων διέρχεται νερό με την βοήθεια ενός κυκλοφορητή με ελεγχόμενη ροή, όπου το νερό αποθηκεύεται σε μια δεξαμενή 100 λίτρων. Επίσης, έχει τοποθετηθεί ροόμετρο για την ένδειξη της ροής του νερού. Ο τύπος του σωλήνα που έχει χρησιμοποιηθεί είναι PPR σωλήνα θερμοκόλλησης. Η πίεση είναι αρκετά χαμηλή και αναπτύσσεται από τον κυκλοφορητή, καθώς και η ροή δεν έρχεται από το δίκτυο αλλά από την δεξαμενή οπότε έχουμε χαμηλές τιμές σε bar. Η συγκεκριμένη κατασκευή όπως φαίνεται και στην φωτογραφία παρακάτω δημιουργήθηκε για τον σκοπό μιας ξεχωριστής Διπλωματικής Εργασίας του συμφοιτητή μας Νίκου Μπαξεβάνου, όπου πάνω στην συγκεκριμένη κατασκευή θα ενσωματωθεί το δικό μας σύστημα αισθητήρων.



Σχήμα 4.1 : Κατασκευαστική Διπλωματική του συμφοιτητή μας Νίκου Μπαξεβάνου, πάνω στην οποία θα εγκατασταθεί η δική μας εργασία, έτος 2021.



Σχήμα 4.2 : Ο κυκλοφορητής που χρησιμοποιεί το παραπάνω σύστημα (Διπλωματική Εργασία του Νίκου Μπαξεβάνου, 2021) .

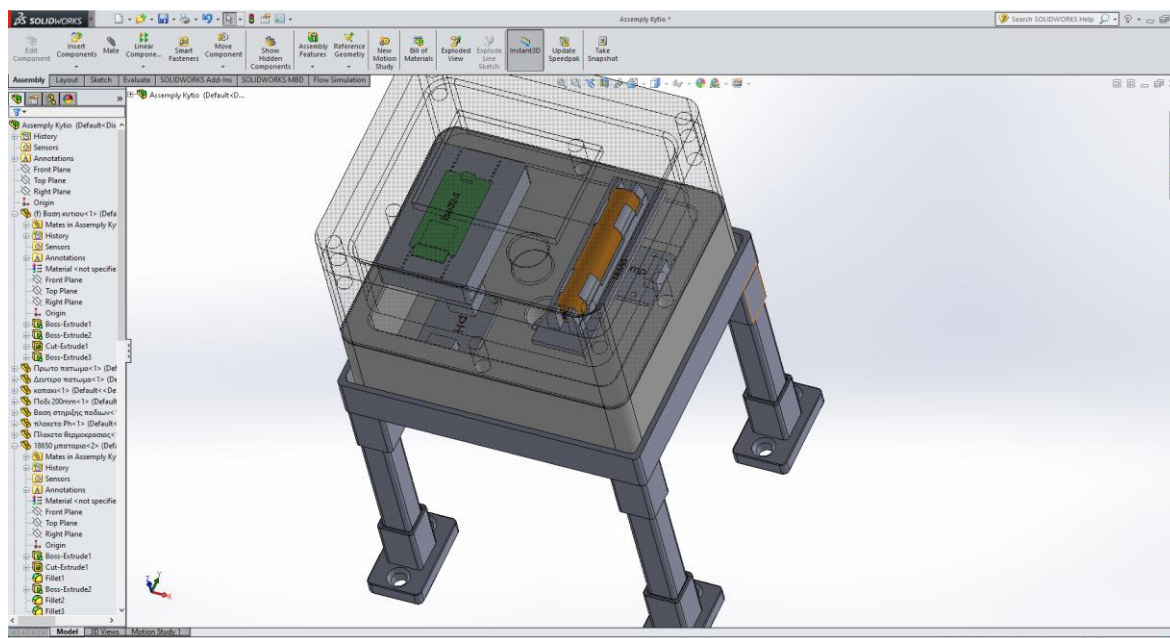


Σχήμα 4.3 : Το σημείο στο οποίο θα προσαρμοστεί το σύστημα αισθητήρων μας, (Διπλωματική Εργασία του Νίκου Μπαξεβάνου, 2021)..

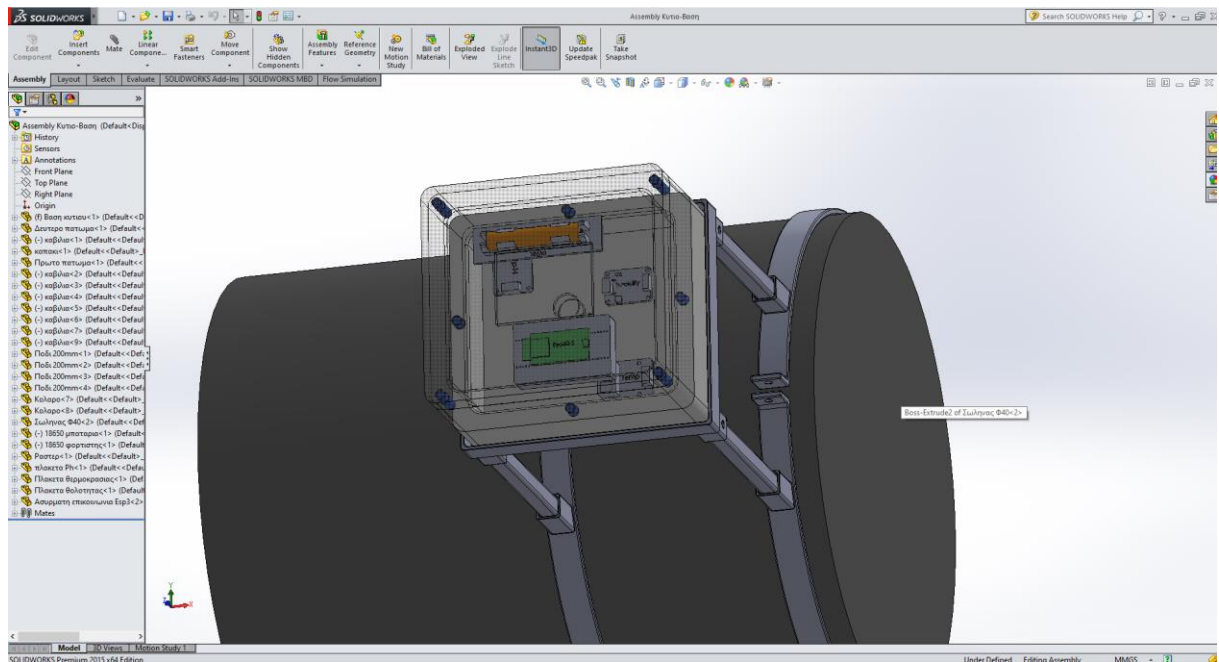
4.2 Σχεδίαση κελύφους ηλεκτρονικών εξαρτημάτων

Στην συγκεκριμένη υπό-ενότητα θα αναλύσουμε το μέρος μέσα στο οποίο θα φυλαχθούν τα ηλεκτρονικά εξαρτήματα του συστήματος αισθητήρων. Σε αυτό το κομμάτι εργαστήκαμε παράλληλα μαζί με τον συμφοιτητή μας Νίκο Παπάζογλου, ο οποίος έχει αναλάβει εκ μέρους της δικής του Διπλωματικής Εργασίας, την σχεδίαση αυτού του κελύφους στο πρόγραμμα SOLIDWORKS αλλά και την υλοποίησή του κατασκευάζοντάς το με την βοήθεια εργαλειομηχανής CNC. Αρχικά, στο εσωτερικό αυτού θα πρέπει να εγκατασταθούν ο μικρό-ελεγκτής μας ESP32, οι πλακέτες διασύνδεσης των αισθητηρίων του pH και της θερμοκρασίας με τον μικρό-ελεγκτή, το τροφοδοτικό μαζί με την επαναφορτιζόμενη μπαταρία αλλά και ένα μικρό ράστερ(breadboard) για την βοήθεια των συνδέσεων. Η σκέψη μας ήταν να δημιουργηθεί κάτι μικρό και εύκολο στη πρόσβαση των εξαρτημάτων, γι' αυτό το λόγο

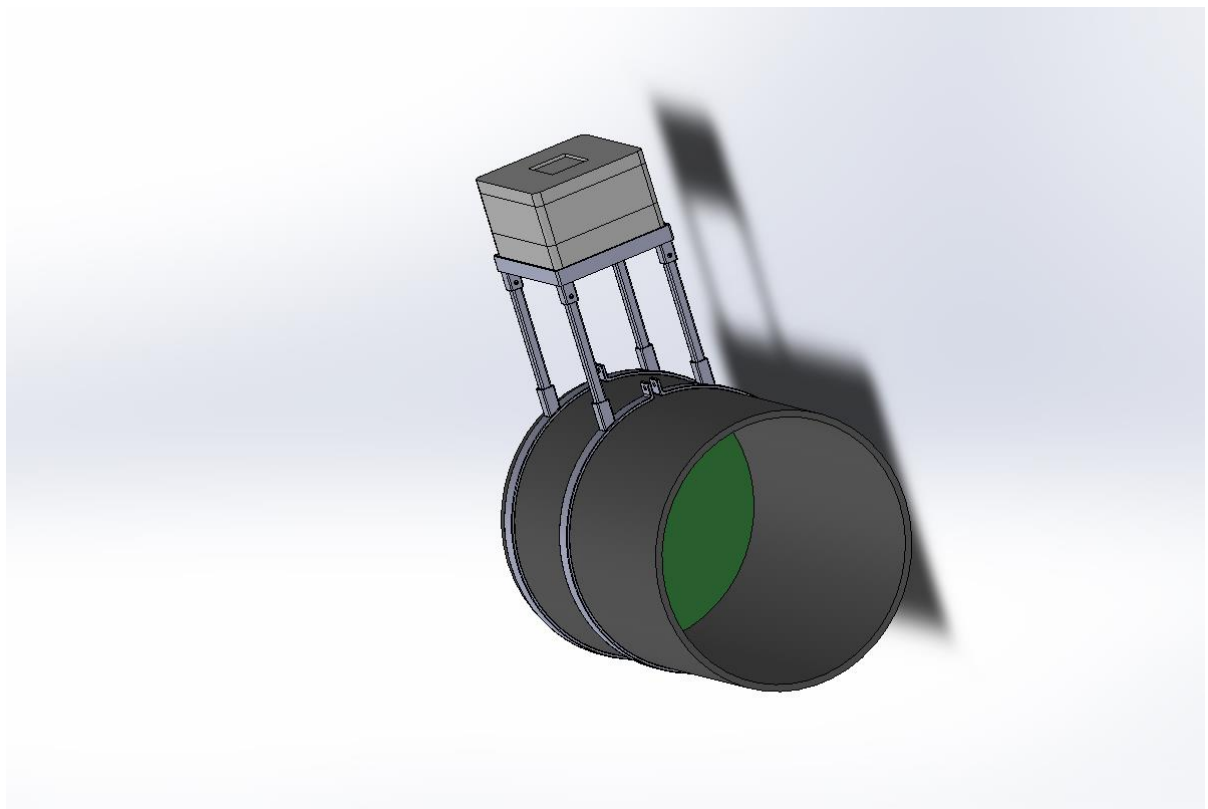
σχεδιάστηκε ένα κουτί με διαστάσεις 10cm ύψος, 10cm πλάτος και 10cm μήκος. Επίσης, στο συγκεκριμένο κουτί αποθηκεύονται τα εξαρτήματα σε έναν χώρο ο οποίος αποτελείται από δυο πατώματα με αποσπώμενο το χάρισμα των δύο αυτών πατωμάτων. Η λογική στην εν λόγω ενεργεία έγινε έτσι ώστε να υπάρχει καλύτερη διευθέτηση τόσο των πλακετών, όσο και των καλωδίων, αλλά και με αυτόν τον τρόπο η χωροταξία είναι μεγαλύτερη καθώς υπάρχει προοπτική επέκτασης του συστήματός μας. Στο πρώτο πάτωμα θα τοποθετηθούν ο ESP32 με το ράστερ για τις απαραίτητες συνδεσμολογίες και το τροφοδοτικό του μικρό-ελεγκτή μαζί με τη μπαταρία λιθίου και στο δεύτερο πάτωμα θα ενταχθούν οι πλακέτες διασύνδεσης των αισθητηρίων με τον μικρό-ελεγκτή, από όπου θα υπάρχει και δίοδος για την όδευση των καλωδίων από τις πλακέτες προς τα αισθητήρια που είναι προσαρμοσμένα στο τμήμα της σωλήνας θερμοκόλλησης. Το κέλυφος θα κατασκευαστεί από πλαστικό τεφλόν, ένα υλικό ελαφρύ και ανθεκτικό ως προς την υγρασία και άλλους διάφορους εξωγενείς παράγοντες, ώστε να προσφέρει μέγιστη στεγανότητα στα ηλεκτρονικά εξαρτήματα. Παράλληλα, η προσαρμογή του θα γίνει επάνω στο κάθετο μέρος του τροχήλατου τραπέζιού της Διπλωματικής Εργασίας που αναφέραμε στην παραπάνω ενότητα.



Σχήμα 4.4 : Το κέλυφος προσαρμογής του συστήματος αισθητήρων σχεδιασμένο στο πρόγραμμα SOLIDWORKS, (Διπλωματική Εργασία του Νίκου Παπάζογλου, 2021, με θέμα την σχεδίαση και την κατασκευή ενός κελύφους με χρήση CNC).



Σχήμα 4.5 : Παλαιότερη προσέγγιση στήριξης του κελύφους επάνω στην σωλήνα, (Διπλωματική Εργασία του Νίκου Παπάζογλου, 2021, με θέμα την σχεδίαση και την κατασκευή ενός κελύφους με χρήση CNC).



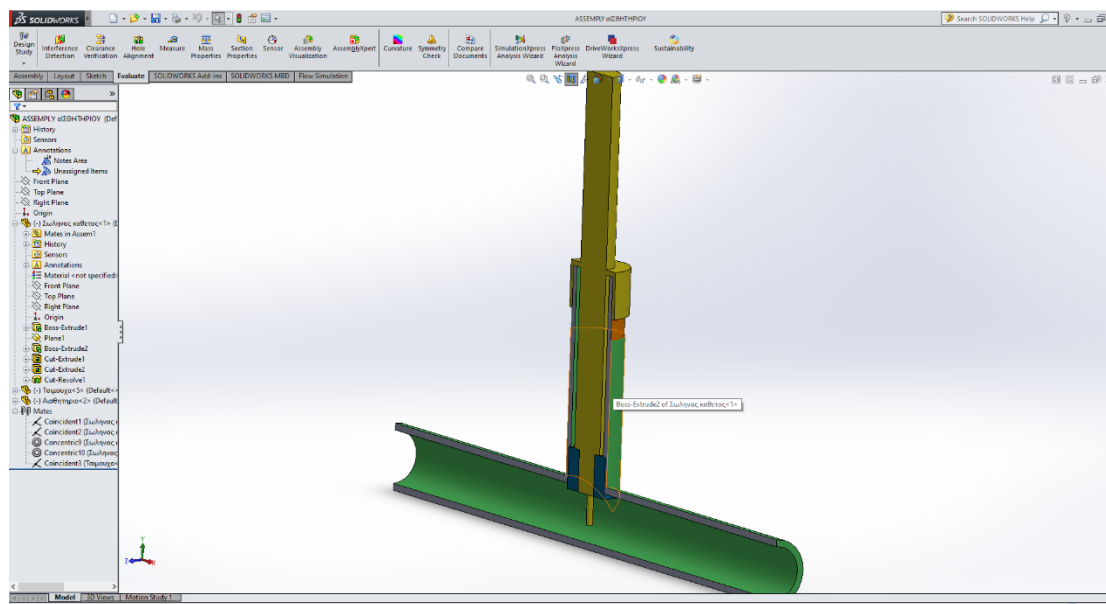
Σχήμα 4.6 : Μακρινή οπτική παλαιότερης προσέγγισης, (Διπλωματική Εργασία του Νίκου Παπάζογλου, 2021, με θέμα την σχεδίαση και την κατασκευή ενός κελύφους με χρήση CNC).

4.3 Προσαρμογή Αισθητήρων

Στο κεφάλαιο αυτό θα αναπτύξουμε το στάδιο που έγινε πριν την υλοποίηση της κατασκευής, δηλαδή την σχεδίαση των αισθητηρίων πάνω στη σωλήνα, τα απαραίτητα εργαλεία και εξαρτήματα που χρησιμοποιήθηκαν για την προσαρμογή των αισθητηρίων αλλά και το τελικό στάδιο της κατασκευής μας.

4.3.1 Σχεδίαση αισθητήρων στο SOLIDWORKS

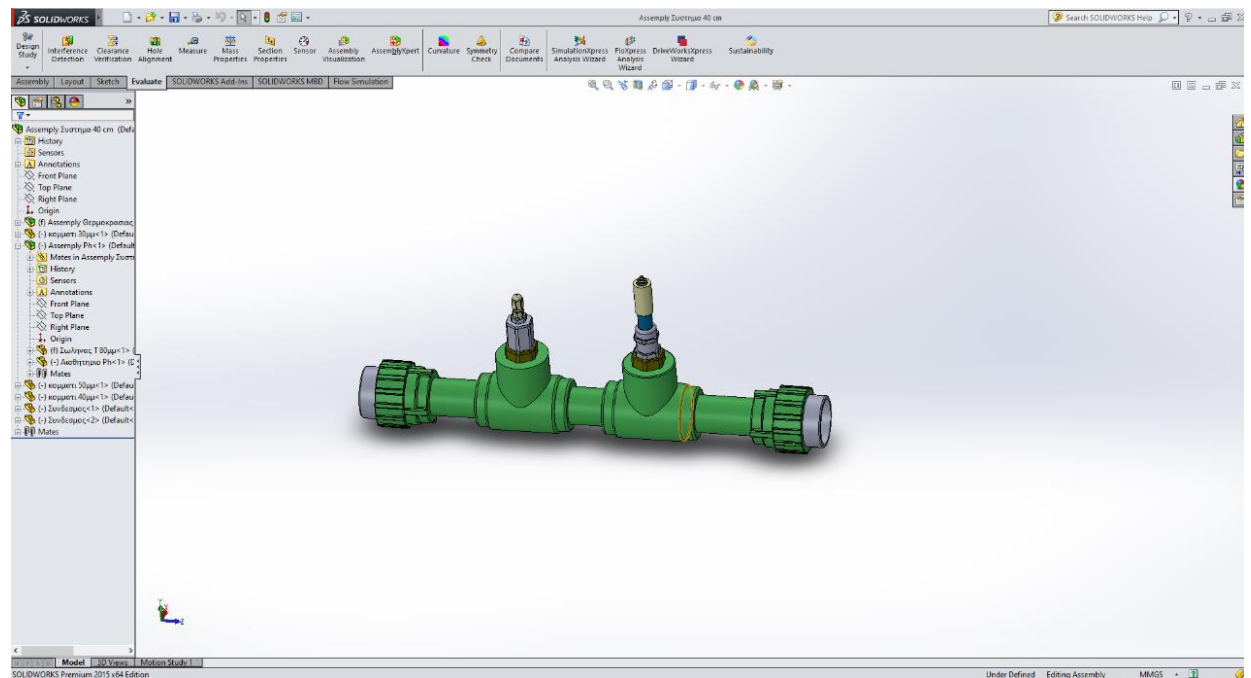
Αρχικά, για να μπορέσουμε να τοποθετήσουμε στην πράξη τα αισθητήρια πάνω στη σωλήνα θα έπρεπε αν έχουμε ένα σχέδιο που θα μας οδηγήει με ευκολότερο τρόπο στον στόχο μας. Γι' αυτό το λόγο δημιουργήθηκε ένα σχέδιο σε συνεργασία με τον συμφοιτητή μας Νίκο Παπάζογλου στο πρόγραμμα SOLIDWORKS, καθώς η απευθείας προσαρμογή των αισθητήρων στη σωλήνα θα ήταν ιδιαίτερα δύσκολη για εμάς διότι δεν έχουμε επιχειρήσει την δημιουργία κάποιας παραπλήσιας κατασκευής στο παρελθόν.



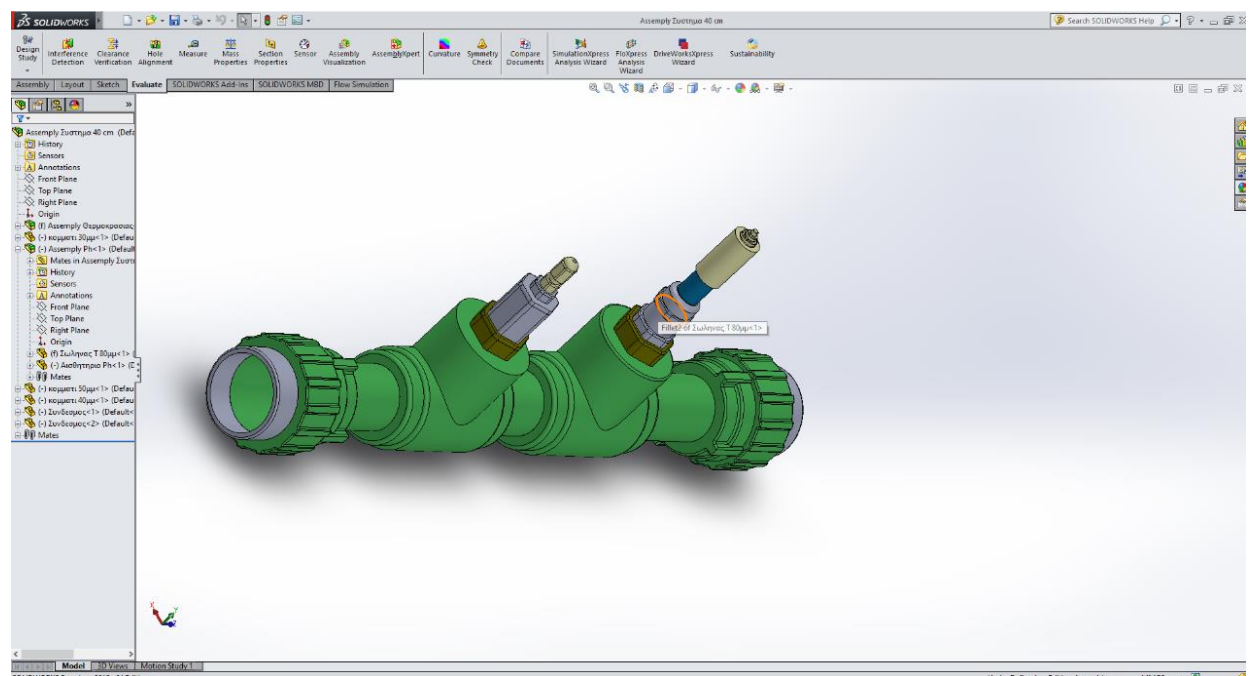
Σχήμα 4.7 : Ενέργειες σχεδίασης του αισθητήρα κάθετα στη σωλήνα, (Διπλωματική Εργασία του Νίκου Παπάζογλου, 2021, με θέμα την σχεδίαση και την κατασκευή ενός κελύφους με χρήση CNC).

Η αρχική μας σκέψη ήταν να προσαρμοστεί τόσο το αισθητήριο του pH όσο και το αισθητήριο της θερμοκρασίας κάθετα στην σωλήνα στηριζόμενα πάνω σε ένα τμήμα κάθετης σωλήνας με τα ακροφύσια και των δύο αισθητήρων να βρίσκονται σε επαφή με το νερό για την λήψη των κατάλληλων μετρήσεων. Έπειτα από πολλές συσκέψεις

και συνομιλίες σχεδιάστηκε το παρακάτω σχέδιο από τον Νίκο Παπάζογλου που ήταν και το καθοριστικό για να ακολουθήσει η πρακτική υλοποίηση.



Σχήμα 4.8 : Τελικό σχέδιο προσαρμογής των αισθητήρων στη σωλήνα (αριστερά είναι το αισθητήριο θερμοκρασίας και δεξιά το αισθητήριο pH), (Διπλωματική Εργασία του Νίκου Παπάζογλου, 2021, με θέμα την σχεδίαση και την κατασκευή ενός κελύφους με χρήση CNC).



Σχήμα 4.9 : Μια διαφορετική οπτική του σχεδίου των αισθητήρων στη σωλήνα, (Διπλωματική Εργασία του Νίκου Παπάζογλου, 2021, με θέμα την σχεδίαση και την κατασκευή ενός κελύφους με χρήση CNC).

4.3.2 Εργαλεία και εξαρτήματα που χρησιμοποιήθηκαν για την υλοποίηση

Αρχικά, τα υλικά που χρειάστηκαν για την κατασκευή είναι :

- 1) Ένα κομμάτι 30cm Φ 32 σωλήνας θερμοκόλλησης PPR.



Σχήμα 4.10 : Σωλήνα θερμοκόλλησης PPR Φ 32.

- 2) Δύο αρσενικοί μαστοί για σωλήνα θερμοκόλλησης διαμέτρου Φ 32.



Σχήμα 4.11 : Μαστοί για σωλήνα θερμοκόλλησης διαμέτρου Φ 32.

3) Δύο Ταφ $\Phi 32 \times 1/2''$ πολυπροπυλενίου PPR θηλυκά.



Σχήμα 4.12 : Ταφ $\Phi 32 \times 1/2''$ πολυπροπυλενίου PPR.

4) Δύο ρακόρ ορειχάλκινα $1/2''$ θηλυκά.



Σχήμα 4.13 : Ρακόρ ορειχάλκινα $1/2''$.

5) Έναν μεταλλικό στυπιοθλίπτη με λάστιχο $1/2''$.



Σχήμα 4.14 : Μεταλλικός στυπιοθλίπτης σύσφραξης.

6) Έναν πλαστικό στηπιοθλίπτη με λάστιχο.



Σχήμα 4.15 : Πλαστικός στυπιοθλίπτης .

7) Ένα κιάθιο για αισθητήριο 1/2''.



Σχήμα 4.16 : Κιάθιο για αισθητήριο 1/2''.

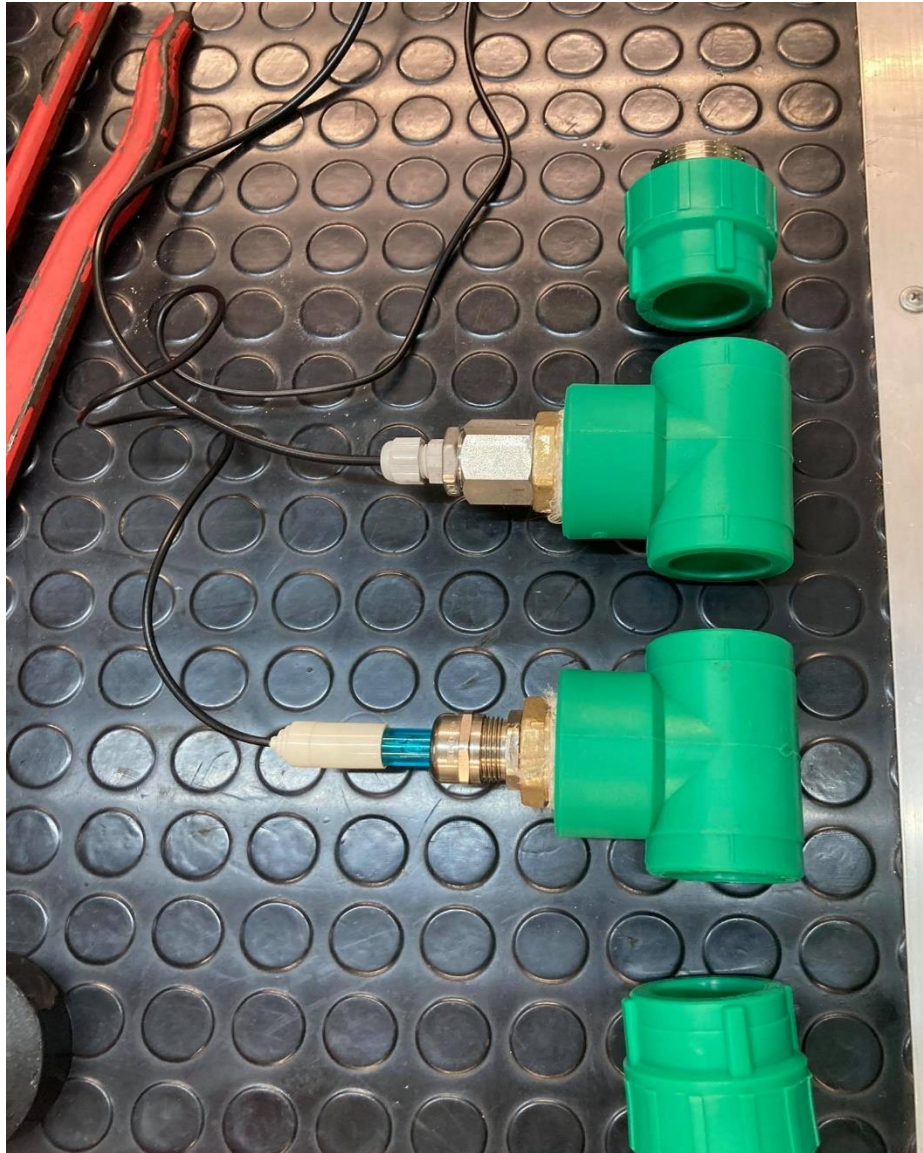
8) Συγκολλητικό μηχάνημα για σωλήνα θερμοκόλλησης.



Σχήμα 4.17 : Συγκολλητικό μηχάνημα.

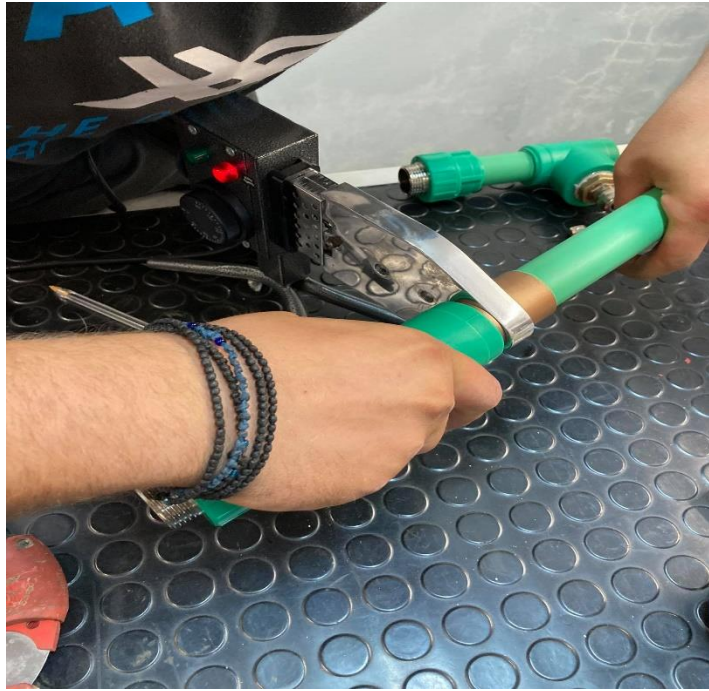
4.4 Υλοποίηση Κατασκευής

Σε αυτήν την υποενότητα θα περιγράψουμε σε ένα σύντομο κείμενο την διαδικασία που ακολουθήσαμε για την προσαρμογή των αισθητήρων. Αρχικά, όπως φαίνεται και στη παρακάτω φωτογραφία ετοιμάστηκαν τα Ταφ με τα αισθητήρια για να έρθει στην συνέχεια η κόλληση της σωλήνας στα επιμέρους διαστήματα μεταξύ των Ταφ και των μαστών. Συγκεκριμένα, επάνω στα Ταφ βιδώθηκαν τα ρακόρ, χρησιμοποιώντας τεφλόν για μεγαλύτερη σύσφιξη, όπου επάνω στα Ταφ βιδώθηκαν αντίστοιχα ο μεταλλικός στυπιοθλίπτης για το αισθητήριο pH και το κυάθιο για την ενσωμάτωση του αισθητηρίου θερμοκρασίας. Τόσο ο στυπιοθλίπτης, όσο και το κυάθιο έχουν αρσενικό σπείρωμα για να μπορέσουν να προσαρμοστούν στα θηλυκά ρακόρ. Για την συγκράτηση του αισθητηρίου θερμοκρασίας βιδώθηκε όπως φαίνεται στην φωτογραφία ένας πλαστικός στυπιοθλίπτης. Στην συνέχεια, με την σύσφιξη και των δύο στυπιοθλιπτών, το λάστιχο που βρίσκεται μέσα σε αυτούς κρατάει στεγανό και συγκρατεί το σύστημα μας στο μέγιστο.



Σχήμα 4.18 : Προσαρμογή αισθητήρων στα Ταφ πολυπροπυλενίου.

Σε δεύτερο στάδιο με την βοήθεια του παρακάτω μηχανήματος ακολουθεί η συγκόλληση την σωλήνας PPR Φ32 στα κενά μεταξύ μαστών και Ταφ έτσι ώστε να συμπληρωθεί ολόκληρο το κομμάτι που πρέπει να προσαρμοστεί στο κύκλωμα σωληνώσεων της Διπλωματικής του συμφοιτητή μας.



Σχήμα 4.19 : Κόλληση σωλήνας με την βοήθεια συγκολλητικού μηχανήματος.



Σχήμα 4.20 : Τελική εικόνα κατασκευής.

5 Ενδεικτικά Πειράματα

Σε αυτήν την ενότητα θα παραθέσουμε κάποια ενδεικτικά πείραμα που πραγματοποιήθηκαν με τα αισθητήρια πριν τα προσαρμόσουμε στην σωλήνα έτσι ώστε να βεβαιωθούμε ότι λειτουργούν ομαλά. Επίσης, θα προστεθεί και μια φωτογραφία η οποία μας δείχνει σε κανονικό χρόνο την λήψη αποτελεσμάτων και την εκτύπωσή τους σε ένα αρχείο Excel Csv μορφής.

5.1 Πειράματα με τα αισθητήρια

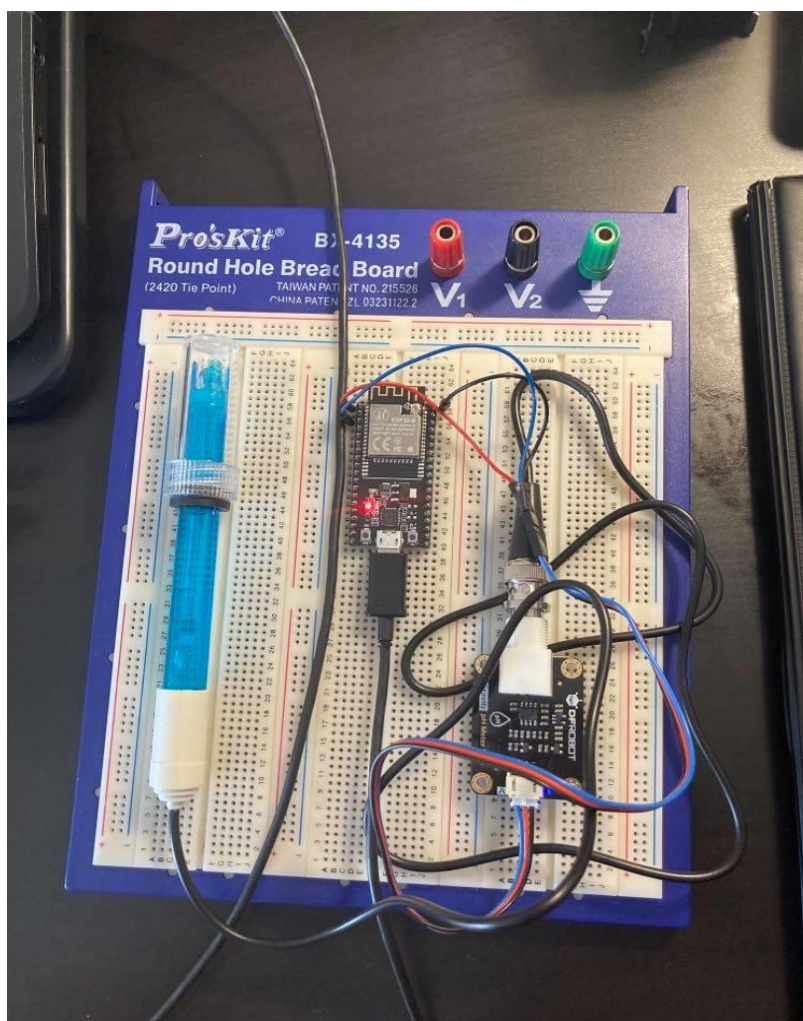
Πρακτικά για να ελέγξουμε την ομαλή λειτουργία των αισθητηρίων πριν τα προσαρμόσουμε στη σωλήνα χρησιμοποιήσαμε διάφορα διαλείμματα για να ελέγξουμε τις μετρήσεις που λαμβάνει το αισθητήριο pH. Συγκεκριμένα, έγινε χρήση κάποιων βασικών διαλειμμάτων όπως οδοντόκρεμα και σαπούνι, αλλά και κάποιων όξινων διαλειμμάτων όπως ξύδι, χυμός λεμόνι.



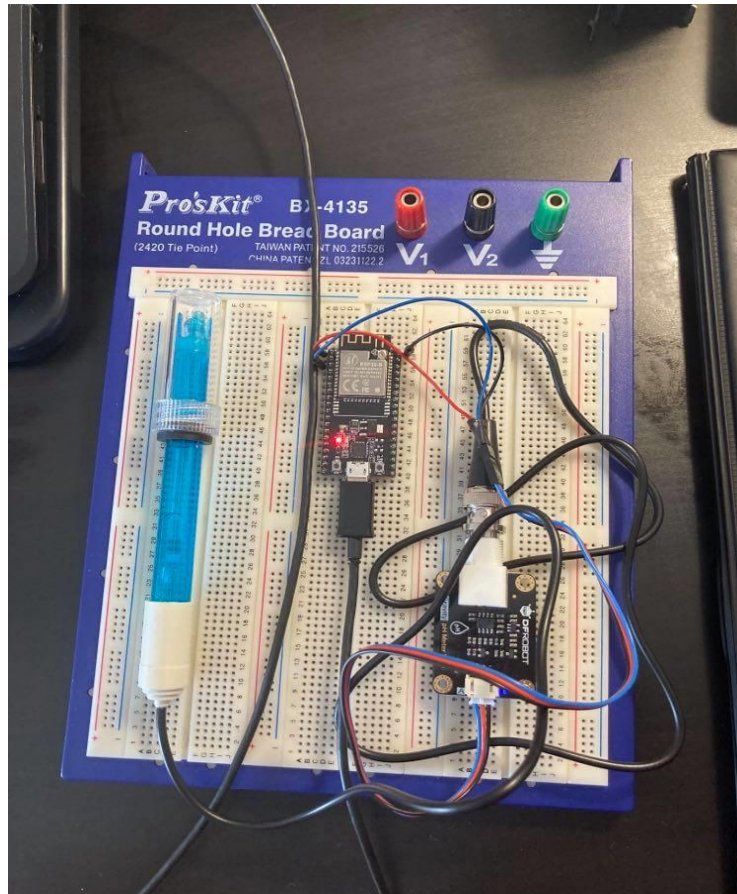
Σχήμα 5.1 : Πειράματα με το αισθητήριο pH.

5.2 Συνδεσμολογίες αισθητήρων μικροελεγκτή

Πριν από την πραγματοποίηση των πειραμάτων έγινε και ο απαραίτητος έλεγχος στις συνδεσμολογίες του συστήματός μας πάνω σε ράστερ δοκιμών, έτσι ώστε να βεβαιωθούμε ότι δεν υπάρχει κάποια λάθος σύνδεση που να επηρεάζει αρνητικά το σύστημά μας. Τοποθετώντας κάθε αισθητήριο σε κάποιες από τις εισόδους του Esp32 και συνδέοντας σωστά τα Vcc (τροφοδοσία) και Gnd (γείωση) ήμασταν έτοιμη να προχωρήσουμε στο στάδιο των πειραμάτων.



Σχήμα 5.2 : Σύνδεση αισθητήρα pH με τον Esp32.



Σχήμα 5.3 : Σύνδεση αισθητήρα θερμοκρασίας με τον Esp32

5.3 Λήψη μετρήσεων

Εφόσον είχαν εγκατασταθεί οι απαραίτητες συνθήκες των αισθητήριων στο τμήμα του κώδικα για τον Esp32 στο Arduino IDE, είχε φτάσει η στιγμή να δημιουργηθεί ένα αρχείο Csv ώστε να αποθηκεύονται εκεί τα αποτελέσματα των μετρήσεων και όχι να τυπώνονται με απλό τρόπο στην σειριακή οθόνη του Arduino IDE. Γι' αυτό το λόγο χρησιμοποιήθηκε ένα τμήμα κώδικα που ετοιμάσαμε σε γλώσσα Python, όπου με την βοήθεια την ασύρματης επικοινωνίας και του πρωτοκόλλου Wi-Fi μετατρέψαμε σε έναν Server τον υπολογιστή μας ο οποίος θα αποθηκεύει τα αποτελέσματα σε ένα Excel αρχείο τυπώνοντας παράλληλα και την ώρα που λήφθηκε κάθε μέτρηση ξεχωριστά. Στην συνέχεια, με κατάλληλες εντολές όπως έχουν περιγράψαμε στο κεφάλαιο Εργαλεία Λογισμικού, αποθηκεύουμε αυτό το Csv αρχείο στην πλατφόρμα του Dropbox, ώστε να μπορεί ο καθένας από οπουδήποτε να έχει πρόσβαση στα αποτελέσματα.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		15:03:58-->TempC:27.19		15:04:04-->PH:7.69		15:04:11-->TempC:27.19		15:04:16-->PH:7.92		15:04:23-->TempC:27.19		15:04:29-->PH:7.91		15:
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														

Σχήμα 5.4 : Λήψη μετρήσεων σε πραγματικό χρόνο.

6 Παράρτημα

Σε αυτό το παράρτημα θα παραθέσουμε τα δύο τμήματα κώδικα που ετοιμάστηκαν τόσο του Arduino IDE για την λειτουργία του συστήματος αισθητήρων, όσο και της Python για την λήψη των αποτελεσμάτων σε Csv αρχείο και την αποθήκευσή τους στο Dropbox.

Κώδικας στο πρόγραμμα Arduino IDE :

```
#include <NTPClient.h>
```

```
#include <WiFiUdp.h>
```

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

```
#include <DFRobot_ESP_PH.h>
```

```
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
```

```
#include <EEPROM.h>
```

```
DFRobot_ESP_PH ph;
```

```
#define ONE_WIRE_BUS 4
```

```
#define PH_PIN 36
```

```
#define ESPADC 4096.0
```

```
#define ESPVOLTAGE 3300
```

```
WiFiUDP ntpUDP;
```

```
NTPClient timeClient(ntpUDP);
```

```
OneWire oneWire(ONE_WIRE_BUS);
```

```
DallasTemperature sensors(&oneWire);
```

```
String formattedDate;
```

```
String dayStamp;
```

```
String timeStamp;
```

```
String TemperatureC="";
```

```
String pHValue="";
```

```
String Label_1="TempC:";
```

```
String Label_2="PH:";
```

```
unsigned long lastTime = 0;
```

```
unsigned long timerDelay = 2000;
```

```
unsigned long timeout = millis();
```

```
float pHVal,voltage,tempC;
```

```
String myurl = "/";
```

```
String line="";
```

```
String IotClientSendWithAnswer(String IPcache, String monmessagecache) {
```

```
line="";
```

```
WiFiClient client;
```

```
client.print(String("GET ") + myurl + monmessagecache + " HTTP/1.1\r\n" +  
    "Host: " + IPcache + "\r\n" +  
    "Connection: close\r\n\r\n");
```

```
while (client.available() == 0) {  
    if (millis() - timeout > 2000) {  
        client.stop();  
        return "Client Timeout!";  
    }  
}
```

```
}  
while(client.available()) {line += client.readStringUntil('\r');}  
return line;  
}
```

```
void setup()  
{
```

```
sensors.begin();

ph.begin();

EEPROM.begin(32);

WiFi.disconnect();

delay(3000);

Serial.println("Start");

WiFi.begin("SSID","PASSWORD");

while (!(WiFi.status() == WL_CONNECTED)){

    delay(300);

    Serial.print("");

}

Serial.println("Connected");

Serial.println("Your IP is");

Serial.println(WiFi.localIP());

timeClient.begin();

timeClient.setTimeOffset(10800);

}
```

```

void loop()
{
  if ((millis() - lastTime) > timerDelay) {

    if (client.connect("192.168.1.11", 8080)) {
      while(!timeClient.update()) {
        timeClient.forceUpdate();
      }
      formattedDate = timeClient.getFormattedDate();

      // Extract date
      int splitT = formattedDate.indexOf("T");

      timeStamp = formattedDate.substring(splitT+1, formattedDate.length()-1);

      sensors.requestTemperatures();
      tempC= sensors.getTempCByIndex(0);

      TemperatureC=String(tempC);
      Serial.println((IoTClientSendWithAnswer("192.168.1.11",timeStamp)));
    }

    if (client.connect("192.168.1.11", 8080)) {
      Serial.println((IoTClientSendWithAnswer("192.168.1.11","-->")));
    }
  }
}

```



```
}
```

```
if (client.connect("192.168.1.11", 8080)) {  
    Serial.println((IotClientSendWithAnswer("192.168.1.11",Label_1)));  
}
```

```
if (client.connect("192.168.1.11", 8080)) {  
  
    Serial.println((IotClientSendWithAnswer("192.168.1.11",TemperatureC)));  
  
}
```

```
if (client.connect("192.168.1.11", 8080)) {  
    Serial.println((IotClientSendWithAnswer("192.168.1.11", "    ")));  
}
```

```
if (client.connect("192.168.1.11", 8080)) {  
    while(!timeClient.update()) {  
        timeClient.forceUpdate();  
    }  
    formattedDate = timeClient.getFormattedDate();  
  
    // Extract date
```

```

int splitT = formattedDate.indexOf("T");

timeStamp = formattedDate.substring(splitT+1, formattedDate.length()-1);

voltage = analogRead(PH_PIN) / ESPADC * ESPVOLTAGE;

phVal = ph.readPH(voltage,tempC);

phValue=String(phVal);

Serial.println((IotClientSendWithAnswer("192.168.1.11",timeStamp)));
}

if (client.connect("192.168.1.11", 8080)) {

Serial.println((IotClientSendWithAnswer("192.168.1.11","-->")));

}

if (client.connect("192.168.1.11", 8080)) {

Serial.println((IotClientSendWithAnswer("192.168.1.11",Label_2)));

}

if (client.connect("192.168.1.11", 8080)){

Serial.println((IotClientSendWithAnswer("192.168.1.11",phValue)));

ph.calibration(voltage, tempC); // calibration process by Serail CMD

}

}

```

```

if (client.connect("192.168.1.11", 8080)) {
  Serial.println((IoTClientSendWithAnswer("192.168.1.11", "
  ")));
}
}
delay(2000);
}

```

Κώδικας στο πρόγραμμα IDLE Python :

```

from http.server import BaseHTTPRequestHandler, HTTPServer

print('Starting Server')

server_address_httpd = ('192.168.1.11',8080)

httpd = HTTPServer(server_address_httpd, RequestHandler_httpd)

httpd.serve_forever()

Request = None

MyFile = None

class RequestHandler_httpd(BaseHTTPRequestHandler):

    def do_GET(self):

        global Request, MyFile

        messagetosend = bytes('Welcome!',"utf")

        self.send_response(200)

        self.send_header('Content-Type', 'text/plain')

```

```
self.send_header('Content-Length', len(messagetosend))

self.end_headers()

self.wfile.write(messagetosend)

Request = self.requestline

Request = Request[5 : int(len(Request)-9)]

print('Your Request is:')

print(Request)

MyFile = open('AnalogReadings.csv','a')

MyFile.write((".".join([str(x) for x in [Request]])))

MyFile.close()

Return
```

7 Βιβλιογραφία

- [1] E. Gregersen, "Britannica," 5 Φεβρουάριος 2009. [Online]. Available: <https://www.britannica.com/technology/pH-meter>. [Accessed 1 Απρίλιος 2021].
- [2] Δ. Ν. Κ. Δρ. Κωνσταντίνος Καλοβρέχτης, Αισθητήρες Μέτρησης και Ελέγχου/ Ηλεκτρικές Μετρήσεις-Τεχνολογίες Μετρήσεων, 3η έκδοση επιμ., Εκδόσεις Τζιόλα, 2018, p. 395.
- [3] G. Boubel, "aquasana.com," Aquasana, [Online]. Available: https://www.aquasana.com/info/tds-meter-what-is-it-and-do-you-need-it-pd.html?fbclid=IwAR2W9i3wXSKBDDvVuejlooaafd6ZVwyiwq9qtM_6HGwuQLd4cAoE6u6fy-p0.
- [4] "Dfrobot," [Online]. Available: https://wiki.dfrobot.com/Turbidity_sensor_SKU__SEN0189?fbclid=IwAR14uLTR-obK9IcqyveTQEFkOTsKLWubidhJMjGUL7ShP8a0mK4m0q6tSQ.
- [5] HBM Company, "Hbm," Hbk Company, [Online]. Available: https://www.hbm.com/en/7646/what-is-a-pressure-sensor/?fbclid=IwAR0wtvrSFIS9m4IVzoVM_piV62zht4OJk_mZHR4D4ZCGEetH2FJfSyGSnZ0.
- [6] "Resources Altium," 27 Φεβρουάριος 2018. [Online]. Available: https://resources.altium.com/p/8-bit-vs-32-bit-mcu-choosing-right-microcontroller-your-pcb-design?fbclid=IwAR0FqDdNcua_MbgV-gL0wf9fE_vpPpVJ7OpJY1Iai09wlXueqGP_E7YBUxk. [Accessed 4 Νοέμβριος 2020].
- [7] Z. Peterson, "Octopart," 25 Οκτώμβριος 2019. [Online]. Available: https://octopart.com/blog/archives/2019/10/selecting-a-high-resolution-or-high-frequency-adc?fbclid=IwAR17VtVQmA8GQt3yr_721ScQvATisc92KqOe2rIXZ0zngipPwhW3oSjWEMo.
- [8] «Electronics Notes,» [Ηλεκτρονικό]. Available: https://www.electronic-notes.com/articles/electronic_components/semiconductor-ic-memory/dynamic-ram-dram-technology.php?fbclid=IwAR1U1uJiIteUa-tKtSSjxx73tcVlnTL7_G3-u_mWi6LF2yKuGZC-T7t4HDA.
- [9] "computer.howstuffworks," 24 Αύγουστος 2000. [Online]. Available: https://computer.howstuffworks.com/question452.htm?fbclid=IwAR0FqDdNcua_MbgV-gL0wf9fE_vpPpVJ7OpJY1Iai09wlXueqGP_E7YBUxk. [Accessed 14 Μάιος 2021].
- [10] N. Koumaris, "Electronics Lab," [Online]. Available: https://www.electronic-lab.com/project/using-esp32s-flash-memory-for-data-storage/?fbclid=IwAR1hSAFmSIC_A38Y1mDCI9VP4f99pootM-OagZtf19b32-Mwxa5yDWEX-S0.

- [1 «Techopedia,» [Ηλεκτρονικό]. Available: https://www.techopedia.com/definition/24481/flash-memory?fbclid=IwAR3MtSyE5FwXdwPAkyj64MLr_r0YM3ZdE7OcENBp2XiBFb2yZOSLtd7OqSI.
- [1 «Random Nerd Tutorials,» [Ηλεκτρονικό]. Available: https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/?fbclid=IwAR1W_kUZex57b-085MyLz6cxo2XVzJaU8t7f4EsWBSsVTY9jtf33UEptGqc.
- [1 J. Weinstein, «Career Karma,» 12 Σεπτέμβριος 2020. [Ηλεκτρονικό]. Available: https://careerkarma.com/blog/what-python-is-used-for/?fbclid=IwAR1eDaz9Bzk2sqkK4mQdXc5lRZPa6f-G7UpngLI-k4PZDrt34EdN_kSNvU4.
- [1 «End Run Technologies,» End Run Technologies, [Ηλεκτρονικό]. Available: https://endruntechnologies.com/products/ntp-time-servers/ntp-client-software?fbclid=IwAR1eDaz9Bzk2sqkK4mQdXc5lRZPa6f-G7UpngLI-k4PZDrt34EdN_kSNvU4.
- [1 A. Kuchling, "docs.python.org," [Online]. Available: https://docs.python.org/3/howto/unicode.html?fbclid=IwAR142MnWPvKb2Tw0fdb3J_RdR2UuUgIY_BGHvivyMaMb2K6uRReaeBkkJTTg.
- [1 B. Kumar, "Linked In," 15 Απριλίου 2017. [Online]. Available: <https://www.linkedin.com/pulse/simplest-way-create-web-server-your-local-machine-banty-kumar?fbclid=IwAR3U3cfRdr0My8tw6No1EOGUJlj9ADXgGT8G7BHgrkWAoTsXfdSrSpqLZSA>.
- [1 «docs.python.org,» [Ηλεκτρονικό]. Available: https://docs.python.org/3/library/http.server.html?fbclid=IwAR1EtXo5GjYGGFFOxDguP9NXEjDNWFa_tQxRO3iyzTa4K5MMCxa9WFcdrwc#http.server.BaseHTTPRequestHandler.handle_one_request.
- [1 "docks.python.org," docks.python.org, [Online]. Available: <https://docs.python.org/3/library/functions.html?fbclid=IwAR1I8RNK5dLFPYzIKM8bDAMWK1Y1RJQHW3kpcyXwEqFTnBvflwpqqDtMxWg#print>.
- [1 A. King, "The internet of things: a survey," *Springer Science+Business Media*, 26 Απριλίου 2014. 9]
- [2 z. ec, "Bluetooth 5.0 Throughput Comparison for Internet of Things Usability A Survey," 0] *Academia*, pp. 2,3,6, 2017.
- [2 "Bluetooth," Bluetooth, 2021. [Online]. Available: www.bluetooth.com/learn-about-1/bluetooth/tech-overview/. [Accessed 2021].
- [2 J. G. Sponas, «Nordic Semi,» 7 Φεβρουάριος 2018. [Ηλεκτρονικό]. Available: blog.nordicsemi.com/getconnected/things-you-should-know-about-bluetooth-range. [Πρόσβαση 2016].

- [2 Lora-Alliance, "Lora-Alliance," [Online]. Available: lora-alliance.org/about-lorawan/.
3]
- [2 «Lora-Developers,» [Ηλεκτρονικό]. Available: lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/.
- [2 "The Things Network," 23 Ιανουάριος 2017. [Online]. Available: www.thethingsnetwork.org/forum/t/lora-gateway-overall-power-consumption/5071/3. [Accessed 2018].
5]
- [2 D. Delfino, "Businessinsider," 20 Απριλίου 2020. [Online]. Available: [www.businessinsider.com/what-is-a-good-internet-speed#:~:text=The%20average%20internet%20download%20speed,internet"%20by%20the%20FCC\)..](http://www.businessinsider.com/what-is-a-good-internet-speed#:~:text=The%20average%20internet%20download%20speed,internet)
6]
- [2 B. Mitchell, "Lifewire," 6 Νοεμβρίου 2020. [Online]. Available: www.lifewire.com/range-of-typical-wifi-network-816564.
7]
- [2 "Wi-Fi," [Online]. Available: [www.wi-fi.org/discover-wi-fi/wi-fi-halow](http://www.wi-fi.org/discover-wi-fi/wi-fi-halov).
8]
- [2 " 4G Wireless Standard," Οκτώμβριος 2001.
9]
- [3 D. Jones, «Lightreading,» 3 6 2019. [Ηλεκτρονικό]. Available: www.lightreading.com/mobile/5g/power-consumption-5g-basestations-are-hungry-hungry-hippos/d/d-id/749979.
- [3 P. Holslin, "Highspeedinternet," 1 Ιούλιος 2021. [Online]. Available: www.highspeedinternet.com/resources/4g-vs-5g.
1]
- [3 M. Courtney, "Datacenterdynamics," 8 February 2021. [Online]. Available: www.datacenterdynamics.com/en/analysis/power-5g-people/.
- [3 C. Craven, "Sdxcentral," 19 Ιανουαρίου 2020. [Online]. Available: www.sdxcentral.com/5g/definitions/what-is-5g-spectrum/.
- [3 A. Παναγιώτης, «Security Report,» 21 5 2019. [Ηλεκτρονικό]. Available: <https://securityreport.gr/magazine-archive/etos-2019/item/7090-asyrmata-protokolla-epikoinonias-gia-efarmoges-iot?fbclid=IwAR34x30W-03h4D9dsLIYncJitl-PcYsEwk2fVsxAlFXR2YG-4Q9sb8tqZ3w>.
4]
- [3 "rs-online," 20 Απριλίου 2015. [Online]. Available: <https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about?fbclid=IwAR1eqEcUygtPnW9U2D4fyE5McwrA3fJ0fXNDIzVEEqGC9h0nhEmbZx-nrdg>.
5]
- [3 J. G. Sponas, "Nordicsemi," 7 Φεβρουαρίου 2018. [Online]. Available: https://blog.nordicsemi.com/getconnected/things-you-should-know-about-bluetooth-range?fbclid=IwAR26xprrznuhQDW_an05YRSiFQxDQ18bCwhQkXkuz9AmcVPDGBPAwPmns3E.
6]

[3 "AVSystem," 24 Μαΐου 2019. [Online]. Available: <https://www.avsystem.com/blog/iot-protocols-7>] and-
standards/?fbclid=IwAR0OH05eeNjAW6xuC4ZFRc6_7f4fWRrWYWecoJtAPH_9UwR5PbdIY
A5lL3Y. [Accessed 4 Μαρτίου 2020].