A LONG RANGE IOT COMMUNICATIONS NETWORK FOR WATER DISTRIBUTION MANAGEMENT IN RURAL AREAS

by

LAMPROS FRANTZESKAKIS

Master in Telecommunication & Automation Systems, Department of Electronics Engineering, Hellenic Mediterranean University ,Chania,Crete, 2020

A THESIS

submitted in partial fulfillment of the requirements for the degree of

MASTER OF TELECOMMUNICATION & AUTOMATION SYSTEMS

DEPARTMENT OF ELECTRONICS ENGINEERING

Hellenic
Mediterranean
University

2020

Approved by:

Professor
Emmanouel Antonidakis

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ 2
ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ 3
ΠΙΝΑΚΕΣ 4
ΚΩΔΙΚΑΣ 4
Abstract 6
Περίληψη 7
Τι ειναι το internet of things. 11
Προτάσεις για την λύση του προβλήματος 13
MQTT βασικές αρχές λειτουργιάς 14
MQTT QOS 15
MQTT Authentication - Encryption 17
Προσδιορισμός Αναγκών 19
Γνωριμία με τον AtMega 328 20
Τι είναι το arduino 21
Arduino IDE 23
Γνωριμία με το Esp8266 26
Η σύνθεση της Πλατφόρμας MqMax 28
Λειτουργείς του MqMax 34
Esplink Firmware 36
Προγραμματισμός της MCU 41
Δίκτυο Broker και η κατασκευή του 48
Τι είναι το Node red 52
Εγκατάσταση και παραμετροποίηση Node red 53
Εισαγωγή στο Node-Red. 56
Κατασκευή επαφής στο Node Red , Pump Control 62
Εγκατάσταση και λειτουργία. 69
Το Μέλλον 72
Ευρετήριο 73
Βιβλιογραφία 74
Ιστότοποι 74

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1 : Στην υψομετρική φαίνεται καθαρά ότι δεν μπορεί να υπάρξει ασύρματη ζεύξη μετάξι των σημείων A1 - A2 9
Εικόνα 2 :Προβολή των σημαιών πρόσβασης και Αντλιοστασίων 10
Εικόνα 3 περιγραφή λειτουργίας pub/sub 15
Εικόνα 4 Περιγραφικό μοντέλο που εξηγεί τα διάφορα Mode του qos 17
Εικόνα 5 το module που επιλεγούμε για την πλατφόρμα μας 20
Εικόνα 6 το κόστος είναι ιδιαίτερα χαμηλό ανά μονάδα 7€/5=1.4€ 20
Εικόνα 7 Το module esp8266 εκδοση 07 μαζι με τις εξοδους του 26
Εικόνα 8 Το esp8266-05 και το esp8266-12e εχουν τεραστιες διαφορες ως πρως τις εξοδους 27
Εικόνα 9 Το περιβάλλον του προγράμματος EasyEDA , σχεδιασμός PCB από Schematic 29
Εικόνα 10: Το σχηματικό του MqMax 29
Εικόνα 11 :Το Σχηματικό της μονάδας τροφοδοσίας για AC/DC 6-28V 30
Εικόνα 12: Power Board για το MqMax PCB view 30
Εικόνα 13 Κατασκευή Πλακετών στο Εργαστήριο του ΕΛ.ΜΕ.ΠΑ 32
Εικόνα 14 Συναρμολόγηση πλακετών μετά την εκτύπωση για δόκιμες. 33
Εικόνα 15 Οι πλακέτες όπως έφτασαν από την JLCPCB 33
Εικόνα 16 Τελικό αποτέλεσμα , η γέννηση του MQmax 0.4 34
Εικόνα 17 Προγραμματισμός για το MqMax 35
Εικόνα 18 Καλώδιο για τον πρώτο προγραμματισμό της πλατφόρμας μετά τον προγραμματισμό μπορούμε να αλλάζουμε το firmware του μικροελεγκτή ασύρματα 36
Εικόνα 19 Η σελίδα το Esplink στο gitHub 37
Εικόνα 20 Προβολή του MqMax στα ασύρματα δίκτυα 39
Εικόνα 21 Η κεντρική Σελίδα από το Esplink firmware στην πλατφόρμα μας MqMax 39
Εικόνα 22 Συνδεση στο ασηρματο δικτυο 40
Εικόνα 23 Αλγόριθμος Λειτουργιάς εκκίνησης. 40
Εικόνα 24 Ρύθμιση του broker 41
Εικόνα 25 Το interface του NoIP 49
Εικόνα 26 Υπολογιστική πλατφόρμα Raspberry pi 3 στην όποια εγκαταστήσαμε τις υπηρεσίες μας 50
Εικόνα 27 παράγωγη κωδικού για το interface του node red 54
Εικόνα 28 Οθανή εισόδου στο node - red 55
Εικόνα 29 Η διαπάλη προγραμματισμού του Node-Red 56
Εικόνα 30 Προβολή σύνδεσης δυο κόμβων για μεταφορά δεδομένων 57
Εικόνα 31 Μέθοδος πολλαπλών εισόδων σε κόμβο. 58
Εικόνα 32 Function Node 58
Εικόνα 33 Το Mqtt node 60
Εικόνα 34 το switch node με 3 επιλογές, μπορούμε να προσθέσουμε όσες θέλουμε 60
Εικόνα 35 Οι επιλογές node για το dashboard 61
Εικόνα 36 Dashboard για την εφαρμογή μας με προστασία των λειτουργιών με κωδικό 61
Εικόνα 37 ολόκληρο το σχέδιο ροής για την λειτουργία και τον έλεγχο των αντλιών..... 63

Εικόνα 38 Διαχωρισμός λειτουργιών ανάλογα με το topic και το payload 63
Εικόνα 39 Γράφημα διαιρετή τάσης και η τιμή της μπαταρίας όπως παρουσιάζεται. 64
Εικόνα 40 Για κάθε έξοδο έχουμε και ένα μήνυμα το οποίο μας μεταφέρει την κατάσταση 65
Εικόνα 41 Έλεγχος λειτουργίας δικτύου. 65
Εικόνα 42 το δικτύωμα ροής για τις εντολές προς τις αντλίες 66
Εικόνα 43 Το Δικτύωμα ροής που λύνει το πρόβλημα της πιστοποίησης χρηστή. 67
Εικόνα 44 Δόκιμη για το σενάριο λειτουργίας με διακόπτες 69
Εικόνα 45 Αυτονομη μοναδα δεξαμενης 2 69
Εικόνα 46 Κατά την εγκατάσταση και σύνδεση των αισθητήρων στάθμης στην δεξαμενή 2 70
Εικόνα 47 Εγκατάσταση στην δεξαμενή 1 και στο πηγάδι 71

ΠΙΝΑΚΕΣ

Πίνακας 1 BOM List για την πλατφορμα και το DC Τροφοδοτικο της 31
Πίνακας 2 Πρωτόκολλο μηνυμάτων 48

ΚΩΔΙΚΑΣ

Κώδικας 1 Απλος κωδικας σε πλατφορμα Arduino 24
Κώδικας 2 Κωδικας σε AVR C 24
Κώδικας 3 Esp-Link Docker image 37
Κώδικας 4 Docker Compile 37
Κώδικας 5 Esptools install 38
Κώδικας 6 Παραμετροποίηση esptools για προγραμματισμό Esp07 38
Κώδικας 7 Βιβλιοθήκες για τον MCU 42
Κώδικας 8 Βασικά αντικείμενα για την λειτουργία του Esplink MQTT 42
Κώδικας 9 συγχρονισμού Esplink με Mcu 42
Κώδικας 10 έλεγχου σωστής λειτουργίας της σύνδεσης μας στο δίκτυο 42
Κώδικας 11 Αντικείμενα MQTT Client 43
Κώδικας 12 . επιστροφή True όταν έχουμε συνδεθεί με τον broker 43
Κώδικας 13 Αποστολή τάσης μπαταριάς στο topic /pumps/bat/ δεξαμενής 2 44
Κώδικας 14 Έλεγχος στάθμης δεξαμενής 2 44
Κώδικας 15 Ρουτίνα ελεγχου δεξαμενής 2 45
Κώδικας 16 Έλεγχος της στάθμης για την εκκίνηση τις αντλίας δεξαμενής 1 45
Κώδικας 17 Ενημέρωση εκκίνησης αντλίας πηγαδιού με τα μηνύματα B_start και B_stop 45
Κώδικας 18 State της δεξαμενής 1 46
Κώδικας 19 έλεγχος σε επανάληψη για ύπαρξη δικτύου και κατάσταση στάθμης δεξαμενής 1 46
Κώδικας 20 έλεγχος μηνυμάτων για την εκκίνηση της αντλίας πηγαδιού 47
Κώδικας 21 έλεγχος εισερχομένων μηνυμάτων αν είναι B ο πρώτος χαρακτήρας τότε αφορούν το πηγάδι. Επίσης ελέγχει αν υπάρχει μήνυμα από την δεξαμενή 1 για την κατάσταση της 47
Κώδικας 22 Έλεγχος Λειτουργίας και αποκατάστασης δικτύου 47
Κώδικας 23 Έλεγχος Για μήνυμα Παύσης Λειτουργίας 48

Κώδικας 24 Εγκατάσταση Broker 50
Κώδικας 25 Επεξεργασία Ρυθμίσεων Broker 50
Κώδικας 26 Προσθήκη κωδικού και χρηστή 51
Κώδικας 27 Εγκατάσταση Node.JS 53
Κώδικας 28 Εγκατάσταση Node-Red 53
Κώδικας 29 Ρύθμιση Firewall 53
Κώδικας 30 Προσθήκη του Node- Red στις υπηρεσίες 53
Κώδικας 31 Εξαγωγή Κωδικού για το Node-Red 54
Κώδικας 32 Εγκατάσταση Κωδικού 54
Κώδικας 33 Τύποι Δεδομένων που μεταφέρει το msg 56
Κώδικας 34 Δημιουργία μεταβλητής στο msg 56
Κώδικας 35 Διπλή έξοδος από ένα Node 59
Κώδικας 36 Ασύγχρονη εξαγωγή δεδομένων 59
Κώδικας 37 Έλεγχος στάθμης Μπαταρίας , προβολή τάσης 64
Κώδικας 38 Έλεγχος Απώλειας δικτύου 65
Κώδικας 39 Δημιουργία Έλεγχου Λογαριασμού χρηστή για το DashBoard 67
Κώδικας 40 Εκκίνηση Χρονομέτρησης σύνδεσης και αποκάλυψης των επίλογων 68
Κώδικας 41 Έλεγχος χρονικού περιθωρίου 68

Abstract

Communication between devices is a general problem since the automation began. Wired communications is a solution for transferring data to nodes located in a small area. When distances grow , then the use of wireless communication is necessary.

Wireless communications have some disadvantages. They are weak over very long distances and have trouble with physical obstacles. When a system consists of many different units that are scattered over long distances using sensors and actuators , wireless connection is a good way to communicate. Such a scenario is the transfer of water to rural areas as well as the management for watering and exploitation by residential and craft industries.

So the solution is to use Internet Of Things in order to have low implementation costs, low power consumption, a wide range of power systems for our system depending on location, customized solution for every usage scenario and many other benefits.

In this work, an integrated water transport communications system will be designed, based on MQTT protocol. A command messaging system will need be designed for all information that need to be exchanged. An MQTT Server will need to be implemented. A communications transceiver will also be designed and implemented to be used for each location on the network. A microcontroller based platform will be designed for the needs of this thesis.

Περίληψη

Η εργασία αυτή πραγματεύεται την δημιουργία ενός ολοκληρωμένου Internet of Things συστήματος για την μεταφορά νερού σε απομακρυσμένες περιοχές . Κατά την διάρκεια της εργασίας δημιουργήσαμε διάφορα εργαλεία και εκμεταλλευτήκαμε τις δυνατότητας της κοινότητας ανοικτού λογισμικού για να ολοκληρώσουμε το έργο μας . Όπως θα δούμε και παρακάτω χρησιμοποιήσαμε μόνο ανοικτού κώδικα εργαλεία και αποφύγαμε κάθε χρήση κλειστού κώδικα η κλειστού υλικού πλατφόρμες . Το αποτέλεσμα ήταν να μειώσουμε το κόστος δραματικά και να κατασκευάσουμε ένα σύστημα ολοκληρωμένο και λειτουργικό.

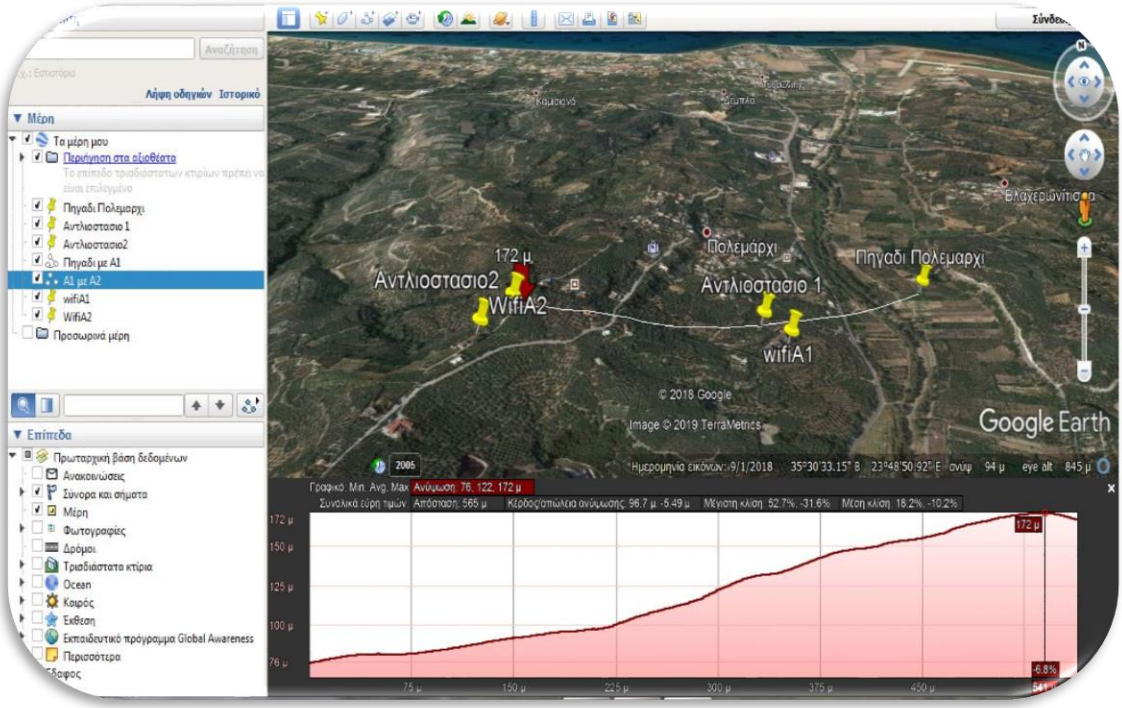
Η ανάγκη για την κατασκευή ενός τέτοιου συστήματος έρχεται από πραγματικές δυσκολίες που αντιμετωπίζει ένας μηχανικός στην κατασκευή αυτοματισμού για ανάλογες περιπτώσεις εφαρμογής . Το πρόβλημα είναι σύνθετο και έχει να κάνει με τα φυσικά εμπόδια που βρίσκουμε σε δύσκολες μορφολογικά περιοχές . Τέτοιες περιοχές είναι όσες έχουν συχνές εναλλαγές από βουνά και πεδιάδες . Συνήθως τα δίκτυα νερού δεν εξυπηρετούν κοντινές περιοχές αλλά μακρινές περιοχές που είναι ορεινές και δυσπρόσιτες από τεχνολογίες όπως το δίκτυο τηλέφωνου/ dsl η ακόμα και από το ασύρματο δίκτυο επικοινωνιών 3G . Το τελευταίο μπορεί να προκαλεί έκπληξη αλλά στην ουσία δεν είναι . Οι εταιρίες κινητής τηλεφωνίας έχουν μελετήσει να έχουν καλύτερη κάλυψη σε αστικές περιοχές που υπάρχει μεγάλη ανάγκη σήματος γιατί θα συνδεθούν πολλοί συνδρομητές, αντί τις περιοχές που δεν υπάρχει καθόλου κινητικότητα.

Η περιοχή υλοποίησης είναι στο Πολεμάρχη του νομού Χανιών Κρήτης. Η περιοχή είναι από τις ποιο δύσκολες που μπορεί να φανταστεί κάποιος από πλευράς επικοινωνίας . Η μορφολογία του εδάφους παρουσιάζει μεγάλες υψομετρικές διαφορές με αποτέλεσμα να μην είναι εύκολο να δομήσουμε ένα δίκτυο . Επίσης δίκτυο μέσω γραμμής δεν είναι διαθέσιμο σε όλες τιε περιοχές . Όπου υπάρχει όμως είναι αρκετά καλό μετά την αναβάθμιση που έγινε το 2016 σε Rural Optic Network, με μεγάλη διαθεσιμότητα σε γραμμές VDSL 50/30 (Rx/Tx). Το πρόβλημα είναι ότι αυτό το δίκτυο είναι διαθέσιμο στα χωριά της περιοχής , όπως και το πολεμάρχη , αλλά δεν είναι διαθέσιμο σε ακτίνα μεγαλύτερη από 500 μετρά από το κέντρο

αναμετάδοσης. Αυτό δημιουργεί μεγάλα κενά τα όποια πρέπει να καλυφτούν με κάποιο τρόπο . Ένας αποδοτικός και οικονομικός τρόπος είναι η ασύρματη απομακρυσμένη σύνδεση μέσω WIFI . Το ότι οι περιοχές αυτές δεν είναι πυκνοκατοικημένες είναι πολύ θετικό γιατί σημαίνει ότι υπάρχουν και κενά κανάλια ασύρματης επικοινωνίας σε όλες τις συχνότητες . Αυτό μας δίνει την λύση που ζητάμε για εύκολη και αξιόπιστη επικοινωνία .

Ας δούμε όμως και την μορφολογία του εδάφους ώστε να αποφασίσουμε με ποια μέθοδο μπορούμε να μεταφέρουμε τα δεδομένα και σε ποιους κόμβους μπορούμε να συνδεθούμε ώστε να κάνουμε ένα η πολλά δίκτυα επικοινωνίας . Ένα βασικό χαρακτηριστικό που πρέπει να καλύπτει το δίκτυο είναι να μας παρέχεται δωρεάν ώστε να μειώσουμε το κόστος λειτουργιάς της άντλησης νερού. Με ένα GIS σύστημα θα κάνουμε μελέτη ώστε να δούμε πως μπορούμε να συνδέσουμε τους κόμβους μας στο δίκτυο και να έχουμε σταθερή σύνδεση για την μεταφορά δεδομένων (Εικόνα 1). Πάντα πρέπει να έχουμε κατά νου και τον κανονισμό της EIRT που αφορά τα ασύρματα δίκτυα .Έχουμε την άδεια να έχουμε σημεία πρόσβασης με μεγίστη ισχύει τα 100mW και για συνδέσεις point to point 29dbi που αντιστοιχεί περίπου στα 800mW.

Για την μελέτη μπορούμε να χρησιμοποιήσουμε το Google Earth που μας παρέχει και το εργαλείο υψομετρικών. Σημαδεύουμε την περιοχή στην όποια βρίσκονται τα αντλιοστάσια και η δεξαμενή. Από εκεί μπορούμε να δούμε εμπειρικά την γύρο περιοχή και βάση γνώσης και εμπειρίας, του περιβάλλοντα χώρου να αποφασίσουμε τα σημεία που μπορούμε να έχουμε πρόσβαση .



Εικόνα 1 : Στην υψομετρική φαίνεται καθαρά ότι δεν μπορεί να υπάρξει ασύρματη ζεύξη μετάξι των σημείων A1 - A2

Στην Εικόνα 1 βλέπουμε την απόσταση μεταξύ του αρχικού και του τελικού σημείου να υπολογίζεται στα 541 μετρά σε ευθεία . Η διάφορα η υψομετρικη σε αυτη την αποσταση ειναι 174 μετρα, που σημενει πολυ αποτομη κληση . Βλέπουμε επίσης ότι το A2 Αντλιοστάσιο είναι πίσω από ένα μικρό βουνό της τάξης των 10 μέτρων προσεγγιστικά.

Αυτό σημαίνει ότι δεν μπορεί να λειτουργήσει καμία μέθοδος ασύρματης σύνδεσης απευθείας και πρέπει να περάσουμε σε μια λύση που θα μας δώσει πρόσβαση τμηματικά. Αυτό μπορεί να γίνει μέσω του δικτύου 3G που λειτούργει στην περιοχή αλλά αυτό θα αυξήσει το κόστος την υλοποίησης .Επίσης στην περιοχή του πηγαδιού δεν υπάρχει κάλυψη του 3G δικτύου. Κοντά στην δεξαμενή έχουμε ένα δωρεάν ιδιόκτητο ασύρματο σημείο πρόσβασης WIFI το όποιο μπορεί να εξυπηρετήσει τον σκοπό μας. Επίσης για την δεξαμενή A1 και το πηγάδι μπορούμε να χρησιμοποιήσουμε το ίδιο σημείο πρόσβασης που είναι ακριβώς στην περιοχή του A1 .

Εικόνα 2 :Προβολή των σημαιών πρόσβασης και Αντλιοστασίων

Λαμβάνοντας υπόψη τα αποτελέσματα της παραπάνω μελέτης βλέπουμε ότι έχουμε ένα πρόβλημα που πρέπει να λύσουμε . Το δίκτυο μας πρέπει να έχει πρόσβαση στο WAN καθώς ο μονός τρόπος επικοινωνίας όλων τον κόμβων , είναι το internet . Και έτσι βλέπουμε φανερά μια πρώτη ανάγκη χρήσης της τεχνολογίας IOT η αλλιώς internet of things .

Τι ειναι το internet of things.

Για πολλά χρόνια ,από την εποχή της τεχνολογικής επανάστασης , άρχισε να γεννάτε ένα ερίτιμα . Πως θα μπορούσαμε να ελέγχουμε ένα μηχανισμό από οπουδήποτε στον κόσμο αξιόπιστα και με ταχύτητα. Πως θα μπορούσαμε να κάνουμε αυτοματισμούς σε απομακρυσμένες περιοχές να επικοινωνούν μετάξι τους και να εκτελούν διεργασίες συνεργατικά με η και χωρίς την εποπτεία απίου φυσικού προσώπου. Αυτό είναι και το βασικό πρόβλημα και το μέλλον της εξέλιξης . Πλέον το διαδίκτυο ενώνει τον κόσμο με υψηλές ταχύτητες επικοινωνίας και με πολλές διαφορετικές επιλογές δικτύωσης. Η ανάπτυξη των ασύρματων επικοινωνιών είναι ένα μεγάλο επίτευγμα για την κοινωνία του σήμερα και προσφέρει την ελευθερία της δικτύωσης αισθητήρων και επενεργών σε σημεία και θέσεις που δεν ήταν εύκολο να γίνει παλαιοτέρα.

Έτσι σήμερα είμαστε στην αρχή μιας νέας μορφής διαδικτύου , ένα διαδίκτυο που δεν είναι άμεσα για τον άνθρωπο αλλά Εμέσα , άμεσα είναι για της συσκευές που μας περιβάλουν . Το λεγόμενο "δίκτυο των πραγμάτων" είναι η μονή λυση για την ανάπτυξη στην επομένη τεχνολογική επανάσταση και τον επαναπροσδιορισμό των απαιτήσεων που έχουν οι σύγχρονες κοινωνίες , για μεγαλύτερη ασφάλεια , καλύτερη υγεία , άμεσο έλεγχο υπηρεσιών , καλύτερη απόδοση στην εκμετάλλευση των φυσικών πόρων.

Επίσης για πρώτη φορά γεννάτε η ιδέα της αυτόνομης διαχείρισης απομακρυσμένων συστημάτων με τα δίκτυα "μηχανή προς μηχανή M2M" που για πρώτη φορά παρουσιάζουν την αποδοτική διαχείριση πόρων και λειτουργιών που είναι χρονοβόρες για τον άνθρωπο , απαιτούν φυσική παρουσία ατόμου σε χορούς επικινδύνους για την υγεία , μιάνουν την ελευθερία . Όταν οι συσκευές μπορούν να επικοινωνούν, έχουν και την δυνατότητα να παράγουν έργο για τον άνθρωπο, χωρίς αυτός να καταναλώνει τον χρόνο του για να εποπτεύει ένα αυτοματισμό η να λειτούργει σαν συντονιστής .

Έτσι άδω πρέπει να ορίσουμε τι είναι το δίκτυο των πραγμάτων "internet of things". Δεν είναι απλά ένα σύστημα με το όποιο εμείς ενημερωνόμαστε και συντονίζουμε τις εργασίες , γιατί αυτό δεν λύνει το πρόβλημα του χρόνου που είναι πολύτιμο αγαθό στην ζωή του ανθρώπου . Δεν λύνουμε το πρόβλημα της απόδοσης, καθώς σαν άνθρωποι κάνουμε λάθη και είμαστε λιγότερο συντονισμένοι σε ένα

σύστημα από ότι είναι τα ενεργά του στοιχειά . Ένα πραγματικό δίκτυο των πραγμάτων πρέπει να έχει χαρακτηρίστηκα, έλεγχου , αυτόνομης λειτουργιάς και επικοινωνίας , συντήρησης ,ασφαλούς επαναφοράς από σφάλματα , συντονισμού, εκπαίδευσης, και νοημοσύνης, ώστε να εκτελεί διεργασίες για τον άνθρωπο χωρίς αυτός απαραίτητα να χρειάζεται να παρέμβει. Απλά να επιλεγεί το κατάλληλο σενάριο βάση των αναγκών που έχουν οριστεί από τον άνθρωπο και αυτό αυτόματα να κανονίζει να γίνει σωστά η διεργασία ,να ενημερώνει μόνο όταν δεν μπορεί να εκτελέσει αποδοτικά το έργο του ώστε ο άνθρωπος να θεραπεύει το πρόβλημα και να επαναφέρει την λειτούργει του συστήματος.

Οι περιπτώσεις χρήσης για το δίκτυο των πραγμάτων είναι άπειρες και περιλαμβάνουν από πραγματικά έξυπνα σπίτια , δίκτυα επικοινωνιών , δίκτυα νερού, έξυπνα αυτοκίνητα, συσκευές διάγνωσης πραγματικού χρόνου, διαχείριση ενεργείας, αγροτική παράγωγη και αλλά πολλά σενάρια .

Αυτό θα έχει μεγάλο αντίκτυπο στην ποιότητα της ζωής των ανθρώπων και στην διαχείριση των πόρων . Μέχρι και σήμερα τέτοια συστήματα δεν είναι ακόμα ολοκληρωμένα και δεν αποτελούν πραγματικότητα για τον μέσο χρηστή . Δεν έχουμε την δυνατότητα να δώσουμε τον έλεγχο στις μηχανές γιατί υπάρχουν προβλήματα ηθικού και κοινωνικού χαρακτήρα .

Στην παρούσα εργασία μας απασχολεί η μεταφορά νερού σε απομακρυσμένο μέρος. Η άντληση του από την γεώτρηση μέχρι την μεταφορά του στο σημείο προορισμού είναι μια διεργασία που απαιτεί ενεργεία και χειρισμούς ώστε να μην υπάρχει σπάταλη νερού και ενεργείας . Επίσης η ενημέρωση για την ορθή λειτουργία του δικτύου μας είναι σημαντική μιας που προσφέρει ηρεμία και ελεύθερο χρόνο για τον χρηστή.

Ανατρέχοντας στην βιβλιογραφία και στο διαδίκτυο παροιμία προβλήματα πάντα έκαναν την εμφάνιση τους σε διαφορές περιπτώσεις και περισσότερο σε παρόμοιες εφαρμογές άντλησης είτε νερού είτε πετρελαίου η υγραερίου. Πάντα όταν υπάρχει δίκτυο μεταφοράς που ξεπερνά τα 400 μετρά τότε αρχίζει να υπάρχει η ανάγκη για την επικοινωνία των αυτοματισμών που ελέγχουν την μεταφορά. Παλιά η λύση ήταν ιδιαίτερα δύσκολη και απαιτούσε την εγκατάσταση ολόκληρων υπολογιστικών μονάδων για τον απομακρυσμένο έλεγχο των αυτοματισμών . Αυτό φυσικά είναι ακριβό και από πλευράς χρημάτων και από πλευράς ενεργείας , αποτελώντας απαγορευτική την τεχνολογία αυτή από τους απλούς αγρότες και πολίτες. Σήμερα όμως με την εξέλιξη της τεχνολογίας και την άνοδο των

μικροϋπολογιστών στο προσκήνιο αυτό το σενάριο αρχίζει να γίνεται όλο και ποιο ελκυστικό , σε βαθμό τέτοιο που στο άμεσο μέλλον όλες οι συσκευές θα είναι διασυνδεδεμένες. Το κόστος της επικοινωνίας μέσω διαδικτύου και η εξέλιξη της λιθογραφίας μας επιτρέπει να έχουμε αυτή την τεχνολογία σε πολύ μικρό κόστος και οικονομικό και ενεργειακό.

Στην συνεχεία θα αναλύσουμε την λύση που προτείνουμε , πως θα υλοποιηθεί και γιατί μας εξυπηρετεί. Γιατί είναι καλύτερη από άλλες που υπάρχουν στην αγορά και γιατί επιλεγούμε να προχωρήσουμε στην υλοποίηση δίκης μας πλατφόρμας αντί να επιλέξουμε μια έτοιμη λύση από την αγορά .

Προτάσεις για την λύση του προβλήματος

Όπως αναλύσαμε στο προηγούμενο κεφάλαιο μετά από την μελέτη που κάναμε , αποφανθήκαμε ότι η ιδανική λύση είναι το WIFI για την δικτύωση και για την υλοποίηση του αυτοματισμού , να αναπτύξουμε μια νέα πλατφόρμα που θα υποστηρίζει WIFI σύνδεση και το πρωτόκολλο MQTT . Ποιοι λόγοι μας οδήγησαν σε αυτές τις επιλογές .

Το Wifi 802.11.xx ειναι ένα ώριμο τεχνολογικά πρωτόκολλο δικτύωσης . Έχει τεραστία υποστήριξη και έχει αποδεδειγμένη αξιοπιστία λειτουργιάς . Αρκετοί κομβόι που παρέχουν WIFI λειτουργούν για χρόνια χωρίς προβλήματα και απώλεια σήματος χωρίς καμία συντήρηση. Είναι ένας αρκετά φτηνός τρόπος να μεταφέρουμε δίκτυο όπου θέλουμε χωρίς καλώδια . Έχει πολύ άψιλο bandwith που ξεκινά από το 0.5Mbit και μπορεί να φτάσει και τα 300Mbit . Άλλες τεχνολογίες ασύρματης δικτύωσης όπως το LORA που είναι σχετικά λίγα χρόνια στο χώρο , δεν μπορούν να προσφέρουν αυτό το bandwith αλλά έχουν αλλά πολύ καλά χαρακτηριστικά που θα μπορούσαν να είναι αρκετά ανταγωνιστικά . Ο Λόγος που το LORA δεν είναι η τελική μας επιλογή , παρότι στην αρχή ελπίζαμε να είναι , είναι το μεγάλο κόστος που έχουν οι μονάδες πρόσβασης στο δίκτυο που υποστηρίζουν LORA , το λεγόμενο LORAWAN . Στο σενάριο μας θα χρειαστούμε 2 κόμβους πρόσβασης με αποτέλεσμα να είναι ιδιαίτερα ακριβό στην υλοποίηση ενώ με την Wifi πρόσβαση οι κομβόι είναι δωρεάν καθώς υπάρχουν ήδη και παρέχουν δίκτυο σε περιοχές που δεν υπάρχει . Στο μέλλον φυσικά υπάρχει περίπτωση να υπάρχουν δίκτυα LORA σε κάθε περιοχή και έτσι να έχουμε ξανά την επιλογή πρόσβασης σε τέτοιο δίκτυο .

MQTT βασικές αρχές λειτουργιάς

Ιστορικά το MQTT εφευρέθηκε από τους Dr.Andy Stanford-Clark της IBM και Arlen Nipper της Arcom (τώρα Eurotech) το 1999. Ο στόχος τους ήταν να δημιουργήσουν ένα πρωτόκολλο για παρακολούθηση σωληνώσεων πετρελαίου μέσω δορυφορικής σύνδεσης, όπως βλέπουμε είναι ένα iot σύστημα στην εποχή του 1999, το οποίο θα είχε ελάχιστη ανάγκη από μπαταρίες και εύρος ζώνης (bandwidth). Ξεκινώντας την σχεδίασή του είχαν τις εξής προδιαγραφές στο μυαλό τους:

- Απλό στην εφαρμογή.
- Να παρέχει Quality of Service (QoS).
- Αποδοτικό ως προς την κατανάλωση ενέργειας και εύρους ζώνης
- Λειτουργικό ανεξαρτήτως δεδομένων (που αποστέλλονται)
- Συνεχής σύνδεση

Η IBM μετά από χρήση του MQTT σε αρκετές δικές τις εφαρμογές έβγαλε το 2010 την έκδοση 3.1 δωρεάν και έτσι ο καθένας μπορεί να το χρησιμοποιήσει και να υλοποιήσει εφαρμογές.

Από τότε αρκετές εταιρίες και οργανισμοί έχουν φτιάξει ποικίλες εφαρμογές στηριζόμενες σε αυτό, άλλες ανοιχτού-κώδικα και άλλες κλειστού-κώδικα, κάποιες κερδοσκοπικά κάποιες όχι.

Ποιο κάτω είναι μερικές από τις μεγαλύτερες υλοποιήσεις βασισμένες στο MQTT πρωτόκολλο:

- Facebook Messenger.
- WarmDirt - Ένα έργο που ελέγχει τη θερμοκρασία στο χώμα.
- homA - Framework για αυτοματισμό σπιτιού .
- Relayr - Μία IoT εταιρία που χρησιμοποιεί το MQTT για να συνδέσει IoT συσκευές.

και πολλές άλλες εφαρμογές το έχουν κάνει σήμερα ένα από τα ποιο χρησιμοποιημένα πρωτοκολλά επικοινωνίας.

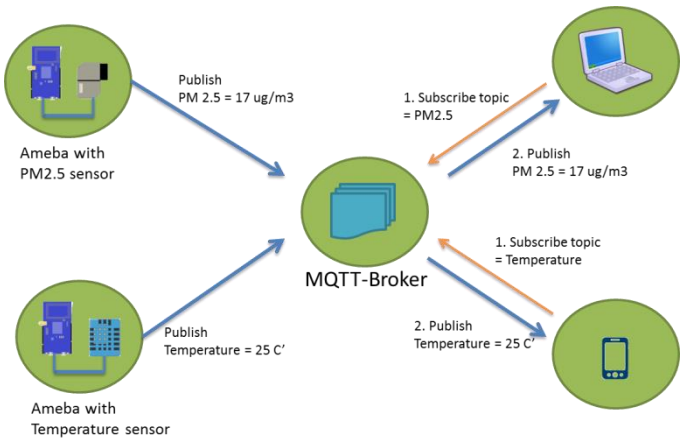
Το publish/subscribe (pub/sub) μοντέλο έρχεται ως εναλλακτική στο παραδοσιακό client-server μοντέλο όπου 2 συσκευές επικοινωνούν απευθείας. Εδώ ο client που στέλνει ένα μήνυμα (pub) δεν ξέρει την ύπαρξη του παραλήπτη- client (sub). Υπάρχει ένα σημιτικό κομμάτι του πάζλ, ο broker, ο οποίος είναι γνωστός τόσο στον αποστολέα (publisher) όσο και στον παραλήπτη (subscriber).

Η δουλειά του broker είναι να λαμβάνει όλα τα εισερχόμενα μηνύματα και να τα προωθεί κατάλληλα ώστε συγκεκριμένα μηνύματα να φτάνουν σε συγκεκριμένους παραλήπτες.

Χωρική αποσύνδεση. Ο publisher και ο subscriber δεν γνωρίζουν ο ένας την ύπαρξη του άλλου.

Χρονική αποσύνδεση. Ο publisher και ο subscriber δεν χρειάζεται να τρέχουν την ίδια χρονική στιγμή.

Αποσύνδεση συγχρονισμού. Όλα τα μέρη (publisher, subscriber, broker) δεν διακόπτουν την λειτουργία τους κατά την αποστολή ή παραλαβή μηνυμάτων



Εικόνα 3 περιγραφή λειτουργίας pub/sub

MQTT QOS

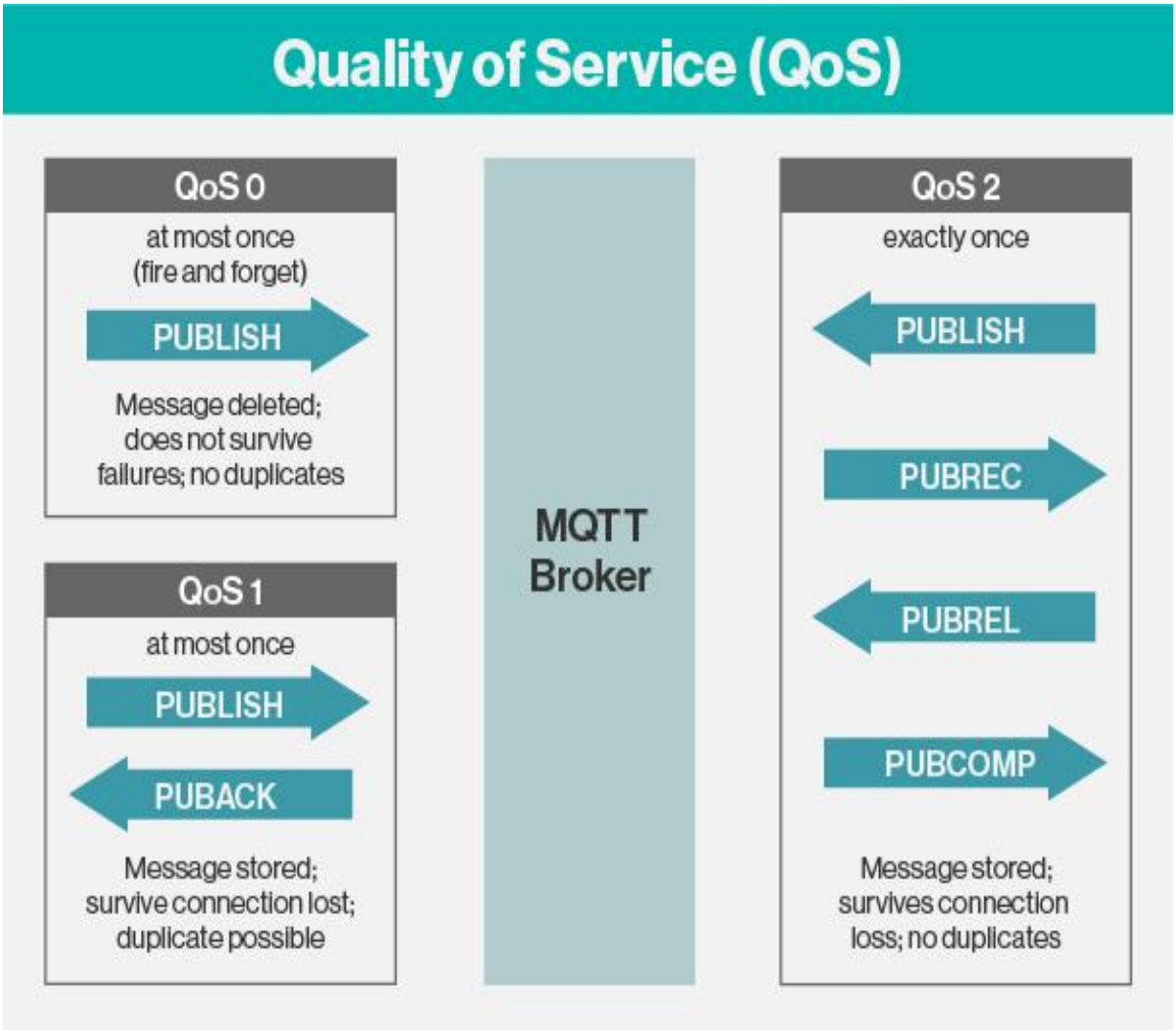
Το επίπεδο QoS είναι μια συμφωνία μεταξύ αποστολέα και παραλήπτη για το πόσο εγγυημένη είναι η παράδοση ενός μηνύματος. Όσο αφορά το QoS η διαδρομή του μηνύματος από τον publisher στο subscriber μπορεί να εξεταστεί από 2 πλευρές. Κατά πρώτον από το Qos που αφορά την αποστολή μηνύματος από τον Publisher στον broker, όπως έχει καθοριστεί από την εντολή publish. Κατά δεύτερο από το Qos

για την αποστολή μηνύματος από τον broker στο subscriber, όπως έχει καθοριστεί από την εντολή subscribe. Συνεπώς, το QoS μπορεί να υποβιβαστεί για κάποιο παραλήπτη ο οποίος έκανε subscribe με μικρότερο QoS. Στο MQTT υπάρχουν 3 QoS επίπεδα:

QoS 0 : Το πολύ μια φορά. Το ελάχιστο επίπεδο QoS, το μήνυμα α- ποστέλλεται μια φορά και δεν λαμβάνεται επιβεβαίωση παράδοσης ούτε αποθηκεύεται. Συνήθως είναι όσο αξιόπιστο είναι και το TCP πρωτόκολ- λο πάνω από το οποίο γίνεται η αποστολή. Αυτό το επίπεδο επιλέγεται όταν υπάρχει σταθερή σύνδεση broker και client (συνήθως ενσύρματη) ή όταν δεν υπάρχει πρόβλημα να χάσουμε λίγη πληροφορία.

QoS 1 : Τουλάχιστον, μια φορά. Ο αποστολέας αποθηκεύει το μήνυμα μετά την αποστολή του μέχρι να πάρει το PUBACK μήνυμα. Αν περάσει συγκεκριμένο χρονικό διάστημα και δεν το λάβει προσπαθεί να αποστεί- λει το μήνυμα ξανά. Με αυτό το τρόπο το μήνυμα μπορεί να φτάσει 24 MQTT περισσότερες φορές. Αυτό το επίπεδο επιλέγεται όταν στόχος είναι να φτάσει σίγουρα η πληροφορία και εναπόκειται στο χρήστη αν θα χειριστεί την επιπρόσθετη πληροφορία (τα αντίγραφα).

QoS 2 : Σίγουρα μια φορά. Είναι ο ασφαλέστερος, αλλά και ο πιο αργός τρόπος να σταλεί ένα μήνυμα. Για να συμβεί αυτό γίνονται 2 ανταλλαγές μηνυμάτων. Αν ο client στείλει δεύτερη φορά ένα μήνυμα, γιατί καθυστέρησε η απάντηση ο broker θα προωθήσει μόνο το ένα. Αυτό το επίπεδο επιλέγεται όταν είναι πολύ σημαντικό για την εφαρμογή να λαμ- βάνονται όλα τα μηνύματα ακριβώς μια φορά και η επιβάρυνση από το διπλό έλεγχο δεν μας είναι τόσο μειονέκτημα.



Εικόνα 4 Περιγραφικό μοντέλο που εξηγεί τα διάφορα Mode του qos

Το QoS είναι σημαντικό χαρακτηριστικό του πρωτοκόλλου MQTT. Με τον έλεγχο της παράδοσης του μηνύματος να είναι στα χέρια του MQTT μπορεί να κάνει την επικοινωνία σε αναξιόπιστα δίκτυα ευκολότερη. Επιπλέον δίνει την ευχέρεια στο χρήστη να διαλέξει το QoS επίπεδο ανάλογα με το είδος της εφαρμογής του.

MQTT Authentication - Encryption

Σε μεγάλα συστήματα επικοινωνίας τα οποία χρησιμοποιούν το MQTT πρωτόκολλο, η εξουσιοδότηση μπορεί να φανεί πολύ χρήσιμη. Η εξουσιοδότηση, που δίνεται στο client αφορά τα δικαιώματα του στα topics και πρέπει να μπορεί να

ρυθμιστεί, ενώ ο broker τρέχει (runtime), για ευνόητους λόγους. Μια εξουσιοδότηση σε ένα topic μπορεί να συμπεριλαμβάνει τα εξής:

- Το ακριβές topic ή κάποιο το οποίο συμπεριλαμβάνει wildcards.
- Δικαίωμα ή όχι για την εκτέλεση των εντολών publish και/ή subscribe.
- Το επίπεδο QoS που έχει δικαίωμα να χρησιμοποιήσει (0,1,2 ή όλα)

Ας πάρουμε ένα σενάριο στο οποίο στέλνουμε ένα γράμμα μέσω ταχυδρομείου. Έχουμε καθορίσει τον παραλήπτη και ο ταχυδρόμος θα φροντίσει να φτάσει στον προορισμό του. Κανείς όμως δεν μας εγγυάται, ότι ο ταχυδρόμος ή όποιος άλλος εμπλέκεται στην μεταφορά δεν θα διαβάσει το γράμμα ή ακόμη χειρότερα να αλλάξει το περιεχόμενο του.

Το ίδιο σενάριο ισχύει και όταν στέλνονται δεδομένα σε απλό κείμενο μέσω ενός δικτύου γενικά και πιο συγκεκριμένα στο διαδίκτυο. Τα TCP πακέτα θα περάσουν από διάφορες υποδομές (routers, firewalls, Internet Exchange Points κλπ) πριν φτάσουν στο προορισμό τους. Το πακέτο με τα δεδομένα μπορεί να διαβαστεί και να τύχει κακόβουλης επεξεργασίας από κάθε σημείο.

Για αποφυγή του πιο πάνω κίνδυνου υπάρχουν τα πρωτόκολλα κρυπτογραφίας TLS (Transport Layer Security) και SSL (Secure Sockets Layer). Τα πρωτόκολλα TLS/SSL χρησιμοποιούν μηχανισμούς χειραψίας ώστε να ρυθμίσουν κατάλληλες παραμέτρους για να δημιουργήσουν μια ασφαλή σύνδεση ανάμεσα σε client και server. Σε αυτή την σύνδεση πλέον τα δεδομένα μεταφέρονται κρυπτογραφημένα και με την παρούσα τεχνολογία θεωρείται αδύνατο να διαβαστούν.

Η πιστοποίηση και εξουσιοδότηση είναι πρακτικές ασφαλείας σε επίπεδο εφαρμογής. Σε επίπεδο μεταφοράς δεδομένων στο MQTT υπάρχει η δυνατότητα TLS σύνδεσης αντί απλής TCP, στη θύρα 8883. Σαφώς και αυτή η σύνδεση επιφέρει επιβάρυνση σε υπολογιστικούς πόρους, αλλά συστήνεται ειδικά όταν γίνεται χρήση username και password.

Προσδιορισμός Αναγκών

Για να αξιοποιήσουμε τις τεχνολογίες του MQTT και του WIFI πρέπει να βρούμε κάποια πλατφόρμα που να τα υποστηρίζει . Μετά από αναζήτηση βρήκαμε διαφορές πλατφόρμες όπως Sonoff , Micro:bit και άλλες πολλές . Καμία δεν μπορούσε να ικανοποιήσει τιε απαιτήσεις μας και έτσι αποφασίσαμε να κατασκευάσουμε την δική μας πρόταση που θα πρέπει να είναι φτηνότερη από αυτές της αγέρας και να αξιοποιεί αυτά που ζητάμε που συνοπτικά είναι οι εξής απαιτήσεις.

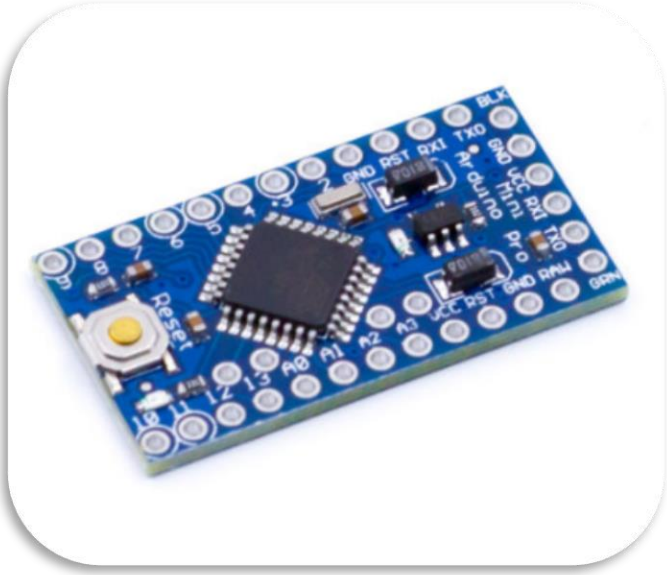
- Ανοιχτού κώδικα.
- Εύκολη επιλογή δικτύου χωρίς επαναπρογραμματισμό.
- Εύκολη αλλαγή Broker.
- Ασύρματο Προγραμματισμό.
- Τουλάχιστον 5 αναλογικές /ψηφιακές εισόδους /εξόδους για αισθητήρες και επενέργησες.
- Χαμηλό κόστος κατασκευής και συντήρησης , κάτω των 20 ευρώ.
- Αφαιρούμενο κύκλωμα τροφοδοσίας για άμεση επιδιόρθωση.
- Υποστήριξη χαμηλής τάσης εναλλασσόμενο ρεύμα για μεταφορά ρεύματος σε μεγάλες αποστάσεις.

Επιλέξαμε για επεξεργαστή τον ATMega 328 MCU σε μορφή module που θα είναι υπεύθυνος για την επικοινωνία με τον φυσικό κόσμο και τον ESP8266 για την ασύρματη επικοινωνία και την λειτούργει του προγραμματισμού καθώς έχει πολύ δυνατό επεξεργαστή και αρκετή μνήμη .

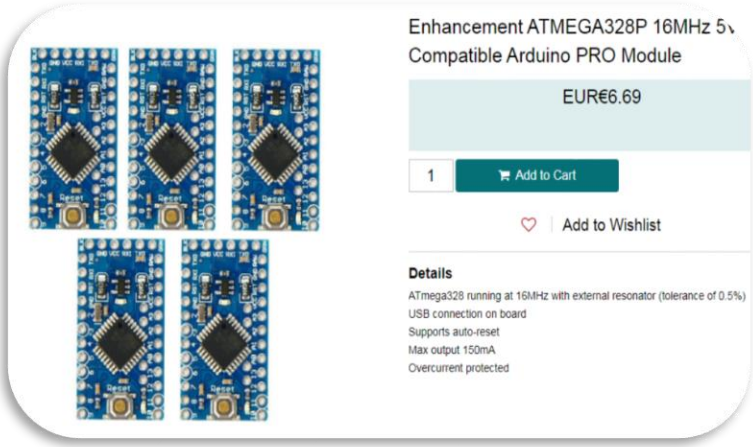
Γνωριμία με τον AtMega 328

Επιλέξαμε ένα 5V 16MHz επεξεργαστή γιατί είναι αξιόπιστος δέχεται αρκετά περιφερικά και μπορούμε εύκολα να ελέγξουμε επενέργησες άμεσα από τις πόρτες του επεξεργαστή .

Έχει πολλά περιφερικά και GPIO που είναι χρήσιμα για να έχουμε σύνθετους αυτοματισμούς. Περιληπτικά έχει 13 GPIO με 2 interrupt και 2 timers 7 αναλογικές 10bit , I^2c , Serial Bus και ISP. Είναι αρκετά φτηνός αφού κοστίζει μόλις 1,5€ σε μορφή πλακέτας module δηλαδή με τον κρύσταλλο και ένα γραμμικό ρυθμιστή τάσης. Αν τον αγοράσουμε σε κομμάτια θα μας έρθει ποιο ακριβά από αυτή την τιμή γιατί δεν περνούμε ποσότητα. Έτσι επιλεγούμε να τον αγοράσουμε σε μορφή module.



Εικόνα 5 το module που επιλεγούμε για την πλατφόρμα μας



Εικόνα 6 το κόστος είναι ιδιαίτερα χαμηλό ανά μονάδα 7€/5=1.4€

Αυτός είναι και ο ποιο διάσημος μικροελεγκτης για την πλατφόρμα προγραμματισμού Adriano.

Τι είναι το arduino

Το 2005 φτιάχτικε μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzi και David Cueartielles ονόμασαν το σχέδιο από τον Αρντουνο της Ιβρέας και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρεα, κωμόπολη της επαρχία Τορίνο στην περιοχή Πεδεμοντιο της βορειοδυτικής Ιταλίας - την ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti .

Το Arduino είναι μία διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες), παρόμοια με την C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης IDE.

Σε εννοιολογικό επίπεδο, στην χρήση του Arduino software stack, όλα τα boards προγραμματίζονται με μία RS-232 σειριακή σύνδεση, αλλά ο τρόπος που επιτυγχάνεται αυτό διαφέρει σε κάθε hardware εκδοχή. Οι σειριακές πλάκες Arduino περιέχουν ένα απλό level shifter κύκλωμα για την μετατροπή του σήματος επιπέδου RS-232 σε TTL. Τα σημερινά Arduino προγραμματίζονται μέσω USB· αυτό καθίσταται δυνατό μέσω της εφαρμογής προσαρμογέων chip USB-to-Serial όπως το FTDI FT232.

Κάποιες παραλλαγές, όπως το Arduino mini, χρησιμοποιούν ένα αφαιρούμενο USB-to-Serial καλώδιο ή board ή άλλες μεθόδους. Ο πίνακας Arduino εκθέτει τα περισσότερα microcontroller I/O pins για χρήση από άλλα κυκλώματα.

Το Arduino nano ενδέχεται να παρέχει male header pins στο κάτω μέρος του board προκειμένου να συνδέονται σε Breadboards. Υπάρχουν πολλά boards συμβατά με και προερχόμενα από Arduino boards. Κάποια είναι λειτουργικά ισάξια με ένα Arduino και μπορεί να χρησιμοποιηθούν εναλλακτικά. Πολλοί είναι το βασικό Arduino με την προσθήκη καινοτόμων output drivers, συχνά για την χρήση σχολικής μόρφωσης για να απλοποιήσουν την κατασκευή buggies και μικρών robot. Κάποιες παραλλαγές είναι τελείως διαφορετικοί επεξεργαστές, με ποικίλα επίπεδα συμβατότητας. Η δυνατότητα να περνάμε το Πρόγραμμα στην Flash του μικρούλικη

μέσω της συριακής πόρτας φυσικά δεν είναι και η καλύτερη λύση , αλλά προσφέρει την ευκολία που χρειαζόμαστε για να κατασκευάσουμε μια προγραμματιζόμενη πλακέτα που προσφέρει ευκολία στην σύνδεση και μας αποδεσμεύει από την αγορά ακριβού και εξειδικευμένου προγραμματιστή. Για να μπορεί να γίνει αυτή η ενεργεία πρέπει να εγκατασταθεί ένα μικρό πρόγραμμα στην flash του μικροελεγκτη που στην ουσία ξεκινά μόλις εμείς δώσουμε τάση στον μικροελεγκτη . Περιμένει μερικά κλάσματα του δευτερόλεπτου και μετά μπαίνει στην κανονική λειτουργία του αποδεσμεύοντας τις πόρτες τις σειριακής .

Αυτό έχει ένα κόστος από πλευράς χώρου μνήμης και ταχύτητας εκκίνησης αφού πρέπει να περιμένει για να δει αν θέλουμε να περάσουμε πρόγραμμα. Ο bootloader είναι μικρό πρόγραμμα και καταλαμβάνει 1,5 Kbyte από την flash που το συνολικό της μέγεθος είναι 32Kbyte για τον AtMega328. Το 2019 έγινε μια ανανέωση στον κώδικα και κατάφεραν να μειώσουν το μέγεθος στα 0,5Kbyte με τον νέο bootloader , Optiboot.

Φυσικά τα παραγόμενα αρχεία από το arduino δεν χρειάζεται να έχουν εγκατεστημένο bootloader για να λειτουργήσουν και γ αυτό έχουμε την επιλογή άμα θέλουμε να πάρουμε το πρόγραμμα χωρίς τον bootloader, αλλά θα χρειαστεί ISP προγραμματιστής για να περάσουμε το πρόγραμμα στον μικροελεγκτη μας .

Arduino IDE

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει στον προγραμματισμό τους καλλιτέχνες και τους νέους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστείτε αρχεία make ή να τρέξετε προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται sketch.

Τα Arduino προγράμματα είναι γραμμένα σε C ή C++ αλλά μας παρέχεται η δυνατότητα να προσθέσουμε και απλό κώδικα από AVR η ακόμα και Assebly αν χρειαστεί. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring, γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες. Οι χρήστες πρέπει μόνο να ορίσουν δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:

-setup():μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις.

-loop():μία συνάρτηση που καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί.

Ένα τυπικό πρώτο πρόγραμμα για έναν μικροελεγκτή αναβοσβήνει απλά ένα LED. Στο περιβάλλον του Arduino, ο χρήστης μπορεί να γράψει ένα πρόγραμμα σαν αυτό:

```
void setup () {
pinMode (LED_PIN, OUTPUT); // ενεργοποιηση pin 13  σαν ψηφιακη εξωδο
}
void loop () {
digitalWrite (LED_PIN, HIGH); // εναυση LED
delay (500); // Αναμονη (500milliseconds)
digitalWrite (13, LOW); // Σβεση LED
delay (500); //  Αναμονη (500milliseconds)
}
```

Κώδικας 1 Απλος κωδικας σε πλατφορμα Arduino

Αντίστοιχος κώδικας σε AVR προγραμματισμό είναι πολύ ποιο δύσκολος και απαιτεί να έχουμε τον οδηγό του επεξεργαστή κοντά μας για να ανατρέχουμε ώστε να βρούμε τα ονόματα και τους διακόπτες που πρέπει να ενεργοποιήσουμε .Αυτό αυξάνει τον χρόνο κατασκευής του προγράμματος μας και το κάνει δύσκολο να το διαβάσουμε ξανά και να καταλάβουμε τι έχουμε κάνει . Φυσικά και αν γράψουμε αυτόν τον κώδικα στο Arduino θα μεταγλωττιστεί κανονικά.

```
int main (void){ //παντα η βασικη συναρτηση ειναι η main
        DDRD |= (1 << PB5); // ορισμος του LED σαν εξοδο
        TCCR1B |= ((1 << CS10) | (1 << CS11)); //ρυθμηση του Timmer
        for (;;) {                  //ατερμονας βροχος
                if (TCNT1 >= 7812){ // ελεγχος αν ο timer εφτασε τα 500ms
                        PORTD ^= (1 << PB5); // Αλλαγη καταστασης του LED
                        TCNT1 = 0; // Μηδενησμος του timer
                }}}
```

Κώδικας 2 Κωδικας σε AVR C

Οπως βλέπουμε το Arduino προσφέρει ένα ποιο εύκολο τρόπο να κάνουμε τις ίδιες διεργασίες άπλα και κατανοητά χωρίς να έχουμε ανάγκη το φυλλάδιο οδηγιών συνεχεία μαζί μας, για να προγραμματίσουμε τον μικροελεγκτή μας . Το ότι κάνει τον κώδικα ποιο απλό οφείλεται στην τεραστία δουλεία που έχει γίνει για να γίνουν ποιο απλές οι εντολές μέσω κατασκευής συναρτήσεων η και αντικείμενων , αφού η Wiring είναι ένα υποσύνολο της γνωστής και αμιγώς αντικειμενοστραφής γλώσσας C++.

Αυτό σημαίνει ότι τα αντικείμενα σαν τεχνική προγραμματισμού ήρθαν στον κόσμο των μικροελεγκτή για να φέρουν την επανάσταση και να δημιουργήσουν προγράμματα που δεν ήταν πριν εφικτό να γίνουν με τον παραδοσιακό τρόπο της Assembly και της ANCI C γιατί από πλευράς ανθρώπου είναι ιδιαίτερα σύνθετο και πολύπλοκο και χρονοβόρο , στοιχειά που το κάνουν μη ανταγωνιστικό να γραφούμε πλέων κώδικα με τις παραδοσιακές μεθόδους.

Το Arduino άνοιξε τον δρόμο στην απλότητα και την ευκολία να χρησιμοποιούμε κώδικα εύκολα και γρήγορα , να φτιάχνουμε ευανάγνωστα προγράμματα και να έχουμε δωρεάν λύσεις για τον προγραμματισμό των μικροελεγκτή που παλαιοτέρα ήθελε ένα σεβαστό ποσό για να αρχίσουμε να προγραμματίζουμε ένα τύπο μικροελεγκτή.

Με την πλατφόρμα του Arduino μπορούμε να έχουμε την ευελιξία και να δουλεύουμε πολλούς διαφορετικούς τύπους μικροελεγκτή καθώς είναι εύκολο για μια εταιρία να προσθέσει υποστήριξη για την πλατφόρμα της καθώς το Arduino είναι ανοικτού κώδικα και δεν απαιτεί αδεία ούτε γραφειοκρατία για να προσθέσει μια ακόμα εταιρία στην λίστα της . Έτσι πλέων πολλές εταιρίες έχουν μπει στην πλατφόρμα και έχουν βγάλει και τα πρώτα πακέτα ανάπτυξης. Ονομαστικά οι σημαντικότερες είναι :

- St
- Renesas
- SiFive
- FreeScale
- Microchip
- Pic
- Arm
- Intel

Επίσης το Arduino IDE πλέων έχει προσθέσει και επαγγελματικού επίπεδου Debugger και υποστήριξη ζωντανής εκσφαλμάτωσης. Επίσης μπήκε και στον κόσμο των FPGA με την πλακέτα MKR Vidor 4000 .

Γνωριμία με το Esp8266

Το ESP8266 είναι ένας επεξεργαστής της Espressif. Αποτελεί ένα module που παρέχει σε όλες τις απαραίτητες ρουτίνες για δικτυακές εφαρμογές καθώς και ασύρματο δίκτυο 802.11.b/g/n . Έχει SPI Flash που μπορεί να φτάσει και τα 8Mbyte για το Firmware ,έχει 7 GPIO και μια αναλογική που είναι στα 10bit 3v3.

Οι GPIO είναι δεσμευμένες και για την λειτούργει του ανάλογα με το τι θέλουμε να κάνει . Γενικά δεν είναι καλός για να τον βάλουμε να επικοινωνεί με το φυσικό περιβάλλον. Έχει μια σειριακή θύρα η όποια μπορεί να χρησιμοποιηθεί για να δέχεται εντολές από έναν επεξεργαστή ώστε να του παρέχει ασύρματη δικτύωση μέσω AT εντολών.

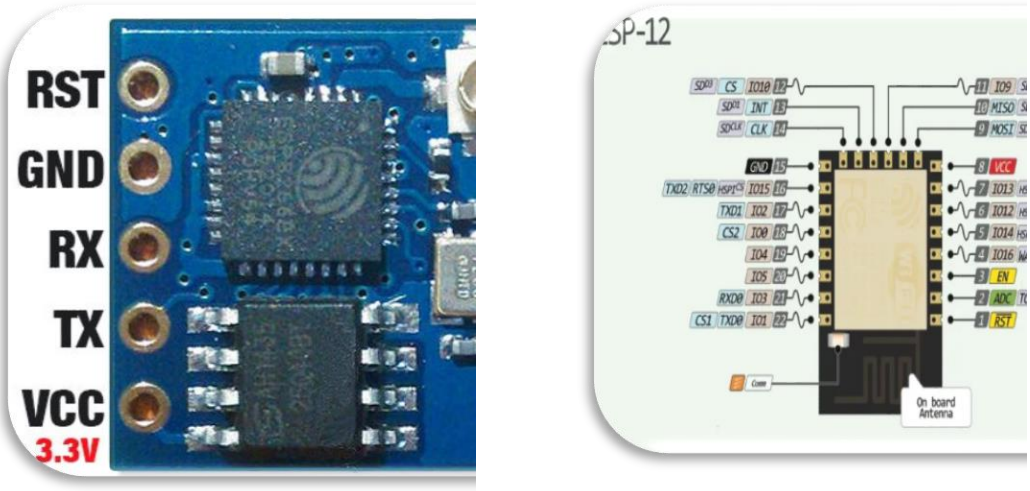
Εικόνα 7 Το module esp8266 εκδοση 07 μαζι με τις εξοδους του

Η Espressif παρέχει και ένα SDK που ονομάζει IDF μέσω του οποίου μπορούμε να κατασκευάσουμε τα δικά μας make ώστε να μπορούμε να εκμεταλλευτούμε την επεξεργαστική του ισχύει . Είναι ένας πολύ φτηνός και καλός επεξεργαστής του οποίου το κόστος είναι και αυτό στ 1.5€. Επίσης έχει και την πιστοποίηση FCC που τον κάνει ικανό να τοποθετηθεί σε προϊόντα χωρίς να χρειάζεται να ξαναβγάλουμε αυτή την πιστοποίηση . Υπάρχει στην αγορά σε πολλές διαφορετικές μορφές και με διαφορετικές μνήμες. Για την περίπτωση μας επιλέξαμε την έκδοση 07 η όποια παρέχει και εσωτερική και σύνδεση IPX για εξωτερική κεραία. Η μεγίστη εκπεμπόμενη ισχύεις είναι 16dbi λόγο περιορισμού από τις ευρωπαϊκές χώρες.

Τα χαρακτηριστικά του είναι πολύ καλά και έχει υπερβολικά χαμηλό κόστος αν θέλουμε να τον συγκρίνουμε με λύσεις από άλλες εταιρίες . Ας δούμε τα χαρακτηριστικά όπως τα περιγράφει ο κατασκευαστής.

- Επεξεργαστής L106 32-bit RISC microprocessor βασισμένος στον Tensilica Xtensa 106Micro χρονισμένο στα 80 MHz.
- 32 KiB εκτέλεσης εντολών RAM, 32 KiB αποθηκευμένων εντολών ,80 KiB δεδομένων εκτέλεσης RAM, 16 KiB ETS παραμετροποίησης συστήματος .
- Εξωτερική QSPI flash: μέχρι και 16 MiB IEEE 802.11 b/g/n Wi-Fi.
- Ενσωματωμένος TR διακόπτης, LNA, Ενισχυτής ισχύος μεταβλητός και δυνατότητα για δίκτυα πλέγματος (mesh netwokrking). WEP η WPA/WPA2 σε επίπεδο υλικού.
- 16 GPIO πόρτες ,SPI, I²C, I²S ,UART
- 10-bit ADC

Οι έξοδοι που παρέχονται αλλάζουν αμάλαγα με την πλακέτα που θέλουμε να δουλέψουμε ,γιατί υπάρχουν πολλές εκδώσεις που δεν έχουν καθόλου GPIO και αναλογική έξοδο. Τα στοιχειά που βλέπουμε παραπάνω είναι τα γενικά χαρακτηριστικά που έχει ο επεξεργαστής Esp8266EX, αλλά το πως θα υλοποιεί το module είναι θέμα του κατασκευαστή και σε τι εφαρμογή απευθύνεται.



Εικόνα 8 Το esp8266-05 και το esp8266-12e εχουν τεραστιες διαφορες ως πρως τις εξοδους

Ο Esp8266 έχει πλέων αναπτυχθεί αρκετά και οι εκδώσεις που κυκλοφορούν στην αγορά είναι ώριμες και αξιόπιστες πάρα το χαμηλό κόστος.

Η σύνθεση της Πλατφόρμας MqMax

Το MqMax αποτελεί το όνομα της πλατφόρμας που δημιουργήσαμε στο εργαστήριο του Ελληνικού Μεσογειακού Πανεπιστήμιου στο εργαστήριο Πληροφοριακών συστημάτων και Μικροελεγκτών. Αποτελείται από δυο modules το Mini Pro 5v και το Esp8266-07.

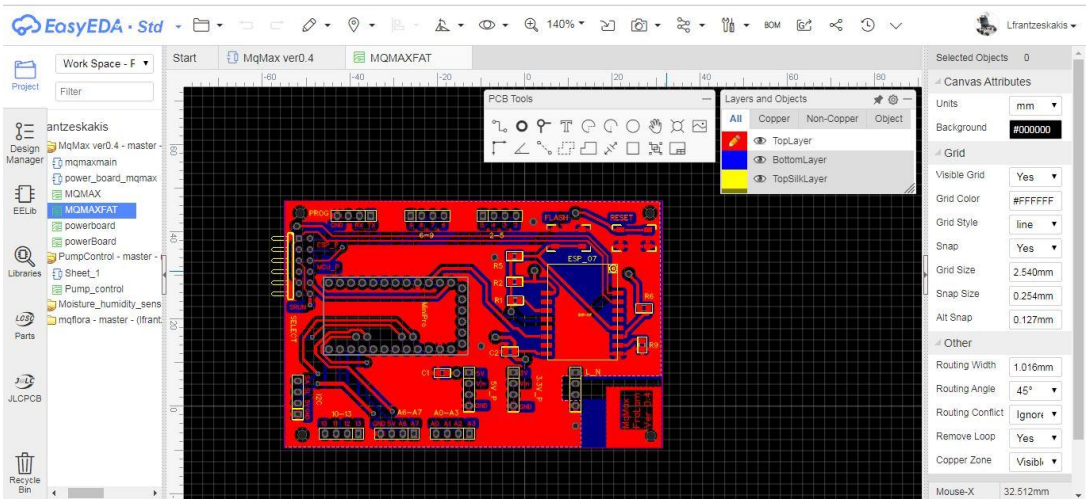
Η επικοινωνία των δυο μονάδων θα γίνει μέσω της σειριακής θύρας που έχουν . Και οι δυο μονάδες όμως προγραμματίζονται μέσω της σειριακής θύρας όποτε πρέπει να προβλέψουμε να έχουμε ένα τρόπο να μπορούμε να προγραμματίσουμε χωριστά τις δυο μονάδες μας . Άρα μια διαπάλη με διακόπτες είναι απαραίτητη για τον προγραμματισμό και την επικοινωνία τους.

Τέλος μένει να δούμε πως θα τροφοδοτήσουμε τις δυο μονάδες . Το πρόβλημα που υπάρχει στα αντλιοστάσια είναι ότι έχουμε μεγάλα ρεύματα υψηλές τάσεις και υπάρχει αρκετή υγρασία . Αυτό προκαλεί ολίσθηση πολλές φόρες τις μονάδες ισχύος στις συσκευές αυτοματισμού, με αποτέλεσμα να μην μπορούν να αντέξουν για πολύ καιρό και να χαλινέ . Αυτό μιάνει την αξιοπιστία και πολλές φόρες οδηγεί στην καταστροφή ολόκληρης περιοχής τροφοδοσίας με αποτέλεσμα να χαλάσει η πλακέτα έλεγχου. Λόγο του ότι τα εξαρτήματα είναι σε πακέτο SMD ,είναι δύσκολο να βρεθούν και συχνά έχουμε να κάνουμε με απομακρυσμένες περιοχές και δύσβατα μέρη που έχουν ανάγκη από άμεση επιδιόρθωση .

Για αυτόν το λόγο αποφασίσαμε την κατασκευή προθέτεις πλακέτας τροφοδοσίας . Αυτό επιπρόσθετα μας προσφέρει την δυνατότητα να φτιάξουμε διαφορετικές πλακέτες τροφοδοσίας ανάλογα με την περίπτωση . Για παράδειγμα θέλουμε να είναι αυτόνομο ενεργειακά το σύστημα τότε μπορούμε να προσθέσουμε μια πλακέτα τροφοδοσίας που να είναι με μπαταρία και φορτιστή φωτοβολταϊκής κυψέλης, η να μην έχουμε την δυνατότητα να έχουμε ρεύμα συνεχές και να έχουμε μόνο εναλλασσόμενο στα 220 η στα 24. Δεν θα χρειαστεί να σχεδιάσουμε διαφορετικές πλατφόρμες άπλα θα αλλάζουμε το σύστημα τροφοδοσίας και θα μπορούμε να δουλέψουμε κανονικά τον αυτοματισμό μας η το σύστημα συλλογής δεδομένων ανάλογα το σενάριο χρήσης.

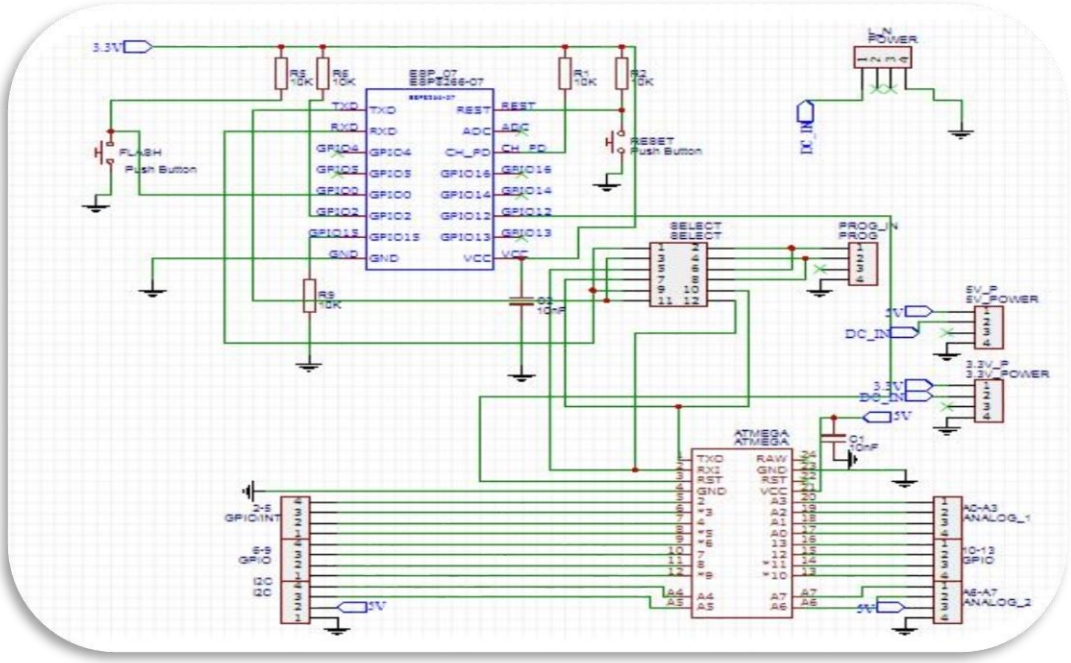
Λαμβάνοντας όλα αυτά υπόψη αρχίσαμε τον σχεδιασμό της ηλεκτρονικής πλακέτας με ένα πρόγραμμα EDA . Επιλέξαμε το EasyEDA (Εικόνα 9) γιατί έχει πολλά σχεδία από εξαρτήματα καθώς και πολλές ευκολίες άμα θέλαμε να περάσουμε

στην παράγωγη. Το πρόγραμμα είναι make και δεν απαιτεί εγκατάσταση ,είναι πλήρες και δωρεάν.



Εικόνα 9 Το περιβάλλον του προγράμματος EasyEDA , σχεδιασμός PCB από Schematic

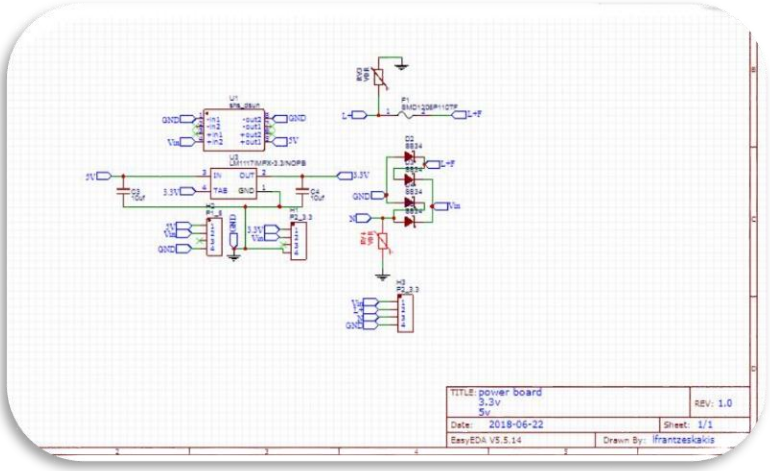
Προσθέσαμε τα δυο Module , το Mcu ATmega382 (arduino mini pro) και το SoC ESP 8266 -07. Στην πλακέτα προσαρμόσαμε όλα τα στοιχειά που θα βοηθήσουν στον εύκολο προγραμματισμό των μονάδων χωριστά .



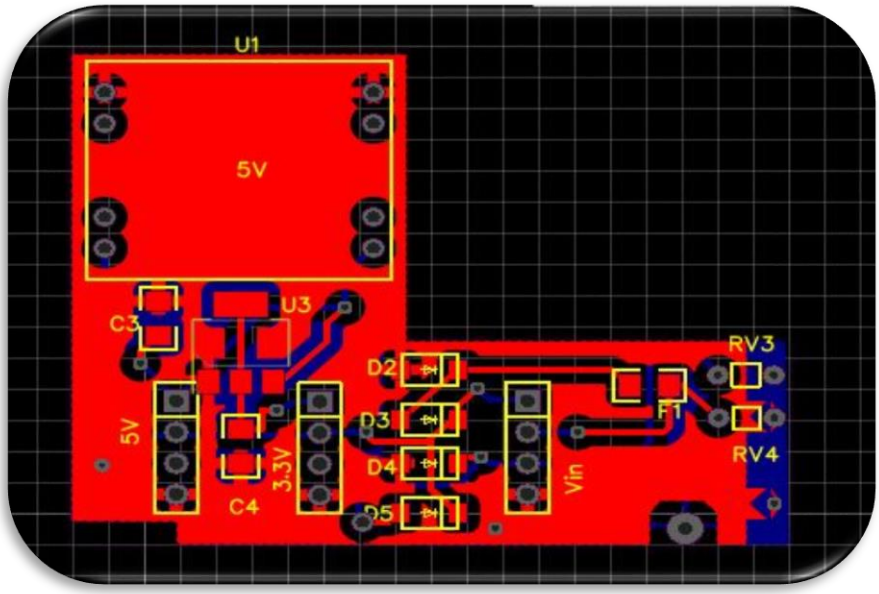
Εικόνα 10: Το σχηματικό του MqMax

Αφού σχεδιάσαμε πρώτα τα σχηματικά (εικόνα 10) μετά προχωρήσαμε στην αποτύπωση της πλακέτας σε PCB (εικόνα 9). Έγιναν διαφορές εκδώσεις μέχρι να φτάσουμε στην 0.4, όπου είναι και η έκδοση με την όποια έγινε η εργασία.

Σχεδιάσαμε χωριστά μια έκδοση για την πλακέτα έλεγχου και μια για την πλακέτα τροφοδοσίας.



Εικόνα 11 :Το Σχηματικό της μονάδας τροφοδοσίας για AC/DC 6-28V



Εικόνα 12: Power Board για το MqMax PCB view

αφού ολοκληρώθηκε ο σχεδιασμός στο CAD πρόγραμμα, άρχισε η διαδικασια δοκιμης για τις προτοτυπες πλακετες ωστε να επιβεβαιωθει η ορθη λειτουργεια της πλατφορμας .

Εδω πρεπει να αναφερουμε μερικα στοιχεια για την λειτουργεια της πλακετας τροφοδοσιας , το λεγομενο PowerBoard.

- Οι θέσεις σύνδεσης έχουν τοποθετηθεί με τέτοιο τρόπο ώστε να μπορεί να μπει μόνο με μια κατεύθυνση.

- Χρησιμοποιήθηκε switching τροφοδοτικό για να μετατρέψουμε την τάση από 6-28 Volt σε 5Volt σταθερά.
- Από τα 5Volt στα 3.3Volt έχουμε ένα Linear Regulator LM1117 το όποιο έχει καλή απόδοση στα mA που χρειαζόμαστε για να τροφοδοτήσουμε τον Esp8266.
- Υπάρχει μια ασφάλεια επαναφοράς τον 1A την είσοδο της τροφοδοσίας.
- Γείωση και VDR για αντικεραυνική προστασία της κατασκευής . Αν δεν συνδέσουμε γείωση αυτό το χαρακτηριστικό καταργείται.
- Μια γέφυρα που επιτρέπει την σύνδεση και εναλλασσόμενου ρεύματος χαμάλης τάσης , ρεύμα αυτοματισμού , όπως έλεγαν στους παλιούς πινάκες . επίσης αυτό μας δίνει την ευελιξία να παρέχουμε ρεύμα με λεπτά καλώδια και σε μεγάλες αποστάσεις.

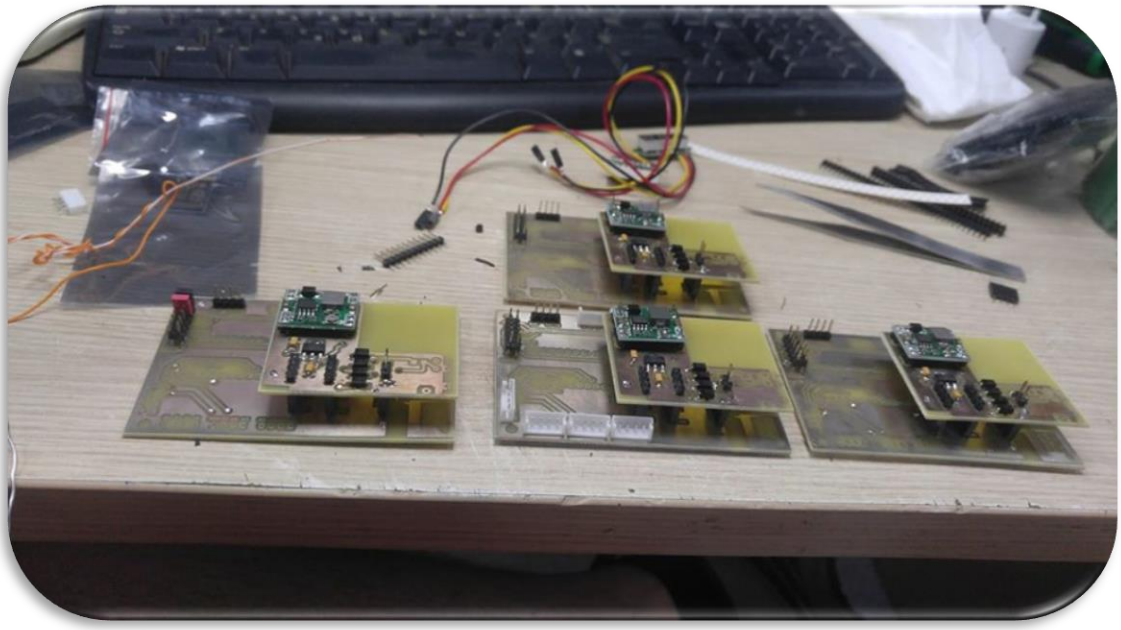
Η συγκέντρωση των υλικών για την κατασκευή των πλακετών με το BOM List που βλέπουμε παρακάτω αφόρα και τις δυο πλακέτες ,PowerBoard και MqMax. Η επιλογή των υλικών έγινε από διαφορετικές πήγες και σκοπό είχε να ατυχούμε το χαμηλότερο κόστος κατασκευής χωρίς να έχουμε χαμηλή ποιότητα.

ID	Name	Quantity	Price/prt	Overall
1	SMD1206P110T Fuse	1	0,12	0,12
2	VDR 80v	2	0,1	0,2
3	dsun mp1853 dc-dc	1	0,7	0,7
4	ESP8266-07	1	1,6	1,6
5	LM1117 -3.3v	1	0,2	0,2
6	ATMEGA 328p	1	1,5	1,5
7	Push Button	2	0,15	0,3
8	10K	5	0,01	0,05
9	10nF	2	0,1	0,2
10	10uf	2	0,1	0,2
11	4pin header	12	0,02	0,24
12	SS34 diode	4	0,1	0,4
13	PCB board 10X12	1	1	1
Συνολο				**6,71**

Πίνακας 1 BOM List για την πλατφορμα και το DC Τροφοδοτικο της

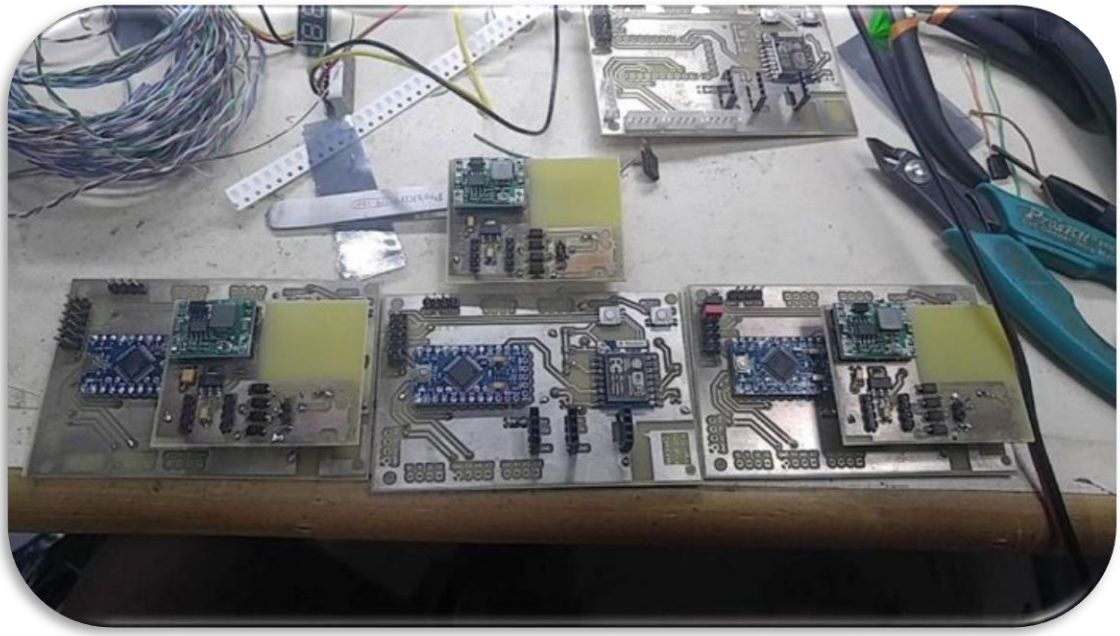
Η εκτύπωση των πλακετών εγίνε με την τεχνική Positive UV και εγίνε στο εργαστήριο του ΕΛ.ΜΕ.ΠΑ (Εικόνα 13).

Το αποτέλεσμα ήταν το αναμενόμενο μετά την εκτύπωση των πρώτων πλακετών και την συμπλήρωση τους με εξαρτήματα. Λειτούργησαν όλα τα υποσυστήματα αρκετά καλά μετά από τις πρώτες δόκιμες



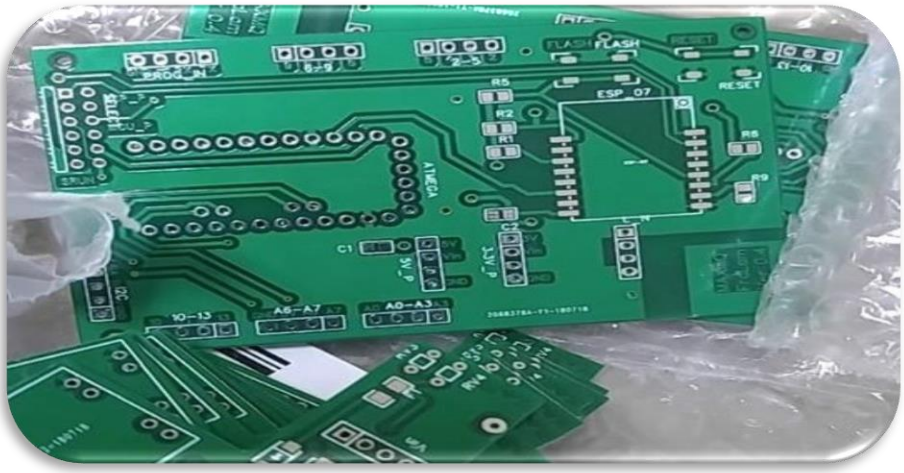
Εικόνα 13 Κατασκευή Πλακετών στο Εργαστήριο του ΕΛ.ΜΕ.ΠΑ

Η συναρμολόγηση έγινε επίσης στο εργαστήριο του ΕΛ.ΜΕ.ΠΑ με εξαρτήματα που λάβαμε από διαφορετικές πήγες.



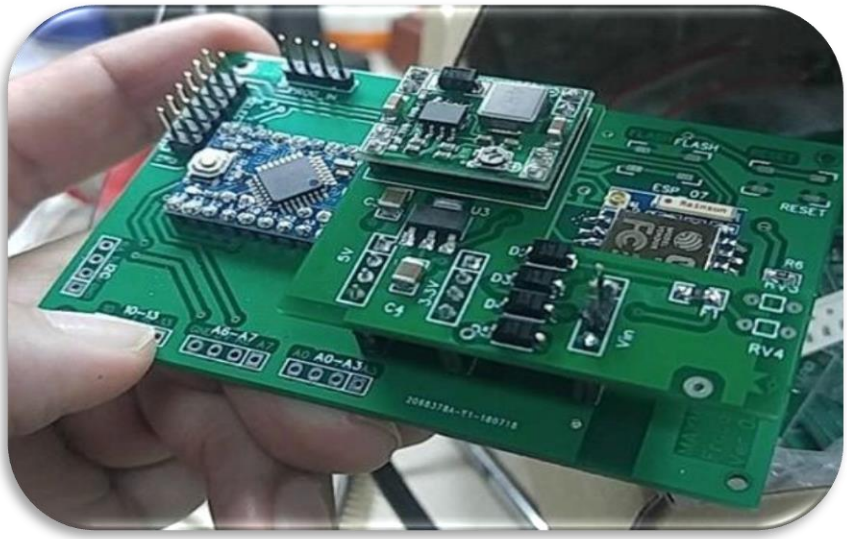
Εικόνα 14 Συναρμολόγηση πλακετών μετά την εκτύπωση για δόκιμες.

Όταν ολοκληρωθήκαν οι δόκιμες σε πραγματικό περιβάλλον περάσαμε στην αγορά κατασκευασμένων πλακετών σε γραμμή παράγωγης κατά παραγγελία. Η παραγγελία έγινε στην JLCPCB με πολύ χαμηλό κόστος παράγωγης , χαμηλότερο από την κατασκευή στο εργαστήριο. Μετά από μερικές μέρες έφτασε το πακέτο με τις πλακέτες για συναρμολόγηση.



Εικόνα 15 Οι πλακέτες όπως έφτασαν από την JLCPCB

Έτσι το τελικό αποτέλεσμα είναι επαγγελματικού επίπεδου με υψηλή ποιότητα και χαμηλό κόστος παράγωγης.



Εικόνα 16 Τελικό αποτέλεσμα , η γέννηση του MQmax 0.4

Φυσικά και θα συνεχίσει η εξέλιξη της πλατφόρμας αλλά για την υλοποίηση της εργασίας είναι σε ένα ώριμο στάδιο στην έκδοση 0.4 .

Λειτουργείς του MqMax

Με την ολοκλήρωση του σχεδιασμού περάσαμε στην υλοποίηση του λογισμικού που θα κάνει την πλατφόρμα μας λειτουργική. Σκοπός μας είναι να προσφέρει κάποιες ευκολίες που θα κάνουν ποιο ευέλικτη την εγκατάσταση της σε περιοχές ενδιαφέροντος. Μετά από διασταύρωση πληροφοριών θέσαμε τους έξεις στόχους .

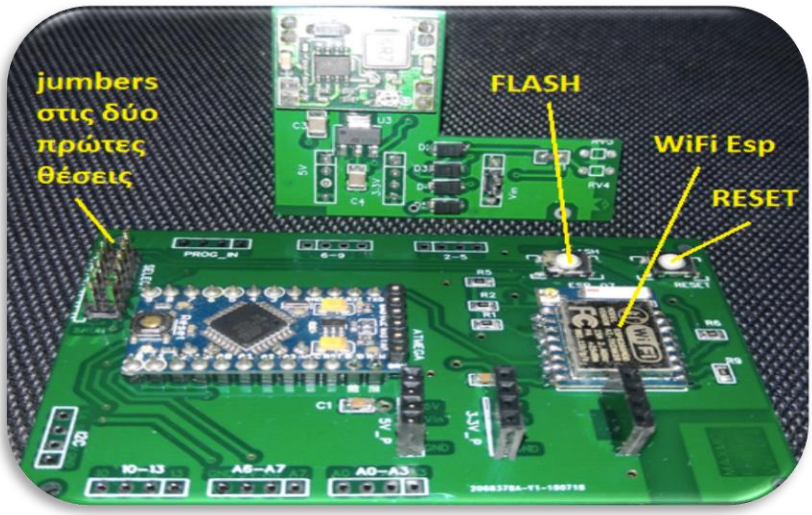
- Να μπορεί να συνδεθεί με εύκολο τρόπο σε ασύρματα δίκτυα
- Να υποστηρίζει MQTT καθώς είναι και το πρωτόκολλο επιλογής μας
- Να μπορεί να προγραμματίζεται ασύρματα η αλλιώς Over the Air
- Να υποστηρίζει εισόδους/εξόδους ψηφιακές και λειτουργία I^2C
- Να έχει αναλογικές Εισόδους
- Να προγραμματίζεται εύκολα με καλώδιο Usb

Αυτές είναι οι βασικές απαιτήσεις που έχουμε από την λειτουργικότητα της πλατφόρμας μας .

Μετά από αναζήτηση καταλήξαμε ότι όλες οι απαιτήσεις μπορούσαν να καλυφτούν από το firmware Esplink της Jeelabs το όποιο είναι Opensource . Το firmware αυτό μας προσφέρει πολλές ελευθερίες και έχουμε την δυνατότητα να το παραμετροποιησουμέ όπως θέλουμε κάνοντας Fork στο gitHub .

Το Esplink ήταν το Firmware που δούλευε για το Arduino Uno WIFI ver1.0 οπότε είχαμε την σιγουριά ότι θα λειτουργήσει αξιόπιστα αφού είναι και το firmware επιλογής και από άλλους κατασκευαστές.

Για τον προγραμματισμό του MqMax χρειαζόμαστε ένα Usb to TTL μετατροπέα που μπούμε να προσαρμόσουμε στις ανάγκες μας . Έχουμε διαθέσιμες τις πόρτες της σειριακής μέσω του επιλογέα προγραμματισμού. Τοποθετούμε τα 2 Jumper στις πρώτες θέσεις και τοποθετούμε το PowerBoard ξανά στην πλακέτα .



Εικόνα 17 Προγραμματισμός για το MqMax

Συνδέουμε το USB/TTL μετατροπέα στην πλακέτα και στον υπολογιστή. Βλέπουμε τις συριακές συσκευές στο σύστημα και κρατάμε την διεύθυνση της συσκευής.

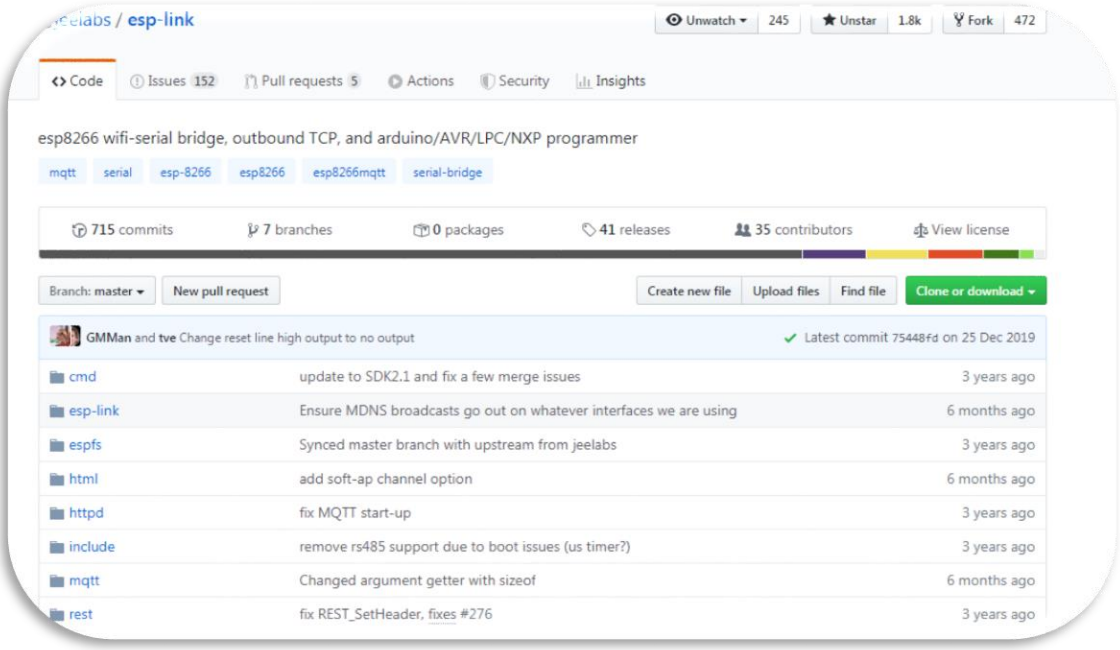
Εικόνα 18 Καλώδιο για τον πρώτο προγραμματισμό της πλατφόρμας μετά τον προγραμματισμό μπορούμε να αλλάζουμε το firmware του μικροελεγκτή ασύρματα

Αυτή η διεύθυνση θα χρησιμοποιηθεί παρακάτω όταν θα παραμετροποιησουμέ το πρόγραμμα που θα χρησιμοποιήσουμε για το ESP το esptools. πρώτα όμως πρέπει να κατασκευάσουμε το firmware που θα εγκαταστήσουμε .

Esplink Firmware

Το Esplink είναι ένα λογισμικό ανοικτού κώδικα το οποίο παρέχει την δυνατότητα μεταφοράς της σειριακής επικοινωνίας μέσω του ασύρματου δικτύου. επίσης έχει ενσωματωμένο Mqtt Client που επικοινωνεί με τους μικροελεγκτές κάνοντας όλες τις δύσκολες λειτουργιές επικοινωνίας και αποθήκευσης στο Esp αφήνοντας μόνο τα μηνύματα για τον μικροελεγκτή. Επίσης μπορεί να κάνει και επικοινωνία με Server μέσω WebSockets και REST πρωτόκολλο . Τέλος παρέχει και την δυνατότητα να προγραμματίσουμε ένα Mcu αν έχει τον Bootloader του Arduino. Αρά μπορεί να προγραμματίσει μόνο AVR και NXP.

Είναι ανοικτού κώδικα και έχει οδηγίες χρήσης και αλλαγής παραμέτρων . Για να το περάσουμε στο Esp πρέπει πρώτα να κατεβάσουμε τον κώδικα και να τον μεταγλωττίσουμε ανάλογα την έκδοση της πλακέτας που χρησιμοποιούμε .



Εικόνα 19 Η σελίδα το Esplink στο gitHub

Η μεταγλώττιση απαιτεί ένα σύνθετο σύστημα εργαλείων και γ αυτόν τον λόγο υπάρχει έτοιμο Docker που αναλαμβάνει να κάνει αυτή την δουλεία πολύ ποιο εύκολη.

```
docker pull jeelabs/esp-link
```

Κώδικας 3 Esp-Link Docker image

Αφού τραβήξουμε τις ρυθμίσεις μεταγλώττισης από το docker , εκτελούμε την εντολή στο περιβάλλον του docker

```
docker run -v c:\<θέση αρχείων>\esp-link:/esp-link jeelabs/esp-link:lates
```

Κώδικας 4 Docker Compile

Η εντολή θα ξεκινήσει την μεταγλώττιση και θέλει αρκετή ώρα μέχρι να ολοκληρωθεί . Όταν ολοκληρωθεί θα εξάγει δυο αρχεία , το user1.bin και το user2.bin. Το user1.bin περιέχει τον βασικό κώδικα για την λειτουργία του Esplink. Το user2.bin περιέχει τις πληροφορίες και κράτα θέση στην flash για να έχουμε την λειτουργία OTA (over The Air update).Έτσι ολοκληρώσαμε την κατασκευή του λογισμικού για το Esp8266.

Την πρώτη φόρα που θα το περάσουμε στο Module πρέπει να το κάνουμε με κάποιο TTL to USB μετατροπέα. Επίσης το Esp8266 07 δεν έχει όλους τους

διακόπτες που χρειάζονται για τον προγραμματισμό του ,κλειστούς με αποτέλεσμα να πρέπει να τους κλείσουμε εμείς στην πλακέτα . Στο σχέδιο έχουμε προσθέσει δυο κουμπιά που έχουν το σήμα Flash και Reset , πρέπει να τα πατήσουμε ταυτόχρονα όσο η πλακέτα τροφοδοτείται για νια να μπει το ESP σε λειτουργία προγραμματισμού επίσης έχουν τοποθετηθεί κατάλληλα οι αντιστάσεις στις εξόδους του esp ώστε να λειτούργει το Esplink σωστά και χωρίς κολλήματα.

Το περιβάλλον προγραμματισμού του για το Esp8266 είναι γραμμένο σε Python . Αρά για να λειτουργήσει πρέπει να έχουμε εγκατεστημένη στον υπολογιστή μας την γλωσσά και τον packet manager pip . Για να ανεβάσουμε το firmware πρέπει να κατεβάσουμε τα esptools, που είναι και το πακέτο προγραμματισμού των ESP. Από το τερματικό τον Windows εκτελούμε την εντολή .

```
pip install esptools
```

Κώδικας 5 Esptools install

Αφού εγκατασταθεί το πακέτο φτιάχνουμε ένα .bat αρχείο , στο οποίο θα περάσουμε τον κώδικα κελύφους που περιγράφεται παρακάτω.

```
python esptool.py --port <όνομα πόρτας> --baud 115200 write_flash --flash_size=detect 0x0000 boot_v1.6.bin 0x1000 user1.bin 0xFC000 esp_init_data_default.bin 0xFE000 blank.bin
```

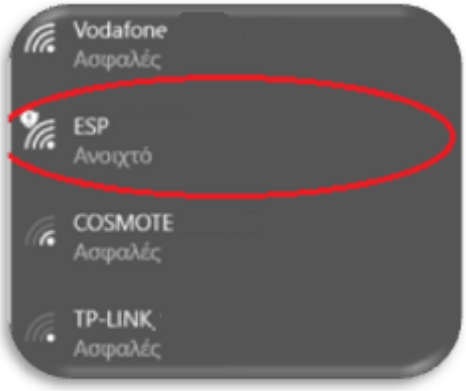
Κώδικας 6 Παραμετροποίηση esptools για προγραμματισμό Esp07

Τοποθετούμε το αρχείο στον φάκελο που έχουμε τα αρχεία από το EspLink . φροντίζουμε να έχουμε τα εξής αρχεία .

- ***boot_v1.6.bin*** είναι το αρχείο που ενεργοποιεί τους κατάλληλους διακόπτες
- ***user1.bin*** είναι το firmware και ο κώδικας που θα εκτελέσει το esp
- ***esp_init_data_default.bin*** είναι οι default λειτουργιές που έχει το esp κατά την εκκίνηση
- ***blank.bin*** αρχείο που στοιχειοθετεί την θέση που τελειώνει η εγράφη στην μνήμη

Έχοντας αυτά τα αρχεία μπορούμε να κάνουμε την φόρτωση του firmware άπλα εκτελώντας το αρχείο .bat .

Αφού ολοκληρωθεί η φόρτωση κάνουμε Reset στο EspLink και βλέπουμε τα ασύρματα δίκτυα που μας περιβάλουν από το κινητό η από τον υπολογιστή μας . Εκεί θα παρατηρήσουμε ότι πλέων υπάρχει ένα νέο δίκτυο με όνομα ESP_XXX .



Εικόνα 20 Προβολή του MqMax στα ασύρματα δίκτυα

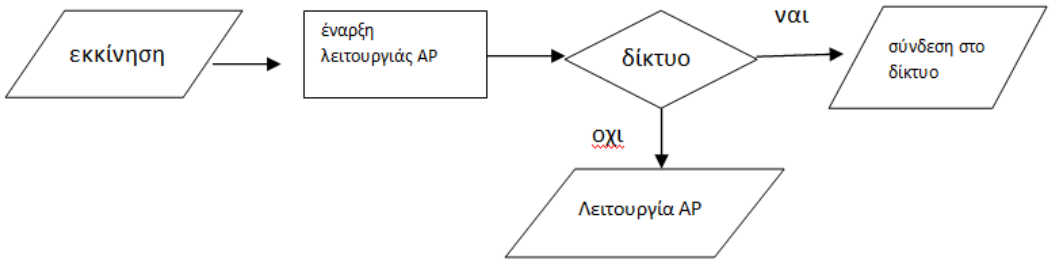
Συνδεόμαστε σε αυτό το δίκτυο και βάζουμε την διεύθυνση 192.168.4.1 σε ένα browser . Αν είμαστε σε κινητό, κάλο είναι να ενεργοποιήσουμε την επιλογή προβολής σαν υπολογιστή.

Εικόνα 21 Η κεντρική Σελίδα από το Esplink firmware στην πλατφόρμα μας MqMax

Στον περιβάλλον του Esplink, πηγαίνουμε στο "WiFi Station" και εντοπίζουμε το WiFi του δρομολογητή (Router) του παρόχου Internet και συνδέουμε τον ελεγκτή με το δίκτυο του. Στη θέση "password" πληκτρολογούμε τον κωδικό ασφαλείας του WiFi του παρόχου.

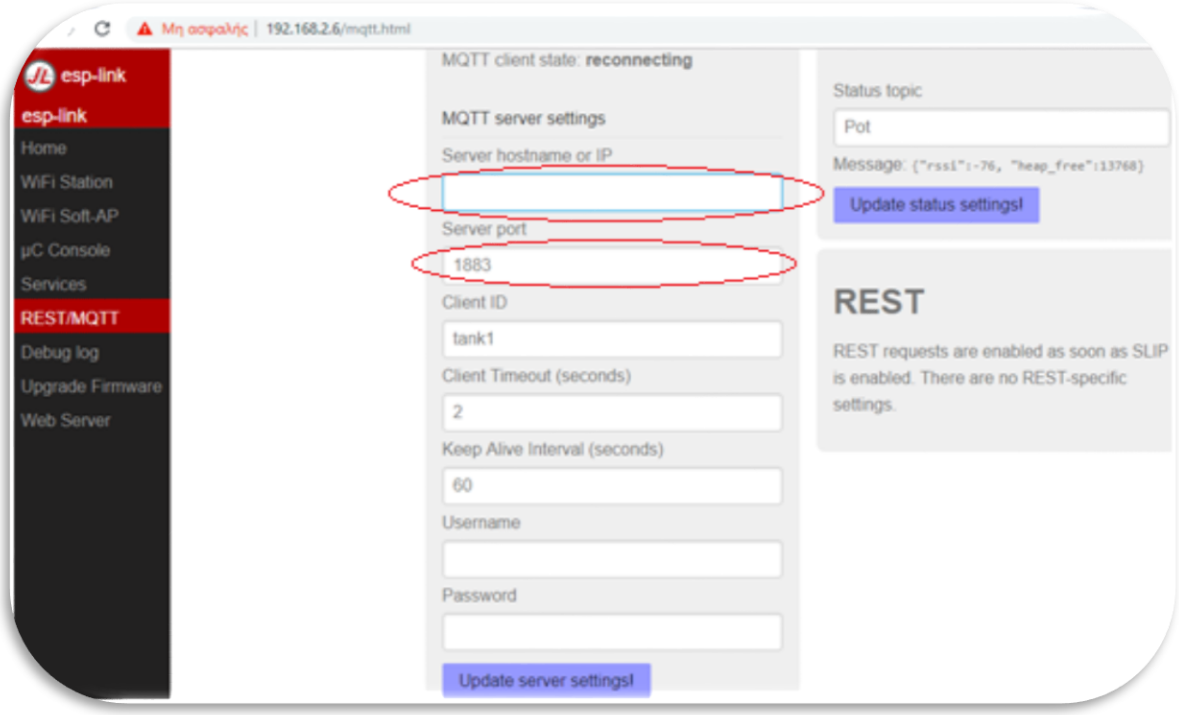
Εικόνα 22 Συνδεση στο ασηρματο δικτυο

Τώρα όπως φαίνεται και στην παραπάνω εικόνα το esp-link θα φαίνεται σε άλλη IP διεύθυνση του φυλλομετρητή και συγκεκριμένα στην http://192.168.2.6/ (όπως φαίνεται στην παραπάνω εικόνα). Ενώ δεν εντοπίζεται ως ασύρματο δίκτυο πλέων, όπως είχε ειπωθεί παραπάνω κατα την διαρκεια της λειτουργειας του αλλα μονο κατα την εκκινηση του συστηματος για μερικα δευτερολεπτα . Εάν δεν συνδεθούμε τότε συνδέεται αυτόματα στο γνωστό δίκτυο. Αν το δίκτυο δεν είναι ενεργό τότε παραμένει σε κατάσταση AP για να μπορέσουμε να συνδεθούμε και να ρυθμίσουμε άλλο δίκτυο.



Εικόνα 23 Αλγόριθμος Λειτουργιάς εκκίνησης.

Για να ρυθμίσουμε τον MQTT Broker στο ίδιο περιβάλλον του Esplink πηγαίνουμε στο “REST/MQTT” και πληκτρολογούμε την διεύθυνση του διακομιστή δηλαδή, Server (Broker) που επιθυμούμε συνδεθουμε, στα πεδία “Username” και “Password” βάζουμε τον χρηστή και τον κωδικό που έχουμε για τον Broker και πατάμε “Update Server settings”.



Εικόνα 24 Ρύθμιση του broker

Εδω πρέπει να επισημάνουμε ότι πρέπει να έχουμε ενεργοποιημένο το πρωτόκολλο SLIP από την καρτέλα ρύθμισης του Broker . Το SLIP είναι μια βιβλιοθήκη για την μετατροπή του RS232 σε IP Stack έτσι οι συσκευές μας έχουν Ack και Data Payload . Κατά την εκκίνηση λειτουργίας τους αρχίζει μια διαδικασία συγχρονισμού η όποια δεν πρέπει να διαταχτεί καθόλα την διάρκεια λειτουργίας τους . Γ αυτό στον κώδικα του Arduino δεν μπορούμε να έχουμε delay ρουτίνες γιατί σταματάνε την λειτουργία του Program Counter και έχουμε αποσυντονισμό.

Προγραμματισμός της MCU

Η ομάδα που έχει φτιάξει το firmware Esplink έχει κατασκευάσει και την βιβλιοθήκη για την επικοινωνία μεταξύ του esp και του atmega το ElClient . Είναι γραμμένη σε C++ και είναι συμβατή με το Arduino IDE . Την προσθέτουμε στις βιβλιοθήκες του arduino IDE και μπορούμε να φτιάξουμε τον κώδικα μας .

```
#include <ELClient.h>
#include <ELClientCmd.h>
#include <ELClientMqtt.h>
```

Κώδικας 7 Βιβλιοθήκες για τον MCU

Αφού συμπεριλάβουμε τις βιβλιοθήκες μας στο πρόγραμμα προσθέτουμε και τις βασικές παραμέτρους για να μπορεί να επικοινωνήσει το esp με τον mcu.

Έτσι φτιάχνουμε τα αντικείμενα τύπου ELClient , ELClientCmd, ELClientMqtt με τις παρακάτω εντολές.

```
ELClient esp(&Serial, &Serial);
ELClientCmd cmd(&esp);
ELClientMqtt mqtt(&esp);
```

Κώδικας 8 Βασικά αντικείμενα για την λειτουργία του Esplink MQTT

Πρώτα πρέπει να αρχίσει η επικοινωνία και αυτό γίνεται στην συνάρτηση void setup() του arduino. Μέσα στην setup προσθέτουμε τον κώδικα έλεγχου συγχρονισμού που είναι ο παρακάτω .

```
bool ok;
 do
 {
  ok = esp.Sync();
  if (!ok) Serial.println("EL-Client sync failed!");
 } while(!ok);
 Serial.println("EL-Client synced!");
```

Κώδικας 9 συγχρονισμού Esplink με Mcu

Για να έχουμε σωστή σύνδεση με το Mqtt δίκτυο πρέπει να ξέρουμε αν είναι συνδεμένο το wifi σε κάποιο AP. Γ αυτό τον λόγο κατασκευάζουμε την συνάρτηση wifiCb() και εισάγουμε τον παρακάτω κώδικα .

```
void wifiCb(void* response) {
 ELClientResponse *res = (ELClientResponse*)response;
 if (res->argc() == 1) {
  uint8_t status;
  res->popArg(&status, 1);

  if(status == STATION_GOT_IP) {
   Serial.println("WIFI CONNECTED");
  } else {
   Serial.print("WIFI NOT READY: ");
   Serial.println(status);
  }
 }
}
```

Κώδικας 10 έλεγχου σωστής λειτουργίας της σύνδεσης μας στο δίκτυο

Αφού έχουμε πάρει την IP από το δίκτυο μας , βλέπουμε και αν μπορούμε να επικοινωνήσουμε με τον Broker. Γ αυτόν το λόγο κάνουμε έλεγχο μέσα στην setup μέσω του αντικειμένου mqtt ορίσαμε ποιο πάνω .

Έτσι κάνουμε όλους τους απαραίτητους ελέγχους με τις παρακάτω εντολές.

```
mqtt.connectedCb.attach(mqttConnected);
mqtt.disconnectedCb.attach(mqttDisconnected);
mqtt.publishedCb.attach(mqttPublished);
mqtt.dataCb.attach(mqttData);
mqtt.setup();
```

Κώδικας 11 Αντικείμενα MQTT Client

Αφού ορίσαμε τα αντικείμενα για τον mqtt client , ξεκινάμε τον έλεγχο της επικοινωνίας με τον broker. Με την συνάρτηση void mqttConnected(void *) περνούμε επιστροφή από τον Esp για την κατάσταση σύνδεσης με τον broker που ορίσαμε στο web interface . Όπως θα δούμε και στον κώδικα παρακάτω , στην mcu ορίζουμε το topic στο οποίο θα γίνεται η επικοινωνία στην περίπτωση μας είναι το /pumps/ επίσης υπάρχει και ένα δεύτερο το οποίο είναι το /pumps/bat/ στο οποίο λαμβάνουμε τα δεδομένα μιας μπαταριάς του υπάρχει στην αυτόνομη μονάδα . Στο παρακάτω κεφάλαιο θα εξηγήσουμε ακριβώς την επικοινωνία και τα μηνύματα

```
bool connected;

void mqttConnected(void* response) {
 Serial.println("MQTT connected!");

 mqtt.subscribe("/pumps/");
 mqtt.subscribe("/pumps/bat");

 connected = true;
}
```

Κώδικας 12 . επιστροφή True όταν έχουμε συνδεθεί με τον broker

Στην void loop() συνάρτηση του arduino εισάγουμε τις συναρτήσεις και τους ελέγχους για την λειτουργία του αλγόριθμου μας . Κάνουμε συνεχή έλεγχο στις εισόδους των πόρτων του Mcu όπου έχουμε συνδέσει τα αισθητήρια στάθμης. Επίσης από εκεί κάνουμε και την αποστολή των μηνυμάτων στο topic.

Για κάθε μέρος είχαμε διαφορετικές απαιτήσεις οπότε κάναμε και μικρές μεταβολές στον κώδικα που έχει το κάθε module . Στον αυτοματισμό του πηγαδιού είχαμε μόνο την λειτουργία open/close καθώς και ενημέρωση για το αν δουλεύει η αντλία. Στην δεξαμενή στην μέση της διαδρομής έχουμε δυο αισθητήρια στάθμης τα όποια ελέγχουν το σενάριο λειτουργίας της αντλίας προστατεύοντας την από έλλειψη νερού καθώς και προστασία από υπερχείλιση της δεξαμενής .

Στην τελική θέση έχουμε δυο αισθητήρες στάθμης για χαμηλή και υψηλή στάθμη όπως επίσης και ένα μετρητή τάσης μπαταριάς καθώς είναι αυτόνομη η λειτουργία στην θέση αυτή καθώς δεν υπάρχει ρεύμα από δίκτυο . Έτσι το σίτεμα ενημερώνετε αυτόματα για την τάση της μπαταριάς και σε περίπτωση που η στάθμη της μπαταριάς είναι χαμηλή να μπορεί να ειδοποιήσει και να κλείσει το σύστημα για να αποφύγουμε την υπερχείλιση.

```
void T_battery()
{
 char finalStage[5];
 float volt;
 String pub;
 volt = analogRead(analog);
 pub = String(volt);
 pub.toCharArray(finalStage , 5);
 mqtt.publish("/pumps/bat",finalStage);
}
```

Κώδικας 13 Αποστολή τάσης μπαταριάς στο topic /pumps/bat/ δεξαμενής 2

Για την Άνω δεξαμενή μας ενδιαφέρει όπως προείπαμε ,η τάση της μπαταριάς του αυτοματισμού και η στάθμη του νερού της δεξαμενής . Σε περίπτωση που είναι χαμηλή τότε στέλνουμε μήνυμα εκκίνησης της αντλίας της μεσαίας δεξαμενής.

```
void Tmsg(int katw , int panw , char *okMsg){
 if (katw  == 0){
 mqtt.publish("/pumps/","M_start");
 }
else if(panw == 1){
mqtt.publish("/pumps/","M_stop");
 }
 else{
 mqtt.publish("/pumps/",okMsg);
 }
}
```

Κώδικας 14 Έλεγχος στάθμης δεξαμενής 2

Ο έλεγχος της στάθμης όπως και της τάσης της μπαταρίας γίνεται κάθε ένα λεπτό χωρίς να σταματάμε όμως τον επεξεργαστή γ αυτό χρησιμοποιούμε τεχνικές για real time έλεγχο και συγχρονισμό .

```
void loop() {
  esp.Process();
  if (connected && (millis()-last) > 60000)  {
    ligoNero = digitalRead(KATWflot);
    polyNero = digitalRead(PANWflot);
    T_battery();
    Tmsg(ligoNero , polyNero ,msg1);
    last = millis();
  }
}
```

Κώδικας 15 Ρουτίνα ελεγχου δεξαμενής 2

Αφού ολοκληρώσαμε την περιγραφή της δεξαμενής 2 προχωρήσαμε στην περιγραφή της λειτουργίας της δεξαμενής 1 .

Σε αυτή την δεξαμενή ελέγχουμε μόνο την στάθμη αλλά αυτή είναι παρεμβατική στο αν θα κάνει εκκίνηση της αντλίας της . Αν η στάθμη είναι κάτω από την χαμηλή πρέπει να σταλεί εντολή στο πηγάδι να σταλεί νερό αλλά και να σταματήσει την αντλία για να μην έχουμε πρόβλημα ξηράς λειτουργίας που είναι καταστροφική για τις πολυβάθμιες αντλίες.

```
void Mpump(char *fup , char *fdown , int katw){
  if(fup != NULL  && katw == 1) {
    digitalWrite(RELAY,HIGH);
    rele = true;
  }
  else if(fdown != NULL || katw == 0) {
    digitalWrite(RELAY,LOW);
    rele = false;}}
```

Κώδικας 16 Έλεγχος της στάθμης για την εκκίνηση τις αντλίας δεξαμενής 1

```
int apotelesmaFloter(int panw ,int katw){
  if (katw == 0){
    mqtt.publish("/pumps/","B_start");
    return 1;
  }
  else if(panw == 1 ){
    mqtt.publish("/pumps/","B_stop");
    return 2;
  }
  else{
    return 0;
  }
}
```

Κώδικας 17 Ενημέρωση εκκίνησης αντλίας πηγαδιού με τα μηνύματα B_start και B_stop

Για να μπορούμε να γνωρίζουμε την κατάσταση λειτουργίας της αντλίας κάθε στιγμή πρέπει να υπάρχει ενημέρωση κατάστασης . Έτσι κάναμε τον κώδικα της συνάρτησης που ακολουθεί .

```
int state(bool rele,char *msg1,char *msg2,int prev){
 if (prev != 0){
  return 0;
 }
 else if (rele){
  mqtt.publish("/pumps/",msg1);
 }
 else if(!rele){
  mqtt.publish("/pumps/",msg2);
 }
}
```

Κώδικας 18 State της δεξαμενής 1

Αυτή είναι μια συνάρτηση που λειτουργεί και ενημερώνεται συνεχεία μέσω της loop. Στην loop προσθέσαμε και μια ακόμα παράμετρο . Σε περίπτωση που χαθεί το δίκτυο τότε η αντλία να κλείνει μέχρι να επιστρέψει το δίκτυο . Ο έλεγχος αυτός γίνεται reset κάθε 2 λεπτά.

```
void loop() {
 esp.Process();
 if (connected && (millis()-last) > 60000 && !alrt)  {
  Mpump(up , down ,digitalRead(KATWflot));
  miza = apotelesmaFloter(digitalRead(PANWflot),digitalRead(KATWflot));
  state(rele,msg1,msg2,miza);
  last = millis();
 }
 else if(!connected || alrt || (millis()-Wdog) > 180000){
  digitalWrite(RELAY,LOW);
  rele = false;
 }
}
```

Κώδικας 19 έλεγχος σε επανάληψη για ύπαρξη δικτύου και κατάσταση στάθμης δεξαμενής 1

Για το πηγάδι έχουμε άπλα την λήψη των μηνυμάτων για να γίνει η εκκίνηση η σβέση της αντλίας . Θα μπορούσαμε να προσθέσουμε έλεγχο στάθμης και στο πηγάδι αλλά λόγο μεγάλου βάθους δεν ήταν μέσα στις ανάγκες . Έτσι ο κώδικας μας περιορίζεται στην ανάγνωση μηνυμάτων από το topic /pumps/.

```
void Bpump(char *fdata){
  char *down = strstr(fdata,"stop");
  char *up = strstr(fdata,"start");
  if(up != NULL){
    digitalWrite(RELAY,HIGH);
    rele = true;
  }
  if(down != NULL){
    digitalWrite(RELAY,LOW);
    rele = false;
  }
}
```

Κώδικας 20 έλεγχος μηνυμάτων για την εκκίνηση της αντλίας πηγαδιού

Και εδώ μας ενδιαφέρει αν υπάρχει δίκτυο και γ αυτό ελέγχουμε την λειτουργία του δικτύου και ανάλογα κρατάμε ενεργή την λειτουργία. σε περίπτωση απώλειας σύνδεσης με τον broker η με την δεξαμενή 1 τότε το σύστημα μπαίνει σε κατάσταση αναμονής.

```
void mqttData(void* response) {
  ELClientResponse *res = (ELClientResponse *)response;
  String topicS = res->popString();
  String dataS = res->popString();
  topicS.toCharArray(topic,16);
  dataS.toCharArray(data,20);
    if (data[0] == 'B'){
    Wdog = millis();
    Bpump(data);
  }
  if (strcmp(data,"M_ON") == 0 || strcmp(data,"M_OFF") == 0) {
    Wdog = millis();
  }
}
```

Κώδικας 21 έλεγχος εισερχομένων μηνυμάτων αν είναι Β ο πρώτος χαρακτήρας τότε αφορούν το πηγάδι. Επίσης ελέγχει αν υπάρχει μήνυμα από την δεξαμενή 1 για την κατάσταση της .

Έτσι στην loop ελέγχουμε αν υπάρχει επικοινωνία με τον Broker η με την δεξαμενή 1.

```
void loop() {
  esp.Process();
  if (connected && (millis()-last) > 60000) {
    state(rele,msg1,msg2,0);
    last = millis();
  }
  if(!connected || (millis() - Wdog) > 180000){
    digitalWrite(RELAY,LOW);
    rele = false;}}
```

Κώδικας 22 Έλεγχος Λειτουργίας και αποκατάστασης δικτύου

Σαν πρόσθετη λειτουργία έχουμε ακόμα δυο εντολές που ισχύουν για όλες τις δεξαμενές και αντλίες ,τις ALERT και SAFE . Αυτές οι εντολές υπάρχουν για να κλείνουν την λειτουργία των αντλιών άμεσα και χωρίς καθυστέρηση σε περίπτωση προβλήματος , από τον χρηστή .

```
void alrtCheck(char *fdata){
  char *fmsg = strstr(fdata,"ALERT");
  char *Smsg = strstr(fdata,"SAFE");
  if(fmsg != NULL){
    mqtt.publish("/pumps/","B_stop");
    alrt = true;
  }
  else if(Smsg != NULL){
    alrt = false;
  }
}
```

Κώδικας 23 Έλεγχος Για μήνυμα Παύσης Λειτουργίας

Έτσι ολοκληρώσαμε την περιγραφή του κώδικα και της λειτουργίας του IOT δικτύου μας μέσω Mqtt broker.

Πινάκας εντολών και μηνυμάτων.

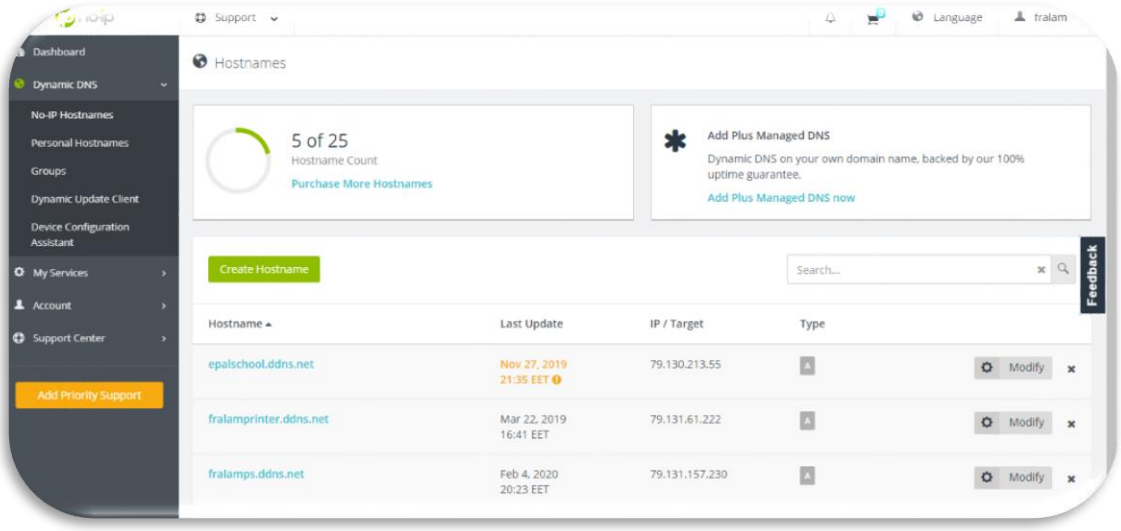
ΠΕΡΙΟΧΗ	ΤΥΠΟΣ ΜΗΝΥΜΑΤΟΣ	Εντολη
Δεξαμενή 1	Ενημέρωση λειτουργίας	M_ON
Δεξαμενή 1	Ενημέρωση αναμονής	M_OFF
Δεξαμενή 1	Εντολή σβέσης πηγαδιού	B_stop
Δεξαμενή 1	Εντολή εκκίνησης πηγαδιού	B_start
Δεξαμενή 2	Εκκίνηση Δεξαμενή 1	M_start
Δεξαμενή 2	Σβέση Δεξαμενή 1	M_stop
Πηγάδι	Ενημέρωση λειτουργίας	B_ON
Πηγάδι	Ενημέρωση αναμονής	B_ OFF
Γενικές εντολές	Τερματισμός λειτουργίας	ALERT
Γενικές εντολές	Κανονική λειτουργία	SAFE

Πίνακας 2 Πρωτόκολλο μηνυμάτων

Δίκτυο Broker και η κατασκευή του.

Για την λειτουργία της επικοινωνίας είναι απαραίτητη η λειτουργία και ενός διακομιστεί για την μεταφορά και αποστολή των μηνυμάτων. Συμφώνα με την αρχιτεκτονική του MQTT χρειαζόμαστε ένα σταθερό σημείο πρόσβασης με γνωστή

θέση στο διαδίκτυο για να αναλάβει τον ρολό αυτό . Υπάρχουν πολλές λύσεις που μπορούν να μας εξυπηρετήσουν δωρεάν αλλά δεν έχουν ούτε ασφάλεια ούτε ρύθμιση χρηστών . Αυτές οι υπηρεσίες είναι με προπληρωμή συνδρομητικά . Έτσι το κόστος μπορεί να αυξηθεί αρκετά , γ αυτό επιλέξαμε την κατασκευή δικού μας Broker για να εξυπηρετήσουμε τις ανάγκες μας . Η επιλογή μας είναι ένα σταθερό υπολογιστικό σύστημα που θα έχει συνεχή λειτουργία και θα είναι συνδεμένο στο δίκτυο του σπιτιού μας . Το μόνο κόστος είναι ένας δυναμικός κατανεμητής ονομάτων δικτύου ,DDNS , στην περίπτωση μας είναι η εταιρία NOIP που μας παρέχει την υπηρεσία ονόματος δικτύου ώστε να έχουμε σταθερή θέση στο διαδίκτυο .



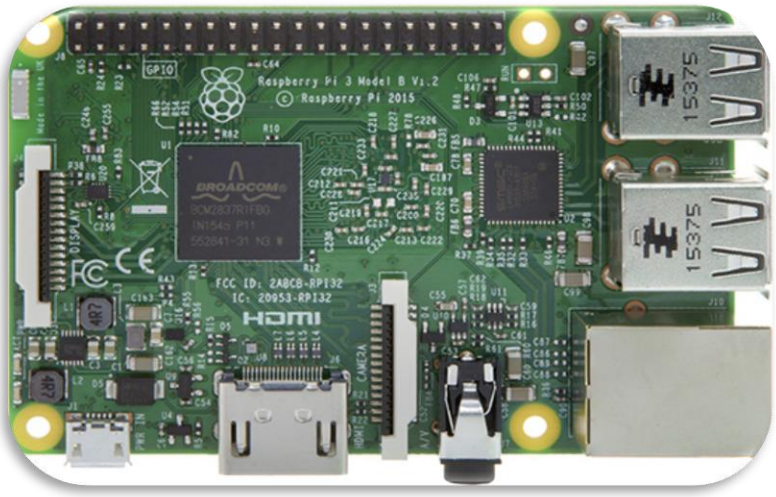
Εικόνα 25 Το interface του NoIP

Για να λειτουργήσει χρειάζεται να στηθεί σε υπολογιστή με λειτουργικό σύστημα συμβατό με δικτυακή λειτουργία . Επιλεγούμε την διανομή ubuntu server 16.04 LTS καθώς μας παρέχει υποστήριξη και έχει αποδεδειγμένη ασφάλεια , μικρές απαιτήσεις σε υλικό αρά και χαμηλές απαιτήσεις σε ενεργεία . Σαν υπολογιστική πλατφόρμα επιλεγούμε το Raspberry pi που είναι μια υπολογιστική πλατφόρμα χαμηλής κατανάλωσης ανεργίας και άψιλης αξιοπιστίας. Χρησιμοποιείται σε πολλές εφαρμογές που δεν χρειαζόμαστε υψηλή επεξεργαστική ισχύ καθώς έχει μικρό κόστος λειτουργίας και αγοράς, κάνοντας το ιδανικό για τέτοιες εφαρμογές.

Για την εγκατάσταση του λειτουργικού ακλουθήσαμε τις οδηγίες που παρέχονται στον ισότοπο της εταιρίας , και αφού ολοκληρώθηκε αρχίσαμε την παραμετροποίηση.

Στο σύστημα αυτό θα προσθέσουμε τον Broker αλλά και την πλατφόρμα , προγραμματισμού ροής, Node-Red . Ετσι καταναλώνοντας ελάχιστη ενεργεία και με

μικρό κόστος θα έχουμε όλες τις υπηρεσίες μας σε μια φτηνή και αξιόπιστη συσκευή που μπορούμε εύκολα να αντικαταστήσουμε σε περίπτωση βλάβης .



Εικόνα 26 Υπολογιστική πλατφόρμα Raspberry pi 3 στην όποια εγκαταστήσαμε τις υπηρεσίες μας .

Αφού εγκαταστήσαμε το λειτουργικό άρχισε και η εγκατάσταση των υπηρεσιών στην πλατφόρμα με πρώτη την υπηρεσία του Mqtt.

Για την εγκατάσταση χρησιμοποιήσαμε τον packet manager του λειτουργικού ubuntu που είναι ο *apt-get* . Για να γίνει η εγκατάσταση πρέπει να έχουμε δικαιώματα διαχειριστή στο σύστημα και γ αυτό εκτελούμε την εντολή *sudo.*

```
sudo apt update
sudo apt install mosquitto
```

Κώδικας 24 Εγκατάσταση Broker

Ο Mosquitto είναι ένας Opensource απλός Broker εύκολος στην παραμετροποίηση και στην εγκατάσταση, παρέχει ασφάλεια τύπου TLS , SSL και ρυθμίζεται εύκολα μέσω αρχείου παραμετροποίησης. Πηγαίνοντας στον φάκελο /etc/mosquitto θα βρούμε το αρχείο

```
mosquitto.conf
```

είναι και το αρχείο που θα παραμετροποιησουμέ ώστε να προσθέσουμε ασφάλεια στο δίκτυο μας.

Ανοίγοντας το με τον nano editor βλέπουμε το αρχείο έτοιμο προς επεξεργασία.

```
sudo nano mosquitto.conf //εντολη επεξεργασιας του αρχειου
```

Κώδικας 25 Επεξεργασία Ρυθμίσεων Broker

Για να προσθέσουμε ένα χρηστή και κωδικό πρόσβασης στην υπηρεσία εκτελούμε την παρακάτω εντολή τερματικού.

```
sudo mosquitto_passwd -c passwordfile <ονομα χρηστη>
```
Κώδικας 26 Προσθήκη κωδικού και χρηστή

Αυτή η εντολή θα δημιουργήσει ένα αρχείο με όνομα passwordfile στην θέση που βρισκόμαστε, το οποίο το αναγνωρίζει ο Broker μας. Τώρα πρέπει να προσθέσουμε την θέση του στο αρχείο παραμετροποίησης με την εντολή που είδαμε παραπάνω . Επίσης πρέπει να αφαιρέσουμε το δικαίωμα ανώνυμης πρόσβασης στην υπηρεσία .

Για να γίνουν όλα αυτά πρέπει να βάλουμε τις παρακάτω παραμέτρους στο αρχείο mosquitto.conf.

```
allow_anonymous false
password_file <πληρη θέση αρχείου>
```

Τώρα για να συνδεθούμε θα πρέπει να βάλουμε τον κωδικό μας και το όνομα χρηστή . Μπορούμε να φτιάξουμε πολλούς διαφορετικούς χρηστές όπως και να έχουμε πολλούς listener στο ίδιο τερματικό ώστε να μπορούμε να εξυπηρετήσουμε πολλούς διαφορετικούς αυτοματισμούς χωρίς να επηρεάζονται μεταξύ τους.

Αφού ολοκληρώσουμε την παραμετροποίηση πρέπει να κάνουμε επανεκκίνηση της υπηρεσίας με τις εντολές.

```
ps -aux | grep mosquitto <εμφανιση ID διεργασίας για το mosquitto>
kill -HUP <ID διεργασίας>
sudo mosquitto <επανεκκίνηση υπηρεσίας>
```

Επίσης και μια επανεκκίνηση του λειτουργικού είναι αρκετή ώστε να εφαρμοστούν οι αλλαγές.

Τι είναι το Node red

Το Node-RED είναι ένα εργαλείο προγραμματισμού βασισμένο σε flow programming, το οποίο αναπτύχθηκε αρχικά από την ομάδα της IBM (Emerging Technology Services) και τώρα αποτελεί μέρος του JS Foundation.

Η τεχνικη του Flow Programming δεν ειναι κενουργια καθως υπαρχει πολλα χρονια. Εχει επινοηθεί από τον J. Paul Morrison στη δεκαετία του 1970, ο προγραμματισμός βάσει ροής (Flow Programming) είναι ένας τρόπος περιγραφής της συμπεριφοράς μιας εφαρμογής ως δικτύου μαύρων κουτιών ή "κόμβων"(nodes) όπως ονομάζονται στο Node-RED. Κάθε κόμβος έχει έναν καλά καθορισμένο σκοπό. Του δίδωνται κάποια δεδομένα, κάνει κάτι με αυτά τα δεδομένα και στη συνέχεια μεταδίδει αυτά τα δεδομένα. Το δίκτυο των κομβων είναι υπεύθυνο για τη ροή δεδομένων μεταξύ των κόμβων.

Είναι ένα μοντέλο προγραμματισμου που οδηγει σε μια οπτική αναπαράσταση και το καθιστά πιο προσιτό σε ένα ευρύτερο φάσμα χρηστών. Εάν κάποιος μπορεί να σπάσει ένα πρόβλημα σε ξεχωριστά βήματα, μπορεί να δει μια ροή στο Node Red και να πάρει μια αίσθηση του τι κάνει ,χωρίς να χρειάζεται να κατανοήσει τις επιμέρους γραμμές κώδικα μέσα σε κάθε κόμβο, και αν καποιες φορες χρειαστει να ελεγξει και τον κωδικα μεσα στους κομβους να μπορει να το κανει ευκολα και αμεσα.

Το Node-RED είναι ένα εργαλείο προγραμματισμού που χρησημοποιειται για τη διασύνδεση συσκευών υλικού, APIs και online υπηρεσιών με νέους και ενδιαφέροντες τρόπους, που προσφερουν ευελιξια ταχυτητα και ευκολια.

Παρέχει ένα περιβάλον που βασίζεται σε πρόγραμμα περιήγησης, το οποίο διευκολύνει τη σύνδεση των ροών χρησιμοποιώντας τους κόμβους της παλέτας που μπορούν να αλλαξουν και να βελτιοθουν κατα το χρόνο εκτέλεσης.

Έτσι το Node Red που ξεκίνησε σαν ένα πρόγραμμα παρουσίασης και έλεγχου μηνυμάτων για MQTT δίκτυα έγινε μια νέα πλατφόρμα στην όποια μπορούμε να αναπτύξουμε ολοκληρωμένες εφαρμογές Intrenet Of Things. Είναι ανοικτού κώδικα και έχει μεγάλη συμμετοχή στην ανάπτυξη του. Επίσης εταιρίες κατασκευάζουν εργαλεία και κόμβους για να υποστηρίξουν τα προϊόντα τους . Περά από εφαρμογές για internet of Things χρησιμοποιείται και για εξόρυξη δεδομένων από τον κυβερνοχώρο (data mining).

Εγκατάσταση και παραμετροποίηση Node red

Α node-red είναι ένα Module της γλώσσας δικτυακού προγραμματισμού Node.JS .Η Node.js χρησιμοποιείται για την κατασκευή δυναμικών σελίδων και υπηρεσιών και βασίζεται στην javascript .

Για την εγκατάσταση του node red σαν υπηρεσία σε ένα σύστημα , απαιτεί την εγκατάσταση της γλώσσας Node.JS , συμφώνα με τον οδηγό εγκατάστασης της γλώσσας εκτελούμε πρώτα τις παρακάτω εντολές.

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install nodejs
sudo apt-get install npm
```

Κώδικας 27 Εγκατάσταση Node.JS

Αφού ολοκληρωθεί η εγκατάσταση της γλώσσας εγκαθιστούμε και το πακέτο του Node Red .

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

Κώδικας 28 Εγκατάσταση Node-Red

Το node red λειτουργεί σαν υπηρεσία στο λειτουργικό μας στην πόρτα 1880. Η πόρτα αυτή δεν είναι ανοικτή από το firewall του λειτουργικού και γ αυτό πρέπει να την ανοίξουμε χειροκίνητα με την παρακάτω εντολή.

```
sudo ufw allow 1880
```

Κώδικας 29 Ρύθμιση Firewall

Έτσι για να γίνει εκκίνηση της υπηρεσίας πρέπει άπλα να ζητήσουμε την υπηρεσία με την εντολή node-red.

Η υπηρεσία αυτή όμως δεν θα έχει αυτόματη εκκίνηση σε περίπτωση επανεκκίνησης του λειτουργικού και αυτό δεν είναι κάλο για την λειτουργία του συστήματος μας . Πρέπει να προσθέσουμε το node red στις υπηρεσίες του συστήματος . Για να το κάνουμε αυτό δίνουμε τις παρακάτω εντολές . Σε αυτό θα μας βοηθήσει το εργαλείο για την τακτοποίηση των υπηρεσιών PM2.

```
sudo npm install -g pm2
which node-red  // αυτή η εντολή μας επιστρέφει το path  του node red
pm2 start  <noderedpath>  -- -v
pm2 save
pm2 startup
```

Κώδικας 30 Προσθήκη του Node- Red στις υπηρεσίες

Όταν μια υπηρεσία είναι στο διαδίκτυο πρέπει να την προστατεύουμε με κωδικό αλλιώς είναι εκτεθειμένη σε κακόβουλες επιθέσεις είτε από φυσικά πρόσωπα είτε από αυτομάτους μηχανισμούς που κάνουν αναζήτηση για ανοικτούς ισότοπους .

Για να προστατέψουμε την σελίδα του Node-red πρέπει να ενεργοποιήσουμε τον μηχανισμό έλεγχου χρηστή. Γ αυτόν τον λόγο εγκαταστήσαμε τα εργαλεία διαχείρισης . Εκτελούμε την παρακάτω εντολή για να φτιάξουμε ένα κωδικό με την μέθοδο hash που μας επιτρέπει να έχουμε διπλή κρυπτογράφιση ώστε να μην μπορεί κάποιος να πάρει τον κωδικό ακόμα και αν ανοίξει το αρχείο ρυθμίσεων.

```
node-red-admin hash-pw
```

Κώδικας 31 Εξαγωγή Κωδικού για το Node-Red

Αυτή η εντολή μας παράγει ένα κλειδί το ποιο πρέπει να αποθηκεύσουμε.

```
      @ubuntu:~$ node-red-admin hash-pw
Password:
$2b$08$in4y9l6N8Lkxv6W
      @ubuntu:~$
```

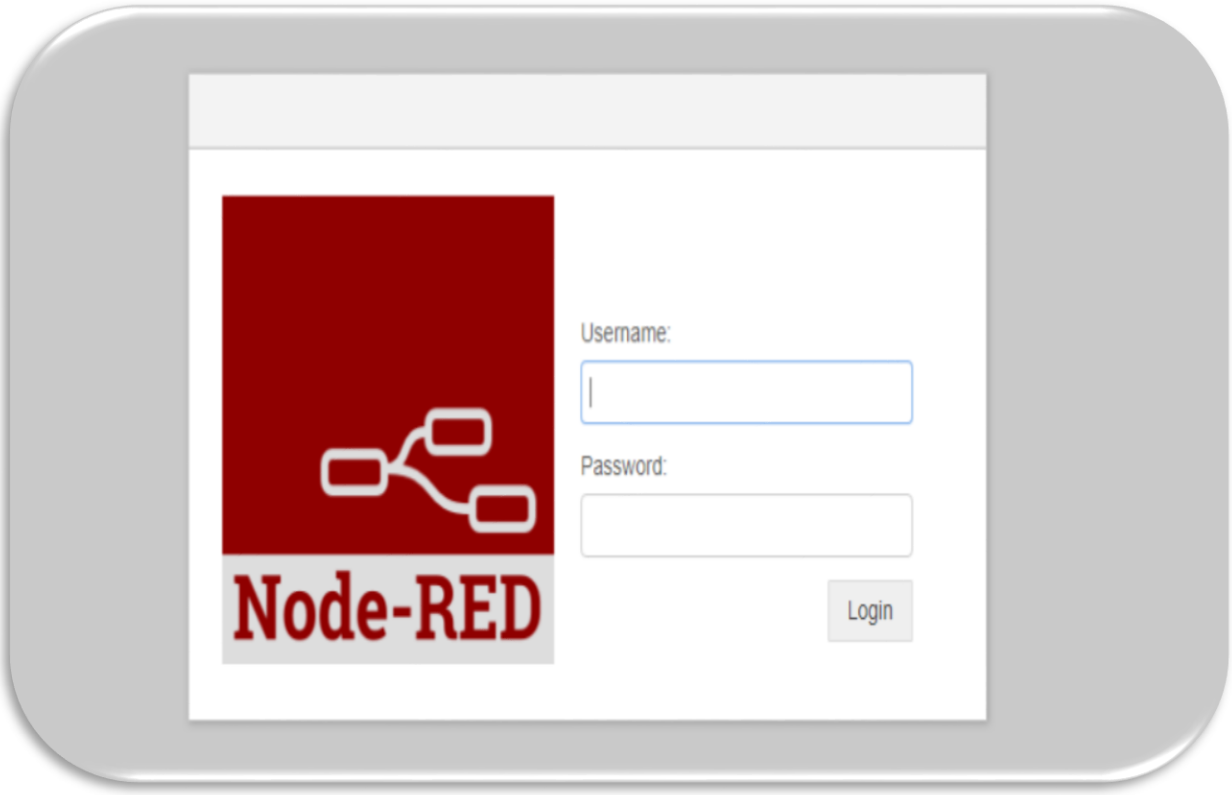
Εικόνα 27 παράγωγη κωδικού για το interface του node red

Αφού φτιάξουμε τον κωδικό πρέπει να τον προσθέσουμε στο αρχείο ρυθμίσεων του node red. Για να το κάνουμε αυτό πρέπει να ενεργοποιήσουμε τον παρακάτω κώδικα από το αρχείο ρυθμίσεων

```
adminAuth: {
    type: "credentials",
    users: [{
        username: "admin",
        password: "$2b$08$in4y9l6N8Lkxv6W" //Εδώ προσθέτουμε τον κωδικό
        permissions: "*"
    }] },
```

Κώδικας 32 Εγκατάσταση Κωδικού

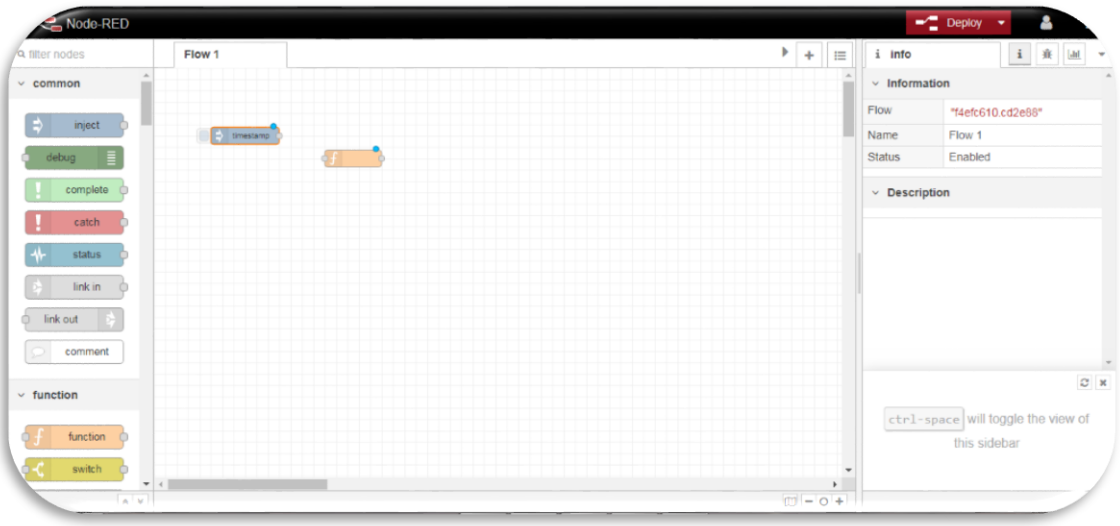
Το σύστημα μας είναι τώρα ασφαλές και μπορούμε να το ανεβάσουμε στο διαδίκτυο με ασφάλεια .



Εικόνα 28 Οθανή εισόδου στο node - red

Εισαγωγή στο Node-Red.

Το Node Red είναι ένα περιβάλλον προγραμματισμού ροής. Αυτό σημαίνει ότι έχουμε μικρά κομμάτια κώδικα τα όποια σχηματίζουν τους κόμβους (nodes). Έτσι ο κάθε κόμβος μεταφέρει μια δομή η όποια έχει το δεσμευμένο όνομα , msg.



Εικόνα 29 Η διαπάλη προγραμματισμού του Node-Red

Ένα msg μπορεί να μεταφέρει στο payload του διαφόρους τύπους μεταβλητών .

```
Boolean - true, false
Number - 0, 123.4
String - "καλησπέρα"
Array - [1,2,3,4]
Object - { "a": 1, "b": 2}
```

Κώδικας 33 Τύποι Δεδομένων που μεταφέρει το msg

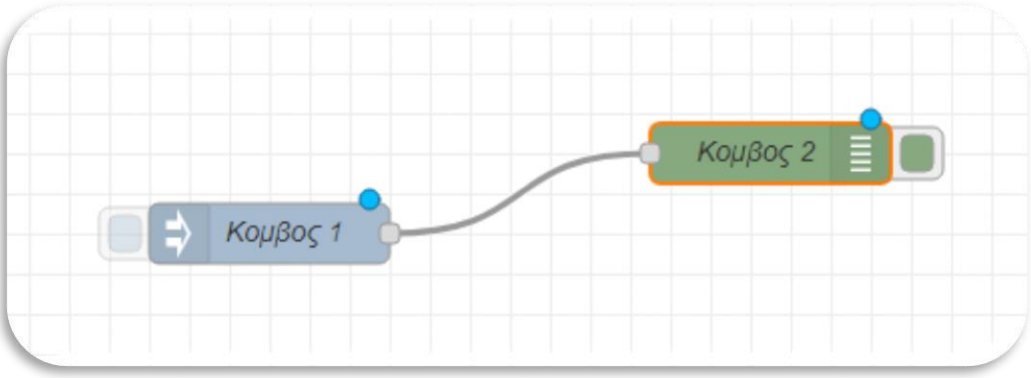
Το αντικείμενο msg δεν περιέχει μόνο το payload σαν περιεχόμενο αλλά έχει και το _msgid και το topic. Επίσης μπορούμε να δώσουμε τις δίκες μας κατηγορίες στο msg όπως για παράδειγμα το όνομα του αισθητήρα με πολύ απλό τρόπο .

```
msg.sensor=sensor;
```

Κώδικας 34 Δημιουργία μεταβλητής στο msg

Έτσι στα επόμενα flow μετά από το σημείο ορισμού έχουμε το πέρασμα τιμών και στο msg.sensor. Αυτό το χαρακτηριστικό μας δίνει τεραστία ευελιξία για να μπορούμε να μεταφέρουμε πολλά δεδομένα ανάμεσα στους κόμβους .

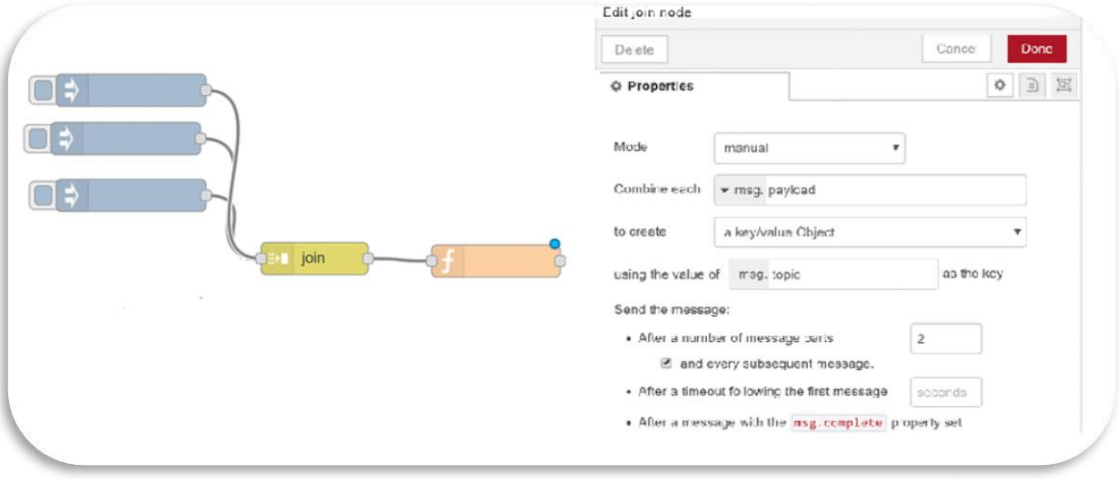
Για να περάσει ένα μήνυμα από τον ένα κόμβο στον άλλο πρέπει οι δυο κομβόι να είναι συνδεμένοι και να υπάρχουν και οι συνθήκες να μεταφερθεί το μήνυμα.



Εικόνα 30 Προβολή σύνδεσης δυο κόμβων για μεταφορά δεδομένων

Οι κομβόι μπορεί να έχουν μόνο εισόδους η μόνο εξόδους η και τα δυο . Στην εικόνα 28 βλέπουμε τον Κόμβο 1 που έχει μόνο έξοδο και τον Κόμβο 2 που έχει μόνο είσοδο.

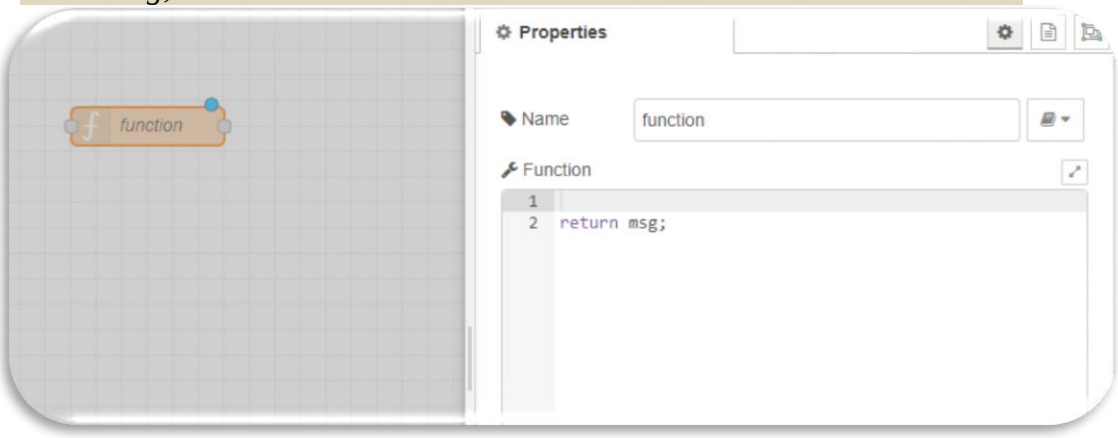
Ένας κόμβος μπορεί να έχει πολλές εξόδους αλλά μια μόνο είσοδο . Αυτό γίνεται γιατί δεν μπορούμε να έχουμε παράλληλες εκτελέσεις στην παρούσα έκδοση καθώς η Node.Js είναι μονονυματική γλωσσά . Με τον καιρό υπάρχει η προοπτική να περάσει σε πολυνυματικές εφαρμογές. Μπορούμε όμως να πάρουμε δυο η και παραπάνω εισόδους με διαχωρισμό των msg.topic και λίγο απλό κώδικα . Αυτό όμως αυξάνει την καθυστέρηση στην επεξεργασία άμα έχουμε να περάσουμε και να επεξεργαστούμε πολλά δεδομένα και δεν προτείνεται σαν μέθοδος . Στην έκδοση 1.0 και μετά προσθετικέ ο κόμβος Join που μπορεί να συνδυάσει δεδομένα βάση φόρμουλας που θα δώσουμε εμείς χωρίς να υπάρχει καθυστέρηση.



Εικόνα 31 Μέθοδος πολλαπλών εισόδων σε κόμβο.

Ένας από τους ποιο χρησίμους τύπους κόμβων είναι ο κόμβος Function. Διαθέτει μια είσοδο και μια έξοδο. Για να επεξεργαστούμε τα δεδομένα του μηνύματος πρέπει πάντα να επεξεργαστούμε το msg. Ο αρχικός κώδικας που βλέπουμε όταν κάνουμε edit τον κόμβο Function είναι ο κώδικας που βλέπουμε παρακάτω.

```
return msg;
```



Εικόνα 32 Function Node

Αυτός ο κώδικας υποδηλώνει ότι πάντα έχουμε να μεταφέρουμε το msg στον επόμενο κόμβο . Σε περίπτωση όμως που έχουμε να μεταφέρουμε σε δυο η και πρέπανε εξόδους γραφούμε την return ως εξής.

```
var msg1 = { payload:"first out of output 1" };
var msg2 = { payload:"second out of output 1" };
var msg3 = { payload:"third out of output 1" };
var msg4 = { payload:"only message from output 2" };
return [ [ msg1, msg2, msg3 ], msg4 ];
```

Κώδικας 35 Διπλή έξοδος από ένα Node

στον παραπάνω κώδικα έχουμε δυο εξόδους , η μια έξοδος μεταφέρει τα msg1,msg2,msg3 σε μορφή πινάκα και η δεύτερη το msg4.

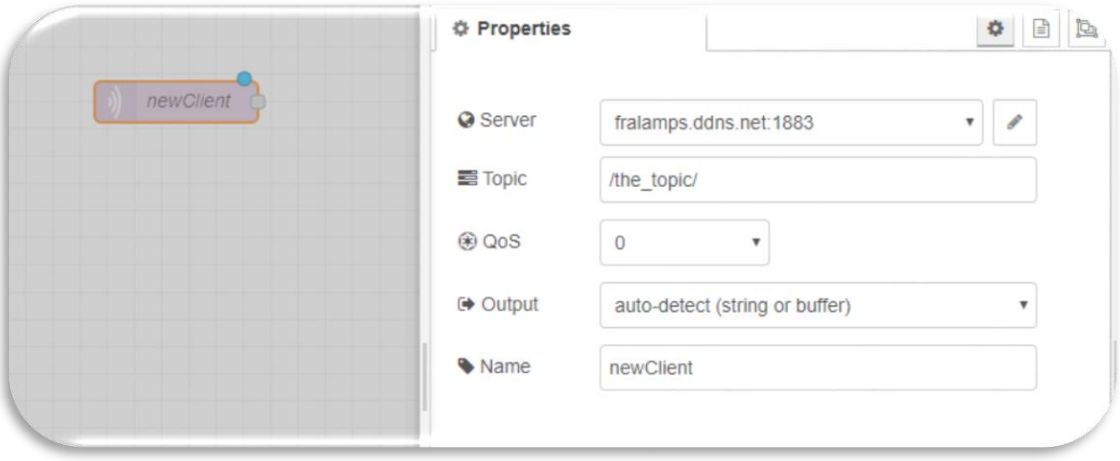
Όταν θέλουμε να μεταφέρουμε ασύγχρονα μηνύματα , δηλαδή όταν θέλουμε να στείλουμε μήνυμα μετά που θα ολοκληρωθεί μια διεργασία σε μια συνάρτηση τότε πρέπει να χρησιμοποιήσουμε την τεχνική που παρουσιάζεται παρακάτω.

```
AsyncWork(msg, function(data) {
   msg.payload = data;
   node.send(msg);
   node.done();
});
return;
```

Κώδικας 36 Ασύγχρονη εξαγωγή δεδομένων

Από ότι βλέπουμε τον ρολό της return τον αναλαμβάνει πλέων η node.send(). Προσοχή θέλει στο ότι αφού στείλουμε το msg , να ολοκληρώσουμε την λειτουργία κλείνοντας την διεργασία αποστολής του κόμβου με την node.done().

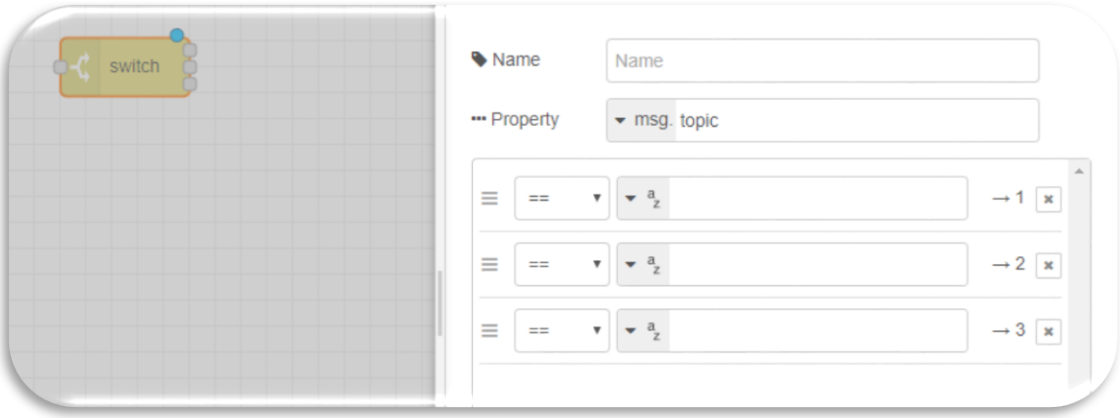
Άλλος ένας σημαντικός τύπος κόμβου που υπάρχει στο Node Red , και που είναι και από τους πρωταρχικούς λογούς κατασκευής του , είναι ο κόμβος του Mqtt. Αυτός ο κόμβος είναι ουσιαστικά ένα μαύρο κουτί προς εμάς καθώς μας παρέχει εύκολη και απλή παραμετροποίηση για να κάνουμε την σύνδεση μας με ένα Mqtt Broker γρήγορα και αποτελεσματικά . Υποστηρίζει όλα τα χαρακτηριστικά της έκδοσης 3.1 και άνω.



Εικόνα 33 Το Mqtt node

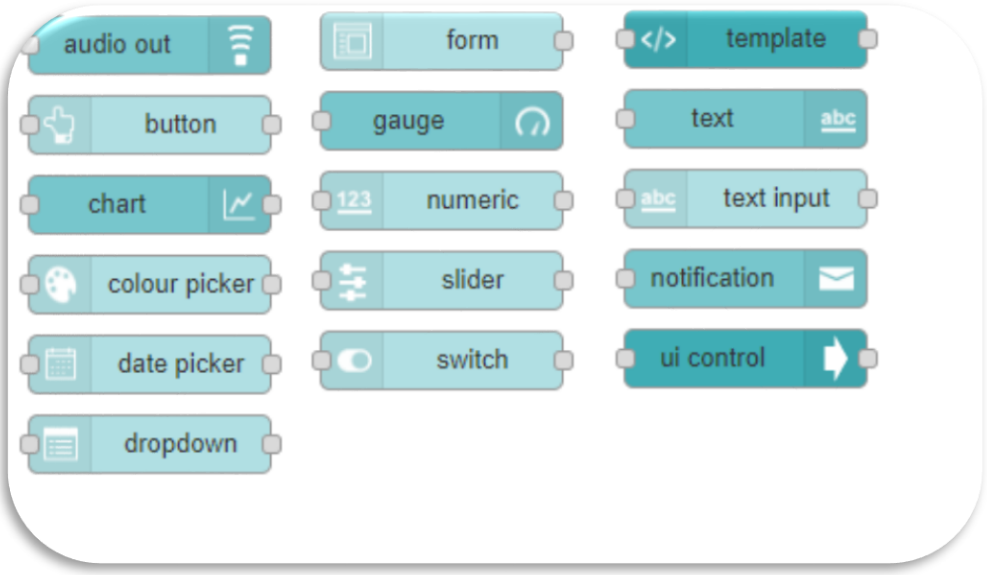
Σε αυτό μπορούμε να ορίσουμε την διεύθυνση του broker , τον κωδικό πρόσβασης όπως και πολλά άλλα στοιχειά που αφορούν την επικοινωνία μας με αυτόν . Επίσης μπορούμε να ορίσουμε τον τύπο των δεδομένων που θα λαμβάνουμε από τον Broker αν είναι String, Buffer η ακόμα και JSON Object που θα δούμε παρακάτω τι είναι ένα αντικείμενο τύπου JSON.

Όταν θέλουμε να δρομολογήσουμε τα δεδομένα μπορούμε να χρησιμοποιήσουμε το switch node . Αυτό το node δουλεύει ακριβώς όπως ένας πολυπλέχτης. Μας δίνει την δυνατότητα να αλλάζουμε ποριά σε ένα msg ανάλογα με το τι μεταφέρει , επιτρέποντας μας να κάνουμε μια αυτόματη επιλογή σεναρίου ανάλογα με το τι μήνυμα έρχεται στον Broker μας και το topic που έχουμε κάνει εγγράφη.



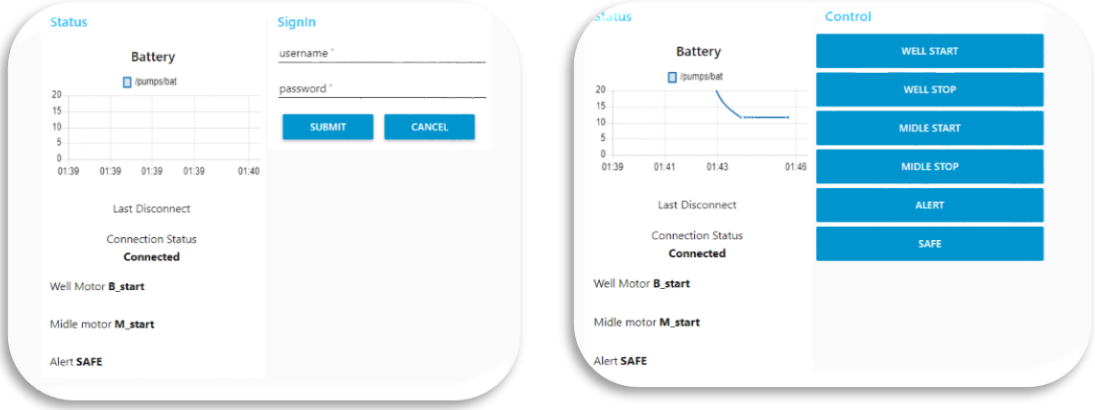
Εικόνα 34 το switch node με 3 επιλογές, μπορούμε να προσθέσουμε όσες θέλουμε

Τέλος άλλο ένα σημαντικό πακέτο από κόμβους είναι το dashboard. Αυτό μας δίνει την δυνατότητα να προσθέτουμε γραφικά στοιχειά και να φτιάξουμε το γραφικό περιβάλλον για την δικτυακή εφαρμογή μας . Παρέχει εργαλεία προβολής δεδομένων όπως γραφήματα, ρολόγια μετρήσεων, κείμενα, ειδοποιήσεις μπάρες επιλογής τιμής , μπάρες προβολής τιμής, κουμπιά, διακόπτες αλλαγής κατάστασης και πολλά αλλά . Μπορούμε επίσης με λίγες γνώσεις html να κατασκευάσουμε τα δικά μας περιβάλλοντα με τα εργαλεία κατασκευής σελίδων .



Εικόνα 35 Οι επιλογές node για το dashboard

Το Dashboard είναι αρκετά απλό στην βασική του λειτουργιά . Επιλεγούμε τα στοιχειά που θέλουμε να προβληθούν και άπλα τα συνδέουμε με τα αντίστοιχα node που θέλουμε για να πάρουν τα δεδομένα που χρειάζεται . Όπως βλέπουμε στην εικόνα 32 έχουν διάφορα είδη αλλά με εισόδους μόνο και αλλά με είσοδο και έξοδο .



Εικόνα 36 Dashboard για την εφαρμογή μας με προστασία των λειτουργιών με κωδικό

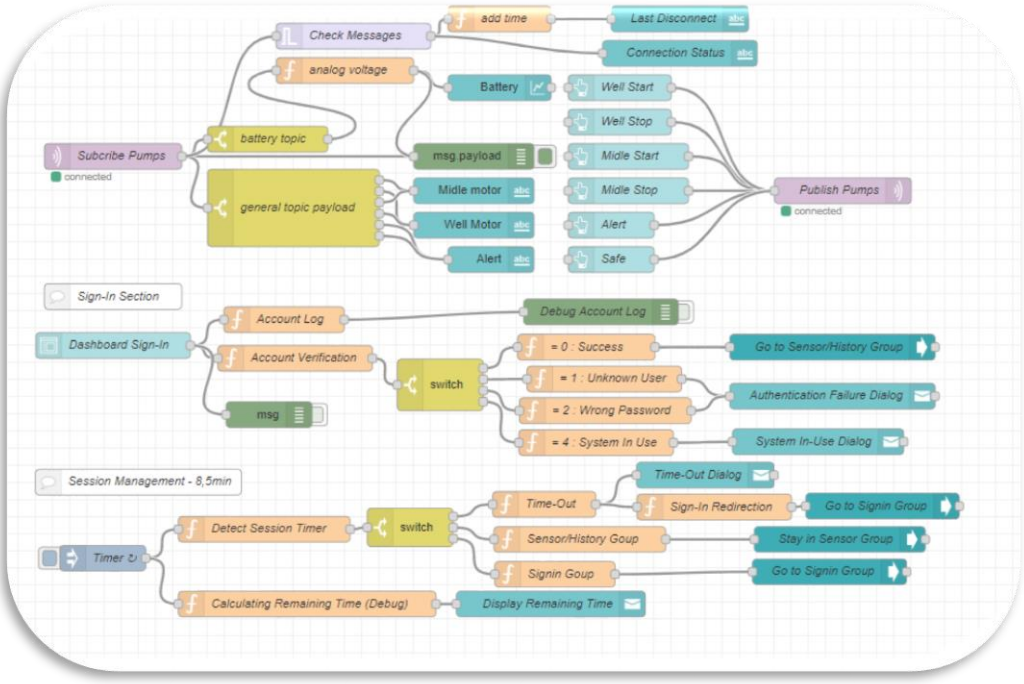
Με το Dashboard κλείνουμε το θέμα με τα χρήσιμα nodes. Φυσικά μπορούμε να προσθέσουμε όσα θέλουμε η και να δημιουργήσουμε και δικά μας ακλουθώντας τις οδηγίες που παρέχονται στο gitHub κανάλι του node Red.

Έτσι από ότι είδαμε έχουμε ένα πολύ δυνατό εργαλείο που μπορεί να μας λύσει τα χεριά όσον αφόρα την δημιουργία επαφής για την εποπτεία της λειτουργίας των αντλιών που έχουμε στο δίκτυο νερού μας . Εύκολα και με κατανοητό τρόπο μπορούμε να δημιουργήσουμε και σενάρια λειτουργίας που θα μπορεί να ελέγχει το Node Red αυτόματα χωρίς εμείς να χρειάζεται να είμαστε σε συνεχή εποπτεία.

Γνωρίζοντας τώρα τον τρόπο λειτουργίας τον Node Red μπορούμε να αναλύσουμε και το πρόγραμμα ροής που κατασκευάσαμε για το δικό μας σενάριο λειτουργιάς.

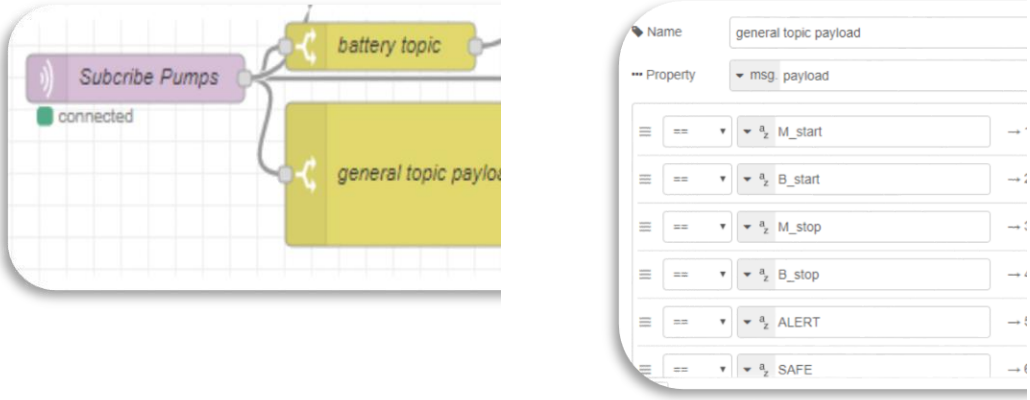
Κατασκευή επαφής στο Node Red , Pump Control

Αφού αναλύσαμε τις βασικές λειτουργιές του Node Red θα δούμε μια πραγματική εφαρμογή με το dashboard και πως μπορούμε να χρησιμοποιήσουμε το Node Red για να επεξεργαστούμε δεδομένα .



Εικόνα 37 ολόκληρο το σχέδιο ροής για την λειτουργία και τον έλεγχο των αντλιών.

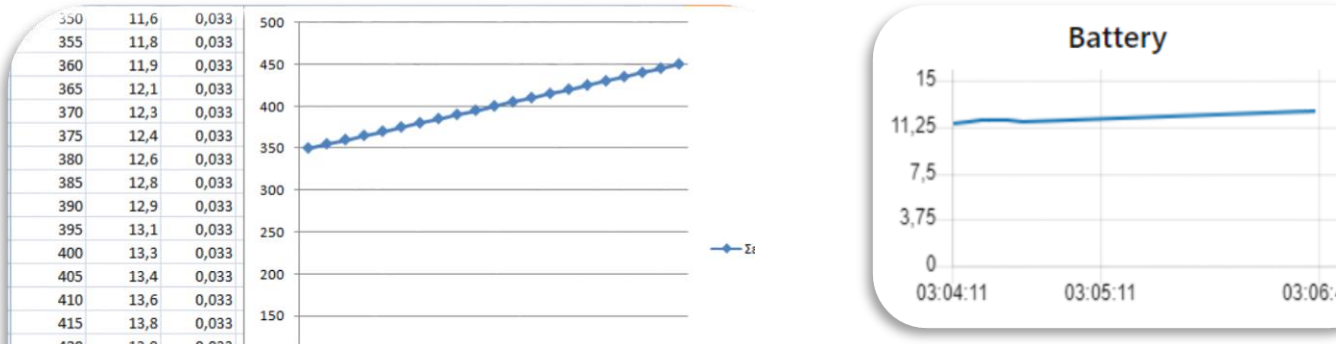
Όπως βλέπουμε και στην εικόνα 36 το σχέδιο ροής είναι σχετικά απλό αν σκεφτούμε ότι έχουμε πλέων μια ολοκληρωμένη δικτυακή εφαρμογή με την όποια μπορούμε να ελέγχουμε δυναμικά τα αντλιοστάσια μας.



Εικόνα 38 Διαχωρισμός λειτουργιών ανάλογα με το topic και το payload

Στην εικόνα 37 έχουμε την είσοδο από τον Broker. Έχουμε κάνει Sub στο topic /pumps/# και λαμβάνουμε όλα τα μηνύματα που λαμβάνει αυτό το κανάλι στον Broker μας .Με το switch node αλλάζουμε έξοδο ανάλογα με το τι θέλουμε να κάνει το πρόγραμμα μας όταν έρθει ένα μήνυμα στο /pumps/ , κάθε εντολή έχουμε και μια διαφορετική έξοδο .

Βλέπουμε και χωριστά το battery topic που είναι ουσιαστικά το /pumps/bat στο οποίο έρχεται η τιμή της αναλογικής εισόδου του MqMax που βρίσκετε στην τελική δεξαμενή που δουλεύει αυτόνομα από ρεύμα . Έχει ένα διαιρετή τάσης και ελέγχει την τιμή της τάσης της μπαταρίας . Ο διαιρετής τάσης συμπεριφέρεται γραμμικά και έτσι είναι εύκολο να πάρουμε την τιμή αυτή και να βρούμε την τιμή της τάσης της μπαταρίας χωρίς να κάνουμε άλλο υπολογισμό στο πρόγραμμα του μικροελεγκτή. Περνώντας δυο τιμές από την αναλογική είσοδο και κάνοντας δυο μετρήσεις την ιδία στιγμή με ένα πολυμερή κάνουμε στην ευθεία για την τάση και βρίσκουμε ότι το α είναι 0.033.



Εικόνα 39 Γράφημα διαιρετή τάσης και η τιμή της μπαταρίας όπως παρουσιάζεται.

Με τον παρακάτω κώδικα έχουμε την σωστή τιμή της τάσης στο διάγραμμα που έχουμε προσθέσει στο Dashboard.

```
var battery=msg.payload;
batteryvoltage=parseInt(battery);
var perc = batteryvoltage * 0.033;
msg.payload = perc.toFixed(2);
return msg;
```

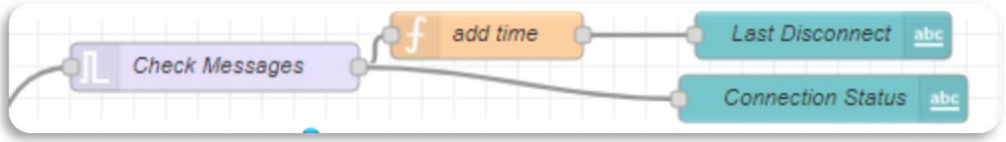
Κώδικας 37 Έλεγχος στάθμης Μπαταρίας , προβολή τάσης

Στην συνεχεία έχουμε την ενημέρωση για την κατάσταση των αντλιών , εάν βρίσκονται σε λειτουργία η αν είναι σε αναμονή.

Εικόνα 40 Για κάθε έξοδο έχουμε και ένα μήνυμα το οποίο μας μεταφέρει την κατάσταση

Με αυτό το πλέγμα κόμβων μπορούμε να γνωρίζουμε την κατάσταση στην όποια βρίσκονται οι αντλίες ανά πασά ώρα.

Πολλές φόρες όμως έχουμε και διακοπές ρεύματος η απώλεια internet στην περιοχή που βρίσκεται το σύστημα μας .Όπως είδαμε στο κεφάλαιο που περιγράφουμε τον κώδικα , γνωρίζουμε ότι για λογούς ασφάλειας το σύστημα σταματήσει να λειτουργεί όταν χάσει το δίκτυο δηλαδή την σύνδεση με τον broker. Σε μια τέτοια περίπτωση πρέπει να γνωρίσουμε πότε έγινε και να έχουμε υπόψη μας ότι μπορεί να μην έχουμε νερό αν δεν επέμβουμε. Για αυτόν το λόγο φτιάξαμε το παρακάτω πρόγραμμα ροής που μέτριοι την ώρα που έφτασε το τελευταίο μήνυμα στον broker μας . Σε περίπτωση που σταματήσουν να έρχονται μηνύματα μας ενημερώνει ότι η σύνδεση χάθηκε.



Εικόνα 41 Έλεγχος λειτουργίας δικτύου.

Στον κόμβο *Check Messages* έχουμε ένα timer που κάθε φόρα που έρχεται ένα μήνυμα μηδενίζει την λειτουργία του . Σε περίπτωση που κάνει overflow στέλνει ένα μήνυμα ότι χάθηκε η σύνδεση . Αυτό με την σειρά του παίρνει μια χρονική στάμπα και προβάλλεται στο dashboard μας στην περιοχή ενημέρωσης. Ο Κώδικας για την λειτουργία αυτή είναι μέσα στον κόμβο *add time* και παρουσιάζεται παρακάτω.

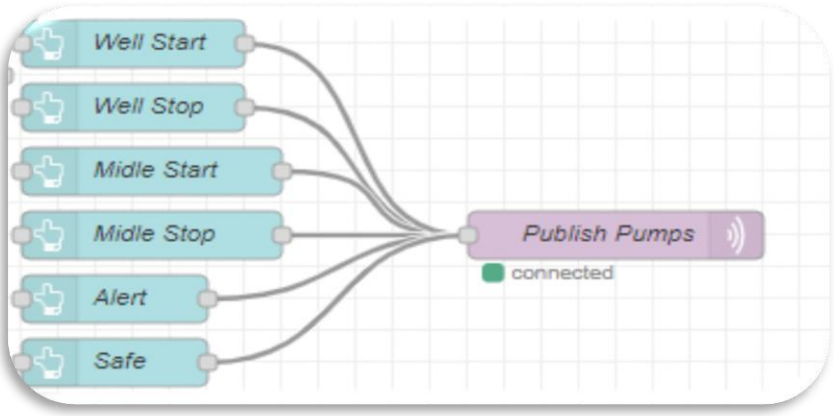
```
if (msg.payload ==="Lost Connection"){
   var time= Date();
   msg.time = time;
}
return msg;
```

Κώδικας 38 Έλεγχος Απώλειας δικτύου

Όπως μπορούμε να παρατηρήσουμε στον κόμβο *Last Disconnect* δεν μεταφέρουμε μόνο το payload που περιέχει το μήνυμα Lost Connection αλλά μεταφέρουμε και την παράμετρο time που περιέχει την ημερομηνία που έγινε το συμβάν . Έτσι μεταφέρουμε δυο δεδομένα χωρίς καθυστέρηση.

Περά από το να ενημερωθούμε μπορούμε και να παρακάμψουμε την λειτουργία των αντλιών με το να έχουμε έλεγχο στις εντολές που λαμβάνουν . Φυσικά ο παρεμβατικός ρόλος δεν μπορεί να παρακάμψει τις λειτουργιές ασφάλειας που έχουμε ορίσει με αποτέλεσμα να μπορούμε να ελέγχουμε και εμείς με ασφάλεια την λειτουργία των αντλιών καθώς γνωρίζουμε ότι αν κάτι πάει στραβά το σύστημα αυτόματα θα σταματήσει η θα ανατρέψει την εντολή μας.

Παρόλα αυτά αρκετές φόρες χρειάζεται να παρέμβουμε στην λειτουργία είτε για να κλείσουμε είτε για να λειτουργήσουμε για λίγο για λογούς συντήρησης. Έτσι δημιουργήσαμε ένα σύνολο από εντολές που στέλνουμε στον Broker μας και εξομοιώνει τις εντολές που στέλνουν τα MqMax μεταξύ τους .



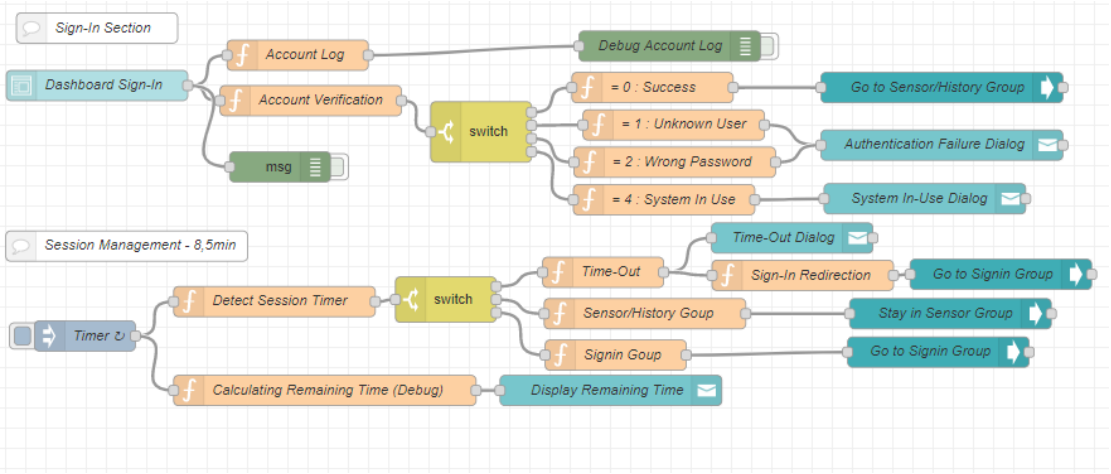
Εικόνα 42 το δικτύωμα ροής για τις εντολές προς τις αντλίες

Φυσικά εδώ γενάτε ένα ερώτημα . Τι θα γίνει αν κάποιος μη εξουσιοδοτημένος από εμάς, προσπαθήσει να πάρει τον έλεγχο των αντλιών . Αν γνωρίζει την διεύθυνση που βρίσκετε η εφαρμογή μας είναι πολύ εύκολο να πάρει τον έλεγχο αφού το Dashboard δεν παρέχει την δυνατότητα να ορίσουμε λογαριασμό χρηστών . Αυτό είναι ένα πολύ σοβαρό θέμα ασφάλειας που πρέπει να αντιμετωπίσουμε .

Γνωρίζοντας τα βασικά από έλεγχο λογαριασμού για χρηστές μπορούμε να φτιάξουμε το δικό μας σύστημα εισόδου και να κρύψουμε τις λειτουργιές που δεν θέλουμε να χρησιμοποιούν μη εξουσιοδοτημένοι χρηστές. Αυτό γίνεται με ασφάλεια καθώς ότι κώδικα φτιάξουμε αυτός τρέχει στον server μας και όχι στον χρηστή όπως γίνεται σε άλλες πλατφόρμες . Επίσης δεν έχουμε Global τιμές οι οποίες μπορεί να

προδώσουν την λειτουργία του συστήματος . Η Node.JS είναι αρκετά δυνατή και ασφαλής γλωσσά για να κάνουμε δικτυακές υπηρεσίες και μπορούμε να ωφεληθούμε από αυτό

.



Εικόνα 43 Το Δικτύωμα ροής που λύνει το πρόβλημα της πιστοποίησης χρηστή.

Στην εικόνα 43 βλέπουμε το πρόγραμμα ροής για την πιστοποίηση με λογαριασμό χρηστή. Παρατηρώντας το Sign-in Section θα δούμε ότι πρώτα φτιάχνουμε μια φόρμα με δυο Text Field με τα εργαλεία του Dashboard τα στοιχεία που θα δώσουμε πέρανε σε δυο συναρτήσεις, τις Account Log και Account Verification. Ας δούμε τον κώδικα της Account Verification.

```
var accounts =  flow.get("accounts") || [ { username : "admin", password :
"admin"},{ username : "guest", password : "guest"}] ;
var username = msg.payload.username ;
var password = msg.payload.password ;
msg.payload = 1;
accounts.forEach(function ( account ){
   if ( account.username == username ) { msg.payload = 2;
     if ( account.password == password ) { msg.payload = 0;
     } }});
if ( msg.payload == 0 ) {
  var currentsocketid = flow.get("clientid") || undefined;
  if ( currentsocketid !== undefined && currentsocketid !== msg.socketid )
msg.payload = 3;}
return msg;
```

Κώδικας 39 Δημιουργία Έλεγχου Λογαριασμού χρηστή για το DashBoard

Όπως βλέπουμε στον κώδικα έχουμε τον ορισμό των χρηστών που έχουν πρόσβαση στο σύστημα . Παρακάτω ελέγχουμε το payload του μηνύματος που μας έρχεται από την φόρμα . Αν ο χρήστης υπάρχει και δεν είναι ενεργός τότε το payload που λαμβάνει το επόμενο node είναι 0 και επιτρέπει την πρόσβαση. Αν είναι 1 τότε

δεν αναγνωρίζει κανένα από τα στοιχεία που δώσαμε. Αν είναι 2 τότε υπάρχει ο χρήστης αλλά με άλλο κωδικό και αν είναι 4 τότε ο χρήστης που προσπαθεί να μπει είναι ήδη σε σύνδεση. Για τις δυο περιπτώσεις που ο χρήστης δεν μπορεί να μπει ενημερώνουμε για το πρόβλημα που προκύπτει. Για την περίπτωση που όλα πάνε καλά κρύβουμε την φόρμα εισόδου και εμφανίζουμε τα κουμπιά έλεγχου ώστε ο συνδεμένος χρήστης να μπορεί να έχει πρόσβαση. Αυτό γίνεται με τα UI Control Nodes και τον κώδικα που έχουμε εισάγει στο node Success.

```
var sessionTimer =  flow.get("sessionTimer") || 0;
var currTime = Date.now();
flow.set("sessionTimer", currTime);
flow.set("clientid", msg.socketid);
/* Εδω περιγράφουμε το ui-control payload */
msg.payload = { group: {
   show : ["Dashboard_Control", ]
   hide : ["Dashboard_Signin"]
   }
};
```

Κώδικας 40 Εκκίνηση Χρονομέτρησης σύνδεσης και αποκάλυψης των επίλογων

Θα παρατηρήσουμε ότι υπάρχει και ένας sessionTimer. Αυτό χρειάζεται για να κλείσει την σύνδεση και την αδεία εισόδου του χρηστή μετά από κάποιο χρόνο που θα ορίσουμε εμείς.

Ο session timer ξεκινά να μετρά από την στιγμή που θα περάσουμε τα σωστά στοιχειά χρηστή. Εμείς αν θέλουμε να ορίσουμε τον χρόνο πρόσβασης μπορούμε να το κάνουμε μεταβαλλόντας μια μεταβλητή στον κώδικα , την *SESSION_TIMEOUT*.

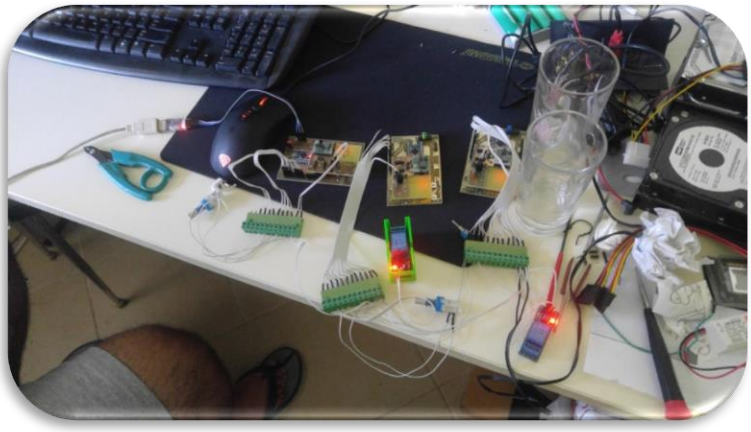
```
var sessionTimer =  flow.get("sessionTimer") || 0;
var currTime = Date.now();
var SESSION_TIMEOUT = 510000; //8,5min
if ( sessionTimer === 0 /* ανενεργός ο timer δεν υπάρχει πρόσβαση */ ) {
   msg.payload = 2;
} else {
   if ( currTime - sessionTimer > SESSION_TIMEOUT ) {
     msg.payload = 0;  }
  else {
     msg.payload = 1; }}
return msg;
```

Κώδικας 41 Έλεγχος χρονικού περιθωρίου

Έτσι ολοκληρώσαμε την κατασκευή και της ασφάλειας της επαφής για την λειτουργία των αντλιών . Έχουμε ολοκληρώσει μια ασφαλή και λειτουργική εφαρμογή που είναι πάντα διαθέσιμη όσο υπάρχει δίκτυο και μας παρέχει τις πληροφορίες που χρειαζόμαστε για να ξέρουμε ότι όλα λειτουργούν καλά .

Εγκατάσταση και λειτουργία.

Όταν ολοκληρώσαμε όλα τα σταδία κατασκευής , δοκίμων και παραμετροποίησης στο εργαστήριο, προχωρήσαμε στην τελική εγκατάσταση στο πραγματικό περιβάλλον.



Εικόνα 44 Δόκιμη για το σενάριο λειτουργίας με διακόπτες

Πρώτα εγκαταστάθηκε η αυτόνομη μονάδα και δοκιμαστικέ στο περιβάλλον της για να εκτιμήσουμε την αξιοπιστία της.

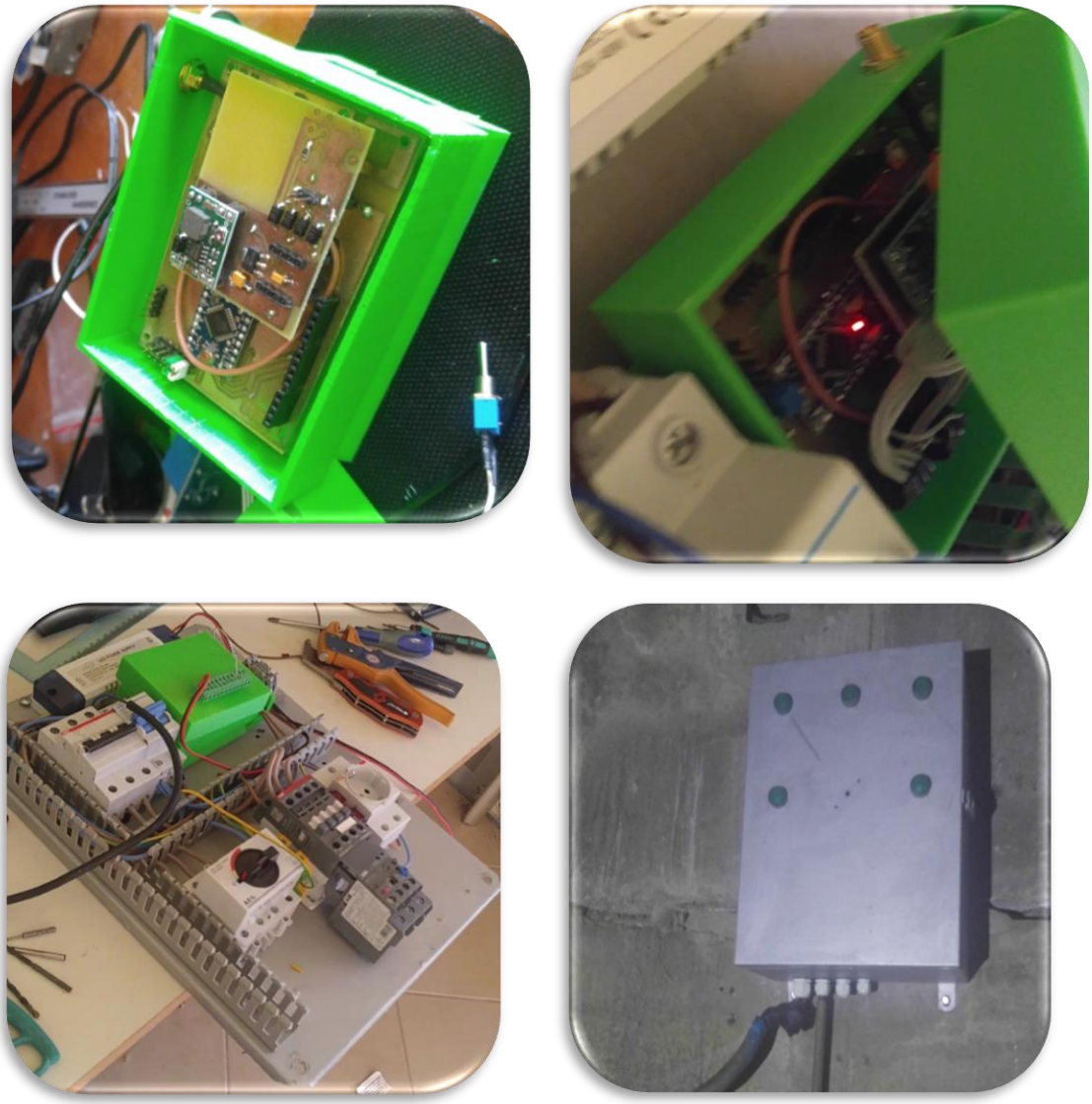
Εικόνα 45 Αυτονομη μοναδα δεξαμενης 2

Εικόνα 46 Κατά την εγκατάσταση και σύνδεση των αισθητήρων στάθμης στην δεξαμενή 2

Μετρισαμε την καταναλωση που ειχε το MqMax για το σεναριο λειτουργειας της δεξαμενης 2 και βρικαμε οτι ειχε συνεχη καταναλωση 30mA με 50mA οταν εστλενε μηνυματα στον Broker . Η διαδικασία αποστολής μηνύματος είχε οριστεί στα 4 δευτερόλεπτα και αυτό μας έδωσε μια μέση κατανάλωση 4 Wh ανά ημέρα. Έτσι μια μπαταρία AGM 12Volt 7Ah είναι αρκετή για αυτόνομη λειτουργία 10 ημερών χωρίς παρουσία ήλιου. Με την επικουρική βοήθεια από το φωτοβολταϊκής δεν είχαμε κανένα πρόβλημα στους καλοκαιρινούς μήνες που είναι και η πάροδος ενδιαφέροντος.

Αφού ολοκληρώσαμε αυτή την εγκατάσταση και είδαμε ότι λειτουργεί καλά εγκαταστήσαμε και τους υπόλοιπους κόμβους.

Για να γίνει σωστά η εγκατάσταση στους ηλεκτρολογικούς πίνακες προχωρήσαμε στον σχεδιασμό και στην 3D εκτύπωση κουτιού που να προσαρμόζεται σε ράγα ηλεκτρολογικού πινάκα..



Εικόνα 47 Εγκατάσταση στην δεξαμενή 1 και στο πηγάδι

Η εγκατάσταση έγινε το καλοκαίρι του 2019 και λειτούργησε όλο το καλοκαίρι χωρίς κανένα πρόβλημα , αξιόπιστα και αποδοτικά.

Το Μέλλον

Η πλατφόρμα μας MqMax στην έκδοση 0.4 έδωσε πολύ θετικά αποτελέσματα για την αξιοπιστία της και την εύκολη κατασκευή δικτύων Mqtt με χαρακτηριστικά M2M δηλαδή πραγματικού Internet Of Things με πολύ φτηνό και απλό τρόπο .

Το κόστος είναι τόσο χαμηλό που μπορούμε να το αυξήσουμε λίγο για να παρέχουμε μεγαλύτερη αξιοπιστία στην λειτουργία ευελιξία στην εγκατάσταση και ποιο μελετημένη τροφοδοσία .

Έτσι στην έκδοση 0.7 πλέων επανασχεδιάσαμε την κλακέτα και την κάναμε μικρότερη . συνεχίζουμε να χρησιμοποιούμε το Esp8266-07 αλλά καταργήσαμε το mini Pro με σκέτη εγκατάσταση του επεξεργαστή AtMega328p .

Έγινε αλλαγή της φιλοσοφίας της λειτουργίας της πλατφόρμας σε αμιγώς Mqtt εργαλείο. Πλέων έχει 5 εισόδους ψηφιακές με οπτική απομόνωση για να μπορούμε να συνδέσουμε μεγαλύτερες τάσεις από 5Volt . Έχει 5 εξόδους PWM που δέχονται τιμές από το 0-253 , 5 αναλογικές εισόδους που έχουν προστασία από υπέρταση και πόρτα με λειτουργία I^2C για αισθητήρες. Επίσης θα ετοιμάσουμε και έκδοση με Current Loop συμβατότητα.

Με το νέο firmware δεν χρειάζεται να προγραμματίζουμε τον Mcu αφού έχει ενσωματωμένο κώδικα που τον κάνει να λειτουργεί μέσω Mqtt . Εμείς άπλα βάζουμε τις Ρυθμίσεις του Mqtt Broker και άμεσα μπορούμε να επικοινωνούμε μαζί του στέλνοντας στο μοναδικό κανάλι που δημιουργεί με τις πόρτες του , έτσι εμείς περνούμε αυτά τα δεδομένα και τα επεξεργαζόμαστε με το Node Red απευθείας.

"IF YOU THINK THAT THE INTERNET HAS CHANGED YOUR LIFE, THINK AGAIN. THE IOT IS ABOUT TO CHANGE IT ALL OVER AGAIN!" — BRENDAN O'BRIEN, CHIEF ARCHITECT & CO-FOUNDER, ARIA SYSTEMS

Ευρετήριο

IOT	Internet of things
Mqttt	MQ Telemetry Transport)
WiFi	wireless fidelity
LoraWan	cloud-based medium access
UART	Universal Asynchronous Receiver/Transmitter
ISP	In-system programming
I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound
RAM	random access memory
FLASH	electrically erasable programmable read-only
CLOUD	accessing data and programs over the Internet
DDNS	Dynamic Domain Name System
IP	Internet Protocol address
TCP	Transmission Control Protocol
SLIP	Serial Line Internet Protocol
C++	high-level programming language
ANCI C	first standardized form of the C language
WIRING	programming language derived from C ++
PORT	interface between the computer and other computers
FIREWALL	network security system
ASSEBLY	low-level programming language
IDE	integrated development environment
COMPILER	a computer program that translates computer code
BOOTLOADER	piece of code that runs before any other
DOCKER	a tool designed to make it easier to create, deploy, and run applications by using containers

Βιβλιογραφία

1. IoT sensor integration to Node-RED platform Milica Lekić ; Gordana Gardašević
2. NOVA: a Knowledge Base for the Node-RED IoT Ecosystem Arne Bröring Darko Anicic
3. Approaching the Internet of things (IoT): a modelling, analysis and abstraction framework A Ikram, A Anjum, R Hill
4. A secure fog-based platform for SCADA-based IoT critical infrastructure T Baker, M Asim, Á MacDermott, F Iqbal
5. Internet of Things (IoT): A literature review S Madakam, V Lake,
6. Internet of things (iot) in agriculture-selected aspects M Stočes, J Vaněk, J Masne
7. Mosquitto: server and client implementation of the MQTT protocol R Light - Journal of Open Source Software, 2017 - joss.theoj.org
8. MQTT-message queuing telemetry transport protocol SA Shinde, PA Nimkar, SP Singh
9. Design and implementation of a reliable message transmission system based on MQTT protocol in IoT HC Hwang, JS Park, JG Shon
10. Introducing usage control in MQTT A La Marra, F Martinelli, P Mori, A Rizos, A Saracino
11. Internet of Things and M2M Communications F Theoleyre, AC Pang

Ιστότοποι

1. https://nodered.org/
2. https://github.com/jeelabs/esp-link
3. https://nodejs.org/en/
4. https://www.arduino.cc/
5. https://www.microchip.com/
6. https://www.espressif.com/
7. http://mqtt.org/
8. https://mosquitto.org/
9. https://www.python.org/