ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Συστήματα αυτόματου ποτίσματος βασισμένα σε arduino

Πτυχιακή εργασία

Νικόλας Δημητρίου

Αθήνα, 2022

HAROKOPIO UNIVERSITY

SCHOOL OF DIGITAL TECHNOLOGY
DEPARTMENT OF INFORMATICS AND TELEMATICS

Irrigation Systems based on arduino

Bachelor thesis

Nikolas Dimitriou

Athens, 2022

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Τριμελής Εξεταστική Επιτροπή

Επιβλέπων

Θωμάς Καμαλάκης

Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Μέλη

Σωτήριος Ξύδης

Επίκουρος Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Δαλάκας Βασίλειος

Ε.ΔΙ.Π, Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Ο Νικόλας Δημητρίου δηλώνω υπεύθυνα ότι:

1. Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.

2. Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

Ευχαριστίες

Με την ολοκλήρωση της πτυχιακής μου εργασίας θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν και με στήριξαν σε όλη τη διάρκεια των σπουδών μου αλλά και της εκπόνησης της εργασίας.

Ευχαριστώ θερμά τον επιβλέπων καθηγητή της πτυχιακής μου εργασίας, κύριο Καμαλάκη Θωμά, για την υποστήριξη, τις πολύτιμες συμβουλές του.

Θα ήθελα, επίσης, να ευχαριστήσω τα μέλη της εξεταστικής επιτροπής κ. Σωτήριο Ξύδη και κ. Δαλάκα Βασίλειο που μου έκαναν την τιμή να αξιολογήσουν την προσπάθεια μου.

Ιδιαίτερα ευχαριστώ στο συμφοιτητή, συγκάτοικο και φίλο μου Κλωνής-Σκέντζος Χαράλαμπος.

Περιεχόμενα

Περίληψη

Σκοπός: Αυτή η πτυχιακή εργασία έχει ως σκοπό την μελέτη των IoT's και με βάση αυτή την μελέτη την δημιουργία ενός αυτόματου ποτιστικού συστήματος και την καταγραφή των δεδομένων που προέρχονται απο αυτό. Για την υλοποίηση χρησιμοποιήθηκε ο μικροεπεξεργαστής ESP32 NodeMCU. Για την απεικόνιση των δεδομένων χρησιμοποιήθηκε το Node-RED μαζί με το Mosquitto τα οποία μας επιτρέπουν την σύνδεση του μικροεπεξεργαστή μας με το λογισμικό, την διαχείριση των δεδομένων με βοήθεια κόμβων και την σύνδεση στο διακομιστή. Τα δεδομένα τα οποία εξετάζουμε είναι πράγματα όπως: υγρασία χώματος, υγρασία, θερμοκρασία και φως.

Λέξεις κλειδιά: [IoT, ESP32 NodeMCU, Node-RED, Mosquitto]

Abstract

The object of this paper is to offer an overview on how IoT applications work, a deeper look into how they are applied and the creation of our own IoT system. The system created in this paper is an automated irrigation system based on Arduino language and the ESP32 NodeMCU micro-controller. In order to visualize the data and the IoT system Node-RED and Mosquitto was used. This software is running on a remote Cyclades Server. The data we are gathering originates from sensors that are connected to the ESP32 NodeMCU and includes data such as: soil moisture, humidity, temperature and light.

Keywords: [IoT, ESP32 NodeMCU, Node-RED, Mosquitto]

Κατάλογος σχημάτων

Συντομογραφίες

IoT	Internet of Things/Διαδίκτυο των Πραγμάτων
WSN	Wireless Sensor Network/Ασύρματα δίκτυα αισθητήρων
MQTT	MQ Telemetry Transport

1 Εισαγωγή

1.1 Εισαγωγή)

Οι προηγμένες τεχνολογίες μπορούν να αποφέρουν οφέλη στην πλειονότητα των ανθρώπων. Τα τελευταία χρόνια, το Διαδίκτυο των Πραγμάτων (IoTs) έχει αρχίσει να παίζει σημαντικό ρόλο στην καθημερινή μας ζωή, διευρύνοντας τις αντιλήψεις και την ικανότητά μας να τροποποιούμε το περιβάλλον γύρω μας. Ιδιαίτερα οι αγροτοβιομηχανικοί και περιβαλλοντικοί τομείς εφαρμόζουν τα IoT τόσο στη διάγνωση όσο και στον έλεγχο. Έτσι, θα μελετηθεί η εφαρμογή των IoT για τη βελτιστοποίηση της γεωργίας τόσο σε μεγάλες καλλιέργειες όσο και σε μικρές φάρμες με τη βοήθεια της τεχνολογίας.

Σε μια τέτοια βελτιστοποίηση της γεωργίας, η εγκατάσταση των IoT εφαρμογών στον τομέα αυτό βελτίωσε την αποτελεσματικότητα και την αποδοτικότητα των αγροτών (Capello et al. (2016), Fang et al. (2014), Hashim et al. (2014), Kodali et al. (2014)). Μπορεί να βοηθήσει στην αξιολόγηση μεταβλητών πεδίου όπως η κατάσταση του εδάφους, οι ατμοσφαιρικές συνθήκες και η βιομάζα φυτών ή ζώων. Μπορεί επίσης να χρησιμοποιηθεί για την αξιολόγηση και τον έλεγχο μεταβλητών όπως η θερμοκρασία και υγρασία. Επιπλέον, τα IoT μπορούν να χρησιμοποιηθούν για την παρακολούθηση και τον έλεγχο παραγόντων που επηρεάζουν την ανάπτυξη και την απόδοση των καλλιεργειών.

Επομένως σε αυτή την εργασία θα εξερευνήσουμε την τεχνολογία που χρησιμοποιείται στα IoT, πως μπορεί να γίνει η συλλογή των δεδομένων από αυτά και μια προσέγγιση ενός δικού μας συστήματος. Στην προσέγγιση του δικού μας συστήματος είναι σημαντικό το κόστος της υλοποίησης να είναι χαμηλό διότι τότε μπορεί να αξιοποιηθεί και από μεγαλύτερο εύρος ανθρώπων.

1.2 Internet of things (IoT)

Το Διαδίκτυο των Πραγμάτων (IoT), που ονομάζεται επίσης Διαδίκτυο των Πάντων ή Βιομηχανικό Διαδίκτυο, είναι ένα νέο τεχνολογικό παράδειγμα που οραματίζεται ως ένα παγκόσμιο δίκτυο μηχανών και συσκευών ικανών να αλληλεπιδρούν μεταξύ τους. Το IoT αναγνωρίζεται ως ένας από τους πιο σημαντικούς τομείς της μελλοντικής τεχνολογίας και κερδίζει τεράστια προσοχή από ένα ευρύ φάσμα βιομηχανιών. Η πραγματική αξία του IoT για τις επιχειρήσεις μπορεί να γίνει πλήρως αντιληπτή όταν οι συνδεδεμένες συσκευές είναι σε θέση να επικοινωνούν μεταξύ τους και να ενσωματωθούν με συστήματα αποθέματος που διαχειρίζονται οι προμηθευτές, συστήματα υποστήριξης πελατών, εφαρμογές επιχειρηματικής ευφυΐας και επιχειρηματικά αναλυτικά στοιχεία (Lee (2015)).

Πέντε τεχνολογίες IoT χρησιμοποιούνται ευρέως για την ανάπτυξη επιτυχημένων προϊόντων και υπηρεσιών που βασίζονται στο IoT:

- Αναγνώριση ραδιοσυχνοτήτων (RFID)
- Ασύρματα δίκτυα αισθητήρων (WSN)
- Ενδιάμεσο λογισμικό
- Cloud computing
- Λογισμικό εφαρμογής IoT.

Σε αυτήν την εργασία θα εστιάσουμε στην τεχνολογία WSN. Τα ασύρματα δίκτυα αισθητήρων (WSN) αποτελούνται από χωρικά κατανεμημένες αυτόνομες συσκευές εξοπλισμένες με αισθητήρα για την παρακολούθηση φυσικών ή περιβαλλοντικών συνθηκών και μπορούν να συνεργαστούν με συστήματα RFID για την καλύτερη παρακολούθηση της κατάστασης πραγμάτων όπως η θέση, η θερμοκρασία και οι κινήσεις τους (Atzori et al. (2010)). Τα WSN επιτρέπουν διαφορετικές τοπολογίες δικτύου και επικοινωνία πολλαπλών βημάτων (multihop communication). Οι πρόσφατες τεχνολογικές εξελίξεις σε ολοκληρωμένα κυκλώματα χαμηλής κατανάλωσης και ασύρματες επικοινωνίες έχουν καταστήσει διαθέσιμες αποδοτικές, χαμηλού κόστους μικρό συσκευές χαμηλής κατανάλωσης για χρήση σε εφαρμογές WSN (Gubbi et al. (2013). Με τέτοιου είδους συσκευές θα ασχοληθούμε και εμείς στην δικιά μας προσέγγιση.

Τα WSN έχουν χρησιμοποιηθεί κυρίως σε εφοδιαστικές αλυσίδες που χρησιμοποιούν μεθόδους θερμικής και ψυκτικής συσκευασίας για τη μεταφορά προϊόντων που είναι ευαίσθητα στη θερμοκρασία (Hsueh and Chang (2010), III and Cheong (2012)). Τα WSN χρησιμοποιούνται επίσης για συστήματα συντήρησης και παρακολούθησης. Για παράδειγμα, η General Electric αναπτύσσει αισθητήρες στους κινητήρες τζετ, στις τουρμπίνες και στα αιολικά της πάρκα. Με την ανάλυση δεδομένων σε πραγματικό χρόνο, η GE εξοικονομεί χρόνο και χρήμα που σχετίζονται με την προληπτική συντήρηση. Ομοίως, η American Airlines χρησιμοποιεί αισθητήρες ικανούς να καταγράφουν 30 terabytes δεδομένων ανά πτήση για υπηρεσίες όπως η προληπτική συντήρηση.

1.3 Εφαρμογή των IoT στην γεωργία

Τα τελευταία χρόνια, τα IoTs έχουν εφαρμοστεί σε πολλές μελέτες, όπως ερευνήθηκαν στους Ojha et al. (2015) και Talavera et al. (2017). Οι εφαρμογές της τεχνολογίας στον τομέα της γεωργίας χρησιμοποιούνται για τη βελτίωση της απόδοσης ή της ποιότητας των καλλιεργειών και για τη μείωση του κόστους. Η εφαρμογή του WSN στη γεωργία ακριβείας βοηθά τους αγρότες με στατιστικό τρόπο, βοηθώντας τους να λαμβάνουν καλύτερες και καλά ενημερωμένες αποφάσεις (Fang et al. (2014), Kodali et al. (2014)).

Οι Fang et al. (2014) εισήγαγαν ένα νέο ολοκληρωμένο σύστημα πληροφοριών (IIS) για περιφερειακή περιβαλλοντική παρακολούθηση και διαχείριση, βασισμένο σε IoT, για τη βελτίωση της αποτελεσματικότητας σε πολύπλοκες εργασίες. Το προτεινόμενο IIS συνδυάζει IoT, Cloud Computing, Γεωπληροφορική (RS, GIS και GPS) και e-Science για περιβαλλοντική παρακολούθηση και διαχείριση, με μια μελέτη περίπτωσης για την περιφερειακή κλιματική αλλαγή και τις οικολογικές αντιδράσεις της, που είναι ένα από τα πιο περιζήτητα ζητήματα στον κόσμο της επιστήμης. Τα αποτελέσματα έδειξαν μεγάλα οφέλη από ένα τέτοιο σύστημα, όχι μόνο στη συλλογή δεδομένων που υποστηρίζεται από τα IoT, αλλά και σε υπηρεσίες web και εφαρμογές που βασίζονται σε πλατφόρμες υπολογιστικού νέφους και ηλεκτρονικής επιστήμης. Η αποτελεσματικότητα της παρακολούθησης και της λήψης αποφάσεων βελτιώθηκαν προφανώς.

Επιπλέον, τα IoTs εφαρμόστηκαν στην αλυσίδα παραγωγής αγροτοβιομηχανίας (Capello et al. (2016). Medela et al. (2013), Li et al. (2013), Ruan and Shi (2016)). Παρουσιάστηκε μια πρωτοποριακή αρχιτεκτονική βασισμένη στην ιδέα των IoT, συνδυάζοντας ασύρματες και κατανεμημένες ειδικές συσκευές αισθητήρων με την προσομοίωση κλιματικών συνθηκών, προκειμένου να παρακολουθείται η εξέλιξη των σταφυλιών για τα οινοποιεία.

Οι Li et al. (2013) παρουσίασαν ένα πληροφοριακό σύστημα για τη γεωργία βασισμένο σε IoT, με κατανεμημένη αρχιτεκτονική. Σε αυτή τη μελέτη, η παρακολούθηση και η ανίχνευση ολόκληρης της διαδικασίας γεωργικής παραγωγής έγινε με κατανεμημένους διακομιστές IoTs. Επιπλέον, ένα σύστημα ανακάλυψης πληροφοριών σχεδιάστηκε για την εφαρμογή, τη συλλογή, την τυποποίηση, τη διαχείριση, τον εντοπισμό και την αναζήτηση επιχειρηματικών δεδομένων από τη γεωργική παραγωγή. Οι Fang et al. (2014) πρότειναν ένα κοινό πλαίσιο σχεδίασης επιχείρησης-τεχνολογίας με επίκεντρο την αξία για την παροχή πληροφοριών στον τελικό χρήστη/καταναλωτή σχετικά με την προέλευση και τις ιδιότητες του προϊόντος. Οι Capello et al. (2016) εφάρμοσαν IoT εφαρμογή για μια υπηρεσία παρακολούθησης σε πραγματικό χρόνο, προκειμένου να καταστεί δυνατή η ιχνηλάτηση των προϊόντων από τον τελικό καταναλωτή πίσω στο πεδίο. Οι Ruan and Shi (2016) παρουσίασαν ένα πλαίσιο IoTs για την αξιολόγηση της φρεσκάδας των φρούτων στις παραδόσεις ηλεκτρονικού εμπορίου, η οποία ήταν μια μη παραδοσιακή υπηρεσία λιανικής που αντιμετωπίζει μοναδικές προκλήσεις στη μεταφορά, λόγω της φθαρτής των προϊόντων και τις ακριβής επιμέλειας.

1.4 Ανάλυση γεωργικών δεδομένων

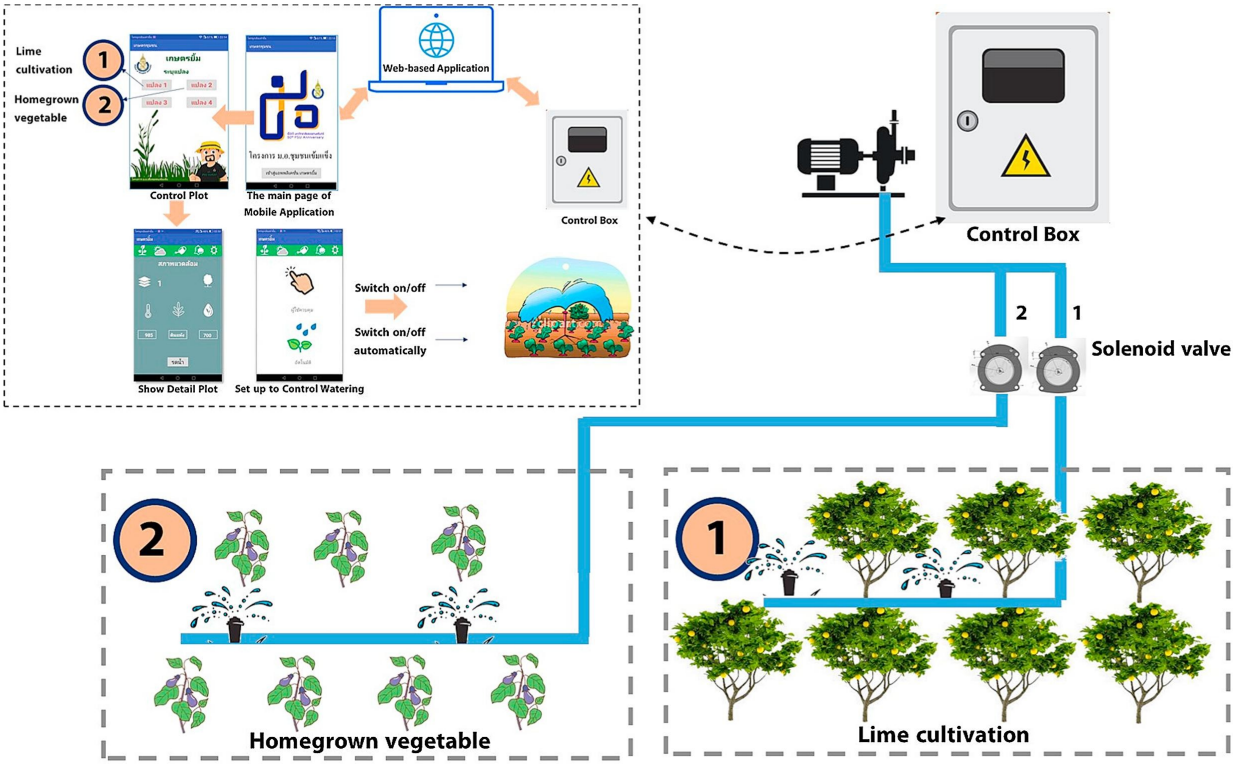
Η χρήση των IoT οδηγεί σε δεδομένα μεγάλης κλίμακας που παρέχουν πολύτιμες πληροφορίες. Για το λόγο αυτό, πολλές μελέτες έχουν προσπαθήσει να μετατρέψουν τέτοια δεδομένα σε χρήσιμες πληροφορίες και γνώσεις, όπως ερευνήθηκε στο (Kamilaris et al. (2017).

Οι Pahuja et al. (2013)) ανέπτυξαν ένα διαδικτυακό σύστημα παρακολούθησης και ελέγχου μικροκλίματος για θερμοκήπια. Το σύστημα υποστηρίχθηκε από ένα WSN για τη συλλογή και ανάλυση δεδομένων αισθητήρων που σχετίζονται με τα φυτά για την παροχή ελέγχου του κλίματος, της λίπανσης, της άρδευσης και των παρασίτων. Οι Tripathy et al. (2014) χρησιμοποίησαν τα δεδομένα που προέρχονται από το WSN για να αποκομίσουν πιο πολλές πληροφορίες μέσω εξόρυξης δεδομένων. Αυτή η μελέτη επικεντρώθηκε στη νόσο των κηλίδων στα φύλλα αξιολογώντας τις σχέσεις καλλιέργειας-καιρού-περιβάλλοντος-ασθένειας, με βάση ασύρματους αισθητήρες και επιτήρηση σε επίπεδο πεδίου. Εκπαιδεύτηκε ένας ταξινομητής για την πρόβλεψη της νόσου. Ο Xian (2017) πρότεινε ένα νέο βολικό διαδικτυακό σύστημα παρακολούθησης για IoT, βασισμένο στο cloud computing. Μετά τη συσσώρευση αρκετών δεδομένων από ένα γεωργικό σύστημα IoTs, αποδείχθηκε η μοντελοποίηση των σχετικών λειτουργικών απαιτήσεων για την προώθηση της εφαρμογής ανάλυσης μεγάλων δεδομένων στη γεωργία.

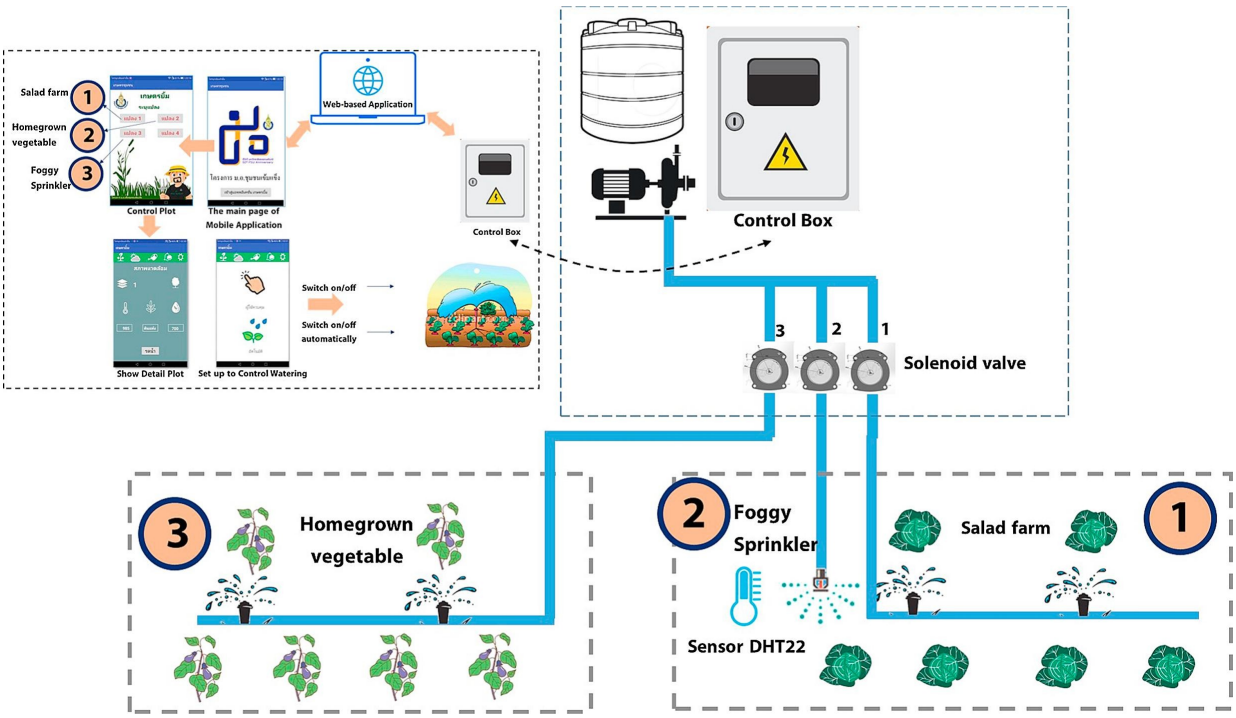
2 Υλοποίηση IoT εφαρμογής

2.1 Εφαρμογή σε πραγματικό παράδειγμα

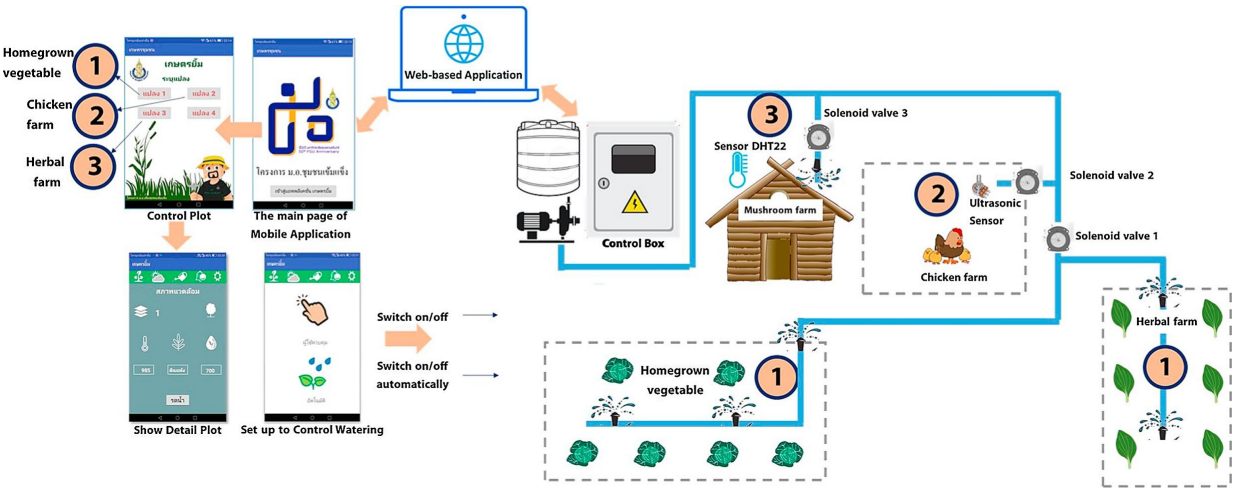
Ας ακολοθήσουμε το παράδειγμα που αναφέρεται στο Muangprathub et al. (2019). Στο παράδειγμα αυτό γίνεται η εγκατάσταση IoT σε 3 διαφορετικά αγροκτήματα που απέχουν μεταξύ τους και οι τεχνικές καλλιέργειας και τα προϊόντα τους διαφέρουν. Το πρώτο αγρόκτημα έχει καλλιέργεια μοσχολέμονου και ντόπιων λαχανικών. Το δεύτερο αγρόκτημα καλλιεργεί σαλάτες και ντόπια λαχανικά. Το τελευταίο αγρόκτημα έχει ένα ολοκληρωμένο σύστημα καλλιέργειας με φάρμα σαλατών, φάρμα κοτόπουλου, φάρμα μανιταριών και φάρμα βοτάνων. Τα τρία παραδείγματα αγροκτημάτων αντιπροσωπεύονται από τα μοντέλα στα Σχήματα 1,2 και 3.



Σχήμα 1: Σχήμα πρώτης φάρμας



Σχήμα 2: Σχήμα δεύτερης φάρμας



Σχήμα 3: Σχήμα τρίτης φάρμας

Τα IoT που εγκαταστάθηκαν στα αγροκτήματα ακολούθησαν μια κοινή λογική. Γίνεται χρήση ενός κουτί ελέγχου το οποίο είναι συνδεδεμένο με τους αισθητήρες, τη αντλία νερού και τις ηλεκτρομαγνητικές βαλβίδες για τον έλεγχο του νερού. Το κουτί ελέγχου στέλνει τα δεδομένα του σε πραγματικό χρόνο σε μια διαδικτυακή εφαρμογή όπου ο διαχειριστής επιλέγει πως θα αξιοποιήσει τα δεδομένα. Τέλος έχει δημιουργηθεί μια εφαρμογή για τον χρήστη που του επιτρέπει την επιλογή του τρόπου ποτίσματος και την προβολή δεδομένων για το αγρόκτημά του.

Στο πρώτο αγρόκτημα το κουτί ελέγχου είναι υπεύθυνο για το πότισμα των λαχανικών και των δέντρων μοσχολέμονου. Ο διαχειριστής ορίζει με βάση τις μετρήσεις των αισθητήρων υγρασίας χώματος πότε να ποτίζουν οι ψεκαστήρες οι οποίοι ελέγχονται από το κουτί ελέγχου μέσω των ηλεκτρομαγνητικών βαλβίδων και κατ' επέκταση την αντλία νερού. Παρόλα αυτά ο χρήστης παραμένει να έχει την επιλογή του τρόπου ποτίσματος σε περίπτωση που το θελήσει.

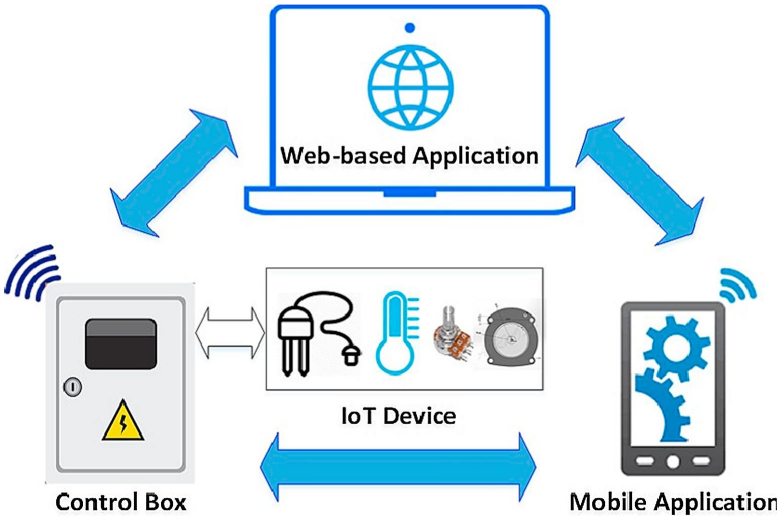
Το δεύτερο αγρόκτημα ακολουθεί τον τρόπο ποτίσματος του πρώτου. Γίνεται όμως η χρήση του αισθητήρας DHT22 ο οποίος μας δίνει τις μετρήσεις τις υγρασίας και τις θερμοκρασίας. Επομένως οι μεταβλητές με τις οποίες ο διαχειριστής επιλέγει την έναρξη του αυτόματου ποτίσματος μπορέι να διαφέρουν. Επίσης υπάρχουν πιο πολλά δεδομένα για τον χρήστη και τον διαχειριστή που μπορούν να φανούν χρήσιμα.

Το τελευταίο αγρόκτημα αξιοποιεί τις τεχνικές ποτίσματος που είδαμε πιο πάνω. Ωστόσο το πότισμα στη φάρμα μανιταριών βασίζεται στις μετρήσεις του αισθητήρα DHT22 και για τη φάρμα κοτόπουλου γίνεται χρήση ενός αισθητήρα υπερήχων. Ο αισθητήρας υπερήχων μετράει αποστάσεις και με την ανάλυση των δεδομένων του μπορεί να μας δείξει την χρονική στιγμή στην οποία εμφανίζεται ένα αυγό.

Τα παραπάνω αγροκτήματα έδειξαν ότι τα IoT μπορούν να χρησιμοποιηθούν για να υποστηρίξουν και να βοηθήσουν τους αγρότες σε κάθε είδους γεωργία. Επιπλέον, οι αγρότες μπορούν να ξοδεύουν τον χρόνο που εξοικονομούν σε άλλες δραστηριότητες για αυξημένο εισόδημα. Στην έρευνα αναφέρεται ότι οι πληροφορίες από τις συσκευές IoTs σε κάθε χωριό χρησιμοποιήθηκαν για τον έλεγχο της ενεργοποίησης-απενεργοποίησης των ψεκαστήρες νερού, αυτόματα. Επίσης έγινε καταγραφή των δεδομένων για 170 μέρες, αυτά τα δεδομένα αποτελούνται από: θερμοκρασία, υγρασία και υγρασία του εδάφους, και συλλεγόντουσαν κάθε 20 λεπτά.

2.2 Σχεδιασμός και επισκόπηση του συστήματος

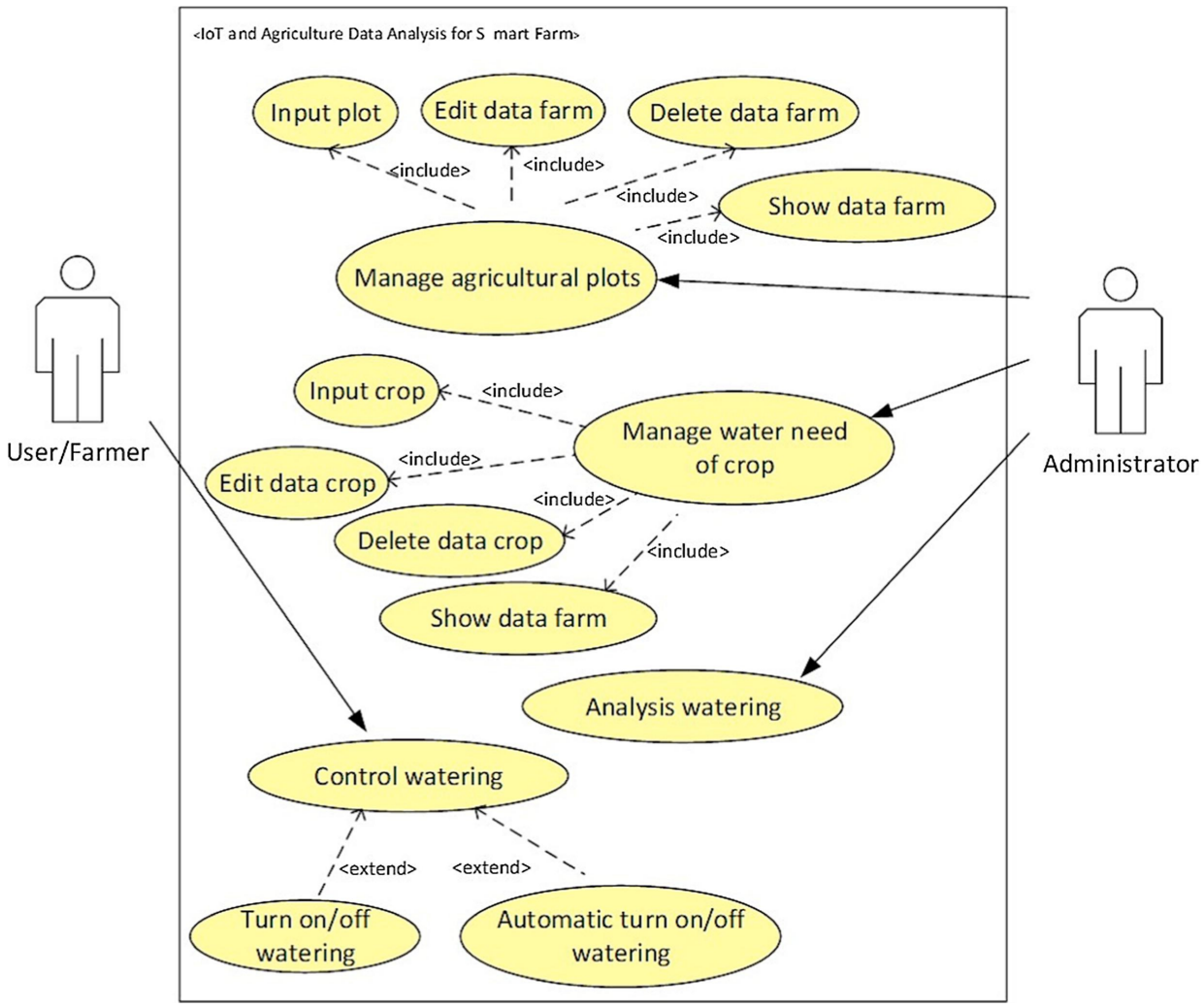
Η προσέγγιση που έγινε στο Muangprathub et al. (2019) είχε ως στόχο το σχεδιασμό και την εφαρμογή συστημάτων με αισθητήρες στον τομέα της καλλιέργειας και τη διαχείριση δεδομένων με χρήση smartphone με εφαρμογή web. Τα τρία στοιχεία είναι το υλικό, η εφαρμογή Ιστού και smartphone εφαρμογή, όπως φαίνεται στο Σχήμα 4.



Σχήμα 4: Μια επισκόπηση του συστήματος.

Αναφερόμενοι στο Σχ. 4, το πρώτο εξάρτημα σχεδιάστηκε και αναπτύχθηκε σε μορφή κουτιού ελέγχου. Αυτό το πλαίσιο ελέγχου έχει σχεδιαστεί για τον έλεγχο των συσκευών IoT και τη λήψη δεδομένων από τις καλλιέργειες. Το υλισμικό και οι συσκευές IoT που χρησιμοποιούνται στο κουτί ελέγχου είναι οι εξής: μικροεπεξεργαστής NodeMCU, αισθητήρας υγρασίας και θερμοκρασίας DHT22, 5 μονάδες ρελέ, ηλεκτρομαγνητική βαλβίδα και αρχιτεκτονική "Crop field".

Το δεύτερο στοιχείο που αναφέρθηκε στο Muangprathub et al. (2019) είναι η διαδικτυακή εφαρμογή. Αυτό το στοιχείο περιλαμβάνει τη διαχείριση των πληροφοριών σε πραγματικό χρόνο από συσκευές IoT σε κάθε αγρόκτημα. Η διαδικτυακή εφαρμογή επιτρέπει σε έναν διαχειριστή να χειριστεί τις συνθήκες για τις ανάγκες σε νερό κάθε καλλιέργειας. Επιπλέον, οι λεπτομέρειες των πληροφοριών από τις συσκευές IoT μπορούν να προβληθούν από έναν διαχειριστή για τη διαχείριση κάθε είδους γεωργίας. Αυτά τα δεδομένα αναλύθηκαν για να προβλεφθεί η ανάγκη του νερού στις καλλιέργειες στο μέλλον. Η διαδικτυακή εφαρμογή για τον χειρισμό δεδομένων IoT από κάθε εγκατάσταση και οι λεπτομέρειες οποιασδήποτε καλλιέργειας φαίνονται στο Σχήμα 5.



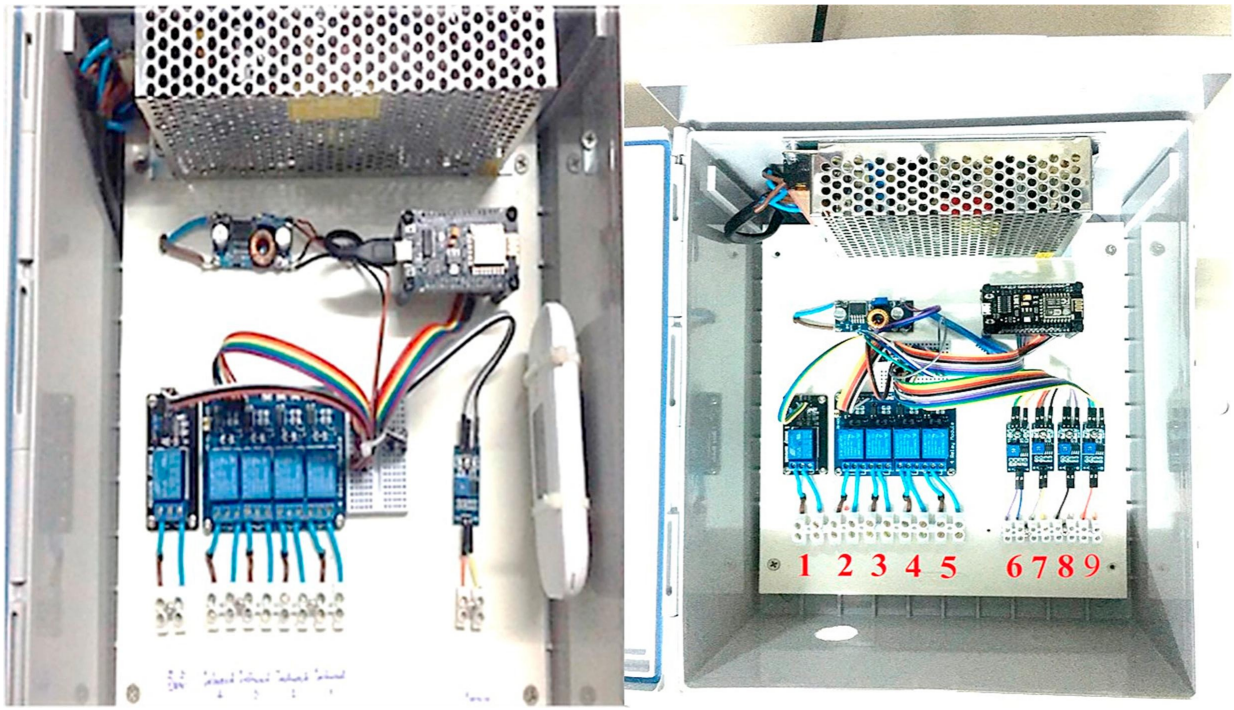
Σχήμα 5: Use case diagram που εμφανίζει τη διεργασία με web-based και mobile εφαρμογές

Το τελευταίο στοιχείο είναι η εφαρμογή που δημιουργήθηκε και χρησιμοποιήθηκε από αγρότη σε κινητό τηλέφωνο. Η εφαρμογή σχεδιάστηκε για να ελέγχει το πότισμα μετά την ανάλυση των δεδομένων. Η εφαρμογή παρείχε δύο λειτουργίες, έτσι ώστε οι αγρότες να μπορούν να ελέγχουν το πότισμα χειροκίνητα ή το προτεινόμενο σύστημα μπορεί να ενεργοποιεί/απενεργοποιεί αυτόματα το πότισμα με βάση τις πληροφορίες IoTs, όπως φαίνεται στο Σχήμα 5.

2.3 Εκτέλεση του συστήματος

Το προτεινόμενο σύστημα υλοποιείται με τρία μέρη, δηλαδή κουτί ελέγχου, web-based εφαρμογή και την smartphone εφαρμογή. Το κουτί ελέγχου διατηρεί τις ηλεκτρονικές συσκευές σε ένα αδιάβροχο κουτί, όπως φαίνεται στο Σχήμα 6. Το κιβώτιο ελέγχου θα μπορούσε να βρίσκεται οπουδήποτε στο αγρόκτημα ή κοντά στο αγρόκτημα, έχοντας τους αισθητήρες υγρασίας εδάφους, την ηλεκτρομαγνητική βαλβίδα, τον αισθητήρα DHT22 και έναν αισθητήρα υπερήχων συνδεδεμένο στο κιβώτιο ελέγχου. Τα IoTs εφαρμόζονται στους αισθητήρες υγρασίας του εδάφους για τη μέτρηση της υγρασίας του εδάφους των καλλιεργειών και για τον έλεγχο της αυτόματης ενεργοποίησης και απενεργοποίηση του ψεκαστήρα του νερού. Η ηλεκτρομαγνητική βαλβίδα χρησιμοποιή-

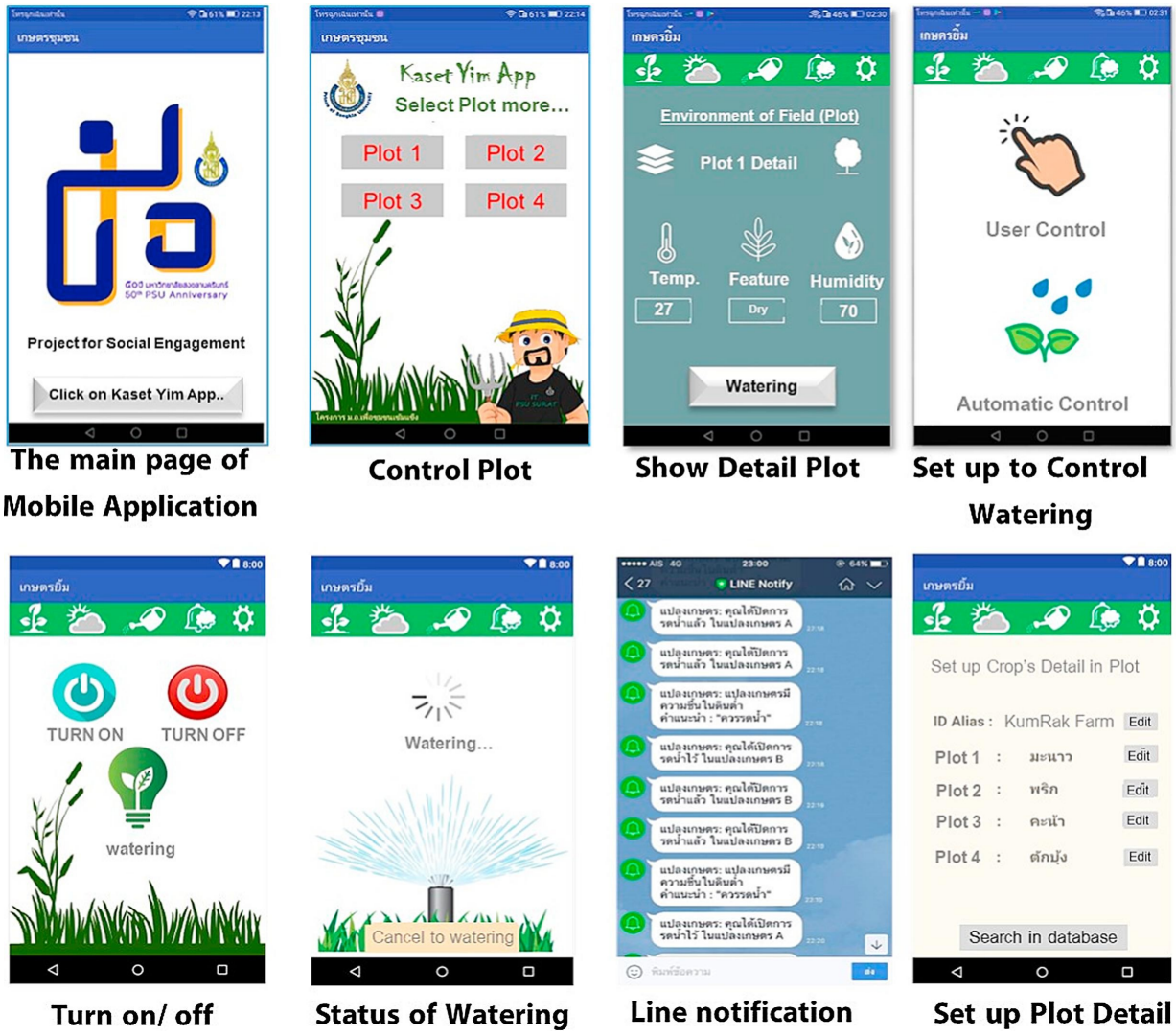
θηκε για τον έλεγχο της ροής του νερού με δράση on/off. Ο αισθητήρας DHT22 χρησιμοποιήθηκε για τον έλεγχο της υγρασίας της φάρμας μανιταριών.



Σχήμα 6: Ο σχεδιασμός του κιβωτίου ελέγχου

Το δεύτερο μέρος είναι μια διαδικτυακή εφαρμογή που λαμβάνει πληροφορίες γεωργίας από το NodeMCU. Έχει πρόσβαση στο διαδίκτυο μέσω σύνδεσης WiFi. Η διαδικτυακή εφαρμογή υλοποιήθηκε για τη διαχείριση αγροτεμαχίων και τη διαχείριση του ποτίσματος της καλλιέργειας ή για την ανάλυση του κατάλληλου ποτίσματος. Επιπλέον, αυτό το μέρος περιλαμβάνει την ανάλυση των γεωργικών δεδομένων.

Το τελευταίο μέρος υλοποιήθηκε προκειμένου να συνδεθεί με τον αγρότη. Η smartphone εφαρμογή χρησιμοποιείται για τον έλεγχο της ενεργοποίησης-απενεργοποίησης του ηλεκτρικού συστήματος από τον αγρότη. Αυτή η εφαρμογή έχει 2 λειτουργίες. αυτόματο και χειροκίνητο. Με χειροκίνητα εννοούμε την επιλογή ποτίσματος μέσω της εφαρμογής όποτε θελήσει ο αγρότης. Το αυτόματο σύστημα ενεργοποιήθηκε όταν εντοπίστηκαν καθορισμένες τιμές απο τους αισθητήρες χωρίς είσοδο από τον χρήστη. Το Σχήμα 7 είναι ένα παράδειγμα της smartphone εφαρμογής για τον έλεγχο του ποτίσματος. Οι κύριες λειτουργίες της εφαρμογής είναι η παρακολούθηση του ποτίσματος, η ρύθμιση των λεπτομερειών της καλλιέργειας σε κάθε οικόπεδο και οι ειδοποιήσεις μέσω της εφαρμογής LINE.



Σχήμα 7: Παράδειγμα smartphone εφαρμογής για τη διαχείριση ποτίσματος

3 Προσέγγιση δικού μας συστήματος

3.1 Πλάνο υλοποίησης

Με βάση αυτά που μελετήσαμε στα δυο εισαγωγικά κεφάλαια θα επιχειρήσουμε να δημιουργήσουμε ένα δικό μας IoT που εφαρμόζει στην γεωργία. Το σύστημά μας θα είναι πιο μικρής κλίμακας και χαμηλού προϋπολογισμού. Ο στόχος είναι η δημιουργία ενός αυτόματου ποτιστικού συστήματος με βάση Arduino, η συλλογή δεδομένων και η απεικόνισή των δεδομένων. Σε κάποια στάδια θα ακολουθήσουμε μεθόδους που αναφέρθηκαν πιο πάνω και σε άλλα θα εφαρμόσουμε την δικιά μας προσέγγιση. Ελπίζουμε ότι το σύστημα και τα αποτελέσματα του θα μπορούν να είναι χρήσιμα για κάποιον χρήστη και να μπορεί να υλοποίηση και ο ίδιος το σύστημα στον κήπο του.

Η σύνδεση μεταξύ της IoT συσκευής και υπολογιστή θα γίνει μέσω του διαδικτύου. Θα έχουμε την ικανότητα να παρακολουθούμε σε πραγματικό χρόνο τις μετρήσεις από τους αισθητήρες, την δυνατότητα επιλογής αυτόματου ή manual ποτίσματος και τέλος την δυνατότητα να απεικονίσουμε τα δεδομένα μιας ημέρας σε έναν γράφο. Όλα αυτά θα τα επιτεύξουμε μέσω προγραμμάτων των Node-RED και Mosquitto που μας επιτρέπουν την επικοινωνία μεταξύ υλισμικό και λογισμικού.

Για την υλοποίηση της σωστής επικοινωνίας υλισμικό με λογισμικού πρέπει να γίνει πρώτα μια προετοιμασία. Επιδιώκουμε να έχουμε μια μόνιμη επικοινωνία με το ποτιστικό σύστημα χωρίς να σπαταλάμε πόρους άλλων συσκευών. Γι' αυτό τρέχουμε τα προγράμματα του Node-RED και του Mosquitto πάνω σε έναν Ubuntu LTS Server. Έτσι έχουμε πρόσβαση στην επεξεργασία των νημάτων αλλά και στην παρακολούθηση των δεδομένων μας όποτε θέλουμε, απο όποια συσκευή θέλουμε. Παρακάτω θα μελετήσουμε σε τι ακριβώς μας οφείλουν αυτά τα προγράμματα.

3.2 Node-RED

3.2.1 Εισαγωγή στο Node-RED

Το Node-RED είναι ένα προγραμματιστικό εργαλείο το οποίο ενώνει IoT συσκευές, διεπιφάνεια προγραμματιστικών εφαρμογών (APIs) και διαδικτυακές υπηρεσίες.

Μας προσφέρει ένα browser-based editor που μας επιτρέπει την εύκολη δημιουργία ροών με ένα μεγάλο εύρος νημάτων τα οποία μπορούν να προγραμματιστούν το καθένα ξεχωριστά σε JavaScript. Με την βοήθεια του mosquitto και της βιβλιοθήκης PubSubClient έχουμε την δυνατότητα να στείλουμε δεδομένα απο τους αισθητήρες οι οποίοι έχουν συνδεθεί στο ESP32 NodeMCU κατευθείαν σε νήματα στον editor Node-RED. Απο εκεί και πέρα είναι στο χέρι μας πως θα αξιοποιήσουμε τα δεδομένα αυτά.

3.2.2 Προετοιμασία του Node-RED

Πριν ξεκινήσουμε να δημιουργούμε flows για τη καταγραφή και τον οραματισμό των δεδομένων θα προετοιμάσουμε τον σέρβερ και κάποιες ρυθμίσεις του Node-RED.

Για αρχή, αφού συνδεθούμε στο σέρβερ, θα κάνουμε εγκατάσταση του Node-RED όπως και του Node-RED Dashboard. Το Node-RED dashboard μας προσφέρει μια ποικιλία από nodes όπως μετρητές και γράφους και θα χρησιμοποιηθούν για τον καλύτερο οραματισμό των δεδομένων. Επειδή δεν έχουμε κάποιο εικονικό περιβάλλον για το σέρβερ που αξιοποιούμε θα εγκαταστήσουμε όλα τα απαραίτητα προγράμματα μέσω του terminal. Για την σωστή εγκατάσταση ακολουθούμε το documentation από την επίσημη ιστοσελίδα του Για την σωστή εγκατάσταση ακολουθούμε το documentation απο την επίσημη ιστοσελίδα του Node-RED.

Εγκατάσταση Node-RED :

```
$ sudo npm install -g --unsafe-perm node-red
```

Εγκατάσταση Node-RED Dashboard:

Πλοηγούμαστε στο directory στο οποίο εγκαταστήσαμε το Node-RED και τρέχουμε.

```
$ cd ~/.node-red
$ npm install node-red-dashboard
```

3.2.3 Είσοδος, Διασφάλιση και αυτοματοποίηση του Node-RED

Για να αποκτήσουμε πρόσβαση στο περιβάλλον του Node-RED θα πρέπει αρχικά να γίνει η εκκίνηση του προγράμματος από το terminal.

```
$ node-red
```

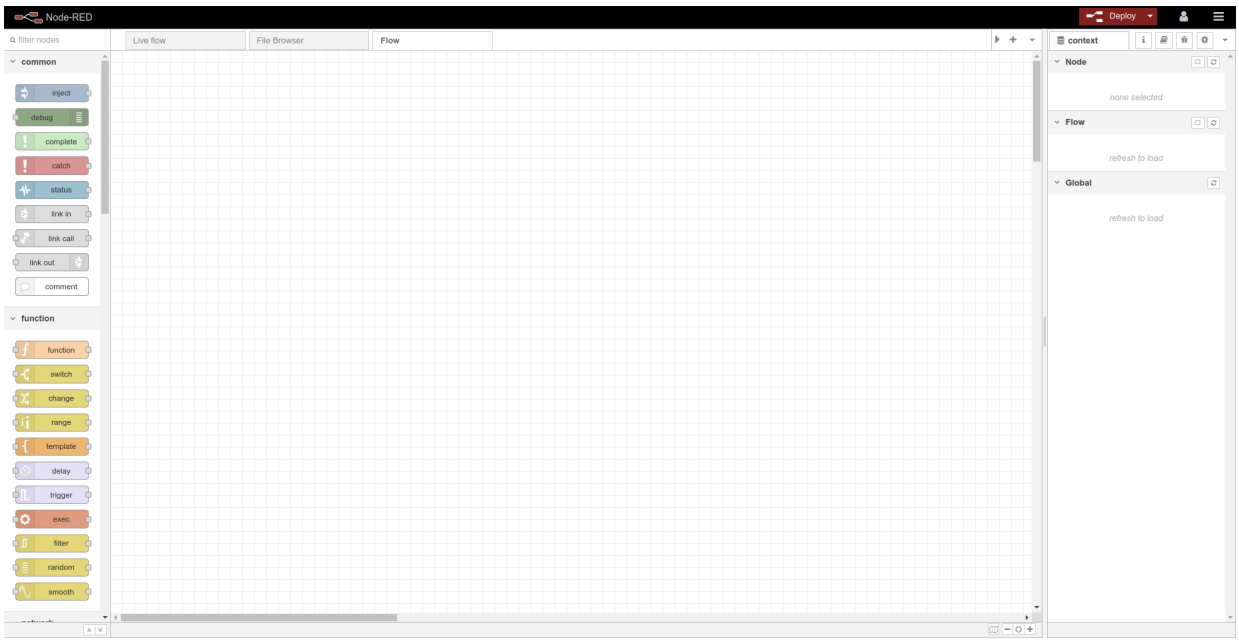
Έπειτα είναι δυνατή η σύνδεση στο περιβάλλον μέσω ενός browser το οποίο πρέπει να τρέξουμε τοπικά. Σε αυτή τη φάση αντιμετωπίζουμε όμως ένα πρόβλημα. Μέχρι στιγμής χειριζόμασταν το σέρβερ μέσω terminal επειδή όπως έχει αναφερθεί πιο πάνω δεν έχουμε κάποιο γραφικό περιβάλλον. Άρα πρέπει να δώσουμε πρόσβαση στο port το οποίο χρησιμοποιεί το Node-RED ώστε να έχουμε πρόσβαση από το δικό μας μηχάνημα.

```
$ sudo ufw allow 1880/tcp
```

*Σημειώνετε σε αυτή τη φάση ότι κάποιες φορές θα πρέπει να δοθεί και πρόσβαση στο port 1883.

Πλέον είναι δυνατή η σύνδεση στο περιβάλλον του Node-RED απο το μηχάνημα μας. Πληκτρολογούμε την IP του σέρβερ μαζί με το Port το οποίο ανοίξαμε. Στη δικιά μας περίπτωση είναι ο ακόλουθος σύνδεσμος:

```
http://83.212.110.251:1880
```



Σχήμα 8: Περιβάλλον του Node-RED

Που μας εμφανίζει την παρακάτω σελίδα όπου θα δημιουργούμε τα flows μας.

Για να διασφαλίσουμε την είσοδο στο σύστημα μπορούμε να ακολουθήσουμε τις οδηγίες που μας παρέχει το Node-RED. Η διασφάλιση είναι σημαντικό κομμάτι από τη στιγμή που δουλεύουμε με ανοιχτά Ports και IoT applications, για τα δεδομένα μας, για την ομαλή λειτουργία των flow και των συσκευών μας. Η διασφάλιση γίνεται μέσω δημιουργία hash κωδικού για τον Admin του συστήματος. Επίσης μπορούν να δημιουργηθούν και Users στην περίπτωση που θέλουμε να μοιραστούμε το πρότζεκτ μας χωρίς όμως να μπορεί ο User να κάνει αλλαγές στα flows.

Το τελευταίο βήμα για να ολοκληρωθεί η προετοιμασία του Node-RED είναι η αυτόματη εκκίνησή του όταν ανοίγει ο σέρβερ. Με αυτόν το τρόπο θα γίνεται η καταγραφή των στοιχείων χωρίς να χρειαστεί να συνδεθούμε στο σέρβερ και να τρέξουμε το πρόγραμμα κάθε φορά. Αυτό επιτυγχάνεται με τα ακόλουθα βήματα. Αρχικά κάνουμε εγκατάσταση το pm2. Το PM2 είναι ένα process manager για το Node.js. Μας διευκολύνει να τρέξουμε εφαρμογές στην εκκίνηση του υπολογιστή μας και να τις επανεκκινήσει εάν χρειαστεί.

```
1  $ sudo npm install -g pm2
```

Στην πορεία πρέπει να δούμε σε ποιο directory έχει γίνει εγκατάσταση το Node-RED. Συνήθως θα είναι στο /usr/bin/node-red αλλά μπορούμε να εκτελέσουμε το which για να σιγουρευτούμε.

```
1  $ which node-red
```

Μετέπειτα λέμε στο PM2 να τρέχει το Node-Red. Στη δικιά μας περίπτωση το directory είναι /usr/bin/node-red.

```
1  $ pm2 start /usr/bin/node-red -- -v
```

Αυτό ήταν το τελευταίο βήμα τις προετοιμασίας του Node-RED. Θα εξερευνήσουμε τις λειτουργίες του σε παρακάτω κεφάλαιο.

3.3 Eclipse Mosquitto

3.3.1 Εισαγωγή στο Mosquitto

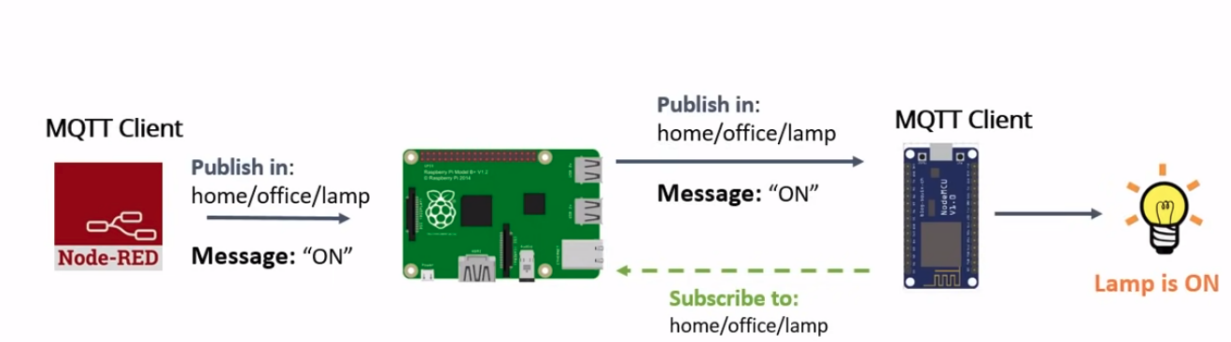
Το Eclipse Mosquitto είναι ένα open source message broker το οποίο εφαρμόζει το πρωτόκολλο MQTT(MQ Telemetry Transport) εκδοχών 5.0, 3.1 και 3.1.1. Το Mosquitto είναι ελαφρύ και κατάλληλο για χρήση σε μεγάλο εύρος συσκευών, είτε αυτές είναι μικρής τάσης επεξεργαστές είτε μεγάλοι σέρβερ.

Το πρωτόκολλο MQTT μας προσφέρει μια lightweight μέθοδος για να μεταφέρουμε ένα μύνημα χρησιμοποιώντας ένα μοντέλο τύπου publish/subscribe. Αυτό το κάνει ιδανικό για την ανταλλαγεί μυνημάτων με Internet of Things συσκευές όπως κινητά, συσκευές μικρής τάσης και μικροεπεξεργαστές.

Στην προκειμένη περίπτωση του αυτόματου σύστημα ποτίσματος, καθιερώνει την επικοινωνία μεταξύ του ESP32 NodeMCU και του Node-RED. Μας επιτρέπει να στείλουμε οδηγίες ώστε να ελέγχουμε κάποιο αποτέλεσμα ή να διαβάσουμε δεδομένα και να τα δημοσιεύσουμε.

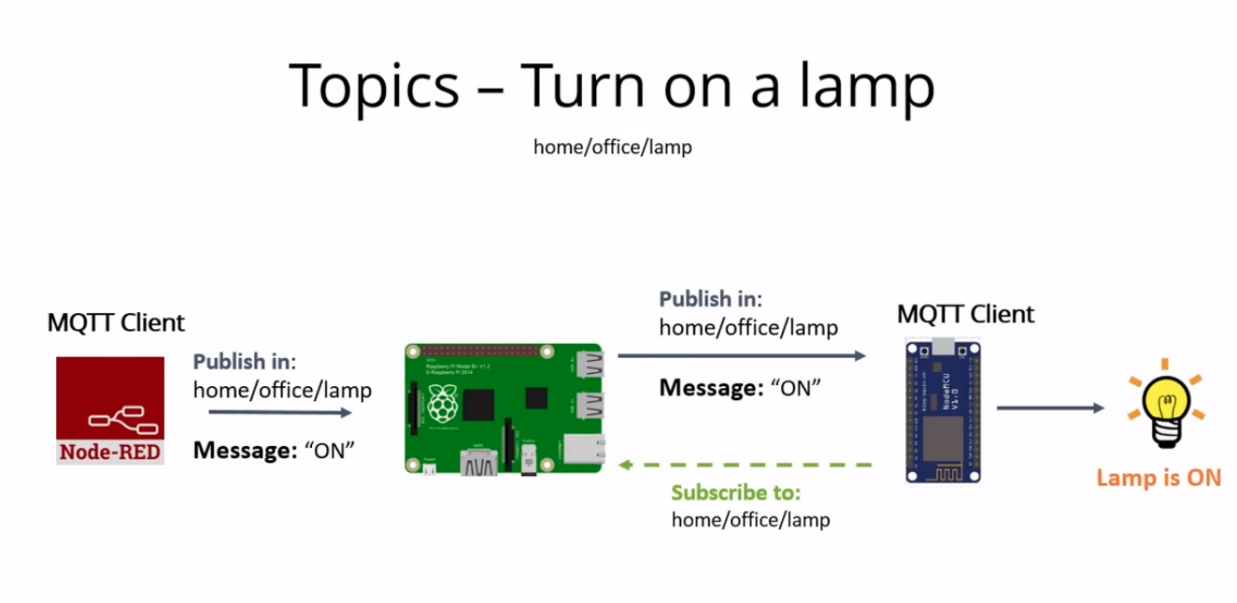
Για την καλύτερη κατανόηση θα εξερευνήσουμε τις παρακάτω έννοιες:

- Publish/Subscribe: Μια συσκευή μπορεί να κάνει publish μηνύματα (Messages) σε άλλες συσκευές ή να κάνει subscribe σε ένα συγκεκριμένο topic ώστε να λαμβάνει μηνύματα.



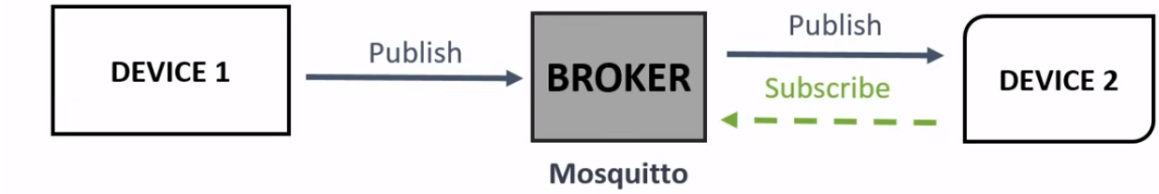
Σχήμα 9: Λειτουργία του Publish and Subscribe

- Messages: Είναι οι πληροφορίες που μοιράζονται μεταξύ των συσκευών - εντολές ή δεδομένα
- Topics: Ο τρόπος με τον οποίο δηλώνουμε ενδιαφέρον για εισερχόμενα μηνύματα ή ορίζουμε που θέλουμε να δημοσιεύσουμε τα μηνύματα. Ένα topic μπορεί να έχει πολλά στάδια τα οποία ξεχωρίζονται με τον χαρακτήρα "/" π.χ. home/office/lamp



Σχήμα 10: Topics

- Broker: Ο MQTT Broker είναι κυρίως υπεύθυνος να λαμβάνει όλα τα μηνύματα, να τα φιλτράρει και να αποφασίζει ποιος ενδιαφέρεται γι αυτά τα μηνύματα και εν τέλη να δημοσιεύει αυτά τα μηνύματα σε όσους ενδιαφέρονται.



Σχήμα 11: mqtt Broker

3.3.2 Προετοιμασία του Mosquito

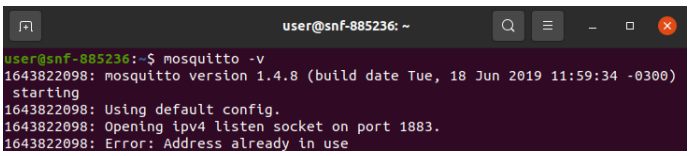
Όπως και στην προετοιμασία του Node-RED θα συνδεθούμε στο σέρβερ μέσω terminal και θα γράψουμε τις ακόλουθες εντολές για την εγκατάσταση του Mosquitto.

```
$ sudo apt install -y mosquitto mosquitto-clients
```

Για να ελέγξουμε αν η εγκατάσταση έγινε σωστά τρέχουμε:

```
$ mosquitto -v
```

Εδώ μας εμφανίζει την έκδοση του προγράμματος. Πέρα όμως από την έκδοση μας εμφανίζει και το port στο οποίο λειτουργεί ο broker, αυτό το port, στην προκειμένη 1883, θα αξιοποιηθεί αργότερα στον κώδικα ώστε να επικοινωνήσουμε με τον broker. Τέλος θα πρέπει, όπως και με το



Σχήμα 12: Εκδοχή του Mosquitto

Node-RED, να σιγουρευτούμε πως η εκκίνηση του προγράμματος γίνεται αυτόματα όποτε ανοίγει ο σέρβερ. Αυτό το επιδιώκουμε με την ακόλουθη εντολή.

```
1   $ sudo systemctl enable mosquitto.service
```

3.4 Cyclades Server

Η δημιουργία του αυτόματου σύστημα ποτίσματος έχει ως στόχο την διευκόλυνση και αυτοματοποίηση μιας καθημερινής πράξης, την φροντίδα ενός φυτού όσο βρισκόμαστε μακριά από αυτό άλλα και την μελέτη των συνθηκών υπό τις οποίες βρίσκεται. Άρα είναι αντιπαραγωγικό να αναγκαζόμαστε να κρατάμε κάποιον υπολογιστή όλη την ώρα ανοιχτό ώστε να τρέχει τα προγράμματα του Node-Red και του Mosquitto. Πολλές IoT εφαρμογές λύνουν αυτό το πρόβλημα με την χρήση ενός Raspberry Pi αλλά δεν είναι εφικτό για πολλούς να έχουν μια τέτοια η παρόμοια συσκευή.

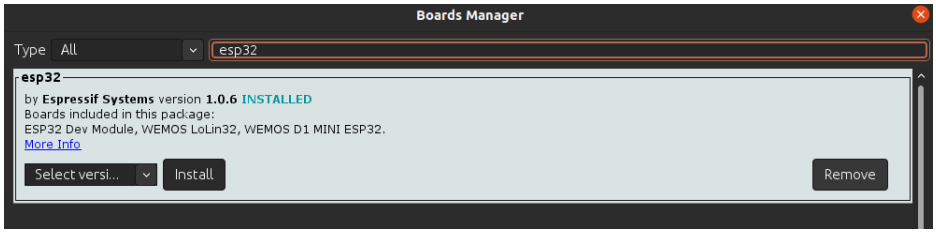
Για αυτό τον λόγω αξιοποιήθηκε η πλατφόρμα του Okeanos η οποία μας παρέχει IaaS (Infrastructure as a Service) υπηρεσίες. Αυτό σημαίνει ότι μπορούμε να δημιουργήσουμε τον δικό μας υπολογιστή, πάντα συνδεδεμένο στο διαδίκτυο χωρίς να νοιαζόμαστε για τυχόν προβλήματα με το υλισμικό ή λογισμικό. Συγκεκριμένα χρησιμοποιήθηκε η υπηρεσία Cyclades με την οποία δημιουργήσαμε ένα VM (Virtual Machine) στο οποίο τρέχουμε τον Ubuntu LTS Server μας. Η υπηρεσίες του Okeanos προσφέρονται δωρεάν στην Ελληνική ακαδημαϊκή κοινότητα. Για άτομα που δεν έχουν πρόσβαση σε τέτοιου είδους υπηρεσία συνιστάται η χρήση IoT application server.

Με την ολοκλήρωση της δημιουργίας του VM μπορούμε να συνδεθούμε σε αυτό και να εγκαταστήσουμε τα προγράμματα που αναφέρθηκαν πιο πάνω. Η σύνδεση γίνεται μέσω της εντολής ssh. Εφόσον συνδεθούμε έχουμε πλέον πρόσβαση στον υπολογιστή μας. Ακολουθώντας το documentation του Node-RED και Mosquitto αντίστοιχα γίνεται η εγκατάστασή τους και μπορούν πλέον να αξιοποιηθούν προς θέλησής μας. Επίσης γίνεται και η εγκατάσταση του Node-RED dashboard το οποίο θα μας βοηθήσει στην πιο ζωντανή απεικόνιση των δεδομένων.

3.5 Arduino IDE

Το Arduino Software IDE μας κάνει εύκολη την συγγραφή του κώδικα και την μεταφόρτωση του κώδικα στο ESP32 NodeMCU. Το ESP32 NodeMCU όπως θα δούμε και παρακάτω είναι ευέλικτο ως προς τις γλώσσες προγραμματισμού που υποστηρίζει. Επιλέγουμε το Arduino IDE κυρίως για την συμβατικότητα με τον μικροεπεξεργαστή ESP32 NodeMCU και για την συμβατικότητα με τις βιβλιοθήκες του Node-Red και του Mosquitto. Επίσης υπάρχει μεγάλο εύρος IoT εφαρμογών που έχουν συγγραφεί σε Arduino επομένως και μεγάλο εύρος πηγών από τις οποίες μπορούμε να συμβουλευτούμε.

Μέσω του IDE μπορούμε να κάνουμε εγκατάσταση βιβλιοθήκες και να τρέξουμε παραδείγματα που μας προσφέρει από μόνο του. Για να γίνει επιτρέπει η μεταφόρτωση κώδικα στον μικροεπεξεργαστή ESP32 NodeMCU θα πρέπει να τον εγκαταστήσουμε στο "Board Manager" του IDE και μετά να επιλέξουμε το μοντέλο μας.



Σχήμα 13: Εγκατάσταση του ESP32 NodeMCU Board στον IDE

4 Εξαρτήματα που χρησιμοποιήθηκαν

4.1 Επιλογή του μικροεπεξεργαστή

Ο μικροεπεξεργαστής είναι η καρδία του IoT project μας. Με αυτόν θα συνδεθούν οι αισθητήρες και με τη σειρά του αυτός θα μας δημοσιεύει τα δεδομένα και θα δέχεται εντολές από το χρήστη μέσω του Node-RED. Η επιλογή του μικροεπεξεργαστή μπορεί να κριθεί από πολλούς παράγοντες. Για να καταλάβουμε ποίοι είναι αυτοί οι παράγοντες πρέπει να μελετήσουμε το πρόβλημα. Παρακάτω θα συγκρίνουμε 3 δημοφιλείς low-budget μικροεπεξεργαστές: Arduino UNO, ESP8266 NodeMCU και ESP32 NodeMCU NodeMCU.

Για αρχή χρειαζόμαστε έναν μικροεπεξεργαστή ο οποίος μπορεί να επικοινωνήσει μέσω ασύρματου δικτύου (WiFi) με τον Broker. Ο πιο δημοφιλής μικροεπεξεργαστής Arduino UNO δεν μας προσφέρει την δυνατότητα αυτή και είναι αναγκαία η προσθήκη ενός WiFi shield για να επιτεύξουμε αυτό το στόχο. Άμα δούμε όμως τον μικροεπεξεργαστή ESP8266 και ESP32 NodeMCU θα παρατηρήσουμε ότι μας προσφέρουν την σύνδεση στο διαδίκτυο. Επίσης σημειώνεται πως οι μικροεπεξεργαστές ESP είναι φθηνότεροι από τον Arduino UNO.

Πέρα από τη σύνδεσή στο διαδίκτυο θα πρέπει όμως ο μικροεπεξεργαστής να υποστηρίζει και τους διάφορους αισθητήρες τους οποίους θέλουμε να χρησιμοποιήσουμε. Η προδιαγραφή που μας ενδιαφέρει περισσότερο είναι αν οι αισθητήρες χρειάζονται αναλογικό η ψηφιακό ώστε να διαβάσουμε σωστά τις μετρήσεις του;. Η κυριότερη διάφορα μεταξύ αναλογικού και ψηφιακού σήματος είναι οτι το αναλογικό σήμα είναι ένα συνεχές κύμα που αλλάζει σε μια χρονική περίοδο ενώ το ψηφιακό είναι ένα διακριτό κύμα που μεταφέρει πληροφορίες σε δυαδική μορφή.

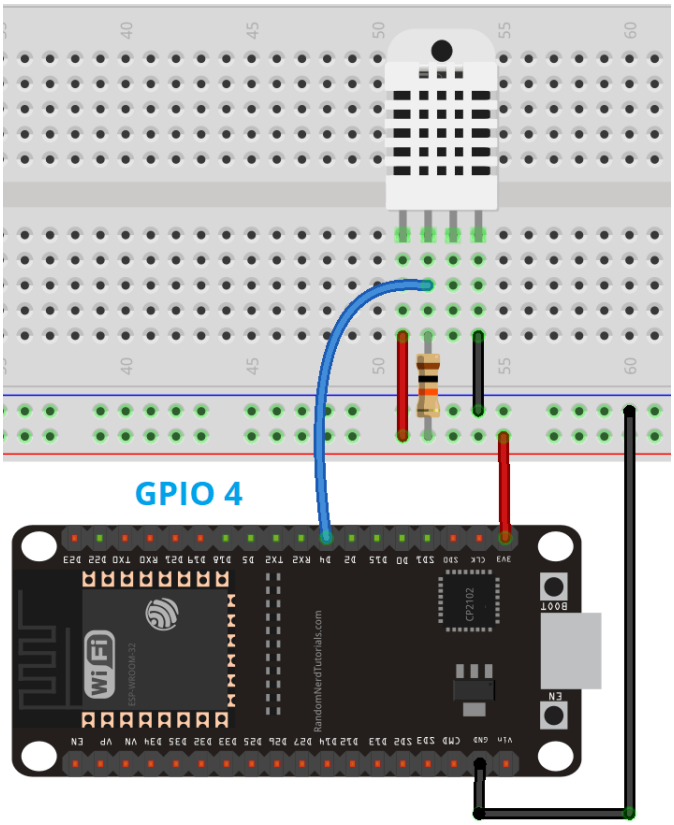
Οι αισθητήρες που έχουμε επιλέξει στέλνουνε κυρίως αναλογικό σήμα. Από τους 3 μικροεπεξεργαστές που αναφέραμε πιο πάνω ο ESP32 NodeMCU φαίνεται να έχει τα περισσότερα αναλογικά Pins με 15. Δυστυχώς ο ESP8266 έχει μόνο ένα αναλογικό Pin και για να γίνει αύξηση των Pins θα πρέπει να γίνει χρήση ενός αναλογικού multiplexer. Τέλος ο Arduino UNO έχει 6 αναλογικά Pins.

Με βάση αυτά τα δυο κριτήρια έγινε η επιλογή του μικροεπεξεργαστή ESP32 NodeMCU NodeMCU ο οποίος μας επιτρέπει την σύνδεση στο διαδίκτυο, υποστηρίζει των αριθμό αναλογικών σημάτων που χρειαζόμαστε και είναι αρκετά οικονομικός. Υπάρχουν όμως και άλλα κριτήρια τα οποία μπορούν να επηρεάσουν την επιλογή μας και εξαρτάται πάντα με το πρότζεκτ που θέλουμε να υλοποιήσουμε.

4.2 Αισθητήρες

4.2.1 DHT 22 αισθητήρας υγρασίας και θερμοκρασίας

Ο DHT22 είναι ένας απλός, οικονομικός ψηφιακός αισθητήρας υγρασίας και θερμοκρασίας. Χρησιμοποιεί έναν χωρητικό αισθητήρα υγρασίας και ένα θερμίστορ για τη μέτρηση του αέρα που βρίσκεται στον περίγυρό του και μας δίνει ψηφιακό σήμα στο data pin του. Θα μπορούμε να δούμε τα δεδομένα του αφού το συνδέσουμε σε ένα αντίστοιχο ψηφιακό Pin του ESP32 NodeMCU. Επίσης θα ήταν καλό να αξιοποιηθεί η χρήση μιας 10κΩ αντίστασης εφόσον θα μας σιγουρέψει την ομαλή λειτουργία του αισθητήρα. Στο παρακάτω σχήμα βλέπουμε το παράδειγμα μιας συνδεσμολογίας του αισθητήρα.



Σχήμα 14: Σχήμα σύνσεσης DHT22

4.2.2 Αισθητήρας υγρασίας χώματος

Ο αισθητήρας υγρασίας χώματος αποτελείται από δύο ανιχνευτές που χρησιμοποιούνται για τη μέτρηση της ογκομετρικής περιεκτικότητας του νερό. Οι δύο ανιχνευτές επιτρέπουν στο ρεύμα να περάσει μέσα από το έδαφος, το οποίο δίνει την τιμή αντίστασης για τη μέτρηση της τιμής υγρασίας.

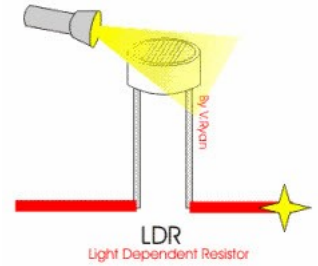
Όταν υπάρχει νερό, το έδαφος θα μεταφέρει περισσότερο ηλεκτρισμό, πράγμα που σημαίνει ότι θα υπάρχει λιγότερη αντίσταση. Το ξηρό έδαφος είναι κακός αγωγός για τον ηλεκτρισμό,

οπότε όταν υπάρχει λιγότερο νερό, τότε το έδαφος θα άγει λιγότερο ηλεκτρισμό, πράγμα που σημαίνει ότι θα υπάρχει μεγαλύτερη αντίσταση. Αυτός ο αισθητήρας έχει την δυνατότητα να συνδεθεί αναλογικά και ψηφιακά, εμείς ωστόσο θα αξιοποιήσουμε το αναλογικό σήμα διότι θέλουμε ακριβές μετρήσεις.

4.2.3 LDR αισθητήρας φωτός

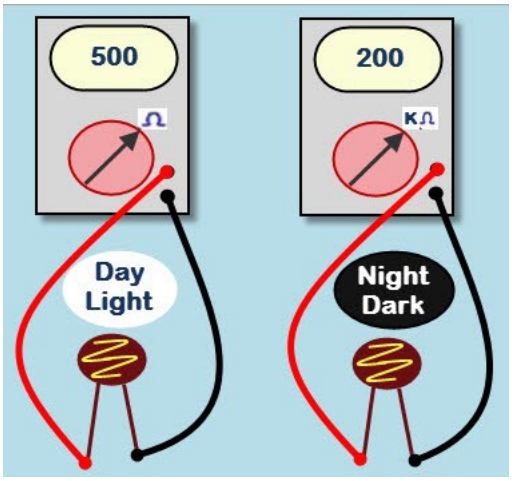
Οι φώτο αντιστάσεις, γνωστές και ως φώτο εξαρτώμενες αντιστάσεις (LDR), είναι συσκευές ευαίσθητες στο φως που χρησιμοποιούνται συχνότερα για να υποδείξουν την παρουσία ή την απουσία φωτός ή να μετρήσουν την ένταση του φωτός.

Αυτή η αντίσταση λειτουργεί με βάση την αρχή της φώτο αγωγιμότητας. Δεν είναι παρά, όταν το φως πέφτει στην επιφάνειά του, τότε η αγωγιμότητα του υλικού μειώνεται και επίσης τα ηλεκτρόνια στη ζώνη σθένους της συσκευής διεγείρονται στη ζώνη αγωγιμότητας. Αυτά τα φωτόνια στο προσπίπτον φως πρέπει να έχουν ενέργεια μεγαλύτερη από το διάκενο ζώνης του υλικού ημιαγωγού. Αυτό κάνει τα ηλεκτρόνια να μεταπηδούν από τη ζώνη σθένους στην αγωγιμότητα. Αυτές οι συσκευές εξαρτώνται από το φως, όταν το φως πέφτει στο LDR τότε η αντίσταση μειώ-



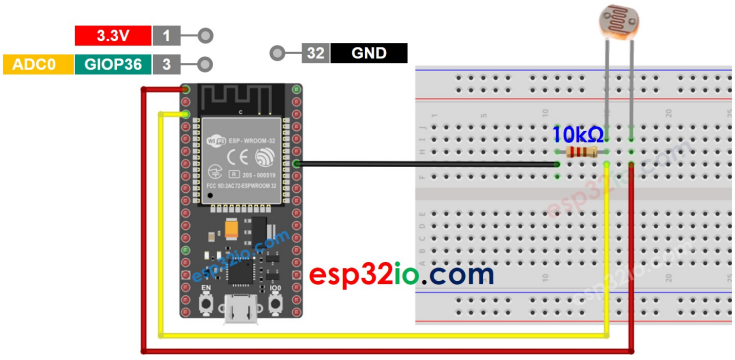
Σχήμα 15: Λειτουργία του LDR

νεται και αυξάνεται στο σκοτάδι. Όταν ένα LDR διατηρείται σε σκοτεινό μέρος, η αντίστασή του είναι υψηλή και, όταν το LDR διατηρείται στο φως, η αντίστασή του θα μειωθεί. Το σήμα που θα



Σχήμα 16: Λειτουργία LDR ύπο διαφορετικές συνθήκες

διαβάσουμε είναι αναλογικό και παρακάτω θα δούμε την συνδεσμολογία του. Θα χρειαστούμε και πάλι όπως με τον αισθητήρα DHT22 μια αντίσταση 10κΩ.



Σχήμα 17: Συνδεσμολογία του αισθητήρα φωτός

4.2.4 Αισθητήρας στάθμης νερού

Ο αισθητήρας έχει μια σειρά από δέκα εκτεθειμένα ίχνη χαλκού, πέντε από τα οποία είναι ίχνη ισχύος και πέντε είναι ίχνη αίσθησης. Αυτά τα ίχνη συμπλέκονται έτσι ώστε να υπάρχει ένα ίχνος αίσθησης ανάμεσα σε κάθε δύο ίχνη ισχύος. Υπάρχει ένα LED τροφοδοσίας στην πλακέτα που θα ανάψει όταν η πλακέτα τροφοδοτηθεί.

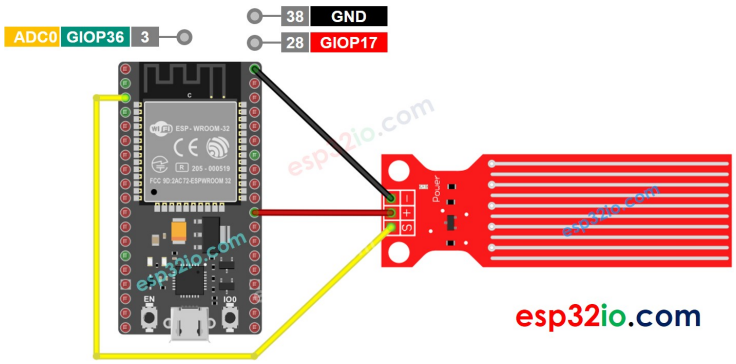
Η λειτουργία του αισθητήρα στάθμης νερού είναι αρκετά απλή. Η σειρά των εκτεθειμένων παράλληλων αγωγών λειτουργούν ως μεταβλητές αντίστασης (ακριβώς όπως ένα ποτενσιόμετρο) της οποίας η αντίσταση ποικίλλει ανάλογα με τη στάθμη του νερού. Συνήθως αυτά τα ίχνη δεν συνδέονται αλλά γεφυρώνονται με νερό όταν βυθίζονται. Η αλλαγή στην αντίσταση αντιστοιχεί στην απόσταση από την κορυφή του αισθητήρα μέχρι την επιφάνεια του νερού.

Η αντίσταση είναι αντιστρόφως ανάλογη με το ύψος του νερού:

- Σε όσο περισσότερο νερό βυθίζεται ο αισθητήρας, οδηγεί σε καλύτερη αγωγιμότητα και θα έχει ως αποτέλεσμα χαμηλότερη αντίσταση.
- Σε όσο λιγότερο νερό βυθίζεται ο αισθητήρας, οδηγεί σε κακή αγωγιμότητα και θα έχει ως αποτέλεσμα μεγαλύτερη αντίσταση.

Ο αισθητήρας παράγει μια τάση εξόδου ανάλογα με την αντίσταση, την οποία μετρώντας μπορούμε να προσδιορίσουμε τη στάθμη του νερού. Το σήμα το οποίο θα διαβάσουμε θα είναι αναλογικό και παρακάτω βλέπουμε τη συνδεσμολογία.

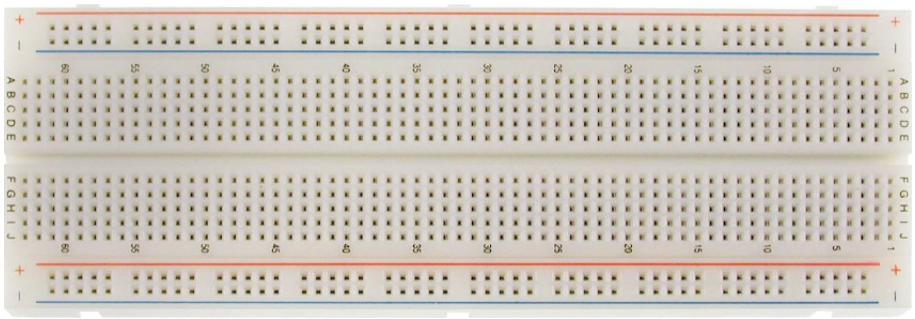
4.3 Άλλα εξαρτήματα



Σχήμα 18: Συνδεσμολογία αισθητήρα στάθμης νερού

4.3.1 Breadboard

Εφόσον μελετήσαμε τους αισθητήρες που θα αξιοποιηθούν θα πρέπει να τους συνδέσουμε με τον μικροεπεξεργαστή ESP32 NodeMCU NodeMCU μαζί με άλλα εξαρτήματα. Αυτή η σύνδεση θα γίνει πάνω σε ένα Breadboard. Το breadboard είναι μια απλή συσκευή που έχει σχεδιαστεί για να μας επιτρέπει να δημιουργούμε κυκλώματα χωρίς να χρειάζεται συγκόλληση. Διατίθενται σε διάφορα μεγέθη και ο σχεδιασμός μπορεί να ποικίλλει, αλλά κατά γενικό κανόνα μοιάζουν κάπως έτσι:



Σχήμα 19: Breadboard

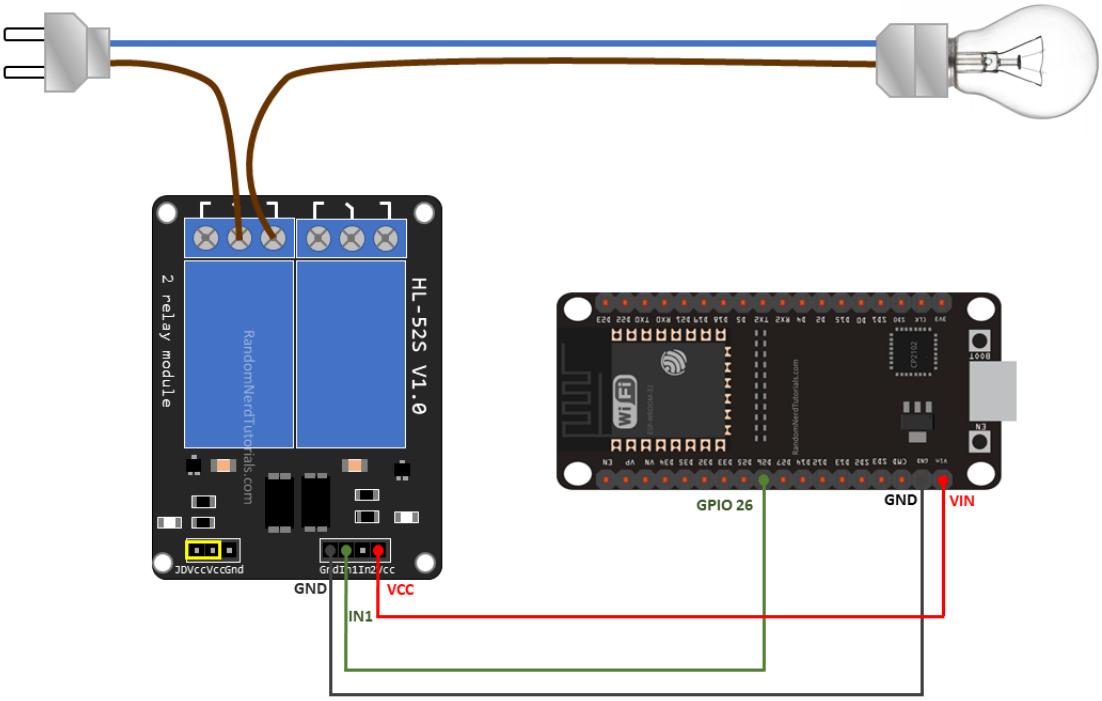
Οι δύο ράγες που βρίσκονται κάτω από κάθε πλευρά χρησιμοποιούνται συνήθως για τη σύνδεση μιας πηγής ρεύματος στην πλακέτα. Συνήθως αναφέρονται ως ράγες ισχύος. Οι άλλες ράγες της πλακέτας χρησιμοποιούνται για εξαρτήματα στο κύκλωμά μας. Κάτω από τις ράγες βρίσκεται σύρμα για την αγωγή του ρεύματος και τη σύνδεση των εξαρτημάτων. Οι ράγες ισχύος μετακινούνται οριζόντια στο Breadboard ενώ οι άλλες ράγες μετακινούνται κάθετα.

4.3.2 Αντλία νερού 3V-6V

Για την διαδικασία του ποτίσματος θα χρησιμοποιηθεί μια μικρή αντλία νερού που τροφοδοτείται στα 3V-6V. Αυτή η αντλία θα τροφοδοτηθεί με μια εξωτερική πηγή ενέργειας και δεν θα μοιράζεται την ίδια πηγή με τον μικροεπεξεργαστή μας.

4.3.3 Ρελέ

Μια μονάδα ρελέ ισχύος είναι ένας ηλεκτρικός διακόπτης που λειτουργεί από έναν ηλεκτρομαγνήτη. Ο ηλεκτρομαγνήτης ενεργοποιείται από ένα ξεχωριστό σήμα χαμηλής ισχύος από έναν μικρο ελεγκτή. Όταν ενεργοποιηθεί, ο ηλεκτρομαγνήτης τραβάει είτε για να ανοίξει είτε να κλείσει ένα ηλεκτρικό κύκλωμα. Στη δικιά μας περίπτωση ο μικρο ελεγκτής είναι το ESP32 NodeMCU. στην άλλη άκρη του ρελέ βρίσκεται η αντλία νερού. Με τη βοήθεια του ρελέ επιλέγουμε πότε να περνάει ρεύμα στο κύκλωμα με αποτέλεσμα να ανοίγει η αντλία. Στη ουσία λειτουργεί ως διακόπτης.



Σχήμα 20: Παράδειγμα συνδεσμολογίας ρελέ με μια λάμπα

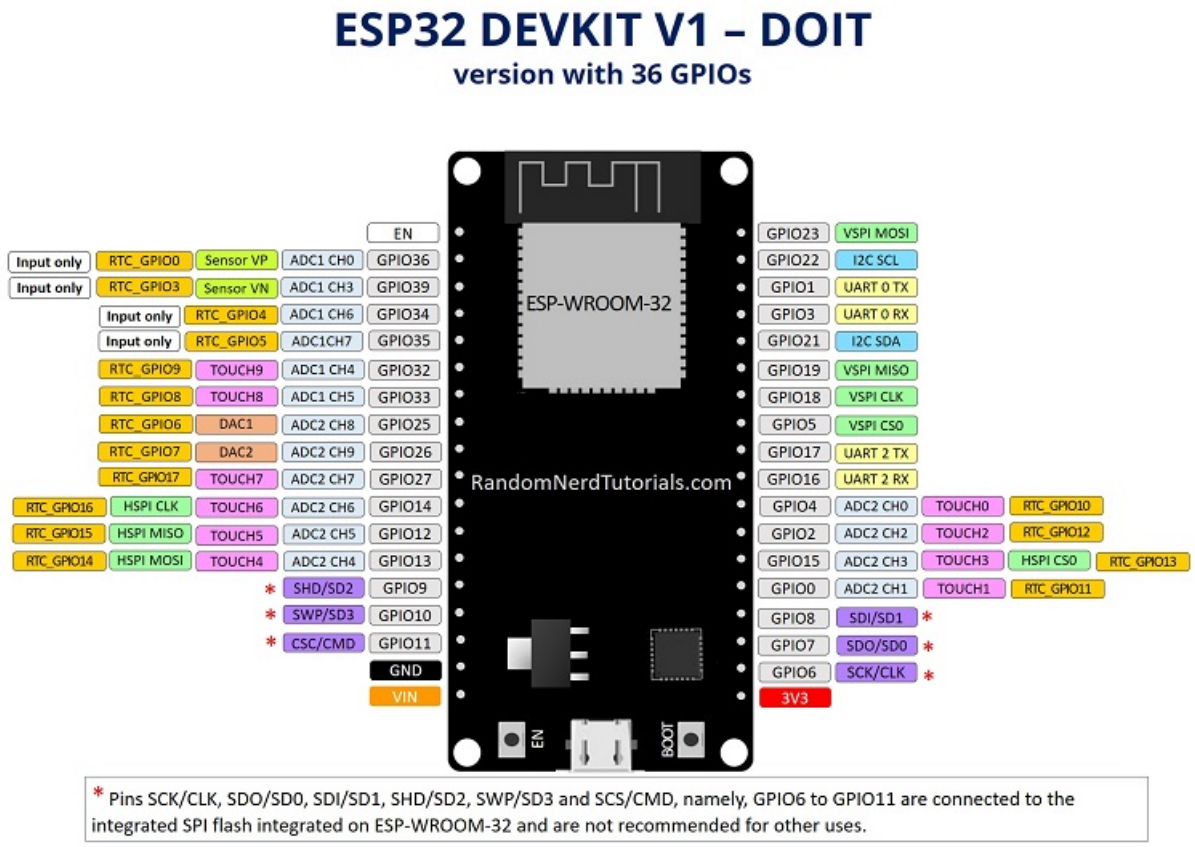
4.3.4 Καλώδια

Τα Καλώδια η αλλιώς jumper wires είναι απλά καλώδια που έχουν ακροδέκτες σύνδεσης σε κάθε άκρο, επιτρέποντάς τους να χρησιμοποιηθούν για τη σύνδεση δύο σημείων μεταξύ τους χωρίς να χρειαστεί συγκόλληση. Με αυτά τα καλώδια θα γίνει η σύνδεση μεταξύ του ESP32 NodeMCU NodeMCU, τους αισθητήρες και τα άλλα εξαρτήματα ώστε να έχουμε επικοινωνία.

5 Κώδικας

5.1 AnalogRead και Pins

Για αρχή θα καθιερώσουμε πως θα επικοινωνούμε με τον μικροεπεξεργαστή και τα Pins του. Στο σχήμα βλέπουμε την ονομασία των Pins και την αρίθμησή τους με την οποία θα τους αναφερόμαστε και στον κώδικα. Τα Pins απο 13 έως 39 μπορούν να αξιοποιηθούν για τα σήματα



Σχήμα 21: Pinouts

που θέλουμε να λάβουμε. Ωστόσο μόνο τα Pins 25 και 26 μας επιτρέπουν την μέτρηση ψηφιακού σήματος. Τα παρακάτω παραδείγματα κώδικα που έχουν γραφτεί σε Arduino θα αναφορτωθούν

Όπως βλέπουμε και στο Σχήμα σύνδεσης παρατηρούμε πως ο αισθητήρας φωτός είναι συνδεδεμένος στο Pin 39 (GPIO 39). Για να διαβάσουμε την τιμή του θα αξιοποιήσουμε την εντολή analogRead.

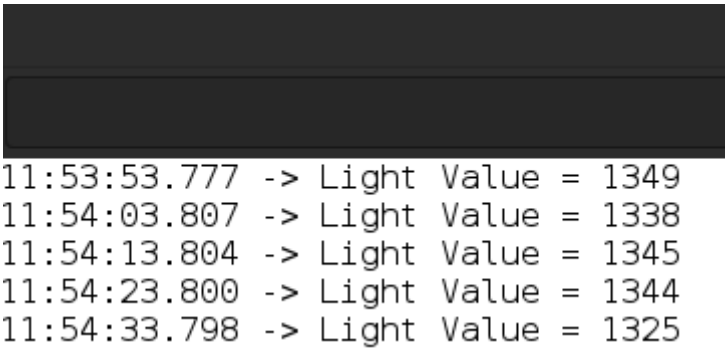
```
1  // GPIO 39 on ESP32 NodeMCU
2  const int light_sensor_pin = 39;
3  int lightValue = analogRead(light_sensor_pin);
```

Πλέον η μεταβλητή lightValue έχει αποθηκευμένη την τιμή που αντιστοιχεί στη μέτρηση του αισθητήρα. Για να την εμφανίσουμε χρησιμοποιούμε την εντολή Serial.print.

```
  Serial.print("Light Value = ");
  Serial.print(lightValue);
```

Για να δούμε εάν δουλεύουν όλα σωστά μπορούμε να αξιοποιήσουμε το Serial monitor του Arduino IDE. Ένα ολοκληρωμένο πρόγραμμα μαζί με τα αποτελέσματα του φαίνεται κάπως έτσι. Η εντολή Serial.println λειτουργεί όπως το Serial.print μόνο που εισάγει μια καινούρια γραμμή στο επόμενο print.

```
const int light_sensor_pin = 39;
int lightValue;

void setup() {
  Serial.begin(115200); // Set baud rate to 115200
}

void loop() {
  lightValue = analogRead(light_sensor_pin); // Read analog Value
  Serial.print("Light Value = ");
  Serial.println(lightValue); // Print light value
  delay(10000);       // Wait 10 seconds before reading again.
}
```



Σχήμα 22: Light sensor values

Η εντολή Serial.begin() ρυθμίζει τον ρυθμό baud για σειριακή επικοινωνία δεδομένων. Ο ρυθμός baud υποδηλώνει τον ρυθμό δεδομένων σε bit ανά δευτερόλεπτο. Ο λόγος που δηλώνουμε 115200 baud είναι επειδή είναι ο προκαθορισμένος ρυθμός του ESP32 NodeMCU. Η εντολή delay() ρυθμίζει την επανάληψη έτσι ώστε να συμβαίνει κάθε 10 δευτερόλεπτα.

Με τον ίδιο τρόπο θα πάρουμε και τις μετρήσεις από τους άλλους αισθητήρες. Σημειώνεται εδώ ότι ο αισθητήρας DHT22 είναι ψηφιακός και θα τον συνδέσουμε με το GPIO 25 (DAC1 - Digital to Analog Converter).

5.2 Κατανόηση των δεδομένων

Αφού λοιπόν καταφέραμε να εμφανίσουμε τα δεδομένα πρέπει να τα κατανοήσουμε κιόλας. Υπάρχουν πολλές πηγές στο διαδίκτυο αλλά και βιβλιοθήκες που μας βοηθάνε στην κατανόηση. Στην περίπτωση του αισθητήρα φωτός με βάση έναν οδηγό από το esp32io.com μπορούμε να κατανοήσουμε τα δεδομένα καλύτερα. Όταν το lightValue < 40 τότε θεωρείτε σκοτάδι, όταν lightValue < 800 το φως θεωρείται αμυδρό, όταν lightValue < 2000 υπάρχει φως, όταν lightValue > 3200 έχουμε αυξημένη φωτεινότητα και σε κάθε άλλη περίπτωση έχουμε πολύ φωτεινότητα.

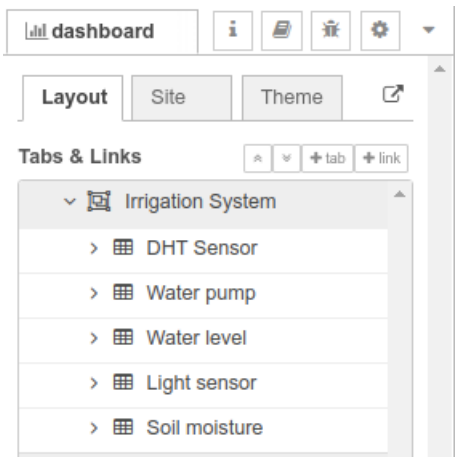
Για τον αισθητήρα DHT22 θα αξιοποιήσουμε την βιβλιοθήκη DHT. Η βιβλιοθήκη μας επιτρέπει την ανάγνωση τον δεδομένων και δεν χρειάζεται κάποια περαιτέρω επεξεργασία των τιμών της θερμοκρασίας και της υγρασίας. Ο τρόπος υλοποίησης βρέθηκε από το randomnerdtutorials.com.

Όταν η μέτρηση του αισθητήρα στάθμης νερού ισούται με 0 τότε ο αισθητήρας είναι στεγνός. Για κάθε άλλη τιμή μέχρι και το 440 θεωρούμε ότι ο αισθητήρας είναι μερικώς βυθισμένος στο νερό και σε κάθε άλλη περίπτωση είναι πλήρως βυθισμένος. Αυτή η μέθοδος βασίστηκε από την ακόλουθη πηγή.

Όσο αφορά τις μετρήσεις του αισθητήρα υγρασίας χώματος θα μετατρέψουμε τα δεδομένα έτσι ώστε να μας τα εμφανίζει σε ποσοστά του 100. Δηλαδή όταν το έδαφος θα είναι στεγνό η ένδειξη θα δείχνει 100%. Η ένδειξη αυτή θα μας φανεί κυρίως χρήσιμη για να ορίσουμε την στιγμή έναρξης του αυτόματου ποτίσματος.

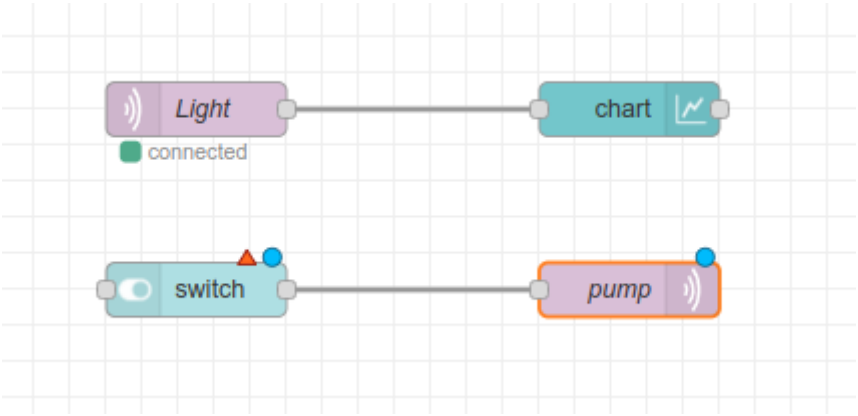
5.3 Δημιουργία επικοινωνίας MQTT με το Node-RED

Το πρώτο βήμα είναι να δημιουργήσουμε τη διάταξη του Dashboard. Στο περιβάλλον του Node-RED έχουμε την επιλογή να φτιάξουμε tabs στην κατηγορία Layout. Εδώ θα δημιουργήσουμε tabs για όλους τους αισθητήρες που έχουμε συνδέσει και θέλουμε να λάβουμε τις μετρήσεις τους. Αρχικά δημιουργούμε μια ενότητα "Irrigation System" και κάτω από αυτήν την ενότητα θα ενσωματώσουμε τους αισθητήρες μας.



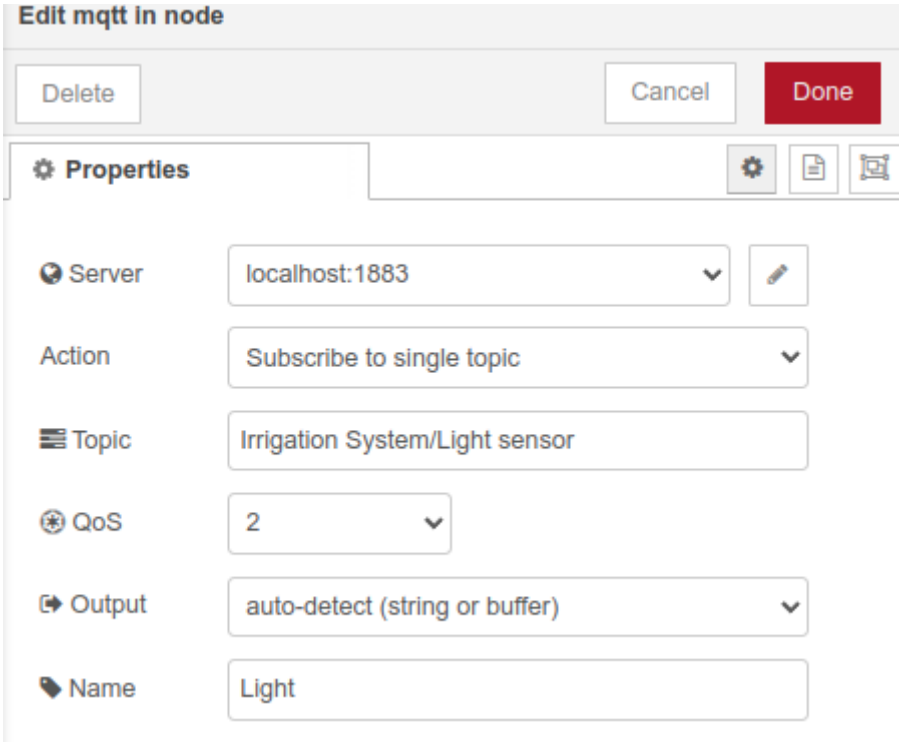
Σχήμα 23: Layout tabs

Στην πορεία θα δημιουργήσουμε flows (ροές) με αποτέλεσμα να λάβουμε το μήνυμα απο τον mqtt broker. Δύο ειδών flows θα δημιουργήσουμε με το Mosquitto, το mqtt in και το mqtt out. Το mqtt in node λειτουργεί σαν λήπτης μηνυμάτων ενώ το mqtt out στέλνει μήνυμα στην IoT συσκευή μας. Άρα το mqtt θα αξιοποιηθεί για όλους τους αισθητήρες και το mqtt out για την αντλία νερού ώστε να δίνουμε την επιλογή στον χρήστη για manual λειτουργία ποτίσματος.



Σχήμα 24: Παράδειγμα του Mqtt in και mqtt out

Το πρώτο flow υποδέχεται ένα μήνυμα από την IoT συσκευή μας και το εμφανίζει σε ένα διάγραμμα το οποίο θα εμφανιστεί στο Dashboard του χρήστη. Συγκεκριμένα λαμβάνουμε την τιμή που μας παρέχει ο LDR αισθητήρας φωτός και την μετατρέπουμε σε αναγνώσιμα δεδομένα. Η διαμόρφωση του mqtt in node που έχουμε ονομάσει "Light" φαίνεται στο παρακάτω σχήμα.



Σχήμα 25: Διαμόρφωση του mqtt in node

Τα δύο σημαντικά πεδία που μας ενδιαφέρουν είναι το πεδίο "Server" και το πεδίο "Topic". Στην επόμενη ενότητα θα δούμε πως μπορούμε να κάνουμε subscribe στο topic που θέλουμε και να

στείλουμε τις μετρήσεις του αισθητήρα και να συνδεθούμε στον mqtt server. Αυτήν την διαδικασία θα την επαναλάβουμε για όλους τους αισθητήρες.

Τέλος η διαμόρφωση του mqtt out node μοιάζει με αυτήν του mqtt in node. Το "switch" στην ουσία είναι ένας διακόπτης που θα στέλνει μήνυμα στον μικροεπεξεργαστή μας για το πότε να ανοίγει την αντλία εάν ο χρήστης έχεις επιλέξει την manual λειτουργία ποτίσματος.

5.4 Δημιουργία επικοινωνίας ESP32 NodeMCU με MQTT

Για να καθιερώσουμε την επικοινωνία με το Mosquitto και το Node-RED θα πρέπει αρχικά να συνδεθούμε στο διαδίκτυο. Θα γίνει χρήση της βιβλιοθήκης WiFi.h του ESP32 NodeMCU NodeMCU.

```
#include <WiFi.h>

// WiFi credentials
const char* ssid = "MyWiFi";
const char* password = "mythesis";

void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password); // Connecting to WiFi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("WiFi connected - ESP IP address: ");
  Serial.println(WiFi.localIP());
}
```

Εφόσον συνδεθούμε στο διαδίκτυο μπορούμε να καθιερώσουμε την σύνδεση με το Mosquitto.

```

// Server IP address
const char* mqtt_server = "myCycladesServer"; // Common server ip
    would be 192.168.1.1

// Initializes the espClient.
WiFiClient espClient;
```

```
PubSubClient client(espClient);

void setup() {
  Serial.begin(115200);
  setup_wifi();    // Connecting to WiFi
  client.setServer(mqtt_server, 1883);    // Establish connection
      with mqtt broker
}
```

Τώρα το μόνο που μας μένει είναι να κάνουμε Publish τα δεδομένα μας στον Broker και να τα λάβουμε στο Node-RED. Στο παρακάτω παράδειγμα θα κάνουμε Publish τα δεδομένα του αισθητήρα φωτός, από το προηγούμενο παράδειγμα, στο ανάλογο topic που έχουμε ορίσει στο Node-RED. Πρέπει να κάνουμε μετατροπή της μεταβλητής lightValue σε const char έτσι ώστε να μπορούμε να κάνουμε Publish στο Node-RED.

```
void loop() {
  static char light[7];
  // Converting float value to string
  dtostrf(lightValue, 6, 2, light);
  // Publishing value to "Irrigation/Light sensor" topic
  client.publish("Irrigation System/Light sensor", light);
  delay(10000);
}
```

Πλέον με την σωστή διαμόρφωση των flows και του κώδικα λαμβάνουμε τις μετρήσεις του αισθητήρα και τις εμφανίζουμε στο Node-Red Dashboard. Η ίδια διαδικασία θα επαναληφθεί και για τους υπόλοιπους αισθητήρες.

5.5 Αυτοματοποίηση ποτίσματος

Η αυτοματοποίηση του ποτίσματος επιτυγχάνεται με βοήθεια του ρελέ και των αισθητήρα υγρασίας χώματος. Στην ουσία όποτε ο αισθητήρας υγρασίας χώματος μας δώσει ένδειξη στεγνού εδάφους θα δίνουμε την εντολή να ανοίγει η αντλία.

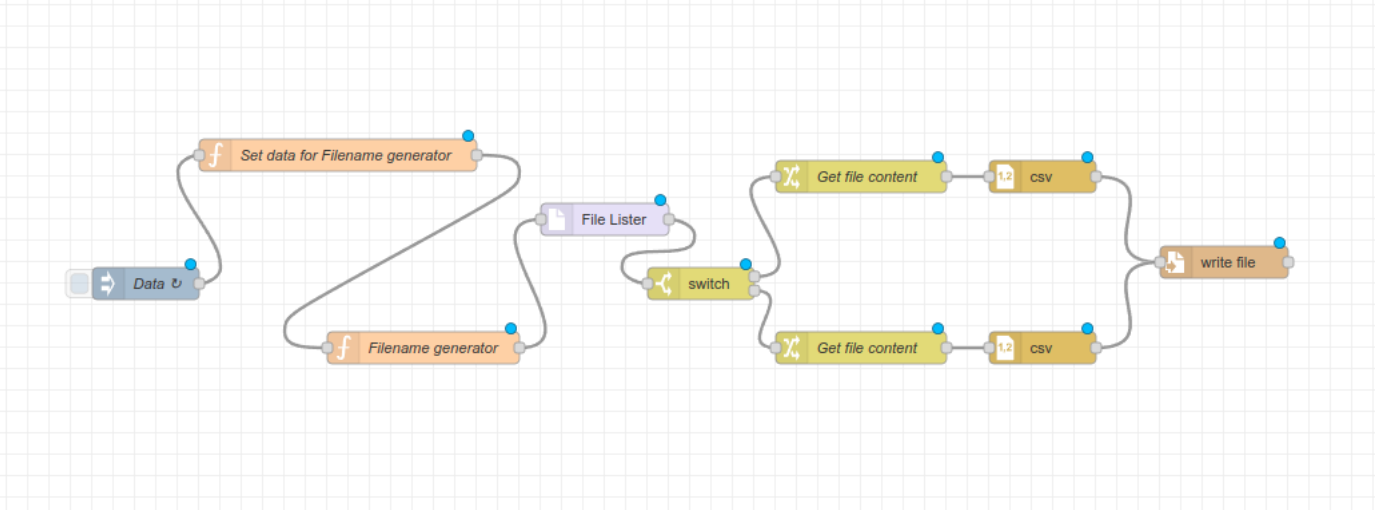
```
const int pump = 25;     // Relay pin connected to pump
int dry = 4000;          // Dry value of our soil moisture sensor

void loop(){
    if (moisture_value > dry){
      digitalWrite(pump, HIGH); // Open water pump for 3 seconds
      delay(3000);
      digitalWrite(pump, LOW);
    }else{
```

```
10        digitalWrite(pump, LOW);  // Close water pump
11      }
12  }
```

5.6 Filename Generator

Πέρα από τα mqtt in και mqtt out το Node-RED μας προσφέρει μια πληθώρα από nodes. Παρακάτω θα εστιάσουμε στα function και storage nodes τα οποία θα μας βοηθήσουν να υλοποιήσουμε λογική στο Dashboard μας.

Αφού καθιερώσαμε πως να λαμβάνουμε τις μετρήσεις από τους αισθητήρες ήρθε η ώρα να αξιοποιήσουμε τα δεδομένα. Η προσέγγιση που έγινε σε αυτό το σημείο είναι η εξής: Μέσω του Node-RED δημιουργούμε ένα flow από nodes το οποίο δημιουργεί ένα αρχείο CSV. Αυτό το CSV αρχείο έχει αποθηκευμένο τις μετρήσεις από τους αισθητήρες για κάθε μισή ώρα μιας ημέρας που περνάει. Το όνομα του αρχείου βασίζεται πάνω στην ημερομηνία και μέσα στο flow εξετάζουμε εάν υπάρχει ένα παρόμοιο αρχείο στο directory του σέρβερ στο οποίο του έχουμε πει να κοιτάξει. Εάν υπάρχει τότε απλά τις καινούριες μετρήσεις στο αρχείο, εάν δεν υπάρχει δημιουργεί καινούριο αρχείο. Έτσι εξασφαλίζουμε ένα αρχείο με τις μετρήσεις μιας ημέρας.



Σχήμα 26: Filename Generator flow

Στο "Data" inject node ορίζουμε κάθε πότε θα γίνεται η δημιουργία/καινούρια είσοδος των δεδομένων. Προηγουμένως αναφέραμε ό,τι θα γίνεται μέτρηση κάθε μισή ώρα επομένως ρυθμίζουμε έτσι και το node. Στα δύο επόμενα JavaScript Function nodes ορίζουμε τα δεδομένα που θέλουμε να περιέχει το CSV αρχείο μας.

Set Data Function

```
// Getting global reading from sensors
var temperature = global.get('temperature');
var humidity = global.get('humidity');
var soil = global.get('soil');
var light = global.get('light');
//Parsing message to next Node
msg.payload = {
    "timestamp" : now.getTime(),
    "temperature" : temperature,
    "humidity" : humidity,
    "soil_moisture" : soil,
    "light": light
}
return msg;
```

Filename Generator Function

```
// Get the current time and convert it to text
var now = new Date();
var yyyy = now.getFullYear();
var mm = now.getMonth() < 9 ? "0" + (now.getMonth() + 1) : (now.getMonth() +
    1); // getMonth() is zero-based
var dd  = now.getDate() < 10 ? "0" + now.getDate() : now.getDate();
var hh = now.getHours() < 10 ? "0" + now.getHours() : now.getHours();
var mmm  = now.getMinutes() < 10 ? "0" + now.getMinutes() : now.getMinutes();
var ss  = now.getSeconds() < 10 ? "0" + now.getSeconds() : now.getSeconds();

// Generate out file name pattern
msg.fname = "IrrigationSystem_"+ dd + "-" + mm + "-" + yyyy + ".CSV";
// Full filename with path for the file node later
msg.filename = "/home/user/datalog/"+ msg.fname;

// We save the current payload into a different place on the msg object
msg.filecontent = msg.payload;

// We are passing the file name search pattern to fs node to tell us if the
    file exists or not
msg.payload = {"pattern":msg.fname};

node.status({fill:"red",shape:"ring",text:msg.fname});
return msg;
```

Στην πορεία λέμε στο "File lister" node που να κοιτάξει όταν να δημιουργεί τα αρχεία ή όταν κοιτάει αν υπάρχει ήδη το αρχείο. Επιλέγουμε το /home/user/datalog directory. Ο φάκελος

datalog θα περιέχει τα καθημερινά δεδομένα των μετρήσεών μας. Τέλος γίνεται η δημιουργία του CSV αρχείου.

5.7 File Browser

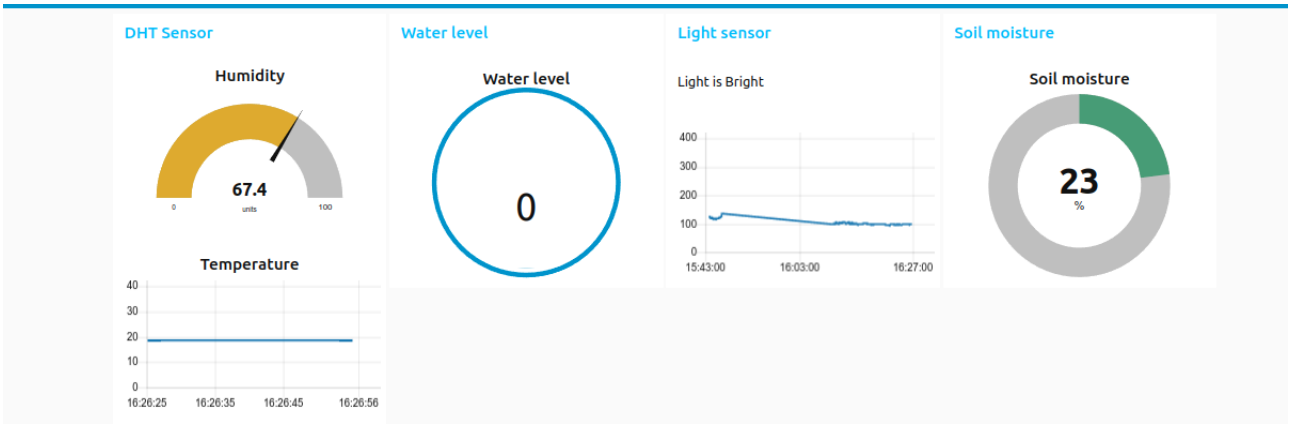
Το τελευταίο στάδιο για να παρουσιάσουμε τα δεδομένα μας είναι η δημιουργία ενός file browser ο οποίος μας επιτρέπει τον οραματισμό των δεδομένων μας. Η δημιουργία του έγινε μέσω της βοήθειας από το git του nygma2004 (2020) ο οποίος έφτιαξε ένα εργαλείο που μας φαίνεται αρκετά χρήσιμο για τον στόχο που θέλουμε να επιτεύξουμε. Ο File browser στην ουσία επιτρέπει στο χρήστη την:

- Εμφάνιση αρχείων σε τοπικούς φακέλους: μέγεθος, ημερομηνίες
- Λήψη αρχείων
- Πλοήγηση σε οποιονδήποτε φάκελο
- Διαγραφή αρχείων
- Εμφάνιση/Απόκρυψη κρυφών αρχείων
- 'Ανοιγμα CSV αρχείων σε μορφή γράφου

Με αυτόν τον τρόπο μπορούμε να προηγηθούμε στο φάκελο datalog και να εμφανίζουμε σε μορφή γράφου τα δεδομένα μας. Έτσι ολοκληρώνεται η διαδικασία οραματισμού των δεδομένων.

6 Αποτελέσματα

Τα αποτελέσματα δείχνουν την τελική μορφή του Dashboard. Ο χρήστης έχει την επιλογή να παρατηρεί τις μετρήσεις των αισθητήρων σε πραγματικό χρόνο. Αυτές οι μετρήσεις γίνονται κάθε 10 δευτερόλεπτα. Η ένδειξη Humidity μας δείχνει τις μετρήσεις του αισθητήρα DHT22 για την υγρασία. Από κάτω ο ίδιος αισθητήρας μας δείχνει την ένδειξη της θερμοκρασίας. Ο αισθητήρας στάθμης νερού αντιπροσωπεύεται από την ένδειξη Water level, στο παράδειγμα που παρατηρούμε η ένδειξη δείχνει 0 επομένως πρέπει να γεμίσουμε με νερό το δοχείο στο οποίο βρίσκεται η αντλία. Ο αισθητήρας φωτός μας δείχνει την ένδειξη του φωτός με κατάλληλο μήνυμα που ορίζουμε στο Node-RED με ένα JavaScript Function Node. Τέλος η υγρασία χώματος φαίνεται από την ένδειξη Soil Moisture.



Σχήμα 27: Επισκόπηση μετρήσεων αισθητήρων σε πραγματικό χρόνο

Πέρα από αυτό ο χρήστης μπορεί να αξιοποίηση το File browser ώστε να μελετήσει τις μετρήσεις των προηγούμενων ημερών.

Σχήμα 28: Γράφος δεδομένων μιας ημέρας

7 Κοιτώντας μπροστά

Η εργασία αυτή είχε ως σκοπό την δημιουργία ενός IoT αυτόματου ποτιστικού συστήματος και την καταγραφή των δεδομένων του. Κοιτώντας μπροστά για το πως θα μπορούσαμε να αξιοποιήσουμε και να επεκτείνουμε τις λειτουργικότητάς του συστήματος εμφανίζονται πολλαπλοί τρόποι.

Για αρχή θα μπορούσαμε να αυξήσουμε των αριθμό των IoT συσκευών. Έτσι θα μεγάλωνε και η κλίμακα μιας φάρμας που θα μπορούσαμε να εφαρμόσουμε το σύστημα. Με παραπάνω αισθητήρες και μάλιστα διαφορετικούς. Ένας αισθητήρας ο οποίος δεν βρήκε εφαρμογή στην εργασία είναι ο αισθητήρας NPK. Ο συγκεκριμένος αισθητήρας μας δίνει την δυνατότητα να εξετάσουμε διάφορα χημικά στοιχεία του εδάφους και να κρίνουμε από αυτά την αποδοτικότητα του εδάφους σε σύγκριση της καλλιέργειας που θέλουμε εφαρμόσουμε. Ο κυρίως λόγος για τον οποίο δεν βρήκε εφαρμογή είναι το κόστος του.

Ένα άλλο στοιχείο το οποίο μπορεί να επεκταθεί είναι η συλλογή των δεδομένων και η αξιοποίησή τους. Με τεχνικές Data mining και σύγκριση των δεδομένων με άλλες βάσεις δεδομένων μπορεί να βελτιωθεί ο τρόπος με τον οποίο εφαρμόζεται η γεωργία με αποτέλεσμα η πιο αποδοτική καλλιέργεια και εξοικονόμηση πόρων.

Οι εφαρμογές των IoT είναι πολλές και ένα μεγάλο πλεονέκτημά τους είναι το κόστος των συσκευών και η προσιτότητά τους. Μπορούν να βελτιώσουν την ποιότητα ζωής απλών ανθρώπων και την περιβαλλοντική ευθύνη που φέρουμε όλοι μας.

Αναφορές

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey.

Capello, F., Toja, M., and Trapani, N. (2016). A real-time monitoring service based on industrial internet of things to manage agrifood logistics.

Fang, S., Xu, L. D., Zhu, Y., Ahati, J., Pei, H., Yan, J., and Liu, Z. (2014). An integrated system for regional environmental monitoring and management based on internet of things.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions.

Hashim, N., Mazlan, S., Aziz, M., Salleh, A., Jaafar, A., and Mohamad, N. (2014). Agriculture monitoring system: a study.

Hsueh, C.-F. and Chang, M.-S. (2010). A model for intelligent transportation of perishable products.

III, C. W. and Cheong, T. (2012). In-transit perishable product inspection.

Kamilaris, A., Kartakoullis, A., and Prenafeta-Bold, F. (2017). A review on the practice of big data analysis in agriculture.

Kodali, R. K., Rawat, N., and Boppana, L. (2014). Wsn sensors for precision agriculture.

Lee, K. (2015). The internet of things (iot): Applications, investments, and challenges for enterprises.

Li, M., Chen, G., and Zhu, Z. (2013). Information service system of agriculture iot.

Medela, A., Cendón, B., González, L., Crespo, R., and Nevares, I. (2013). Iot multiplatform networking to monitor and control wineries and vineyards.

Muangprathub, J., Boonnam, N., Kajornkasirat, S., Lekbangpong, N., Wanichsombat, A., and Nillaor, P. (2019). Iot and agriculture data analysis for smart farm.

nygma2004 (2020). File browser in dashboard: https://gist.github.com/nygma2004.

Ojha, T., Misra, S., and Raghuwanshi, N. (2015). Wireless sensor networks for agriculture: the state-of-the-art in practice and future challenges.

Pahuja, R., Verma, H., and Uddin, M. (2013). A wireless sensor network for greenhouse climate control.

Ruan, J. and Shi, Y. (2016). Monitoring and assessing fruit freshness in iot-based e-commerce delivery using scenario analysis and interval number approaches.

Talavera, J., Tobón, L., Gómez, J., Culman, M., Aranda, J., Parra, D., and Garreta, L. (2017). Review of iot applications in agro-industrial and environmental fields.

Tripathy, A., Adinarayana, J., Vijayalakshmi, K., Merchant, S., Desai, U., Ninomiya, S., and Kiura, T. (2014). Knowledge discovery and leaf spot dynamics of groundnut crop through wireless sensor network and data mining techniques.

Xian, K. (2017). Internet of things online monitoring system based on cloud computing.