ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε
Μεταπτυχιακό Τμήμα Σπουδών
«Μηχανικών Η/Υ και Συστημάτων Ευφυή Συστήματα & IoT»

Μεταπτυχιακή Εργασία:

Προσομοίωση απομακρυσμένου ελέγχου-χειρισμού του συστήματος Festo MPS® PA Compact Workstation μέσω διαδικτύου

Postgraduate Thesis:

Remote control-command simulation of the Festo MPS® PA Compact Workstation system via Internet

Ονοματεπώνυμο Σπουδαστή:

Λελεκάνος Εμμανουήλ

Επιβλέπων καθηγητής

Δρ. Ξενάκης Απόστολος

Λάρισα

Δεκέμβριος 2019

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Περίληψη

Στο παρόν πόνημα επιχειρήσαμε να «συνδέσουμε» έναν εκπαιδευτικό σταθμό μετρήσεων και ελέγχου όπως είναι ο **MPS® PA Compact Workstation** της εταιρίας Festo με το σύγχρονο περιβάλλον επικοινωνίας του Διαδικτύου των Πραγμάτων – IoT. Σκοπός, λοιπόν, είναι να γνωρίσουμε βασικές έννοιες του Internet of Things, καθώς επίσης και του σταθμού μετρήσεων και ελέγχου της Festo. Η μελέτη και η προσομοίωση της λειτουργίας του σταθμού μετρήσεων της Festo πραγματοποιήθηκε από τον συνάδελφό μου Χονδρογιάννη Βλάση, μέχρι το σημείο όπου οι τιμές των μετρήσεων εξήχθησαν σε αρχείο.

Η συνέχεια της πορείας των τιμών μέχρι τον απομακρυσμένο χειριστή και οι εντολές ελέγχου αυτού προς τον σταθμό μετρήσεων αναλύονται στην παρούσα διπλωματική εργασία. Οι τιμές αποθηκεύονται σε αρχείο σε μια κάρτα μνήμης microSD η οποία στη συνέχεια τοποθετείται στην αντίστοιχη θύρα που διαθέτει το Ethernet shield που είναι τοποθετημένο επάνω στο Arduino. Οι τιμές διαβάζονται, ελέγχονται και αποστέλλονται στον MQTT Broker. Η πλατφόρμα Node-Red που σχεδιάσαμε και υλοποιήσαμε διαβάζει τις τιμές από τον Broker και στη συνέχεια τις εμφανίζει στον απομακρυσμένο χρήστη του συστήματος. Επίσης από την πλατφόρμα Node-Red, μέσω των διακοπτών ON-OFF, στέλνουμε μηνύματα εντολών ελέγχου στον Broker, τα οποία μπορεί να διαβάζει το Arduino.

Εφαρμόσαμε Fog Computing τοποθετώντας ελέγχους κατά την ανάγνωση των τιμών, ώστε να ενημερώνεται άμεσα ο τοπικός χειριστής και αυτές οι τιμές να μην αποστέλλονται στον Broker, εξοικονομώντας με αυτόν τρόπο πόρους δικτύου και χρόνο. Αντιμετωπίσαμε ορισμένα προβλήματα που μπορέσαμε και επιλύσαμε, και άλλα, για τα οποία τεχνολογικός περιορισμός δεν μας επέτρεψε να βρούμε λύσεις.

Συνοπτικά, η μελέτη αυτή παρουσιάζει τη διαδικασία προγραμματισμού και τον προγραμματισμό ενός συστήματος έτσι ώστε δύο ροές δεδομένων (τιμές μετρήσεων, εντολές ελέγχου) που χρησιμοποιούν τα ίδια μέσα, να καταλήγουν στους ενδιαφερόμενους χρήστες του.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Abstract

In the present study we attempted to "connect" a measurement and control training station such as the Festo MPS® PA Compact Workstation with the modern Internet of Things - IoT interface. The goal is to get to know the basics of the Internet of Things, as well as the Festo measuring and control station. The study and simulation of the Festo measurement station was performed by my colleague Vlasis Chondrogiannis, up to the point where the measurement values were extracted.

The continuation of the values route to the remote operator and the control commands to the measurement station are analyzed in this thesis. The values are stored in a file on a microSD card which is then inserted into the corresponding port of the Ethernet shield plugged into the Arduino. The values are read, checked and sent to an MQTT Broker. The Node-Red platform we designed and implemented reads the values from the Broker and then displays them to the remote user of the system. Also from the Node-Red platform, via the ON-OFF switches, we send control command messages to the Broker, which the Arduino can read.

We implemented Fog Computing by setting read-through controls so that the local operator is immediately informed and these values are not sent to the Broker, thereby saving network resources and time. We encountered some problems that we were able to resolve and others that, due to technological limitations, we were not able to.

In summary, this study presents the programming process and programming of a system so that two data streams (measurement values, control commands) using the same means, reach the users concerned.

Keywords: Arduino Uno, Node-Red, MQTT Broker, Festo, Publish-Subscribe, Fog & Edge Computing

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Ευχαριστίες

Ευχαριστώ θερμά τον επιβλέποντα καθηγητή Δρ. Απόστολο Ξενάκη για τη βοήθεια, τη στήριξη και την κατανόηση που μου προσέφερε καθ' όλη την διάρκεια της εκπόνησης τούτης της εργασίας.

Ευχαριστώ τον συνοδοιπόρο σ' αυτό το ταξίδι Βλάσιο Χονδρογιάννη για τη βοήθεια και τη συνεργασία που είχαμε.

Τέλος ευχαριστώ τη σύζυγό μου Αθανασία για την υπομονή της και τη στήριξη που μου προσέφερε.

Αφιερώνω αυτήν την εργασία

στη μικρή μου Ρέα.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Περιεχόμενα

Περίληψη 2
Abstract 3
Ευχαριστίες 4
Περιεχόμενα 5
1. Εισαγωγή 7
2. Τεχνολογίες IoT και Industry 4.0 8
2.1 Industry 4.0 8
2.2 Τι είναι το Cloud Computing 10
2.2.1 Καταναλωτής vs Επιχείρηση 11
2.2.2 Γνωστά παραδείγματα Cloud Computing 12
2.2.3 Αντιρρήσεις για το Cloud Computing 12
2.3 Fog και Edge Computing στο Industry 4.0 13
2.4 Πρωτόκολλο επικοινωνίας MQTT 16
2.4.1 Η προέλευση του MQTT 16
2.4.2 Ορολογία MQTT 16
2.4.3 Λειτουργία MQTT 17
2.4.4 Διευκρίνιση: MOM vs MQTT 18
2.4.5 Μειονεκτήματα MQTT 18
2.5 Περιγραφή MQTT Broker 19
2.6 Περιγραφή IoT Platform 20
3. Σύστημα Festo και Unitronics 22
3.1 Περιγραφή διάταξης Festo 22
3.2 Λειτουργία διάταξης Festo 23
3.3 Περιγραφή PLC Unitronics Vision 570 24
3.4 Τεχνικά μέσα σύνδεσης με το σύστημα Festo 25
3.5 Εναλλακτικός τρόπος επικοινωνίας Arduino – Festo. 26
3.5.1 Κάρτα μνήμης microSD 26
3.5.2 Οθόνη Υγρών Κρυστάλλων (LCD) 27
4. Προγραμματισμός Συστήματος Πληροφορίας 29
4.1 Προγραμματισμός Arduino 29
4.1.1 Περιβάλλον προγραμματισμού και Σύνδεση Arduino 29
4.1.2 Ανάλυση απαιτήσεων λογισμικού 30
4.1.3 Περιγραφή του κώδικα Arduino 30
4.2 Προγραμματισμός Node-Red 32
4.2.1 Περιβάλλον προγραμματισμού Node-Red 32
4.2.2 Ανάλυση απαιτήσεων Node-Red 33
4.2.3 Κώδικας πλατφόρμας Node-Red 33
4.3 Επιλογή Broker 34
4.4 Βοηθητικά Προγράμματα – Eclipse Paho 35

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

4.4.1 Λειτουργία Paho 35
4.4.2 Ημιδιαδρομή Node-Red – Broker (Paho = Arduino) 35
4.4.3 Ημιδιαδρομή Broker – Arduino (Paho = Node-Red) 35
4.4.4 Η αξία του Paho 35
5. Διασυνδέσεις 37
5.1 Διεπαφή Χρήστη – User Interface 37
5.2 Σύνδεση Node-Red → Broker 37
5.3 Σύνδεση Arduino → Broker 38
5.3.1 Ενσύρματη σύνδεση 38
5.3.2 Ασύρματη σύνδεση 39
5.4 ESP8266 WiFi module 39
5.4.1 Ανάλυση των ακροδεκτών του ESP 40
5.4.2 Σύνδεση Arduino – ESP (για τη ρύθμιση του ESP) 41
5.4.3 Παραμετροποίηση ESP μέσω Arduino IDE 42
5.4.4 Σύνδεση Arduino – ESP (για την εκτέλεση προγράμματος) 42
5.5 Σύνδεση Arduino – Υποδοχή SD κάρτας 42
5.6 Σύνδεση Arduino – Οθόνης Υγρών Κρυστάλλων (LCD) 43
6. Ροή και Επεξεργασία Δεδομένων 44
6.1 Μορφή Τιμών 44
6.2 Ροή και Επεξεργασία Τιμών 45
6.2.1 Ροή Τιμών 45
6.2.2 Επεξεργασία Τιμών 46
6.3 Ροή Εντολών 46
7. Τεχνικά ζητήματα, Λύσεις και Συμπεράσματα 48
Βιβλιογραφία 53
Ηλεκτρονικές Πηγές 54
Παράρτημα 1 - Κώδικας Arduino 55
Ανάγνωση Δεδομένων από την κάρτα SD 55
Βιβλιοθήκες και μεταβλητές 56
Σύνδεση και επανασύνδεση με τον Broker 57
Ανάγνωση, Έλεγχος και Αποστολή Δεδομένων στον Broker 57
Ανάγνωση Μηνυμάτων από τον Broker – CallBack 74
Setup 75
Loop 75
Παράρτημα 2 - Κώδικας Node-Red 77
Παράρτημα 3 - Συνοπτικός πίνακας εντολών του ESP 95

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

1. Εισαγωγή

Με την παρούσα διπλωματική εργασία θέλουμε να σχεδιάσουμε και να υλοποιήσουμε ένα σύστημα hardware + software (middleware) που να μας επιτρέψει να ελέγξουμε το σύστημα της Festo, απ' οπουδήποτε στον κόσμο, αρκεί να έχουμε σύνδεση στο διαδίκτυο.

Θα πρέπει να συνδέσουμε και να προγραμματίσουμε σωστά ένα Arduino έτσι ώστε να λαμβάνει τις τιμές των μετρούμενων παραμέτρων και να μπορεί να στείλει εντολές ελέγχου στο PLC (ON-OFF). Το ίδιο Arduino θα πρέπει να επικοινωνεί με κάποιον διαδικτυακό broker στον οποίο θα στέλνει τις τιμές του PLC και θα δέχεται τις απομακρυσμένες εντολές ελέγχου. Με λίγα λόγια πρέπει να φτιάξουμε ένα middleware (Arduino) που θα "συνδέει" το σύστημα της Festo (PLC) με το διαδίκτυο.

Θα πρέπει επίσης να σχεδιάσουμε και να προγραμματίσουμε μια κατάλληλη πλατφόρμα IoT, όπου θα αναπαρίσταται η λειτουργία του συστήματος της Festo. Θα μπορούμε να βλέπουμε σε σχεδόν πραγματικό χρόνο τις τιμές των μεγεθών του συστήματος και να στέλνουμε εντολές ελέγχου προς το Arduino – PLC.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

2. Τεχνολογίες IoT και Industry 4.0

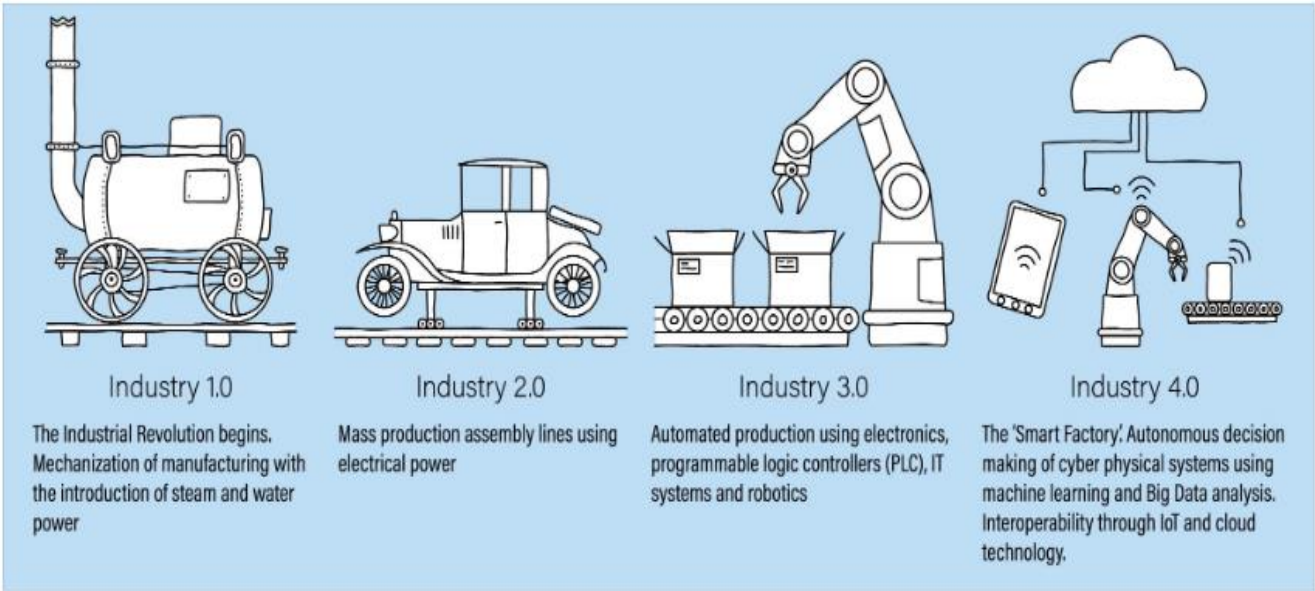
2.1 Industry 4.0

Ο όρος **Industry 4.0** καθιερώθηκε ως η 4η βιομηχανική επανάσταση. Τα στάδια στην ανάπτυξη συστημάτων βιομηχανικής κατασκευής από το χειρωνακτικό έργο προς την κατεύθυνση της Industry 4.0 μπορούν να παρουσιαστούν ως ένα μονοπάτι μέσα από τις τέσσερις βιομηχανικές επαναστάσεις, όπως μπορούμε να δούμε στην εικόνα 1.

-Η πρώτη βιομηχανική επανάσταση ξεκίνησε με τη μηχανοποίηση και την παραγωγή μηχανικής ισχύος στη δεκαετία του 1800. Έφερε τη μετάβαση από τη χειρωνακτική εργασία στις πρώτες διαδικασίες παραγωγής. κυρίως στην κλωστοϋφαντουργία. Η βελτίωση της ποιότητας ζωής ήταν η κύρια κινητήρια δύναμη της αλλαγής.

-Η δεύτερη βιομηχανική επανάσταση προκλήθηκε από την ηλεκτροδότηση που επέτρεψε την εκβιομηχάνιση και τη μαζική παραγωγή στη δεκαετία του 1900. Συχνά αναφέρεται σε αυτό το πλαίσιο ένα απόσπασμα του Henry Ford, ο οποίος είπε σχετικά με το αυτοκίνητο Ford T-Model «Μπορείτε να έχετε οποιοδήποτε χρώμα, αρκεί να είναι μαύρο». Το απόσπασμα καταγράφει καλά την εισαγωγή της μαζικής παραγωγής, αλλά χωρίς τη δυνατότητα προσαρμογής των προϊόντων.

-Η τρίτη βιομηχανική επανάσταση χαρακτηρίζεται από την ψηφιοποίηση με εισαγωγή της μικροηλεκτρονικής και του αυτοματισμού στη δεκαετία του 1960. Στην κατασκευή αυτή διευκολύνεται η ευελιξία παραγωγής, όπου μια ποικιλία προϊόντων κατασκευάζεται σε ευέλικτες γραμμές παραγωγής με προγραμματιζόμενες μηχανές. Τέτοια συστήματα παραγωγής, ωστόσο, δεν έχουν ακόμα ευελιξία όσον αφορά την ποσότητα παραγωγής.



Εικόνα 1 – Τα ορόσημα της Βιομηχανικής Επανάστασης (Ηλ.Πηγή: κεφ.2[1])

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Το Industry 4.0 είναι μια στρατηγική πρωτοβουλία της γερμανικής κυβέρνησης που υποστηρίζει παραδοσιακά την ανάπτυξη του βιομηχανικού τομέα. Η βασική ιδέα παρουσιάστηκε για πρώτη φορά στην έκθεση του Αννόβερου το έτος 2011.

Σήμερα βρισκόμαστε στην τέταρτη βιομηχανική επανάσταση που προκλήθηκε από την ανάπτυξη τεχνολογιών πληροφοριών και επικοινωνιών (ΤΠΕ). Η τεχνολογική βάση της είναι η έξυπνη αυτοματοποίηση των κυβερνο-φυσικών συστημάτων με αποκεντρωμένο έλεγχο και προηγμένη συνδεσιμότητα (λειτουργίες IoT). Η συνέπεια αυτής της νέας τεχνολογίας για τα συστήματα βιομηχανικής παραγωγής είναι η αναδιοργάνωση των κλασσικών ιεραρχικών συστημάτων αυτοματισμού στο αυτοοργανωτικό κυβερνοφυσικό σύστημα παραγωγής, το οποίο επιτρέπει ευέλικτη προσαρμοζόμενη μαζική παραγωγή και ευελιξία στην ποσότητα παραγωγής. [13]

Με τη χρήση υπολογιστών, αυτοματισμού και τεχνολογίας Cloud, τα εργοστάσια καθίστανται όλο και πιο αποδοτικά και «έξυπνα». Το Industry 4.0 αναφέρεται στην τρέχουσα τάση βελτιωμένης αυτοματοποίησης, επικοινωνίας μηχανής με μηχανή και ανθρώπου με μηχανή, τεχνητής νοημοσύνης, συνεχιζόμενων τεχνολογικών βελτιώσεων και ψηφιοποίησης της βιομηχανικής παραγωγής.
Στην εξέλιξη του Industry 4.0 καταλυτικό ρόλο έπαιξαν 4 παράγοντες:

1. Αύξηση των όγκων δεδομένων.
2. Υπολογιστική ισχύ και συνδεσιμότητα.
3. Εμφάνιση αναλυτικών στοιχείων και δυνατοτήτων επιχειρηματικής ευφυΐας — π.χ. νέες μορφές αλληλεπίδρασης ανθρώπου-μηχανής, όπως διεπαφές αφής και συστήματα αυξημένης πραγματικότητας.
4. Βελτιώσεις στη μεταφορά ψηφιακών οδηγιών στον φυσικό κόσμο, όπως η προηγμένη ρομποτική και η εκτύπωση 3D.

<u>Ένα παράδειγμα Industry 4.0</u>

Μια πιλοτική εγκατάσταση, που αναπτύχθηκε από το Γερμανικό Κέντρο Έρευνας Τεχνητής Νοημοσύνης (DFKI) στο Kaiserslautern της Γερμανίας, δείχνει πώς μπορεί να λειτουργήσει ένα «έξυπνο» εργοστάσιο. Αυτή η πιλοτική εγκατάσταση χρησιμοποιεί φιάλες σαπουνιού για να δείξει πώς τα προϊόντα και οι μηχανές παραγωγής μπορούν να επικοινωνούν μεταξύ τους. Οι κενές φιάλες σαπουνιού φέρουν ετικέτες RFID και αυτές οι ετικέτες ενημερώνουν τα μηχανήματα εάν οι φιάλες θα πρέπει να έχουν μαύρο ή λευκό καπάκι. Ένα προϊόν που βρίσκεται στη διαδικασία κατασκευής φέρει μαζί του μια ψηφιακή μνήμη προϊόντος από την αρχή και μπορεί να επικοινωνεί με το περιβάλλον του μέσω ραδιοσήματος. Αυτό το προϊόν γίνεται ένα κυβερνο-φυσικό σύστημα που επιτρέπει στον πραγματικό και στον εικονικό κόσμο να συγχωνευθούν. (Ηλ.Πηγή: κεφ.2[2])

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

2.2 Τι είναι το Cloud Computing

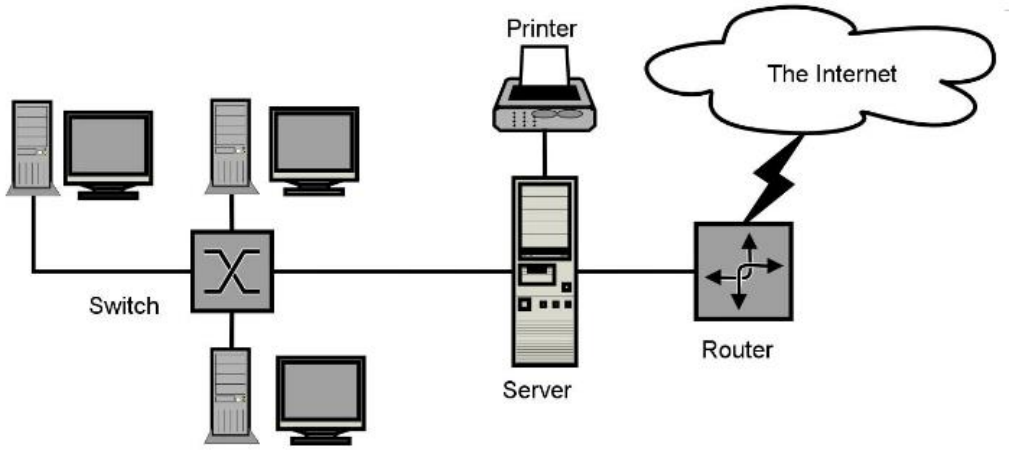
Τι είναι το Cloud; Πού είναι το Cloud; Είμαστε στο Cloud τώρα; Ερωτήσεις που πιθανώς ακούς ή αναρωτιέσαι κι εσύ ο ίδιος.

Εικόνα 2 – Cloud (Ηλ.Πηγή: κεφ.2[3])

Ο όρος Cloud Computing είναι παντού. Με απλούστερους όρους, το Cloud Computing σημαίνει αποθήκευση και πρόσβαση σε δεδομένα και προγράμματα μέσω του Internet αντί του σκληρού δίσκου του υπολογιστή σας. Το Cloud είναι μόνο μια μεταφορά για το Διαδίκτυο. Ανάγεται στις ημέρες των διαγραμμάτων ροής και των παρουσιάσεων που θα αντιπροσωπεύουν τη γιγαντιαία υποδομή των server-farms του Internet σαν ένα φουσκωτό, λευκό σύννεφο cumulus, που δέχεται συνδέσεις και μοιράζει τις πληροφορίες καθώς αιωρείται.

Αυτό με το οποίο το Cloud Computing δεν έχει σχέση, είναι ο σκληρός μας δίσκος. Όταν αποθηκεύουμε δεδομένα ή εκτελούμε προγράμματα από τη μονάδα σκληρού δίσκου, ονομάζεται τοπική αποθήκευση και υπολογισμός. Ό,τι χρειαζόμαστε είναι φυσικά κοντά μας, πράγμα που σημαίνει ότι η πρόσβαση στα δεδομένα μας είναι γρήγορη και εύκολη για έναν συγκεκριμένο υπολογιστή ή για άλλους στο τοπικό δίκτυο. Η επεξεργασία στον τοπικό σκληρό δίσκο αποτελεί τον τρόπο λειτουργίας της βιομηχανίας υπολογιστών εδώ και δεκαετίες. Ορισμένοι θα υποστηρίζουν ότι εξακολουθεί να είναι ανώτερη από το cloud computing, για λόγους που θα εξηγήσω σύντομα.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212



Εικόνα 3 – Από τα πρώτα σχέδια όπου εμφανίστηκε το Cloud (Ηλ.Πηγή: κεφ.2[3])

Το Cloud δεν αφορά επίσης στην κατοχή ενός εξειδικευμένου υλικού αποθήκευσης συνδεδεμένου στο δίκτυο (**N**etwork **A**ttached **S**torage) ή ενός τοπικού διακομιστή. Η αποθήκευση δεδομένων σε οικιακό δίκτυο ή δίκτυο γραφείου δεν θεωρείται ότι χρησιμοποιεί το Cloud. (Ωστόσο, μερικά NAS θα σας επιτρέψουν να έχετε απομακρυσμένη πρόσβαση στα πράγματα μέσω του Διαδικτύου και υπάρχει τουλάχιστον ένα εμπορικό σήμα από τη Western Digital που ονομάζεται "My Cloud", μόνο και μόνο για να παραμείνει η σύγχυση.)

Για να θεωρηθεί ως "cloud computing", πρέπει να έχετε πρόσβαση στα δεδομένα ή τα προγράμματά σας μέσω του Διαδικτύου ή, τουλάχιστον, να έχετε αυτά τα δεδομένα συγχρονισμένα με άλλες πληροφορίες μέσω του Ιστού. Σε μια μεγάλη επιχείρηση, μπορεί να γνωρίζετε όλα όσα πρέπει να ξέρετε για το τι βρίσκεται στην άλλη πλευρά της σύνδεσης. Ως μεμονωμένος χρήστης, ίσως να μην γνωρίζετε ποτέ τι είδους επεξεργασία μαζικών δεδομένων συμβαίνει στο άλλο άκρο. Το τελικό αποτέλεσμα είναι το ίδιο: με μια σύνδεση στο διαδίκτυο, το cloud computing μπορεί να γίνει οπουδήποτε, οποτεδήποτε. (Ηλ.Πηγή: κεφ2.[3])

2.2.1 Καταναλωτής vs Επιχείρηση

Αναφερόμαστε στο Cloud Computing, καθώς επηρεάζει τους μεμονωμένους καταναλωτές — εκείνους από εμάς που βρίσκονται στο σπίτι ή σε μικρά έως μεσαία γραφεία και χρησιμοποιούν το διαδίκτυο σε τακτική βάση. Υπάρχει ένα εντελώς διαφορετικό «σύννεφο» όταν πρόκειται για επιχειρήσεις. Ορισμένες επιχειρήσεις επιλέγουν να εφαρμόσουν το Software-as-a-Service (SaaS), όπου η επιχείρηση εγγράφεται σε μια εφαρμογή στην οποία έχει πρόσβαση μέσω του Διαδικτύου, (για παράδειγμα Salesforce.com). Υπάρχει επίσης η Platform-as-a-Service (PaaS), όπου μια επιχείρηση μπορεί να δημιουργήσει τις δικές της προσαρμοσμένες εφαρμογές για χρήση από όλους στην εταιρεία. Υπάρχει και η ισχυρή Infrastructure-as-a-Service (IaaS), όπου εταιρείες όπως οι Amazon, η Microsoft, η Google και η Rackspace παρέχουν μια υποδομή που μπορεί να «ενοικιαστεί» από άλλες εταιρείες.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

(Για παράδειγμα, η Netflix παρέχει τις υπηρεσίες της επειδή είναι πελάτης των υπηρεσιών cloud της Amazon.)

Φυσικά, το cloud computing είναι μια μεγάλη επιχείρηση: η αγορά δημιούργησε 100 δισεκατομμύρια δολάρια ετησίως το 2012 και θα μπορούσε να ανέλθει σε 127 δισεκατομμύρια δολάρια μέχρι το 2017 και 500 δισεκατομμύρια δολάρια μέχρι το 2020. (Ηλ.Πηγή: κεφ2.[3])

2.2.2 Γνωστά παραδείγματα Cloud Computing

Το local computing και το cloud computing μερικές φορές δεν είναι ευδιάκριτα. Αυτό συμβαίνει επειδή το Cloud είναι μέρος σχεδόν όλων των υπολογιστών μας αυτές τις μέρες. Μπορείτε εύκολα να έχετε ένα τοπικό κομμάτι λογισμικού (για παράδειγμα το Microsoft Office 365) που χρησιμοποιεί μια μορφή cloud computing για αποθήκευση (Microsoft OneDrive).

Η Microsoft προσφέρει επίσης μια σειρά από εφαρμογές που βασίζονται στο Web, το Office Online, οι οποίες είναι μόνο οι εκδόσεις Word, Excel, PowerPoint και OneNote που έχουν πρόσβαση στο Internet μέσω του προγράμματος περιήγησης στο Web χωρίς να εγκατασταθεί τίποτα. Αυτό τους καθιστά μια έκδοση του cloud computing (Web-based = cloud).

Μερικά άλλα σημαντικά παραδείγματα Cloud Computing που χρησιμοποιείτε πιθανώς:
Google Drive, Apple iCloud, Amazon Cloud Drive.
Οι υβριδικές υπηρεσίες όπως το Box, το Dropbox και το SugarSync λένε ότι λειτουργούν στο Cloud επειδή αποθηκεύουν μια συγχρονισμένη έκδοση των αρχείων σας στο διαδίκτυο, αλλά επίσης συγχρονίζουν τα αρχεία αυτά με την τοπική αποθήκευση. Ο συγχρονισμός αποτελεί τον ακρογωνιαίο λίθο της εμπειρίας του cloud computing, ακόμα κι αν έχετε πρόσβαση στο αρχείο τοπικά.

Ομοίως, θεωρείται Cloud Computing αν έχετε μια κοινότητα ατόμων με ξεχωριστές συσκευές που χρειάζονται τα ίδια δεδομένα που συγχρονίστηκαν, είτε πρόκειται για έργα συνεργασίας, είτε για να κρατήσουν την οικογένεια συγχρονισμένη. (Ηλ.Πηγή: κεφ2.[3])

2.2.3 Αντιρρήσεις για το Cloud Computing

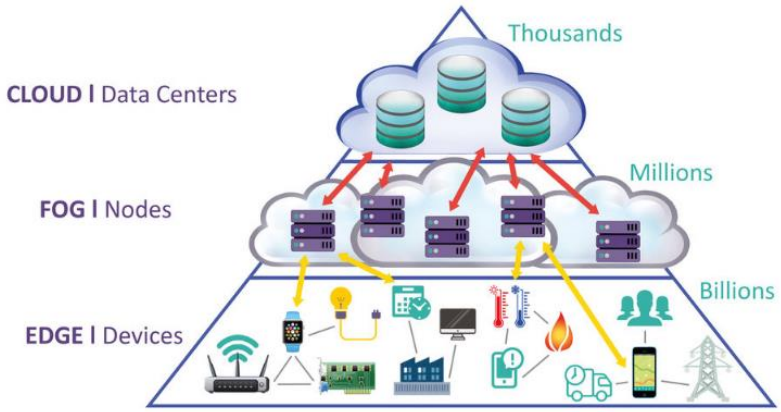
Υπάρχουν βέβαια και οι αντιρρήσεις για τη χρήση του Cloud, όπως για κάθε τι νέο. Κάποιες από αυτές είναι:
-Η πλήρης εξάρτηση των χρηστών από τους παρόχους διαδικτύου και υπηρεσιών Cloud Computing, όσον αφορά το κόστος.
-Τεχνικά προβλήματα ή και κακόβουλες επιθέσεις που κάνουν απροσπέλαστα τα δεδομένα από τους χρήστες τους.
-Ηθικού τύπου προβλήματα: σε ποιον ανήκουν τα δεδομένα που ανεβαίνουν στο Cloud? Στον χρήστη που τα έθεσε εκεί ή στην εταιρεία που παρέχει την υπηρεσία «φιλοξενίας»?
-Θέματα ασφάλειας: ποιοι μπορούν να έχουν πρόσβαση στα δεδομένα που υπάρχουν στο Cloud?

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Σε αυτή τη φάση ανάπτυξής του, το Cloud Computing μοιάζει με την Άγρια Δύση, όπου οι κανόνες φτιάχνονται παράλληλα με την εξέλιξή του. Στην προσπάθεια καθορισμού κανόνων, το IEEE δημιούργησε την πρωτοβουλία IEEE Cloud Computing το 2011 για τη θέσπιση προτύπων για χρήση, ειδικά για τον επιχειρηματικό τομέα. (Ηλ.Πηγή: κεφ2.[3])

2.3 Fog και Edge Computing στο Industry 4.0

Ο όρος Fog Computing προέκυψε από τη σύγκριση με τον όρο Cloud Computing. Μικρό νέφος, χαμηλά στο έδαφος (ομίχλη). Πρόκειται δηλαδή για μικρό Cloud Computing, κοντά στα σημεία ελέγχου (sensors & actuators).



Εικόνα 4 – Κατηγοριοποίηση και αναλογία συσκευών IoT (Ηλ.Πηγή: κεφ.2[4])

Όπως μπορούν να πιστοποιήσουν οι κατασκευαστές, ορισμένες ευρυζωνικές συνδέσεις και συνδέσεις με βάση το Cloud υποφέρουν από μειωμένη ανταπόκριση, αξιοπιστία και διαθεσιμότητα, ειδικά σε απομακρυσμένες βιομηχανικές τοποθεσίες. Ανεξάρτητα από το πόσο μεγάλος ή μικρός είναι ο κίνδυνος, είναι απαράδεκτη οποιαδήποτε διακοπή λειτουργίας σε μια γραμμή παραγωγής.

Επιπλέον, όσο περισσότερο πρέπει να μεταφέρονται τα δεδομένα στο κέντρο δεδομένων, τόσο περισσότερο χρόνο και κόστος χρειάζονται για την αποθήκευση και επεξεργασία τους. Γι' αυτό είναι σημαντικό να τεθούν οι δυνατότητες επεξεργασίας δεδομένων, ειδικά για τα πλέον ευαίσθητα δεδομένα, ακριβώς στην άκρη του δικτύου όπου παράγονται, κοντά σε αισθητήρες, μηχανές και συσκευές IIoT. Αυτό είναι το Edge Computing.

«Στο Edge, [το κρίσιμο για τις επιχειρήσεις δίκτυο] παρέχει έγκαιρη και κοντινή σύνδεση με πόρους τοπικού υπολογισμού και αναλυτικών στοιχείων. Στο Cloud, παρέχει ελαστικό εύρος ζώνης για να μετακινεί δεδομένα μέσα και έξω απ' αυτό και μεταξύ πολλαπλών περιπτώσεων Cloud. Βοηθά επίσης να κατανεμηθούν οι φόρτοι εργασίας σε όλους τους χώρους, τα σωστά δεδομένα και η λειτουργικότητα των εφαρμογών στο σωστό μέρος την κατάλληλη στιγμή».

Οι περιπτώσεις χρήσης είναι πολλές και ποικίλες. Το Edge Computing είναι συχνά μια πιο αξιόπιστη, ισχυρή και αναπτυσσόμενη επιλογή για ορισμένους τομείς, όπως η βιομηχανία πετρελαίου και

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

φυσικού αερίου, η οποία διαθέτει συνήθως εξοπλισμό που βρίσκεται στη θάλασσα ή σε δυσπρόσιτες τοποθεσίες. Και οι κατασκευαστές μπορούν να δρουν πολύ πιο γρήγορα σε δεδομένα IIoT όταν αυτά τα φορτία επεξεργάζονται στην άκρη του δικτύου, ενισχύοντας την παραγωγική απόδοση.

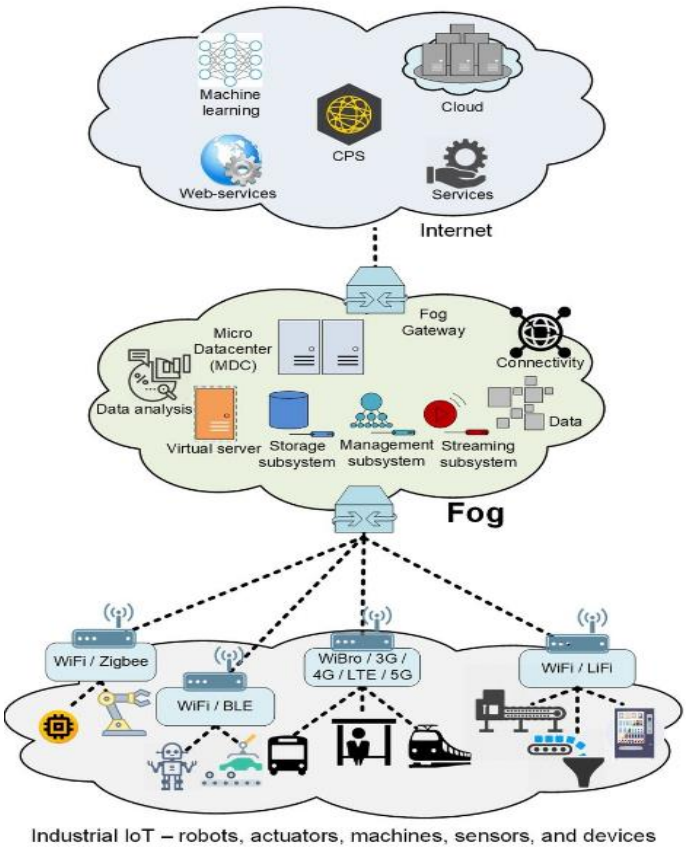
Η λιανική βιομηχανία αναμένεται να δει την ταχύτερη υιοθέτηση του Edge Computing, καθώς οι επιχειρήσεις αυτές χρησιμοποιούν όλο και περισσότερο εργαλεία αγορών, βίντεο και ψηφιακή σήμανση για να συνδεθούν με τους πελάτες. Η ταχύτερη και κοντινότερη πρόσβαση στη βάση πελατών τους δίνει τη δυνατότητα στους εμπόρους λιανικής πώλησης να αποκτήσουν γρήγορη γνώση σχετικά με τις αγοραστικές τους συμπεριφορές, τους δίνει τη δυνατότητα να προσαρμόσουν τις εμπειρίες τους και να βελτιώσουν την επιτήρηση και την ασφάλεια.

Οι οργανισμοί υγείας, επίσης, προσπαθούν να συνδέσουν καλύτερα τις συσκευές στην άκρη του δικτύου. Με αυτόν τον τρόπο μπορούν να βελτιώνουν πιο γρήγορα και αποτελεσματικά τη φροντίδα των ασθενών. Για παράδειγμα, οι διαγνωστικές συσκευές μπορούν να μεταδώσουν άμεσα δεδομένα σχετικά με την κατάσταση ενός ασθενούς, βοηθώντας τους γιατρούς να λαμβάνουν ταχύτερες αποφάσεις. Οι αισθητήρες στις νοσοκομειακές κλίνες και στις αναπηρικές καρέκλες μπορούν να παρακολουθήσουν τη διαθεσιμότητα και τη χρήση, διασφαλίζοντας ότι υπάρχει εξοπλισμός όταν είναι απαραίτητος.

Όπως ακριβώς οι συσκευές IIoT θα χρειαστούν επεξεργασία στην άκρη του δικτύου, τα δεδομένα που συσσωρεύονται θα χρειαστούν ανάλυση. Εδώ χρειάζεται Fog Computing.
«Η ομίχλη παρέχει τον σύνδεσμο που λείπει για τα δεδομένα που πρέπει να ωθηθούν στο σύννεφο και τι μπορεί να αναλυθεί τοπικά στην άκρη», εξηγεί ο Mung Chiang, κοσμήτορας του Κολλεγίου Μηχανικών του Πανεπιστημίου Purdue και ένας από τους κορυφαίους ερευνητές του κόσμου σε Fog και Edge Computing.

Παρόμοια με τα οφέλη του Edge Computing, το Fog Computing παρέχει αναλυτικά στοιχεία με μικρή καθυστέρηση στα τελικά σημεία. Αυτό μειώνει τις απαιτήσεις για το εύρος ζώνης και την απόσταση που πρέπει να μεταφέρουν τα δεδομένα για ανάλυση, πράγμα που ισοδυναμεί με εξοικονόμηση κόστους.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212



Εικόνα 5 – Διάταξη συσκευών στο Industrial IoT (Ηλ.Πηγή: κεφ.2[5])

Ενώ τα Edge και Fog Computing αποτελούν μεγάλες πύλες για την επανάσταση του IIoT, οι οργανισμοί θα πρέπει να δώσουν ιδιαίτερη προσοχή στη χωρητικότητα, τη λειτουργικότητα και τη συνδεσιμότητα του δικτύου, ώστε να αξιοποιήσουν πλήρως το δυναμικό του —όλα τα στοιχεία— ενός κρίσιμου για τις επιχειρήσεις δικτύου.

Καθώς οι επιχειρήσεις ξεκινούν την εποχή του Industry 4.0 και υιοθετούν την τεχνολογία IIoT, Edge και Fog Computing, είναι σημαντικό να επικεντρωθούν στις δυνατότητες του δικτύου. Οι κατάλληλοι πάροχοι λύσεων μπορούν να βοηθήσουν στην οικοδόμηση ενός κρίσιμου για τις επιχειρήσεις δικτύου, το οποίο παρέχει όλα όσα είναι απαραίτητα για να αποκομίσει οφέλη από αυτές τις αναδυόμενες τάσεις.

Συμπερασματικά θα λέγαμε ότι, σε αντίθεση με το Cloud, το Fog Computing παραλληλίζει την επεξεργασία δεδομένων στην άκρη του δικτύου, το οποίο ικανοποιεί τις απαιτήσεις της επίγνωσης της θέσης και της χαμηλής καθυστέρησης. Η πολυεπίπεδη αρχιτεκτονική Fog Computing είναι σε θέση να υποστηρίζει γρήγορη ανταπόκριση στα επίπεδα Edge, παρέχοντας υψηλή υπολογιστική απόδοση και ευφυΐα [8].

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

2.4 Πρωτόκολλο επικοινωνίας MQTT

Στο περιβάλλον IoT που μελετάμε, ο πιο ταιριαστός broker είναι αυτός που ακολουθεί το πρωτόκολλο **M**essage **Q**ueuing **T**elemetry **T**ransport. Ας δούμε λίγα πράγματα γι' αυτό το πρωτόκολλο επικοινωνίας.

Το MQTT (πρώην MQ Telemetry Transport) είναι ένα ελαφρύ πρωτόκολλο που έχει σχεδιαστεί κυρίως για τη σύνδεση συσκευών με περιορισμένη ισχύ σε δίκτυα χαμηλού εύρους ζώνης. Αν και υπήρχε εδώ και πάνω από μια δεκαετία, η έλευση του M2M (επικοινωνία μηχανής με μηχανή) και του Διαδικτύου των Πραγμάτων (IoT) το κατέστησαν δημοφιλές πρωτόκολλο. Οι προγραμματιστές που επιθυμούν να δημιουργήσουν λύσεις IoT πρέπει να μάθουν το MQTT, το οποίο γρήγορα γίνεται το προτιμώμενο πρωτόκολλο για τη σύνδεση συσκευών στο Cloud. Οι επιχειρησιακές πλατφόρμες cloud όπως οι Amazon Web Services, η Microsoft Azure και η IBM Watson εκθέτουν το IoT PaaS μέσω του MQTT. (Ηλ.Πηγή: κεφ2.[6])

2.4.1 Η προέλευση του MQTT

Το MQTT δημιουργήθηκε το 1999 από δύο μηχανικούς, τους Andy Stanford-Clark (IBM) και Arlen Nipper (Eurotech). Έπρεπε να εφεύρουν ένα νέο πρωτόκολλο για τη σύνδεση αγωγών πετρελαίου μέσω αναξιόπιστων δορυφορικών δικτύων. Το κίνητρο για το σχεδιασμό του MQTT ήταν να δημιουργηθεί ένα ελαφρύ και αποδοτικό, όσον αφορά το εύρος ζώνης, πρωτόκολλο, άσχετο με τον τύπο των δεδομένων, με υποστήριξη για πολλαπλά επίπεδα ποιότητας υπηρεσιών (QoS). Είναι ενδιαφέρον ότι ακόμα και σήμερα, αυτοί είναι οι ίδιοι λόγοι για τους οποίους επιλέχθηκε το MQTT για την εφαρμογή λύσεων IoT. Το 2011, η IBM και η Eurotech έδωσαν το MQTT στο προτεινόμενο σχέδιο της Eclipse που ονομάζεται Paho. Το 2013 υποβλήθηκε στο OASIS για τυποποίηση. Η τελευταία έκδοση της προδιαγραφής πρωτοκόλλου, 3.11 έχει γίνει ένα πρότυπο OASIS. (Ηλ.Πηγή: κεφ2.[6])

2.4.2 Ορολογία MQTT

Ας κατανοήσουμε την ορολογία που διέπει το MQTT:

-<u>Client/Πελάτης</u>: Κάθε Publisher (Εκδότης) και Subscriber (Συνδρομητής) που συνδέεται στον κεντρικό broker μέσω δικτύου, θεωρούνται πελάτης. Είναι σημαντικό να σημειωθεί ότι στο MQTT υπάρχουν servers και clients. Και οι Publishers και οι Subscribers καλούνται clients, απ' όταν συνδεθούν στην κεντρική υπηρεσία. Οι clients μπορεί να είναι μόνιμοι ή περιστασιακοί. Οι μόνιμοι clients διατηρούν την επικοινωνία που έχουν με τον broker, ενώ οι περιστασιακοί clients δεν παρακολουθούνται από τον broker. Οι clients συχνά συνδέονται με τον broker μέσω βιβλιοθηκών και SDK. Υπάρχουν διαθέσιμες πάνω από δώδεκα βιβλιοθήκες για C, C++, Go, Java, C#, PHP, Python, Node.js και Arduino.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

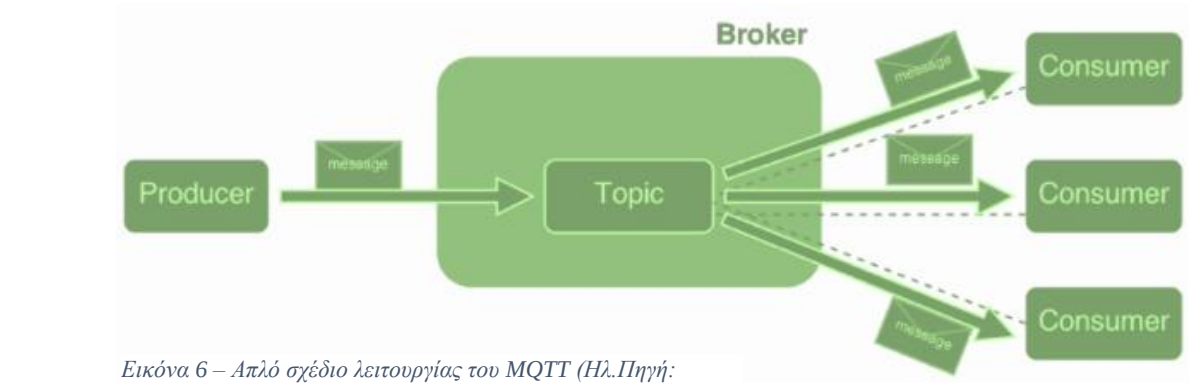
-Broker/Διαμεσολαβητής: Broker είναι το λογισμικό που λαμβάνει όλα τα μηνύματα από τους εκδότες-πελάτες (Pub-clients) και τα στέλνει στους συνδρομητές-πελάτες (Sub-clients). Διατηρεί τη σύνδεση με τους μόνιμους πελάτες. Δεδομένου ότι ο broker μπορεί να γίνει εμπόδιο ή να οδηγήσει σε αποτυχία ενός σημείου, είναι συχνά σε συστοιχία για καλύτερη επεκτασιμότητα και αξιοπιστία. Εναπόκειται στους υπεύθυνους υλοποίησης να αποφασίσουν πώς να δημιουργήσουν ένα επίπεδο στον κλιμακωτό broker. Μερικές από τις εμπορικές εφαρμογές των MQTT broker είναι: HiveMQ, Xively, AWS IoT και Loop.

-Topic/Θέμα: Ένα θέμα στο MQTT είναι ένα τελικό σημείο στο οποίο συνδέονται οι πελάτες. Λειτουργεί ως κεντρικός κόμβος διανομής για τη δημοσίευση και την εγγραφή μηνυμάτων. Σε τυπικό MOM, δημιουργείται ένα θέμα πριν ο εκδότης και ο συνδρομητής συνδεθούν με το τελικό σημείο. Στο MQTT, ένα θέμα είναι μια γνωστή τοποθεσία για τον εκδότη και τον συνδρομητή. Δημιουργείται εκείνη τη στιγμή, όταν κάποιος από τους πελάτες συνδέεται με τον broker. Τα θέματα είναι απλά, ιεραρχικές συμβολοσειρές, κωδικοποιημένα στο UTF-8, οριοθετημένα από μια εμπρόσθια κάθετο. Για παράδειγμα, το building1/room1/temperature και building1/room1/humidity είναι έγκυρα ονόματα θεμάτων. Οι συνδρομητές μπορούν να επιλέξουν να εγγραφούν σε ένα συγκεκριμένο θέμα ή σε όλα τα επιμέρους θέματα μέσω wildcards. Μια συνδρομή στο building1/+/temperature θα εγγραφεί αυτόματα στο θέμα θερμοκρασίας όλων των δωματίων του κτιρίου. Ομοίως, το building1/#/ θα εγγραφεί σε όλα τα διαθέσιμα θέματα κάτω από το building1.

-Connection/Σύνδεση: Το MQTT μπορεί να χρησιμοποιηθεί από πελάτες βάσει του TCP/IP. Η τυπική θύρα που χρησιμοποιούν οι brokers είναι η 1883, η οποία δεν είναι ασφαλής. Αυτοί οι brokers που υποστηρίζουν TLS/SSL χρησιμοποιούν συνήθως τη θύρα 8883. Για ασφαλή επικοινωνία, οι πελάτες και ο broker βασίζονται σε ψηφιακά πιστοποιητικά. Το AWS IoT είναι μία από τις ασφαλείς εφαρμογές του MQTT, που απαιτεί από τους πελάτες να χρησιμοποιούν τα πιστοποιητικά X.509. (Ηλ.Πηγή: κεφ2.[6])

2.4.3 Λειτουργία MQTT

Το MQTT χρησιμοποιεί το πρότυπο pub/sub για να συνδέει τα ενδιαφερόμενα μέρη μεταξύ τους. Το κάνει αυτό αποσυνδέοντας τον αποστολέα (publisher) από τον δέκτη (subscriber).



Εικόνα 6 – Απλό σχέδιο λειτουργίας του MQTT (Ηλ.Πηγή:

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Ο εκδότης στέλνει ένα μήνυμα σε ένα κεντρικό θέμα το οποίο έχει πολλούς συνδρομητές που περιμένουν να λάβουν το μήνυμα. Οι εκδότες (publishers) και οι συνδρομητές (subscribers) είναι αυτόνομοι, πράγμα που σημαίνει ότι δεν χρειάζεται να γνωρίζει ο ένας την παρουσία του άλλου.

Το MQTT διασφαλίζει την αξιοπιστία παρέχοντας την επιλογή τριών επιπέδων QoS:

- QoS0 (το πολύ μία φορά): Το μήνυμα στέλνεται μία φορά χωρίς να απαιτείται αναγνώριση.
- QoS1 (το λιγότερο μία φορά): Το μήνυμα στέλνεται τουλάχιστον μία φορά και απαιτείται αναγνώριση.
- QoS2 (ακριβώς μία φορά): Χρησιμοποιείται μηχανισμός τετραπλής «χειραψίας» για να διασφαλιστεί ότι το μήνυμα παραδόθηκε ακριβώς μία φορά.[6]

2.4.4 Διευκρίνιση: MOM vs MQTT

Η θεμελιώδης διαφορά μεταξύ της εφαρμογής MQTT και του MOM (Message Oriented Middleware) είναι ότι τα μηνύματα αποθηκεύονται και παραδίδονται. Σε αντίθεση με το MOM, το MQTT δεν προορίζεται για την αντιμετώπιση ανθεκτικών και επίμονων μηνυμάτων. Δεν μπορεί να ληφθεί υπόψη για την εφαρμογή του σχεδίου αποθήκευσης και προώθησης. Ενώ το παραδοσιακό MOM έχει σχεδιαστεί για την αξιόπιστη παράδοση μηνυμάτων μεταξύ επιχειρησιακών εφαρμογών, το MQTT είναι ένα απλούστερο πρωτόκολλο με μόλις πέντε API που έχουν σχεδιαστεί για τη σύνδεση συσκευών. Δεν μπορεί να χρησιμοποιηθεί ως middleware για συναλλακτικά συστήματα.

2.4.5 Μειονεκτήματα MQTT

Το MQTT είναι το δημοφιλέστερο πρωτόκολλο επικοινωνίας του IoT δεδομένου ότι τα στοιχεία του είναι περιορισμένων δυνατοτήτων συσκευές. Αλλά έχει κάποιους περιορισμούς που πρέπει να αντιμετωπιστούν, και αυτοί περιγράφονται παρακάτω:

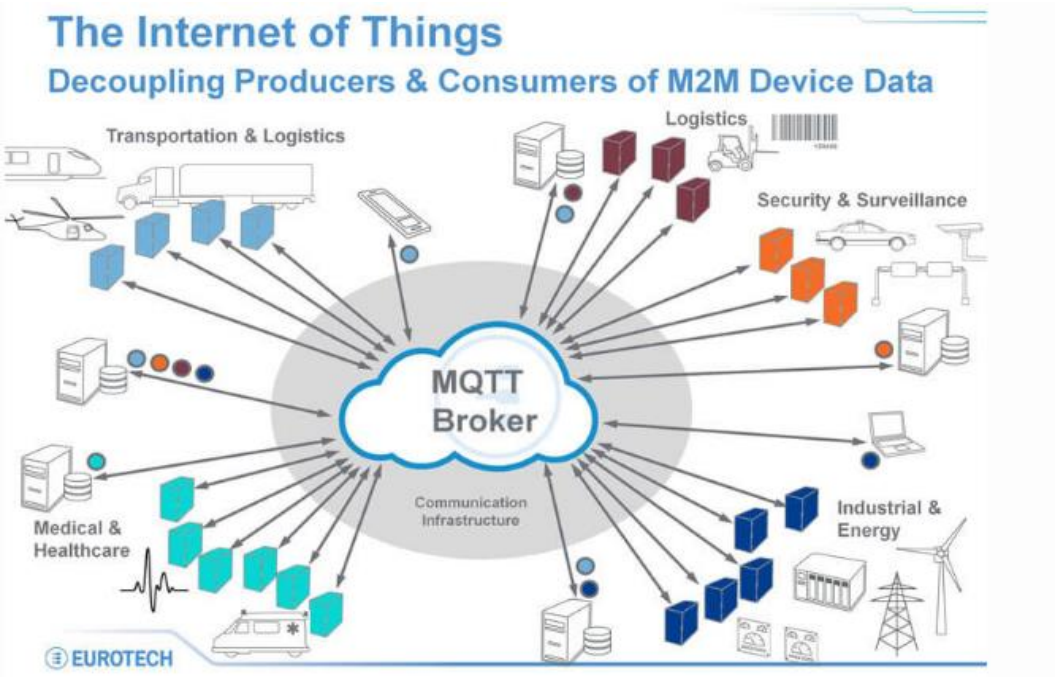
1. <u>Λήξη μηνύματος</u>: Αυτή τη στιγμή στο MQTT δεν υπάρχει λήξη μηνύματος, οπότε σε περίπτωση που βάζετε ένα μήνυμα στον Broker και στη συνέχεια ξεχάσετε να το συγκεντρώσετε ή κανείς δεν έρχεται ποτέ να το λάβει, παραμένει εκεί για πάντα. Ως αποτέλεσμα, ο Broker είναι υπερφορτωμένος με μηνύματα και υποβαθμίζεται η συνολική του απόδοση.
2. <u>Ασφάλεια</u>: Το πρωτόκολλο MQTT παρέχει όνομα χρήστη και κωδικό πρόσβασης για την εξακρίβωση της ταυτότητας, και διάφορες υλοποιήσεις Broker προσθέτουν διαφορετικό μηχανισμό πάνω από αυτό. Έτσι, η ασφάλεια στο MQTT εξαρτάται από την περίπτωση χρήσης και την επιλογή του Broker. Οι περισσότεροι Brokers παρέχουν ασφάλεια με βάση το TLS, αλλά το TLS επηρεάζει σημαντικά την απόδοση, ιδιαίτερα τη χρήση της CPU κατά τη διάρκεια της χειραψίας.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

3. Τακτοποίηση: Οι βασικές προκλήσεις για έναν αξιόπιστο οργανισμό διαβίβασης δεδομένων στο περιβάλλον του IoT είναι η τακτοποίηση μηνυμάτων και η επανάληψη μηνυμάτων που χάνονται κατά τη μετάδοση. Το MQTT παρέχει εγγυημένη παράδοση μηνυμάτων, αλλά η διατήρηση της τακτοποίησης μηνυμάτων στο MQTT είναι μια δύσκολη εργασία.
4. Προτεραιότητα: Το MQTT δεν υποστηρίζει μια λειτουργία που ονομάζεται προτεραιότητα μηνυμάτων. Εάν κάποιο σύστημα έχει πιο σημαντικά δεδομένα τότε πρέπει να είναι άμεσα διαθέσιμο σε όλους τους συνδρομητές, για παράδειγμα τα δεδομένα που συγκεντρώνονται από το σύστημα συναγερμού πυρκαγιάς είναι πιο σημαντικά από τα δεδομένα αισθητήρα θερμοκρασίας ή πίεσης, οπότε πρέπει να είναι διαθέσιμο πρώτο σε όλους τους δέκτες. Έτσι, για εκείνη την προτεραιότητα των μηνυμάτων απαιτείται η αποστολή δεδομένων σε σειρά. [7, VI]

2.5 Περιγραφή MQTT Broker

Ο όρος Broker στην IoT τεχνολογία, είναι δανεισμένος από τον χρηματοοικονομικό τομέα, όπου εκεί σημαίνει τον ενδιάμεσο κρίκο στις συναλλαγές (αγορές – πωλήσεις).



Εικόνα 7 – Η θέση του MQTT broker στο περιβάλλον IoT (Ηλ.Πηγή: κεφ.2[7])

Στην τεχνολογία IoT είναι μια υπηρεσία που προσφέρεται είτε δωρεάν για ερασιτεχνική, πειραματική και εκπαιδευτική χρήση, είτε με ετήσιο κόστος για επαγγελματική και μεγάλου όγκου δεδομένων χρήση.

Ο IoT Broker είναι ένας διαμεσολαβητής μηνυμάτων που διασφαλίζει ότι τα αντικείμενα που είναι συνδεδεμένα στο διαδίκτυο, έχουν ένα ασφαλές, ανοιχτό και διαλειτουργικό περιβάλλον επικοινωνίας.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Ο Broker ελέγχει τη διανομή των πληροφοριών και κυρίως είναι υπεύθυνος για τη λήψη όλων των μηνυμάτων από τον εκδότη, το φιλτράρισμα τους, αποφασίζει ποιος ενδιαφέρεται γι' αυτά και στέλνει τα μηνύματα σε όλους τους εγγεγραμμένους πελάτες. Μπορεί να κάνει τα ακόλουθα πράγματα:

- Αποδοχή αιτημάτων πελατών (publishers/subscribers).
- Λαμβάνει τα εκδοθέντα μηνύματα των χρηστών.
- Επεξεργάζεται διαφορετικά αιτήματα όπως εγγραφή και διαγραφή χρηστών.
- Μετά τη λήψη των μηνυμάτων από τον εκδότη, τα στέλνει στους ενδιαφερόμενους χρήστες.[7]

2.6 Περιγραφή IoT Platform

Για να κατανοήσουμε τι είναι μια πλατφόρμα IoT, θα πρέπει να ξέρουμε πώς λειτουργεί ένα ολοκληρωμένο σύστημα IoT. Εν συντομία θα περιγράψουμε τους τέσσερις πυλώνες στους οποίους βασίζεται ένα σύστημα IoT:

1. Ένα ολοκληρωμένο σύστημα IoT χρειάζεται Υλικό/Hardware, όπως αισθητήρες ή συσκευές. Αυτοί οι αισθητήρες και οι συσκευές συλλέγουν δεδομένα από το περιβάλλον (π.χ. έναν αισθητήρα υγρασίας) ή εκτελούν ενέργειες στο περιβάλλον (π.χ. πότισμα καλλιεργειών).

2. Ένα ολοκληρωμένο σύστημα IoT χρειάζεται Συνδεσιμότητα/Connectivity. Το υλικό χρειάζεται έναν τρόπο να μεταδίδει όλα αυτά τα δεδομένα στο Cloud (π.χ. στέλνοντας δεδομένα υγρασίας) ή χρειάζεται έναν τρόπο να λαμβάνει εντολές από το Cloud (π.χ. νερό στις καλλιέργειες τώρα). Αυτό μπορεί να επιτευχθεί με ώριμες μορφές συνδεσιμότητας όπως η κυψελοειδής, δορυφορική ή WiFi, ή μπορεί να απαιτήσει πιο πρόσφατες, εστιασμένες στο IoT, επιλογές συνδεσιμότητας όπως το LoRa.

3. Ένα ολοκληρωμένο σύστημα IoT χρειάζεται Λογισμικό/Software. Αυτό το λογισμικό φιλοξενείται στο Cloud και είναι υπεύθυνο για την ανάλυση των δεδομένων που συλλέγονται από τους αισθητήρες και για τη λήψη αποφάσεων (π.χ. γνωρίζει από δεδομένα υγρασίας ότι μόλις έβρεξε και μετά δίνει εντολή στο σύστημα άρδευσης να μην ποτίσει σήμερα).

4. Ένα ολοκληρωμένο σύστημα IoT χρειάζεται Διεπαφή Χρήστη/User Interface. Για να γίνει όλο αυτό χρήσιμο, πρέπει να υπάρχει ένας τρόπος για τους χρήστες να αλληλοεπιδρούν με το σύστημα IoT (π.χ. μια εφαρμογή με βάση τον ιστό, με πίνακα ελέγχου που δείχνει τάσεις υγρασίας και επιτρέπει στους χρήστες να ενεργοποιούν ή να απενεργοποιούν χειροκίνητα τα συστήματα άρδευσης).

Επιπλέον, η πραγματική αξία του IoT ξεκλειδώνεται όταν ενσωματώνεται σε υπάρχοντα επιχειρηματικά συστήματα και ροές δεδομένων. Είναι επομένως κρίσιμο το γεγονός ότι όλα αυτά τα διαφορετικά συστατικά συνδέονται μεταξύ τους αποτελεσματικά και κατά τρόπο διαχειρίσιμο. Σε υψηλό επίπεδο, οι πλατφόρμες IoT παρέχουν ένα ξεκίνημα για την κατασκευή συστημάτων IoT παρέχοντας ενσωματωμένα εργαλεία και δυνατότητες για να κάνουν το IoT ευκολότερο και φθηνότερο για τις επιχειρήσεις, τους προγραμματιστές και τους χρήστες. Μια πλατφόρμα IoT

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

διευκολύνει την επικοινωνία, τη ροή δεδομένων, τη διαχείριση συσκευών και τη λειτουργικότητα των εφαρμογών.

Οι πλατφόρμες IoT υπάρχουν στο μέρος 3 και, συχνά, στο μέρος 4 αυτού που περιγράφεται παραπάνω. Με όλα τα διαφορετικά είδη υλικού και τις διαφορετικές επιλογές συνδεσιμότητας, πρέπει να υπάρχει ένας τρόπος να κάνουμε τα πάντα να λειτουργούν μαζί. Οι πλατφόρμες IoT βοηθούν στην επίλυση αυτού του προβλήματος. Συνοψίζοντας θα λέγαμε ότι οι πλατφόρμες IoT:

- Συνδέουν το Υλικό, όπως αισθητήρες και συσκευές.
- Διαχειρίζονται τα πρωτόκολλα επικοινωνίας του Υλικού και του Λογισμικού.
- Παρέχουν ασφάλεια και πιστοποίηση για τις συσκευές και τους χρήστες.
- Συλλέγουν, οπτικοποιούν και αναλύουν τα δεδομένα που οι αισθητήρες και οι συσκευές συγκεντρώνουν.
- Ολοκληρώνουν όλα τα παραπάνω σε υπάρχοντα επιχειρησιακά συστήματα και άλλες υπηρεσίες του ιστού.

Το ερώτημα γιατί υπάρχουν τόσες πολλές πλατφόρμες IoT μπορεί να απαντηθεί με 1) ότι η αγορά εξακολουθεί να είναι τόσο πρωτοπόρα ώστε οι κυρίαρχοι παίκτες δεν έχουν ακόμη αναδειχθεί ή 2) ότι λόγω των άπειρων εφαρμογών IoT σε διάφορες βιομηχανίες, θα υπάρξουν διαφορετικές πλατφόρμες IoT που επικεντρώνονται σε διαφορετικά είδη εφαρμογών. Όπως πάντα, είναι πιθανώς ένας συνδυασμός των δύο, αλλά πιστεύω ότι η πρώτη εξήγηση είναι πιο ρεαλιστική. (Ηλ.Πηγή: κεφ2.[7])

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

3. Σύστημα Festo και Unitronics

Η διάταξη που θέλουμε να ελέγξουμε μέσω ενός συστήματος IoT που σχεδιάζουμε και υλοποιούμε, είναι το **MPS® PA Compact Workstation** συνδεδεμένο με το plc **Unitronics Vision 570**.

3.1 Περιγραφή διάταξης Festo

Στην εικόνα 8 βλέπουμε τη διάταξη της Festo, MPS® PA Compact Workstation.

Εικόνα 8 – Διάταξη Festo, MPS® PA Compact Workstation (Ηλ.Πηγή: κεφ.3[1])

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Αποτελείται από τα εξής εξαρτήματα:

Μηχανικά μέρη:

- 2 δεξαμενές
- Δεξαμενή πίεσης
- Σύστημα σωληνώσεων
- Ρυθμιστική βαλβίδα φίλτρου
- Πλαίσιο στήριξης
- Πλάκα προφίλ

Αισθητήρες:

- 2 αισθητήρες χωρητικότητας
- 2 επιπλέοντες διακόπτες
- Αισθητήρας υπερήχων
- Αισθητήρας ροής
- Αισθητήρας πίεσης
- Αισθητήρας θερμοκρασίας PT100

Ενεργοποιητές:

- Αντλία
- Αναλογική βαλβίδα ελέγχου κατεύθυνσης
- Σφαιρική βαλβίδα 2 κατευθύνσεων με πνευματικό ενεργοποιητή στρέψης τεταρτημόριου και ανίχνευση τελικής θέσης
- Διπλής ενέργειας
- Θέρμανση

Ηλεκτρικά στοιχεία:

- Πίνακας σύνδεσης I/O με μετατροπέα μέτρησης
- Ελεγκτής κινητήρα
- Τερματικό I/O SysLink 8I/8O
- Αναλογικό τερματικό SysLink 15 ακίδων

3.2 Λειτουργία διάταξης Festo

Τα τέσσερα ελεγχόμενα συστήματα του MPS® PA Compact Workstation μπορούν να λειτουργούν μεμονωμένα. Χρησιμοποιώντας έναν αντίστοιχο ελεγκτή, το σύστημα ελέγχου στάθμης και ροής μπορεί να ρυθμιστεί ως κλιμακωτό σύστημα ελέγχου. Ο σχεδιασμός των αισθητήρων και των ενεργοποιητών βαλβίδων επιτρέπει τη διεξαγωγή δοκιμών τόσο με συνεχείς (π.χ., P, I, PI, PID) όσο και σε ασυνεχείς ελεγκτές (π.χ. ελεγκτές δύο σημείων). Οι αντλίες μπορούν να ελέγχονται είτε με άμεση ενεργοποίηση είτε με ρύθμιση ταχύτητας.

Με τα συστήματα ελέγχου ρυθμού ροής και πίεσης, η μεταβλητή χειρισμού του ελεγκτή μπορεί επίσης να χρησιμοποιηθεί για τη λειτουργία μιας αναλογικής βαλβίδας ελέγχου κατεύθυνσης. Μια σφαιρική βαλβίδα δύο κατευθύνσεων με πνευματικό ενεργοποιητή στρέψης τεταρτημόριου τοποθετείται στην επιστροφή μεταξύ της ανυψωμένης δεξαμενής και της κάτω δεξαμενής. Η σφαιρική βαλβίδα διπλής κατεύθυνσης μπορεί να χρησιμοποιηθεί για να προσομοιώνει ένα «φορτίο» διαταραχής μεταβλητής αντιστάθμισης στο σύστημα ελέγχου στάθμης [9].

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

3.3 Περιγραφή PLC Unitronics Vision 570

Προηγμένος προγραμματιζόμενος λογικός ελεγκτής στην οπίσθια πλευρά, μεγάλη και υψηλής ανάλυσης, έγχρωμη οθόνη αφής 5.7" στην εμπρόσθια πλευρά. Μονάδες εισόδου/εξόδου Snap-in για έναν ελεγκτή Όλα σε Ένα. Μπορεί να επεκταθεί μέχρι 1000 I/Os.

Εικόνα 9 – Human Interface Display του Vision 570 (Ηλ.Πηγή: κεφ.3[2])

Παρακάτω θα δούμε περιληπτικά τα χαρακτηριστικά του:

-PLC:

- Αυτόματη ρύθμιση του PID, έως 24 ανεξάρτητους βρόχους.
- Προγράμματα σεναρίων και καταγραφή δεδομένων μέσω πινάκων δεδομένων.
- Κάρτα MicroSD/SD–αρχείο καταγραφής, backup, κλώνος και πολλά άλλα.
- Λειτουργικά μπλοκ.

-Human-Machine Interface:

- Υψηλής ποιότητας οθόνη αφής.
- Πολυγλωσσική απεικόνιση.
- Προεγκατεστημένες οθόνες συναγερμών.

-Επικοινωνία:

- Mini USB για τον προγραμματισμό του V570.
- CANbus.
- Απομονωμένα RS485/ RS232.
- Πρόβλεψη για επιπλέον θύρα Σειριακή ή Ethernet.

-Πρωτόκολλα Επικοινωνίας:

- MODBUS TCP.
- SNMP (SNMP V1 Trap, SNMP community Name).
- CANopen, CANlayer2, UniCAN.
- BACnet, KNX και M-Bus μέσω της πύλης.
- Πρωτόκολλο FB: για οποιοδήποτε πρωτόκολλο τρίτου μέρους.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Εικόνα 10 – Θύρες επικοινωνίας (Ηλ.Πηγή: κεφ.3[2])

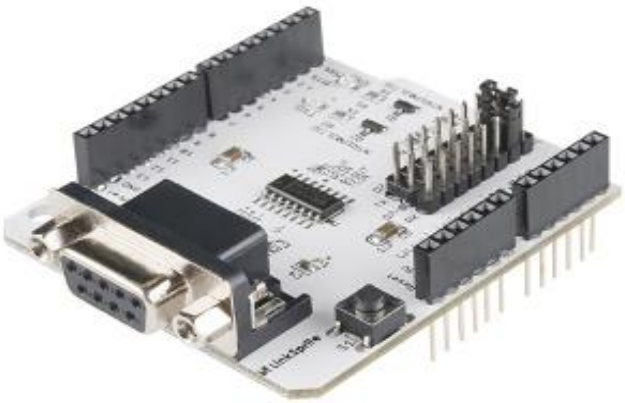
Εικόνα 11 – Θύρες επικοινωνίας (Ηλ.Πηγή: κεφ.3[2])

-Γενικά χαρακτηριστικά: Web server, E-mail & SMS, υποστήριξη 3G Modem, βοηθητικά προγράμματα απομακρυσμένης πρόσβασης. Μπορεί να υποστηρίξει μέχρι 1000 εισόδους και εξόδους και στις επιλογές αυτών περιλαμβάνονται: ψηφιακές, αναλογικές, υψηλής ταχύτητας, θερμοκρασίας και μέτρηση βάρους.(Ηλ.Πηγή: κεφ.3[2])

3.4 Τεχνικά μέσα σύνδεσης με το σύστημα Festo

Η σύνδεση του συστήματος της Festo με το Arduino έχει έναν σκοπό, τον απομακρυσμένο έλεγχο της διάταξης Festo. Άρα με την σύνδεση αυτή πρέπει να διαβάζουμε τις τιμές των μετρήσεων και να στέλνουμε εντολές ελέγχου.

Η σύνδεση του plc με το arduino μπορεί να επιτευχθεί, μέσω της θύρας RS232. Η σειριακή επικοινωνία είναι φθηνότερη και ευκολότερη. Θα χρειαστούμε όμως την κάρτα RS232 shield, που φαίνεται στην εικόνα 12, έτσι ώστε να αποκτήσει σειριακή θύρα το arduino. Οι πληροφορίες μας θα χρησιμοποιήσουν το πρωτόκολλο επικοινωνίας FB, που διαθέτει το plc.



Εικόνα 12 – Arduino RS232 shield shield (Ηλ.Πηγή: κεφ.3[3])

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Η δεύτερη επιλογή μας για τη σύνδεση του plc με το arduino είναι μέσω CANbus. Και γι' αυτή τη σύνδεση βέβαια θα χρειαστούμε επιπλέον κάρτα για το arduino. Μια CAN-BUS shield, όπως φαίνεται στην εικόνα 13.

Εικόνα 13 – Arduino CAN-BUS shield (Ηλ.Πηγή: κεφ.3[4])

3.5 Εναλλακτικός τρόπος επικοινωνίας Arduino – Festo.

Και στις δύο παραπάνω συνδέσεις έχουμε πλήρη (αμφίδρομη) επικοινωνία. Το ιδανικό σενάριο εφαρμογής IoT. Αν για κάποιον λόγο όμως δεν είναι δυνατή καμία από τις παραπάνω συνδέσεις, θα μπορούσαμε εναλλακτικά να λαμβάνουμε τις μετρούμενες τιμές στην microSD card από το plc και να τις διαβάζουμε στο arduino και οι εντολές ελέγχου θα μπορούσαν να εμφανίζονται στην LCD οθόνη του arduino, ώστε να εκτελούνται χειροκίνητα.

3.5.1 Κάρτα μνήμης microSD

Η κάρτα μνήμης που φαίνεται στην εικόνα 14, αντικαθιστά τη σειριακή επικοινωνία του Arduino και του συστήματος της Festo, για τη διαδρομή των τιμών των μετρήσεων από τους αισθητήρες προς το Node-Red.

Η βιβλιοθήκη SD επιτρέπει την ανάγνωση και την εγγραφή σε κάρτες SD, π.χ. στο Arduino Ethernet shield. Στηρίζεται στην sdfatlib από τον William Greiman. Η βιβλιοθήκη υποστηρίζει συστήματα

Εικόνα 14 - κάρτα μνήμης microSD (Ηλ.Πηγή: κεφ.3[6])

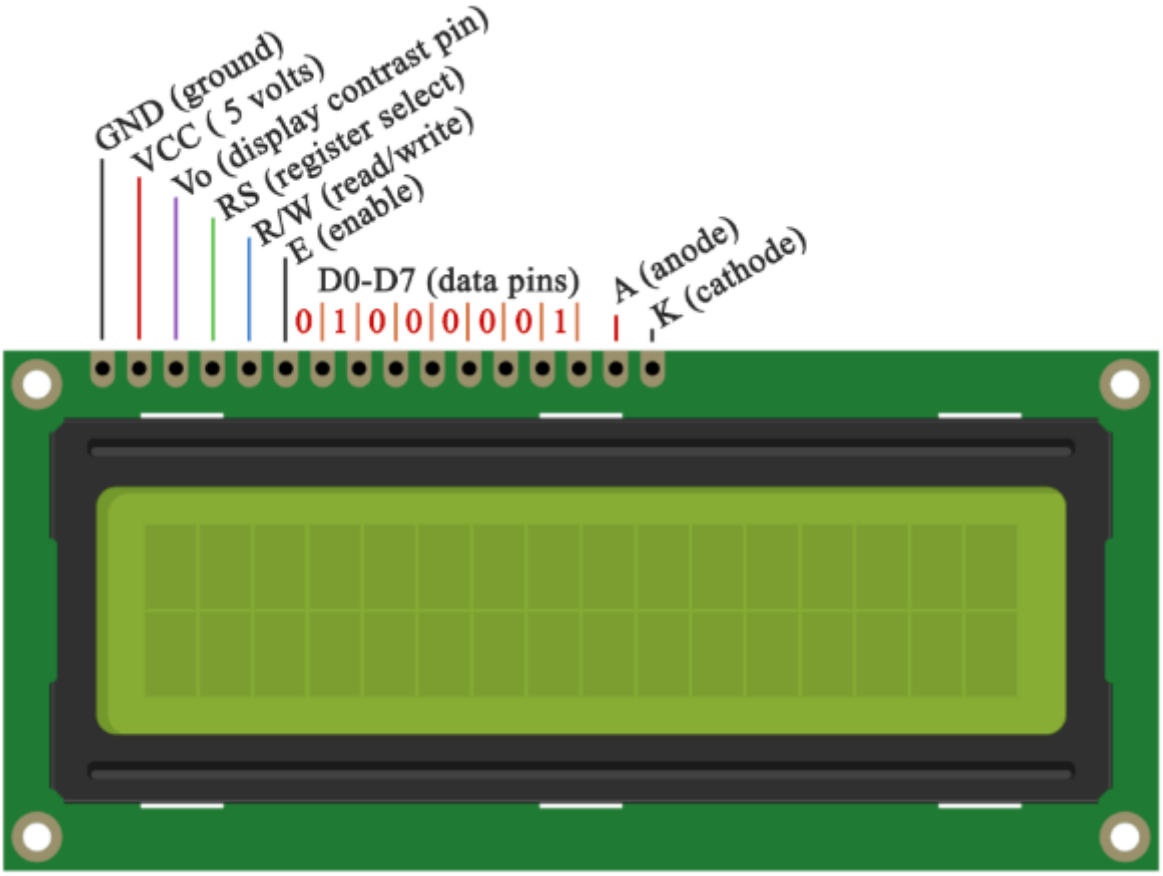
Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

αρχείων FAT16 και FAT32 σε τυπικές κάρτες SD και κάρτες SDHC.

Χρησιμοποιεί για την ταξινόμηση 8.3 ονόματα για αρχεία. Τα ονόματα αρχείων που μεταβιβάζονται στις λειτουργίες της βιβλιοθήκης SD μπορούν να περιλαμβάνουν διαδρομές που χωρίζονται από καθέτους, /, π.χ. "directory/filename.txt". Επειδή ο κατάλογος εργασίας είναι πάντα η ρίζα της κάρτας SD, ένα όνομα αναφέρεται στο ίδιο αρχείο ανεξάρτητα από το αν περιέχει ή όχι μια κάθετο μπροστά (π.χ. "/file.txt" ισοδυναμεί με "file.txt"). Από την έκδοση 1.0, η βιβλιοθήκη υποστηρίζει το άνοιγμα πολλών αρχείων.

3.5.2 Οθόνη Υγρών Κρυστάλλων (LCD)

Η οθόνη LCD αντικαθιστά τη σειριακή επικοινωνία του Arduino και του συστήματος της Festo, για τη διαδρομή των εντολών ελέγχου από το Node-Red προς το σύστημα της Festo. Στην παρακάτω εικόνα 15 βλέπουμε την οθόνη καθώς και τους ακροδέκτες της.



Εικόνα 15 – LCD 16 x 2 (Ηλ.Πηγή: κεφ.3[7])

Η βιβλιοθήκη LiquidCrystal επιτρέπει τον έλεγχο οθονών LCD συμβατές με το πρόγραμμα οδήγησης Hitachi HD44780. Επιλέγουμε οθόνη με 16 ακροδέκτες. Οι οθόνες LCD έχουν παράλληλη διασύνδεση, πράγμα που σημαίνει ότι ο μικροεπεξεργαστής πρέπει να χειριστεί ταυτόχρονα αρκετούς

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

ακροδέκτες διεπαφής για να ελέγξει την οθόνη. Η διεπαφή αποτελείται από τους ακόλουθους ακροδέκτες:

- Ένας **ακροδέκτης καταχωρητή επιλογής (RS)** που ελέγχει τη θέση στην οποία γράφονται δεδομένα στη μνήμη της οθόνης LCD. Μπορεί να επιλέξει είτε τον καταχωρητή δεδομένων, ο οποίος κρατάει ό,τι πηγαίνει στην οθόνη, είτε έναν καταχωρητή εντολών, όπου ο ελεγκτής της οθόνης αναζητά οδηγίες για το τι πρέπει να κάνει στη συνέχεια.
- Ένας **ακροδέκτης ανάγνωσης/εγγραφής (R/W)** που επιλέγει τη λειτουργία ανάγνωσης ή τη λειτουργία εγγραφής.
- Ένας **ακροδέκτης ενεργοποίησης** που επιτρέπει την εγγραφή στους καταχωρητές.
- 8 **ακροδέκτες δεδομένων (D0-D7)**. Οι καταστάσεις αυτών των ακροδεκτών (υψηλή ή χαμηλή) είναι τα bits που γράφονται σε έναν καταχωρητή κατά την εγγραφή ή οι τιμές που διαβάζονται κατά την ανάγνωση.

Υπάρχει επίσης ένας ακροδέκτης (Vo) για την αντίθεση της οθόνης, οι ακροδέκτες τροφοδοσίας (+ 5V και Gnd) και οι ακροδέκτες LED Backlight (Bklt + και BKlt-) που χρησιμοποιούνται για τον έλεγχο της αντίθεσης της οθόνης, για την τροφοδοσία της και για την ενεργοποίηση και απενεργοποίηση της λυχνίας LED οπίσθιου φωτισμού, αντίστοιχα. (Ηλ.Πηγή: κεφ.5[5])

Η διαδικασία ελέγχου της οθόνης περιλαμβάνει την τοποθέτηση των δεδομένων που σχηματίζουν την εικόνα που πρέπει να εμφανίζεται στους καταχωρητές δεδομένων, και στη συνέχεια, η τοποθέτηση οδηγιών στον καταχωρητή οδηγιών. Χρησιμοποιούμε την βιβλιοθήκη LiquidCrystal.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

4. Προγραμματισμός Συστήματος Πληροφορίας

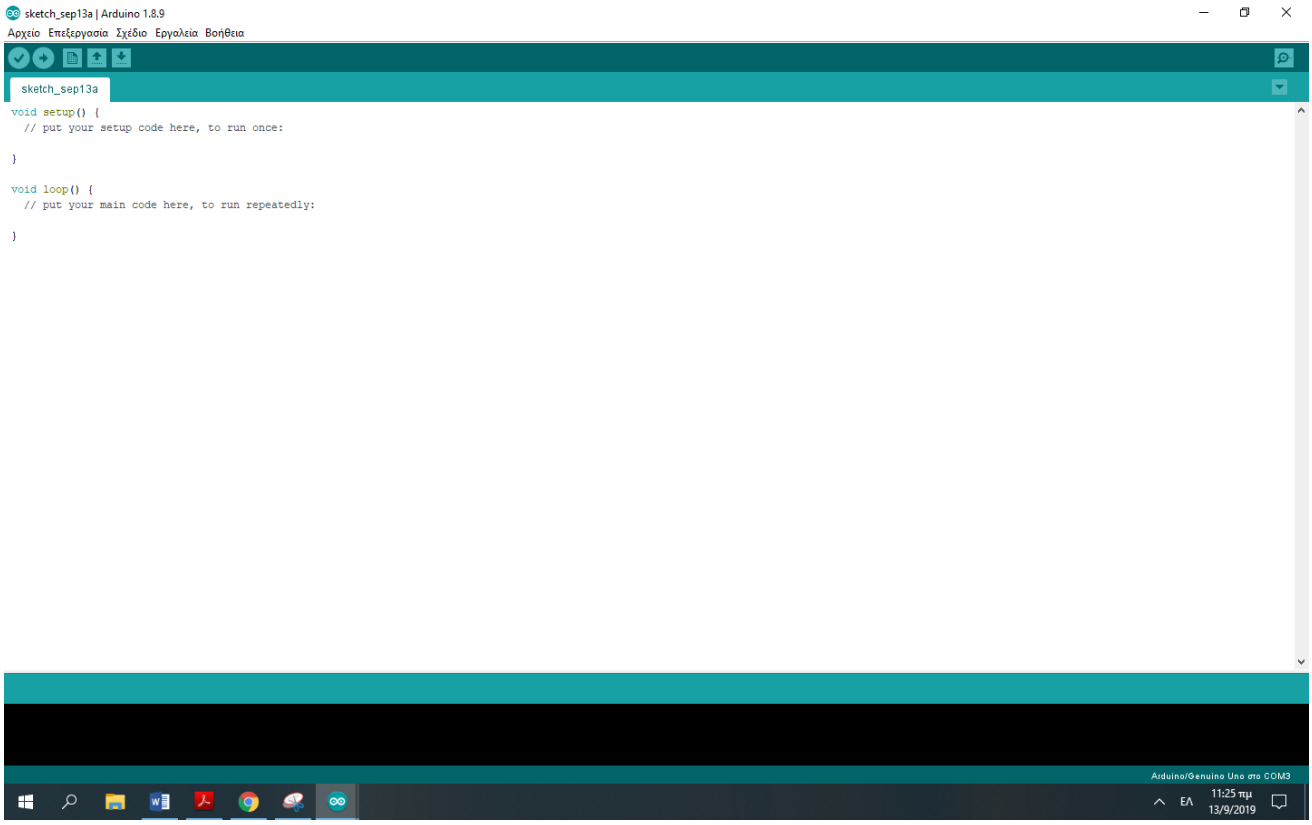
Η σωστή διαχείριση της πληροφορίας είναι το κυριότερο στοιχείο σε ένα σύστημα IoT. Επειδή έχουμε πολλούς κόμβους, από όπου διέρχεται η πληροφορία, έτσι ώστε να μπορούν να τη διαβάζουν διαφορετικές πλατφόρμες υλικού και λογισμικού, είναι αναγκαία συνθήκη η πληροφορία να ταξιδεύει ακέραιη και ορθή σε όλη τη διαδρομή της.

4.1 Προγραμματισμός Arduino

Η γλώσσα προγραμματισμού του Arduino είναι η C++ και μπορούμε να αναπτύξουμε τον κώδικα μέσω της εφαρμογής Arduino IDE με τελευταία έκδοση την 1.8.10.

4.1.1 Περιβάλλον προγραμματισμού και Σύνδεση Arduino

Στην παρακάτω εικόνα 16 μπορούμε να δούμε το περιβάλλον προγραμματισμού του Arduino. Συνδέουμε το Arduino με τον υπολογιστή μας με καλώδιο τύπου usb και επιλέγουμε από το περιβάλλον IDE τη σειριακή θύρα (com3 ή com4) στην οποία φαίνεται η πλακέτα του Arduino. Η επιλογή γίνεται από το menu: Εργαλεία → Πλακέτα, Εργαλεία → Θύρα.



Εικόνα 16 – Arduino IDE programming panel

Πρώτα όμως θα πρέπει να αναλύσουμε τις απαιτήσεις που πρέπει να υλοποιήσουμε για να σχεδιάσουμε σωστά τον κώδικα.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

4.1.2 Ανάλυση απαιτήσεων λογισμικού

Οι απαιτήσεις που πρέπει να ικανοποιεί το Arduino είναι:

1. Να μπορεί να διαβάζει τα εισερχόμενα δεδομένα, τις τιμές δηλαδή που έρχονται από το σύστημα της Festo. Τα δεδομένα θα είναι σε μορφή JSON, οπότε το arduino θα πρέπει να «σπάει» την κάθε συμβολοσειρά (string) και να κρατάει τα ζευγάρια μέτρηση–τιμή, π.χ. θερμοκρασία–23.
2. Να μετατρέπει σε ψηφιακές τιμές, αν είναι απαραίτητο, όσες έρχονται σε αναλογική μορφή (4 – 20 mA), με την προσθήκη κατάλληλου μαθηματικού τύπου.
3. Να ελέγχει τις τιμές αν ανήκουν στο αποδεκτό εύρος τιμών ή αν υπάρχει κάποιο σφάλμα. Να κάνει τον πρώτο έλεγχο στις τιμές που διαβάζει, αν είναι σε λογικό-αποδεκτό εύρος τιμών ή αν υπάρχει υποψία π.χ. χαλασμένου αισθητήρα. Αν υπάρχει τέτοια τιμή δεν φεύγει για τον broker, αλλά εμφανίζει κατάλληλο μήνυμα στην οθόνη του arduino (Fog Computing).
4. Κάθε τιμή ανάλογα με τη μέτρηση που τη συνοδεύει, πρέπει να στέλνεται στο αντίστοιχο πεδίο (topic) του broker.
5. Να μπορεί να διαβάζει εισερχόμενα μηνύματα από τον broker, όταν υπάρχουν, και να προωθεί τις εντολές ελέγχου στο σύστημα της Festo.

4.1.3 Περιγραφή του κώδικα Arduino

Αφού αναλύσαμε τις απαιτήσεις που πρέπει να υποστηρίζει το πρόγραμμα, συνεχίζουμε με τον κώδικα που πρέπει να αναπτύξουμε. Στο Παράρτημα 1 – Κώδικας Arduino υπάρχει ο κώδικας που αναπτύξαμε για το πρόγραμμα, κατηγοριοποιημένος στις επιμέρους λειτουργίες του. Η πρώτη ενότητα Ανάγνωση Δεδομένων από κάρτα SD αφορά έναν αυτόνομο δοκιμαστικό κώδικα για την ανάγνωση δεδομένων από την κάρτα SD. Οι επόμενες ενότητες αφορούν το συνολικό ενιαίο πρόγραμμα:

- Ξεκινά από την ενότητα Βιβλιοθήκες και μεταβλητές, στην οποία φαίνονται οι βιβλιοθήκες που είναι απαραίτητες για την υλοποίηση του κώδικα καθώς και οι μεταβλητές του προγράμματος που πρέπει να είναι universal, να χρησιμοποιούνται δηλαδή από οποιαδήποτε συνάρτηση ή μέθοδο του προγράμματος.
- Ακολουθεί η ενότητα Σύνδεση και επανασύνδεση με τον Broker. Με την μέθοδο–συνάρτηση **reconnect()** εξασφαλίζουμε ότι θα είμαστε συνδεδεμένοι με τον broker σε κάθε περίπτωση, αρκεί να έχει την κατάλληλη θέση στο πρόγραμμα.
- Στην ενότητα Ανάγνωση, Έλεγχος και Αποστολή Δεδομένων ανήκει η μέθοδος–συνάρτηση **ReadSendValues()** που αναλαμβάνει να διαβάσει από το αρχείο που περιέχεται στην κάρτα SD, να ελέγξει τις τιμές που διάβασε (fog computing) αν είναι εντός λογικών ορίων ή αν υπάρχει κάποιο σφάλμα στη μέτρηση και εφόσον δεν υπάρχει, να στείλει τις τιμές στα αντίστοιχα topics του broker. Ο κώδικας που είναι *σε πλάγια γραφή* (σε σχόλια) είναι έλεγχοι που αφαίρεσα στην προσπάθεια να περιορίσω το μέγεθος του προγράμματος (βλ.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Συμπεράσματα: Πρόβλημα Nº 2). Στον Πίνακα 1 που ακολουθεί μπορούμε να δούμε τους συνδυασμούς των strings που πρέπει να ελέγξει η **ReadSendValues**.

Ζευγάρια Strings για την πρώτη γραμμή του αρχείου				
S1=Temp	S1=Flow	S1=Lev1	S1=Lev2	S1=Pres
S2=Flow	S2=Temp	S2=Temp	S2=Temp	S2=Temp
S2=Lev1	S2=Lev1	S2=Flow	S2=Flow	S2=Flow
S2=Lev2	S2=Lev2	S2=Lev2	S2=Lev1	S2=Lev1
S2=Pres	S2=Pres	S2=Pres	S2=Pres	S2=Lev2
Ζευγάρια Strings για τις υπόλοιπες γραμμές				
S11=Temp	S11=Flow	S11=Lev1	S11=Lev2	S11=Press
S22=Flow	S22=Temp	S22=Temp	S22=Temp	S22=Temp
S22=Lev1	S22=Lev1	S22=Flow	S22=Flow	S22=Flow
S22=Lev2	S22=Lev2	S22=Lev2	S22=Lev1	S22=Lev1
S22=Pres	S22=Pres	S22=Pres	S22=Pres	S22=Lev2

Πίνακας 1- Συνδυασμοί Strings

- Ακολουθεί η ενότητα Ανάγνωση μηνυμάτων από τον broker – Callback. Εξίσου σημαντική μέθοδος–συνάρτηση η **Callback** επιτρέπει στο πρόγραμμα να διαβάσει εισερχόμενα μηνύματα από τον broker. Με αυτή τη μέθοδο μπορεί το πρόγραμμα να διαβάσει τις εντολές ελέγχου που έρχονται από την πλατφόρμα Node-Red, άρα από τον απομακρυσμένο χειριστή του συστήματος, προς τη διάταξη Festo. Οποιαδήποτε εντολή προορίζεται για το Arduino → Festo, η Callback θα την εμφανίζει στην LCD οθόνη του Arduino, αφού η σύνδεση Arduino – Festo δεν είναι εφικτή. Ο κώδικας που είναι *σε πλάγια γραφή* (σε σχόλια) προορίζεται για μελλοντική χρήση, όταν θα είναι εφικτό οι εντολές ελέγχου να φτάνουν στο plc.
- Ακολουθεί η ενότητα **Setup** στην οποία περιέχονται οι αρχικοποιήσεις των συσκευών, συνδέσεις, έλεγχοι κι ό,τι άλλο είναι απαραίτητο για ξεκινήσει σωστά το πρόγραμμα. Ό,τι κώδικας είναι στο πεδίο του Setup τρέχει μία φορά μόνο.
- Τελευταία ενότητα είναι η **Loop**, στην οποία τοποθετείται ο κώδικας που πρέπει να τρέχει ατέρμονα. Για παράδειγμα η συνάρτηση **reconnect()** συνοδευόμενη με τις εγγραφές (subscribe) του πελάτη (Arduino) στα πεδία (topics) του broker. Με αυτόν τον τρόπο εξασφαλίζουμε ότι σε κάθε loop που θα εκτελείται το πρόγραμμα, θα ελέγχεται η σύνδεση με τον broker και θα μπορεί το πρόγραμμα στο arduino να βλέπει τα topics που πρέπει, για τυχόν εισερχόμενα μηνύματα.

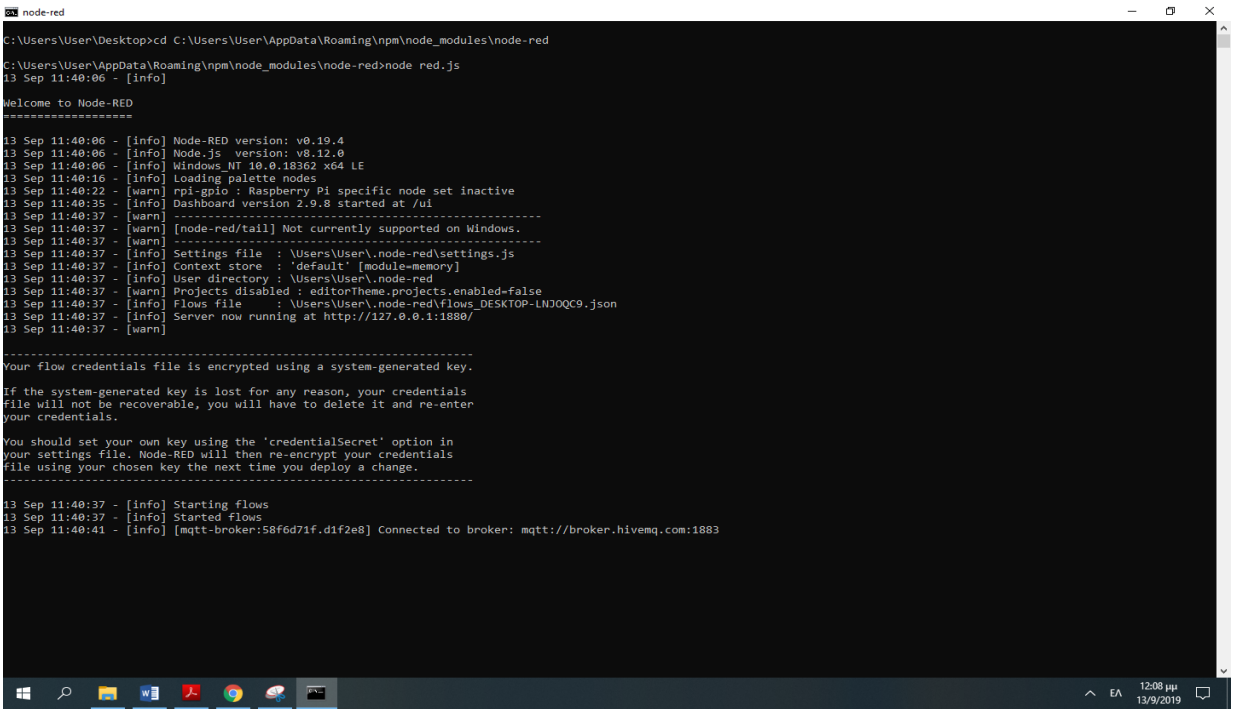
Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

4.2 Προγραμματισμός Node-Red

Ως πλατφόρμα IoT για την υλοποίηση της εργασίας επιλέξαμε τη Node-Red, επειδή είναι απλή και εύκολη στον προγραμματισμό της. Η εγκατάστασή της είναι τοπική, στον υπολογιστή μας και όχι Cloud based. Ο χρήστης του συστήματος IoT τρέχει την πλατφόρμα στον υπολογιστή του και δεν συνδέεται με κωδικούς στο Cloud.

4.2.1 Περιβάλλον προγραμματισμού Node-Red

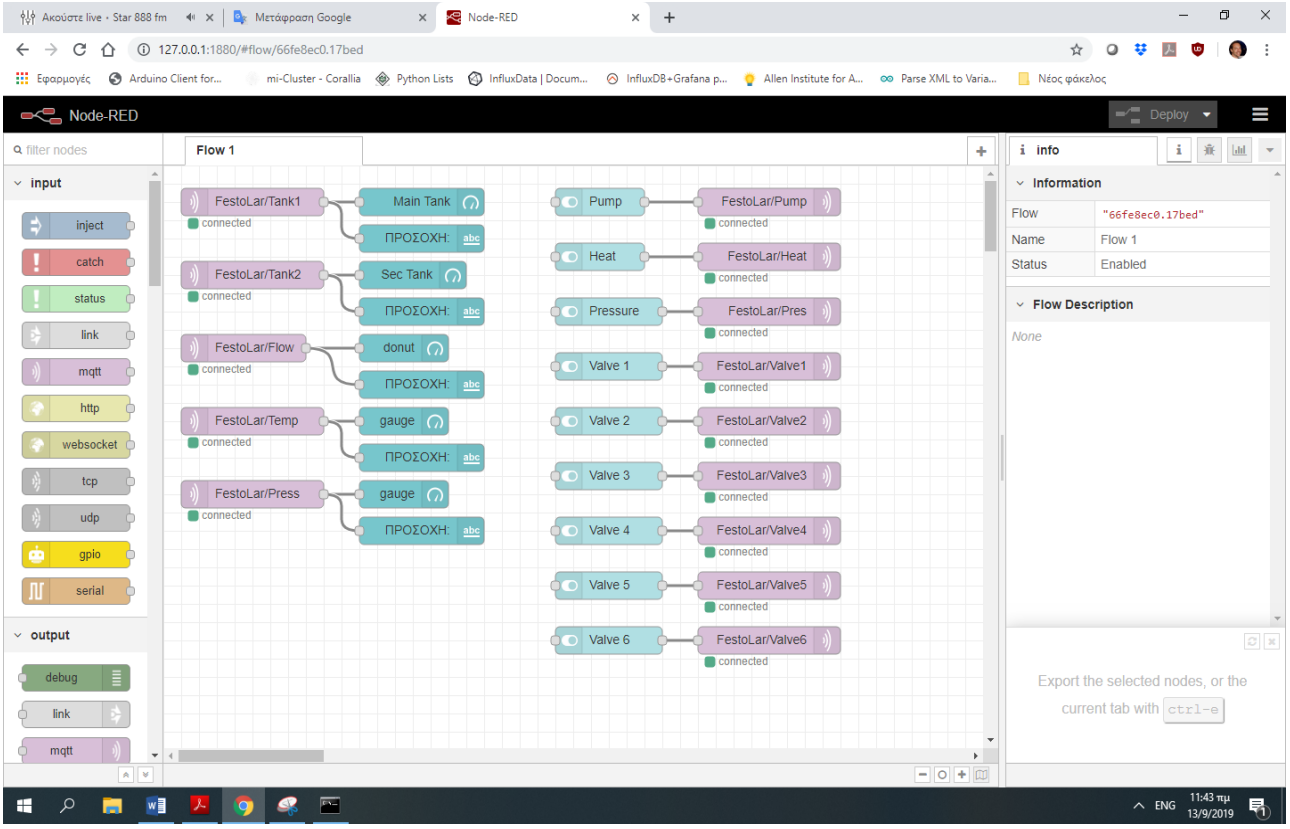
Για να μπούμε στο περιβάλλον προγραμματισμού της Node-Red τρέχουμε το batch αρχείο για να ενεργοποιηθεί η υπηρεσία, όπως φαίνεται στην παρακάτω εικόνα 17.



Εικόνα 17 – Ενεργοποίηση service Node-Red

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Αφού ενεργοποιηθεί η υπηρεσία και συνδεθεί το Node-Red στον broker, ανοίγουμε έναν φυλλομετρητή (Internet browser) και πληκτρολογούμε τη διεύθυνση 127.0.0.1:1880 και βλέπουμε το περιβάλλον προγραμματισμού της Node-Red, όπως φαίνεται στην παρακάτω εικόνα 18.



Εικόνα 18 – Node-Red dashboard

Πρώτα όμως πρέπει να καθορίσουμε τις απαιτήσεις που πρέπει να εξυπηρετεί η πλατφόρμα.

4.2.2 Ανάλυση απαιτήσεων Node-Red

Οι απαιτήσεις που πρέπει να ικανοποιεί η πλατφόρμα Node-Red, είναι:

- Να διαθέτει τα κατάλληλα απεικονιστικά στοιχεία των μετρούμενων τιμών. Από την συλλογή που διαθέτει η Node-Red, μπορούμε να επιλέξουμε απεικονιστικά σχήματα, που να ταιριάζουν καλύτερα στο σύστημα της Festo και φυσικά να τα συνδέσουμε με τα αντίστοιχα topics του broker.
- Να διαθέτει τα κατάλληλα απεικονιστικά στοιχεία ελέγχου, διακόπτες που θα στέλνουν τις εντολές ελέγχου στο Arduino. Φυσικά και αυτοί πρέπει να συνδεθούν στα αντίστοιχα topics του broker, ώστε να στέλνονται οι εντολές ελέγχου (ON-OFF).

4.2.3 Κώδικας πλατφόρμας Node-Red

Στο dashboard που φαίνεται στην εικόνα 18, με τη μέθοδο drag and drop, τοποθετούμε τα στοιχεία απεικόνισης, διασύνδεσης και ελέγχου, που είναι απαραίτητα για την αναπαράσταση του συστήματος Festo. Απαιτείται ομαδοποίηση, παραμετροποίηση και χωροθέτηση των στοιχείων αυτών έτσι ώστε να

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

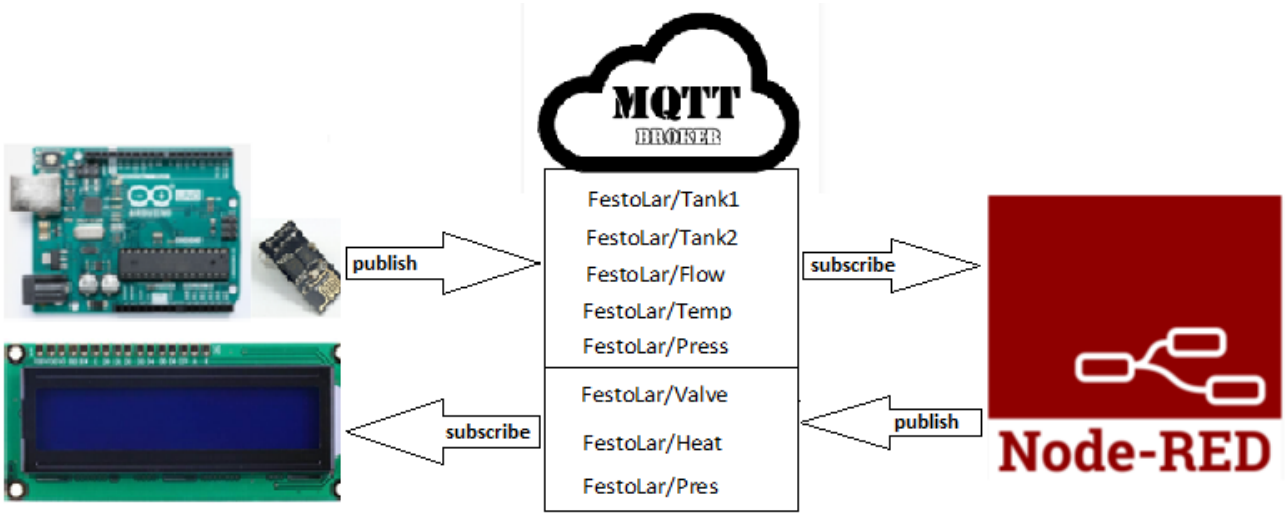
έχουμε το αποτέλεσμα της εικόνας 20. Ο κώδικας που προέκυψε από την παραπάνω διαδικασία που περιγράψαμε μπορεί να εξαχθεί (menu → export → clipboard) και επισυνάπτεται στο Παράρτημα 2.

4.3 Επιλογή Broker

Η επιλογή του broker είναι σημαντική, αν σκεφτούμε ότι είναι ο κεντρικός κόμβος από όπου διέρχονται τα δεδομένα μας προς κάθε κατεύθυνση μεταξύ τον συσκευών και υπηρεσιών IoT. Για την εργασία μας (πειραματική χρήση – μικρός όγκος δεδομένων) μπορούμε να επιλέξουμε μεταξύ δύο γνωστών δωρεάν broker:

- test.mosquitto.org - http://test.mosquitto.org/
- broker.hivemq.com - https://www.hivemq.com/try-out/

Στο παρακάτω σχέδιο 1, μπορούμε να δούμε τον κομβικό ρόλο του Broker καθώς και τα πεδία (topics) που θα χρησιμοποιήσουμε στην εφαρμογή μας.



Σχέδιο 1 – Συσχέτιση των topics του broker με Arduino και Node-Red

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

4.4 Βοηθητικά Προγράμματα – Eclipse Paho

Κύριο βοηθητικό εργαλείο στην εξέλιξη της εργασίας είναι το **Eclipse Paho MQTT utility**. Συνδέοντας το Paho στον ίδιο broker που θα επιλέξουμε για το IoT σύστημά μας, μπορούμε να ελέγξουμε την αμφίδρομη μεταφορά των δεδομένων, μεταξύ Arduino και Node-Red.

4.4.1 Λειτουργία Paho

- Προσθέτουμε μια νέα σύνδεση με τον broker που θέλουμε (π.χ. με τον hivemq.com), γράφοντας στο πεδίο Server URI: tcp://broker.hivemq.com:1883
- Ακολούθως στο πεδίο Subscription μπορούμε να προσθέσουμε topics τα οποία θέλουμε να διαβάζουμε.
- Τέλος στο πεδίο Publication γράφουμε το topic στο οποίο θέλουμε να γράψουμε και το μήνυμα που επιθυμούμε.

Με αυτόν τον τρόπο, μπορούμε να ελέγξουμε αν κάνουμε Publish και Subscribe από το Arduino και το Node-Red, αντικαθιστώντας κάθε φορά το ένα από τα δύο με το Paho.

4.4.2 Ημιδιαδρομή Node-Red – Broker (Paho = Arduino)

Με το Paho έχουμε τη δυνατότητα να κάνουμε publish τιμές στα αντίστοιχα topics του broker και να δούμε αν εμφανίζονται στο Node-Red. Αντίστοιχα μπορούμε να κάνουμε subscribe στα κατάλληλα topics του broker και να δούμε τις εντολές ελέγχου που στέλνουμε από το Node-Red.

4.4.3 Ημιδιαδρομή Broker – Arduino (Paho = Node-Red)

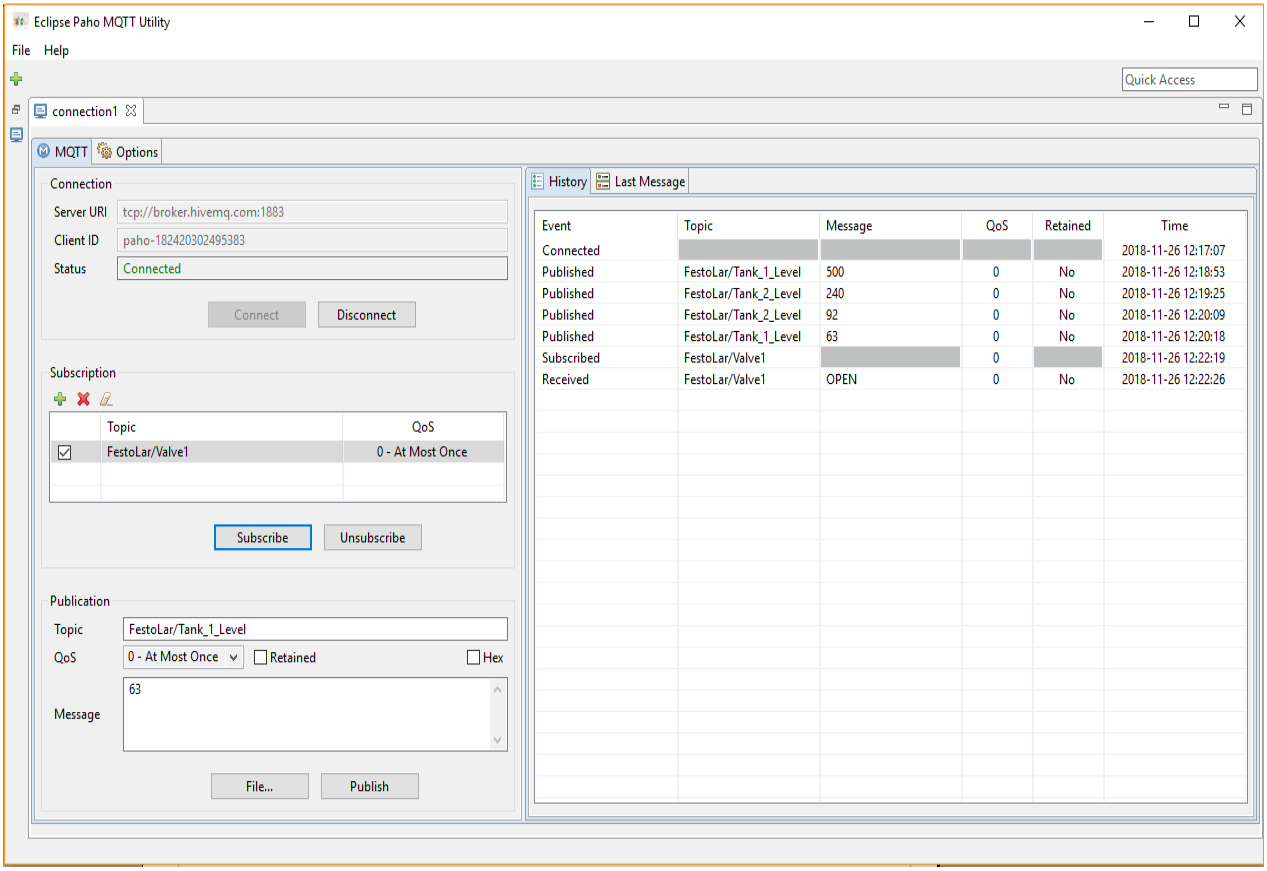
Αντίστροφα τώρα μπορούμε να κάνουμε publish τις εντολές ελέγχου στα αντίστοιχα topics του broker και να δούμε αν τις διαβάζει το Arduino. Αντίστοιχα μπορούμε να κάνουμε subscribe στα κατάλληλα topics του broker και να δούμε τις τιμές που στέλνει το Arduino.

4.4.4 Η αξία του Paho

Με το Paho μπορούμε να ελέγξουμε εύκολα τη ροή των δεδομένων μας, μοιράζοντας τη διαδρομή στη μέση. Μπορούμε να βρούμε τα λάθη μας εύκολα, χωρίς να χαθούμε ταυτόχρονα σε κώδικα Arduino και Node-Red. Είναι ένα εξαιρετικό εργαλείο που βοηθάει πολύ στην ανάπτυξη συστημάτων IoT που χρησιμοποιούν MQTT brokers.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Στην παρακάτω εικόνα 19 μπορούμε να δούμε το περιβάλλον του Paho.



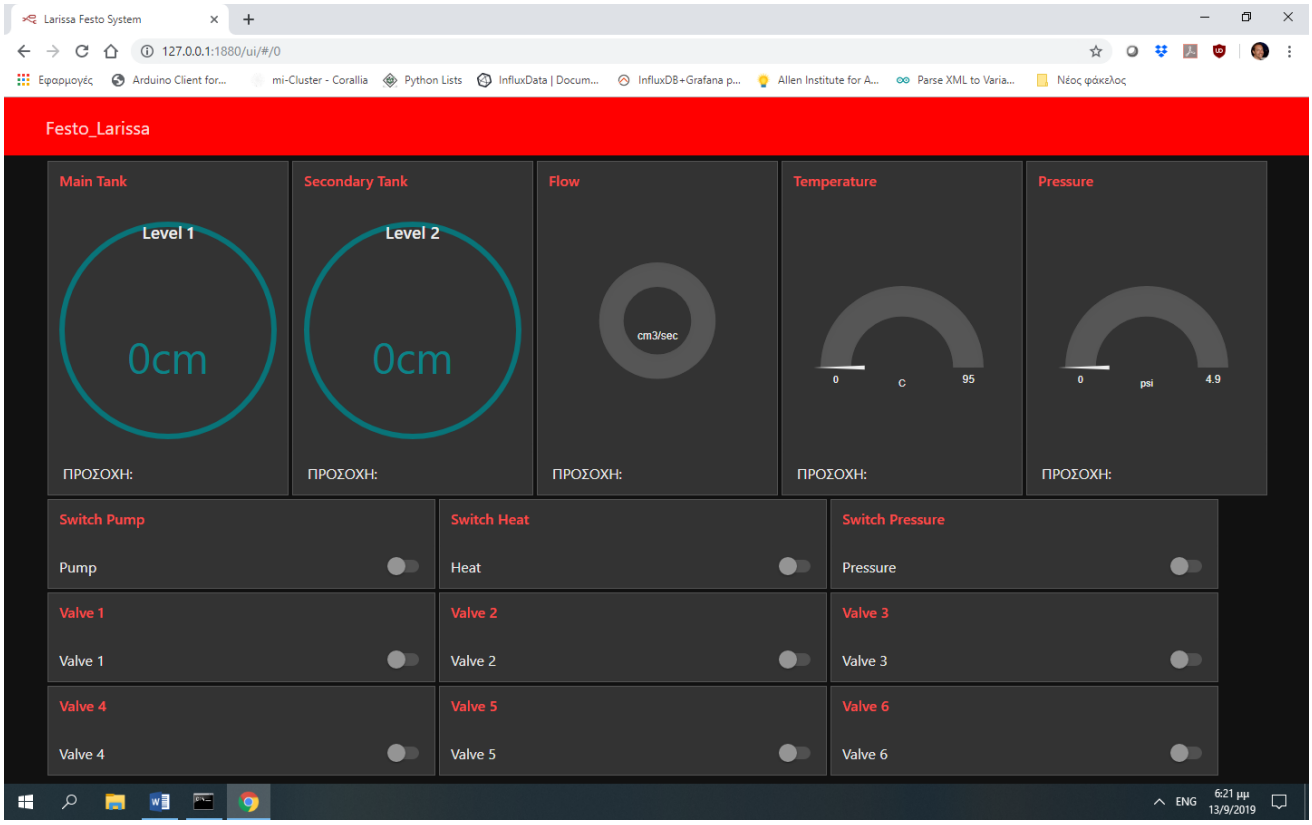
Εικόνα 19 – Paho MQTT Utility

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

5. Διασυνδέσεις

5.1 Διεπαφή Χρήστη – User Interface

Αφού προγραμματίσαμε την πλατφόρμα Node-Red σωστά, μπορούμε να δούμε το περιβάλλον με το οποίο ο χρήστης θα αλληλεπιδρά με το arduino και κατ' επέκταση με το σύστημα της Festo, στην παρακάτω εικόνα 20.



Εικόνα 20 – Control Panel

5.2 Σύνδεση Node-Red → Broker

Η Node-Red είναι εγκατεστημένη στον υπολογιστή μας, ο υπολογιστής μας συνδέεται ενσύρματα ή ασύρματα με τον Router και μέσω αυτού αποκτά πρόσβαση στο Internet. Κατά τον προγραμματισμό της Node-Red δηλώνουμε τον broker, τη θύρα του (1883) καθώς και τα topics με τα οποία θα συνδεθεί η πλατφόρμα. Στο παρακάτω σχέδιο 2 μπορούμε να δούμε τη σύνδεση.

Σχέδιο 2 – Σύνδεση Node-Red – Broker

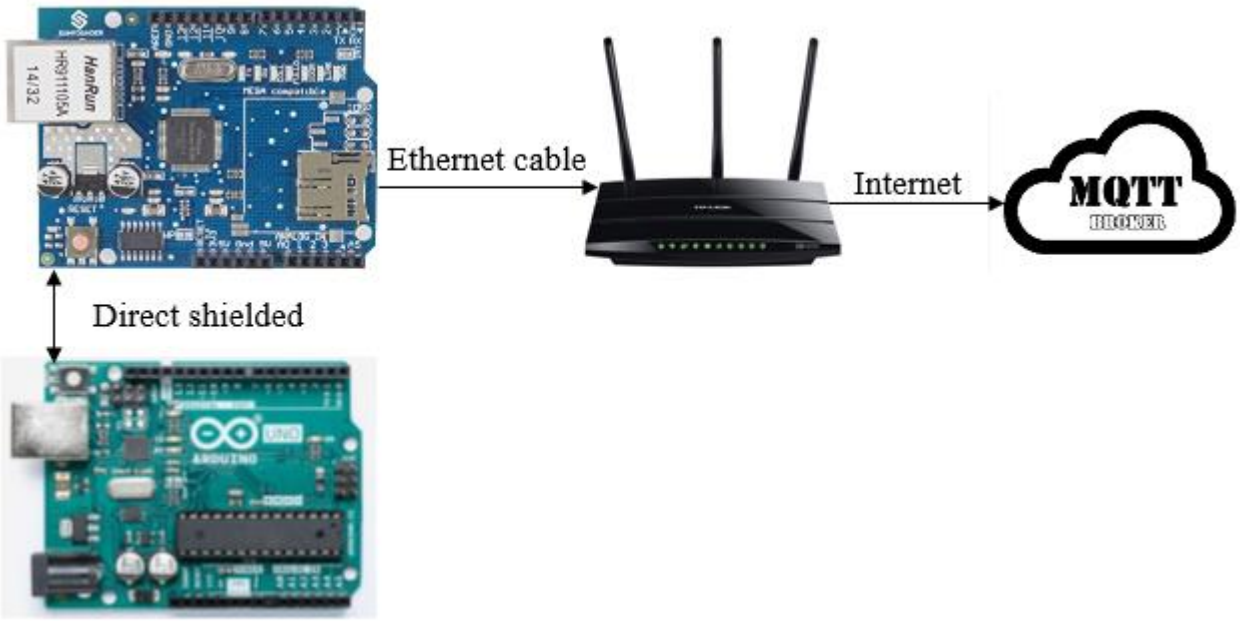
Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

5.3 Σύνδεση Arduino → Broker

Στον προγραμματισμό του Arduino δηλώνουμε τα απαραίτητα στοιχεία του δικτύου που θα συνδεθούμε καθώς και τη διεύθυνση του broker. Αν συνδεθούμε με Ethernet Shield θα πρέπει να δηλώσουμε την mac address της. Αν συνδεθούμε ασύρματα θα πρέπει να δηλώσουμε το όνομα του δικτύου (SSID) και τον κωδικό (password).

5.3.1 Ενσύρματη σύνδεση

Αν έχουμε ενσύρματη σύνδεση Arduino – Router, πρέπει να δηλώσουμε στον κώδικα του Arduino την φυσική διεύθυνση (mac address) καθώς και την ip διεύθυνση της Ethernet shield (επιπλέον πλακέτα άμεσα συνδεδεμένη στο Arduino). Τις δύο αυτές διευθύνσεις μπορούμε να τις βρούμε από το πρόγραμμα διαχείρισης του Router. Η Ethernet shield κουμπώνει άμεσα πάνω στην πλακέτα του Arduino, παρέχοντάς του θύρα Ethernet και διαθέτει υποδοχή για κάρτα microSD. Στο παρακάτω σχέδιο 3 μπορούμε να δούμε την ενσύρματη σύνδεση.



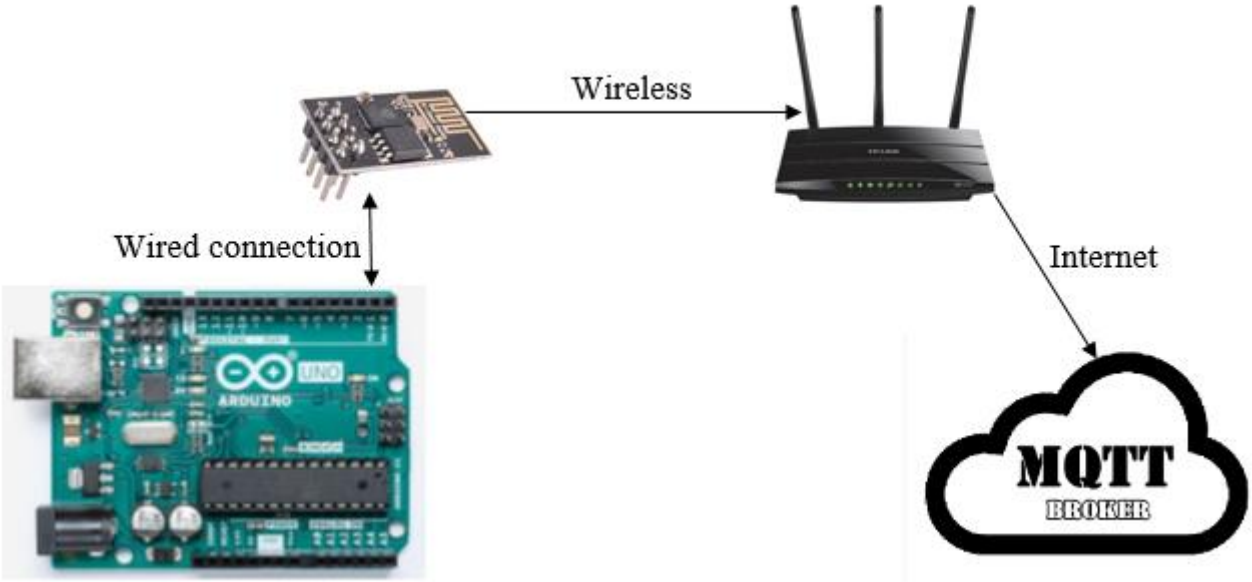
Σχέδιο 3 – Ενσύρματη σύνδεση Arduino - Broker

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

5.3.2 Ασύρματη σύνδεση

Αν έχουμε ασύρματη σύνδεση Arduino – Router, πρέπει να δηλώσουμε στον κώδικα το όνομα του ασύρματου δικτύου (ssid) που θέλουμε να συνδεθούμε και τον κωδικό του (password). Το **D**ynamic **H**ost **C**onfiguration **P**rotocol του Router αναλαμβάνει να δώσει ip διεύθυνση στο Arduino.

Για να συνδέσουμε όμως ασύρματα το Arduino με τον Router χρειαζόμαστε επιπλέον υλικό. Μια φθηνή και αξιόπιστη λύση είναι το ESP8266 WiFi module. Πρέπει να το συνδέσουμε σωστά με τροφοδοσία και pins επικοινωνίας με το Arduino και να το παραμετροποιήσουμε, έτσι ώστε να λειτουργεί ως πομποδέκτης WiFi. Στο παρακάτω σχέδιο 4 μπορούμε να δούμε την ασύρματη σύνδεση.



Σχέδιο 4 – Ασύρματη σύνδεση Arduino - Broker

5.4 ESP8266 WiFi module

Ας δούμε πρώτα τα χαρακτηριστικά του ESP8266 Wifi module. Αυτά είναι:

- Χαμηλού κόστους, συμπαγής και ισχυρή μονάδα Wi-Fi.
- Τροφοδοσία ρεύματος: μόνο 3.3V.
- Κατανάλωση ρεύματος: 100mA.
- I/O τάση: 3.6V (max).
- I/O ρεύμα πηγής: 12mA (max).
- Ενσωματωμένη χαμηλής ισχύος 32-bit MCU @ 80MHz.
- 512kB Flash Memory.
- Μπορεί να χρησιμοποιηθεί ως σταθμός ή σημείο πρόσβασης ή και τα δύο μαζί.
- Υποστηρίζει βαθύ ύπνο (<10uA).
- Υποστηρίζει τη σειριακή επικοινωνία ως εκ τούτου συμβατή με πολλές πλατφόρμες ανάπτυξης όπως το Arduino.
- Μπορεί να προγραμματιστεί χρησιμοποιώντας Arduino IDE ή AT-εντολές ή Lua Script.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

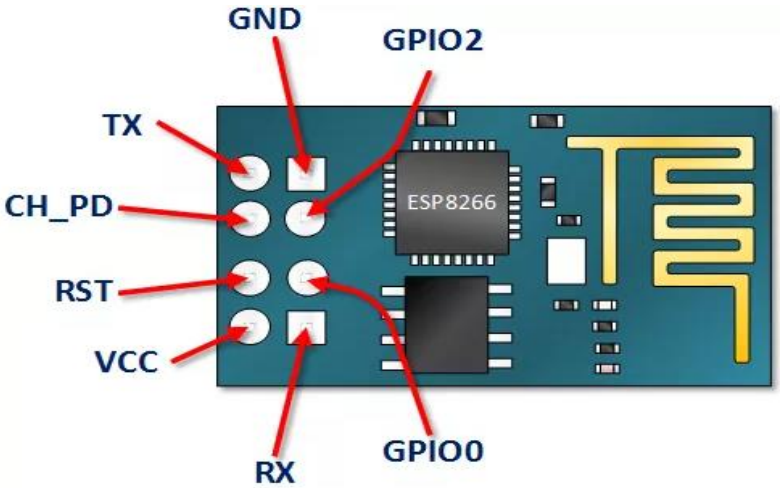
Το ESP8266 διαθέτει τρεις τρόπους λειτουργίας:

1. Station/Σταθμός: να συνδεθεί σε κάποιο Access Point (π.χ. Router) σαν πελάτης/client.
2. Access Point: Να λειτουργήσει το ίδιο σαν Router, όπου μπορούν να συνδεθούν άλλες συσκευές σε αυτό.
3. Οι δύο προηγούμενες λειτουργίες ταυτόχρονα. Station και AP ταυτόχρονα.

Για να συνδέσουμε σωστά το Arduino με το ESP, πρέπει πρώτα τα δούμε την αρχιτεκτονική του ESP και ειδικότερα των ακροδεκτών (pins) του.

5.4.1 Ανάλυση των ακροδεκτών του ESP

Στην παρακάτω εικόνα 21 μπορούμε να δούμε τους ακροδέκτες (pins) του ESP και να τους αναλύσουμε:



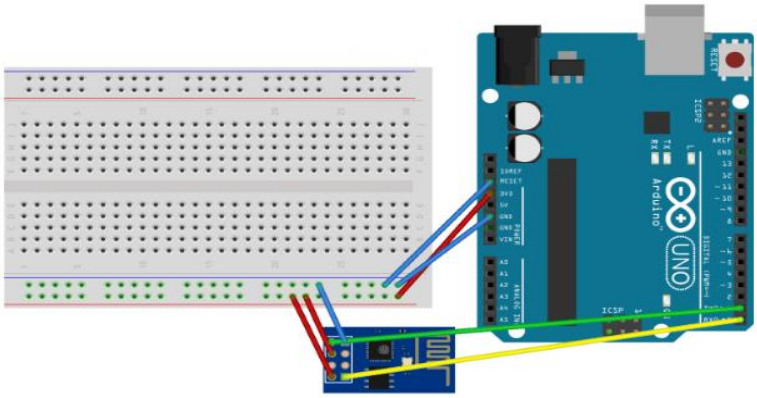
Εικόνα 21 – Ακροδέκτες του ESP8266 (Ηλ.Πηγή: κεφ.5[1])

- TX: Σύνδεση με το RX pin του προγραμματιστή για τη φόρτωση του προγράμματος.
- CH_PD: Ενεργό chip – σε κατάσταση High.
- RST: Επαναφορά του module.
- VCC: Σύνδεση με τάση 3,3V **μόνο**
- GND: Σύνδεση με τη γείωση του κυκλώματος
- GPIO2: Γενικού σκοπού Είσοδος/Εξοδος pin.
- GPIO0: Γενικού σκοπού Είσοδος/Εξοδος pin.
- RX: Γενικού σκοπού Είσοδος/Εξοδος pin.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

5.4.2 Σύνδεση Arduino – ESP (για τη ρύθμιση του ESP)

Η συνδεσμολογία που βλέπουμε στο σχέδιο 5, αφορά την επικοινωνία του χρήστη με το ESP για την παραμετροποίησή του.



Σχέδιο 5 – Σύνδεση Arduino – ESP8266

Τα βήματα είναι τα εξής:

1. Συνδέουμε την έξοδο 3v3 (3.3V) του Arduino στην κόκκινη γραμμή σε ένα breadboard. Το ESP8266 λειτουργεί με 3.3V και όχι με 5V, γι' αυτό είναι απαραίτητο. Αν θέλουμε να συνδέσουμε άλλα εξαρτήματα που χρησιμοποιούν 5V, μπορούμε να συνδέσουμε την έξοδο 5V στην άλλη κόκκινη γραμμή του breadboard, απλά να είμαστε σίγουροι ότι δεν συνδέσαμε τα δύο κόκκινα μαζί.
2. Συνδέουμε το GND (γείωση) στη μπλε γραμμή.
3. Συνδέουμε τον ακροδέκτη RES ή RESET στη μπλε γραμμή. Όταν γειώνουμε τον ακροδέκτη επαναφοράς, το Arduino λειτουργεί ως απλό USB σε σειριακή σύνδεση, κάτι που θέλουμε για να μιλήσουμε στο ESP8266.
4. Συνδέουμε τον ακροδέκτη RXD του Arduino στον ακροδέκτη RX του ESP8266 (κίτρινο χρώμα στο σχέδιο).
5. Συνδέουμε τον ακροδέκτη TXD του Arduino με τον ακροδέκτη TX του ESP (πράσινο χρώμα στο σχέδιο). Συνήθως, όταν θέλουμε δύο πράγματα να μιλάνε το ένα στο άλλο μέσω σειριακής σύνδεσης, συνδέουμε τον ακροδέκτη TX του ενός με τον RX του άλλου (η αποστολή του ενός πηγαίνει στη λήψη του άλλου και το αντίθετο). Εδώ δεν έχουμε συνομιλία του Arduino με το ESP8266, όμως ο υπολογιστής μας μιλάει σε αυτό μέσω του Arduino.
6. Συνδέουμε τον ακροδέκτη GND του ESP στην μπλε γραμμή και τον ακροδέκτη VCC στην κόκκινη γραμμή.
7. Τέλος, το CH_PD πηγαίνει στην κόκκινη γραμμή, υποτίθεται ότι δεν θα λειτουργήσει αν δεν το συνδέσουμε.

Παρατήρηση: Και αφού μιλήσαμε για τον ακροδέκτη CH_PD, πρέπει να θυμόμαστε ότι αν θέλουμε να επανεγγράψουμε το firmware του ESP8266 θα πρέπει να συνδέσουμε τον ακροδέκτη GPIO0 στο GND (μπλε γραμμή), που θέτει το ESP σε λειτουργία flash.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

5.4.3 Παραμετροποίηση ESP μέσω Arduino IDE

Στο Arduino IDE, δεν χρειάζεται να επιλέξουμε πλακέτα, καθώς δεν φορτώνουμε τίποτα στο ESP8266. Απλά επιλέγουμε τη σωστή θύρα στο μενού Εργαλεία και μεταβαίνουμε στην επιλογή Εργαλεία → Σειριακή Παρακολούθηση. Στη συνέχεια, ρυθμίζουμε το ρυθμό μετάδοσης σε 115200 (το βασικό firmware ESP8266 το χρησιμοποιεί) ή 9600 και τα τερματικά γραμμής Both NL & CR.

Εάν πληκτρολογήσουμε AT στο πεδίο μηνύματος και πατήσουμε enter, πρέπει να απαντήσει με το OK. Υπάρχει ένα μεγάλο σύνολο εντολών που συνοδεύει το ESP8266, που μπορούμε να δούμε στον παρακάτω σύνδεσμο: https://room-15.github.io/blog/2015/03/26/esp8266-at-command-reference/#at-commands και μπορούμε με αυτές να επιλέξουμε τη λειτουργία που επιθυμούμε για το ESP8266 καθώς και να ρυθμίσουμε πλήθος άλλων παραμέτρων. Στο Παράρτημα 3, μπορούμε να δούμε έναν συνοπτικό (βασικό) πίνακα εντολών για την παραμετροποίηση του ESP8266.

5.4.4 Σύνδεση Arduino – ESP (για την εκτέλεση προγράμματος)

Κατά την εκτέλεση του προγράμματος του Arduino, πιθανώς να συνδέσουμε το ESP (Rx και Tx) με τους ψηφιακούς ακροδέκτες 2 και 3, ως Tx και Rx του Arduino.

5.5 Σύνδεση Arduino – Υποδοχή SD κάρτας

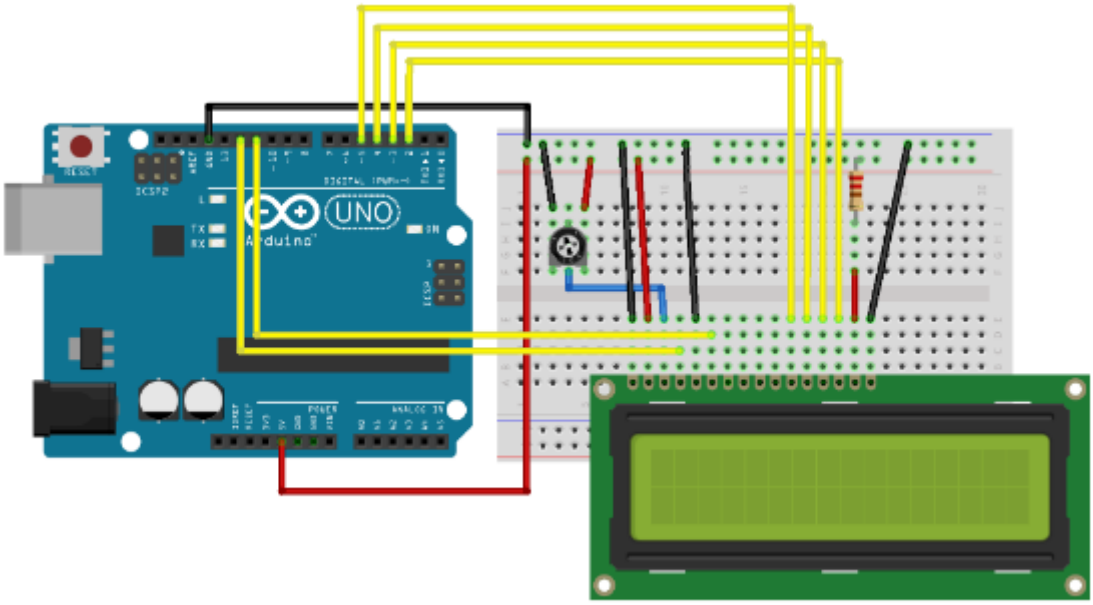
Η υποδοχή της κάρτας μνήμης (microSD) είναι ενσωματωμένη στο Ethernet shield που συνδέεται άμεσα με το Arduino. Η επικοινωνία όμως μεταξύ του μικροελεγκτή και της κάρτας SD χρησιμοποιεί το SPI, το οποίο λαμβάνει χώρα στους ψηφιακούς ακροδέκτες 11, 12 και 13 (στις περισσότερες πλακέτες Arduino) ή 50, 51 και 52 (Arduino Mega). Επιπλέον, πρέπει να χρησιμοποιήσουμε άλλον έναν ακροδέκτη για να επιλέξουμε την κάρτα SD. Αυτός μπορεί να είναι ο ακροδέκτης υλικού SS – ψηφιακός ακροδέκτης 10 (στις περισσότερες πλακέτες Arduino) ή τον ακροδέκτη 53 (στο Mega) - ή άλλον ακροδέκτη που καθορίζεται κατά την κλήση του SD.begin (). Σημειώνουμε ότι ακόμα και αν δεν χρησιμοποιήσουμε τον ακροδέκτη SS υλικού, πρέπει να παραμείνει ως έξοδος αλλιώς η βιβλιοθήκη SD δεν θα λειτουργήσει.

Άρα οι ψηφιακοί ακροδέκτες: 10, 11, 12 και 13 του Arduino δεν πρέπει να συνδεθούν με άλλο κύκλωμα, έτσι ώστε να λειτουργήσει η κάρτα μνήμης.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

5.6 Σύνδεση Arduino – Οθόνης Υγρών Κρυστάλλων (LCD)

Οι συμβατές με Hitachi LCD οθόνες μπορούν να ελέγχονται σε δύο λειτουργίες: 4-bit ή 8-bit. Η λειτουργία 4-bit απαιτεί 7 ακροδέκτες εισόδου/εξόδου από το Arduino, ενώ η λειτουργία 8-bit απαιτεί 11 ακροδέκτες. Στο σχέδιο 6 βλέπουμε τη συνδεσμολογία του Arduino και της LCD.



Σχέδιο 6 – Κύκλωμα σύνδεσης LCD – Arduino (Ηλ.Πηγή: κεφ.5[5])

Για τη σύνδεση της Οθόνης Υγρών Κρυστάλλων θα χρειαστούμε τους 4, 5, 6, 7, 8 και 9 ψηφιακούς ακροδέκτες του Arduino.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

6. Ροή και Επεξεργασία Δεδομένων

Στο σύστημα που μελετάμε και υλοποιούμε έχουμε δύο ροές δεδομένων. Η πρώτη ροή αφορά τις τιμές που μετράει το σύστημα της Festo (Ladder) και μέσω Arduino, broker και Node-Red φτάνουν στον απομακρυσμένο χειριστή, οπουδήποτε στον κόσμο. Η δεύτερη ροή δεδομένων ξεκινά από το περιβάλλον του χειριστή (user interface της Node-Red) και αφορά τις εντολές ελέγχου που καταλήγουν στο σύστημα της Festo.

6.1 Μορφή Τιμών

Τα αρχεία που παράγονται από την προσομοίωση του συστήματος της Festo, στο πρόγραμμα LabView του συναδέλφου μου Βλάση Χονδρογιάννη, έχουν ομαδοποίηση κατά ημέρα και όλες οι ημέρες μετρήσεων του ίδιου μήνα περιλαμβάνονται στον ίδιο φάκελο. Η μορφή του αρχείου είναι ως εξής:

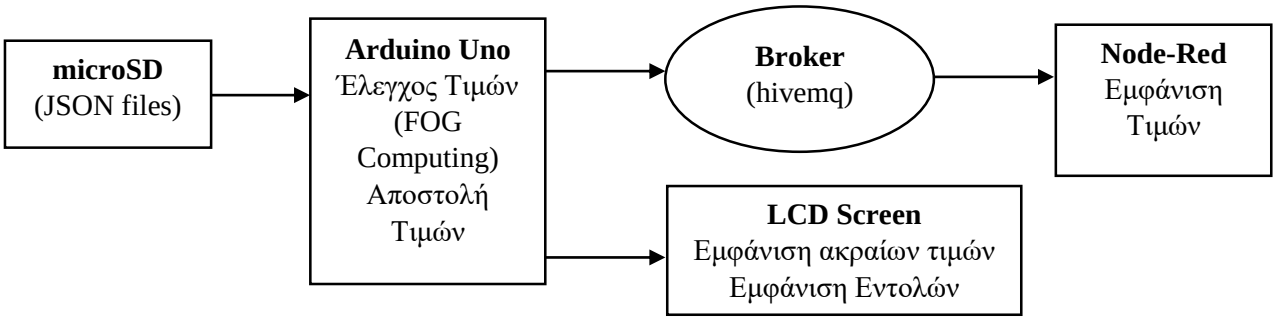
[{"Case 1 ":"Circulation","Date":"02/04/2019","Time":"09:35","Temp":18,"Flow":0.01},
{"Case 1 ":"Circulation","Date":"02/04/2019","Time":"09:36","Temp":18,"Flow":0.09},
{"Case 1 ":"Circulation","Date":"02/04/2019","Time":"09:37","Temp":18,"Flow":0.25},
{"Case 1 ":"Circulation","Date":"02/04/2019","Time":"09:38","Temp":21,"Flow":0.65},
{"Case 1 ":"Circulation","Date":"02/04/2019","Time":"09:39","Temp":21,"Flow":1.25},
{"Case 1 ":"Circulation","Date":"02/04/2019","Time":"09:40","Temp":22,"Flow":1.55},
{"Case 1 ":"Circulation","Date":"02/04/2019","Time":"09:41","Temp":25,"Flow":2.00},
{"Case 1 ":"Circulation","Date":"02/04/2019","Time":"09:42","Temp":26,"Flow":2.00},
{"Case 1 ":"Circulation","Date":"02/04/2019","Time":"09:43","Temp":30,"Flow":2.00},
{"Case 1 ":"Circulation","Date":"02/04/2019","Time":"09:44","Temp":30,"Flow":2.00}]

Είναι τύπου json, αλλά εξαιτίας του Προβλήματος N_o 1 που αναφέρω στο επόμενο κεφάλαιο, έπρεπε να τα μετατρέψω σε txt. Εύκολα διαπιστώνουμε ότι κάθε γραμμή περιέχει δύο μετρήσεις (π.χ. θερμοκρασία και ροή) και τις τιμές αυτών. Παρατηρούμε ότι η πρώτη γραμμή ξεκινά με αγκύλη(‘[‘), αυτό σημαίνει διαφορετικό σημείο εκκίνησης της ανάγνωσης και αυξάνει την πολυπλοκότητα του κώδικα.

Συνοψίζοντας, σε ό,τι αφορά το αρχείο καταγραφής των τιμών, η κάθε γραμμή του περιέχει δύο μετρήσεις, τις οποίες πρέπει το πρόγραμμα να διαβάζει σωστά.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

6.2 Ροή και Επεξεργασία Τιμών



Σχέδιο 7 – Συνολική ροή τιμών

Στο παραπάνω σχέδιο 7 βλέπουμε τη διαδρομή των τιμών. Οι τιμές, βέβαια, παράγονται στους αισθητήρες του συστήματος της Festo ή εναλλακτικά στο πρόγραμμα Ladder, όπου προσομοιώνεται η λειτουργία του Festo, και αποθηκεύονται σε κάρτα μνήμης microSD σε μορφή JSON. Το πρόγραμμα ξεκινά από την κάρτα μνήμης, όπου είναι αποθηκευμένα τα αρχεία με τις τιμές των μετρήσεων. Η ροή και η επεξεργασία των δεδομένων γίνονται από το ίδιο πρόγραμμα, αλλά, για την καλύτερη κατανόηση, τα παραθέτω ξεχωριστά.

6.2.1 Ροή Τιμών

Αφετηρία μας, λοιπόν, είναι η κάρτα μνήμης που περιέχει τις τιμές των μετρήσεων.

- ➔ Τοποθετώντας την κάρτα μνήμης στην κατάλληλη υποδοχή που διαθέτει η Ethernet shield, έχουμε τα δεδομένα μας στο Arduino.
- ➔ Κατόπιν ελέγχου των τιμών, αυτές που είναι εκτός ορίων εμφανίζονται στην οθόνη LCD του Arduino μαζί με εντολή ελέγχου (Fog Computing).
- ➔ Το Arduino, με τον κατάλληλο προγραμματισμό, στέλνει (publish) τα δεδομένα στα αντίστοιχα πεδία (topics) του Broker.
- ➔ Το Node-Red εγγράφεται (subscribe) σε αυτά τα πεδία και διαβάζει τις τιμές κάθε φορά που αυτές αλλάζουν και τις εμφανίζει στο περιβάλλον του χειριστή.

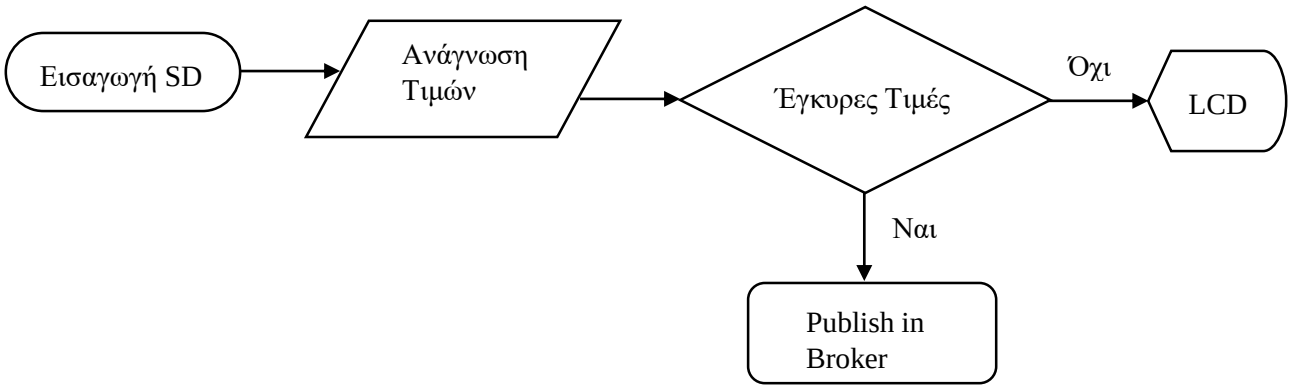
Σε αυτό το σημείο να υπενθυμίσω ότι η κάρτα μνήμης microSD, αντικαθιστά τη λειτουργία της σειριακής επικοινωνίας PLC – Arduino, όσον αφορά τη ροή των δεδομένων.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

6.2.2 Επεξεργασία Τιμών

Η επεξεργασία και ο έλεγχος των τιμών γίνονται στο Arduino. Το Arduino θα πρέπει:

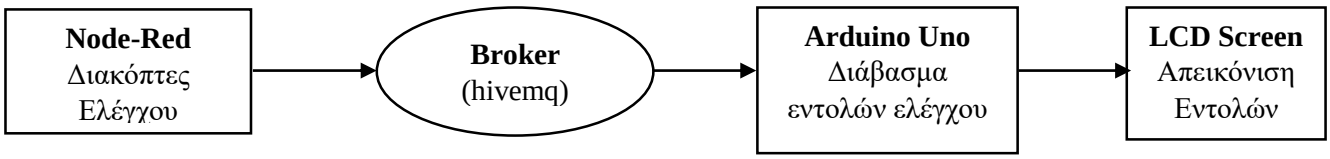
- ➔ Να εξάγει τα δεδομένα από την κάρτα μνήμης, αποκωδικοποιώντας τα αρχεία JSON (διαβάζοντας τα αρχεία txt) και να κρατήσει τα ζευγάρια: μέτρηση – τιμή. Παράδειγμα: Temp – 23.
- ➔ Να ελέγξει για κάθε μέτρηση, με βάση προκαθορισμένα όρια τιμών λειτουργίας, αν η τιμή βρίσκεται εντός ή εκτός αυτών των ορίων:
    - Αν είναι εκτός ορίων τις εμφανίζει στην οθόνη LCD με μήνυμα για πιθανό σφάλμα του αισθητήρα και συνεχίζει με την επόμενη μέτρηση (Fog Computing).
    - Αν είναι εντός ορίων συνεχίζει την αποστολή της στον broker.



Σχέδιο 8 – Ροή και έλεγχος τιμών - Arduino

6.3 Ροή Εντολών

Στο παρακάτω σχέδιο 9 βλέπουμε τη διαδρομή των εντολών (ON-OFF). Αφετηρία αυτής της διαδρομής είναι οι διακόπτες ελέγχου στο περιβάλλον χρήστη της Node-Red.



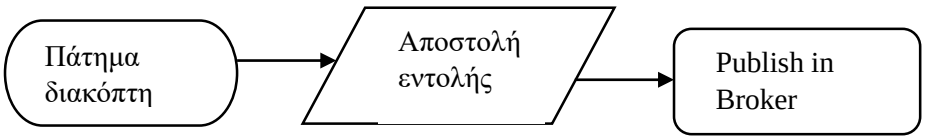
Σχέδιο 9 – Συνολική ροή Εντολών

- ➔ Πατώντας τον διακόπτη ελέγχου στέλνεται (publish) το μήνυμα (ON-OFF) στο αντίστοιχο topic του broker.
- ➔ Το Arduino, με τον κατάλληλο προγραμματισμό (μέθοδος CallBack), είναι σε θέση να διαβάσει (subscribe) τα μηνύματα από τα αντίστοιχα topics.
- ➔ Το Arduino εμφανίζει στην οθόνη LCD μηνύματα, σύμφωνα με τις εντολές που διάβασε από τον broker.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Ο κάθε διακόπτης είναι συνδεδεμένος με στοιχείο publish συγκεκριμένου topic. Άρα όταν πατάμε τον διακόπτη είτε για ON είτε για OFF, φεύγει ένα ζευγάρι: topic + ON/OFF για τον broker. Τα topics φαίνονται στο Σχέδιο 1 της παραγράφου 4.3.

Παράδειγμα: Για να θερμάνουμε το νερό της δεξαμενής 1, πρέπει να πατήσουμε ON τον διακόπτη Heat. Με το πάτημα θα αποσταλεί το μήνυμα **ON** στο topic: **FestoLar/Heat**.



Σχέδιο 10 – Ροή Εντολών Node-Red

Σε αυτό το σημείο να υπενθυμίσω ότι η οθόνη LCD αντικαθιστά τη λειτουργία της σειριακής επικοινωνίας PLC – Arduino, όσον αφορά τις εντολές ελέγχου προς τους drivers του PLC.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

7. Τεχνικά ζητήματα, Λύσεις και Συμπεράσματα

Στην πορεία υλοποίησης της παρούσας εργασίας διδαχθήκαμε πολλά μαθήματα. Ξεκινώντας από την ιδέα της εργασίας, πώς δηλαδή έναν εκπαιδευτικό σταθμό εργασίας μετρήσεων και ελέγχου μπορούμε να τον εντάξουμε στο περιβάλλον IoT. Αναλύσαμε τις απαιτήσεις και οργανώσαμε τα υλικά και τα άυλα μέσα υλοποίησης που χρειαζόμασταν. Αυτά, εν συντομία, ήταν:

- Ηλεκτρονικός υπολογιστής στον οποίο υλοποιήσαμε την πλατφόρμα Node-Red, ως πιστή αναπαράσταση του εκπαιδευτικού σταθμού της Festo· επίσης χρησιμοποιήσαμε και το βοηθητικό πρόγραμμα Eclipse Paho.
- Arduino Uno και τα απαραίτητα περιφερειακά του συστήματα (οθόνη LCD, Ethernet shield, ESP 8266) ως το κύριο μέσο (gateway) διασύνδεσης του σταθμού μετρήσεων με το διαδίκτυο.
- Προγραμματισμός της Node-Red (με τη μέθοδο drag & drop) και παραμετροποίηση των απαραίτητων πεδίων για τη σωστή σύνδεση με τον Broker.
- Προγραμματισμός Arduino μέσω του περιβάλλοντος IDE, έτσι ώστε να εκπληρώνει τον σκοπό του.

Η ουσία λοιπόν αυτής της εργασίας είναι: Οι τιμές των μετρήσεων του σταθμού της Festo να φτάνουν σε κάποιον απομακρυσμένο χειριστή, ο οποίος μέσω της Node-Red θα μπορεί να βλέπει τις τιμές σε σχεδόν πραγματικό χρόνο και να στέλνει πίσω στο σύστημα εντολές ελέγχου (ON-OFF), και ο σταθμός μετρήσεων να λαμβάνει αυτές τις εντολές.

Το πρώτο τεχνικό εμπόδιο εμφανίστηκε όταν μάθαμε ότι ο σταθμός της Festo **δεν** μπορεί να συνδεθεί και να επικοινωνήσει με κανένα εξωτερικό μέσο. Δεδομένου αυτού, έπρεπε να βρούμε λύσεις προσομοίωσης. Έτσι λοιπόν οι τιμές των μετρήσεων ξεκινάνε από μια κάρτα microSD, όπου εξάγονται και αποθηκεύονται τα αρχεία μετά την προσομοίωση διάφορων σεναρίων λειτουργίας του σταθμού από τον συνάδελφο Βλάση Χονδρογιάννη. Οι εντολές ελέγχου καταλήγουν στην οθόνη LCD που είναι συνδεδεμένη στο Arduino.

Η διαδικασία σύνδεσης των περιφερειακών συσκευών του Arduino, αλλά περισσότερο ο προγραμματισμός τους, ήταν πολύπλοκη και επίπονη. Είχαμε να κάνουμε με hardware για το οποίο έπρεπε να επιλέξουμε τις σωστές βιβλιοθήκες υποστήριξης και να μάθουμε τις δυνατότητές τους. Τις λειτουργίες, δηλαδή, που παρέχουν και κατά πόσο αυτές είναι αναγκαίες στον προγραμματισμό του Arduino. Αναζητήσαμε λύσεις σε διάφορα fora και ιστοσελίδες, γι’ αυτό και ο χρόνος που καταναλώσαμε στον προγραμματισμό του Arduino ήταν ίσως και το 80% του συνολικού χρόνου υλοποίησης της εργασίας.

Στην πλατφόρμα Node-Red η επιλογή των κατάλληλων απεικονιστικών στοιχείων ήταν σαφώς μια ευκολότερη διαδικασία. Προσοχή έπρεπε να επιδείξουμε στην επιλογή των topics και στα πεδία σύνδεσης με τον broker. Το πιο δύσκολο κομμάτι στον προγραμματισμό της Node-Red ήταν η

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

χωροθέτηση όλων των απαραίτητων στοιχείων, έτσι ώστε να προκύψει ένα λειτουργικό interface για τον χρήστη.

Στην προσπάθεια υλοποίησης της εργασίας προέκυψαν κάποια τεχνικά προβλήματα, τόσο στην σύνδεση του υλικού (hardware), όσο και στον προγραμματισμό (software) του συστήματος. Θα αναφέρουμε μερικά δύσκολα προβλήματα και τις λύσεις που προτάθηκαν και εφαρμόστηκαν.

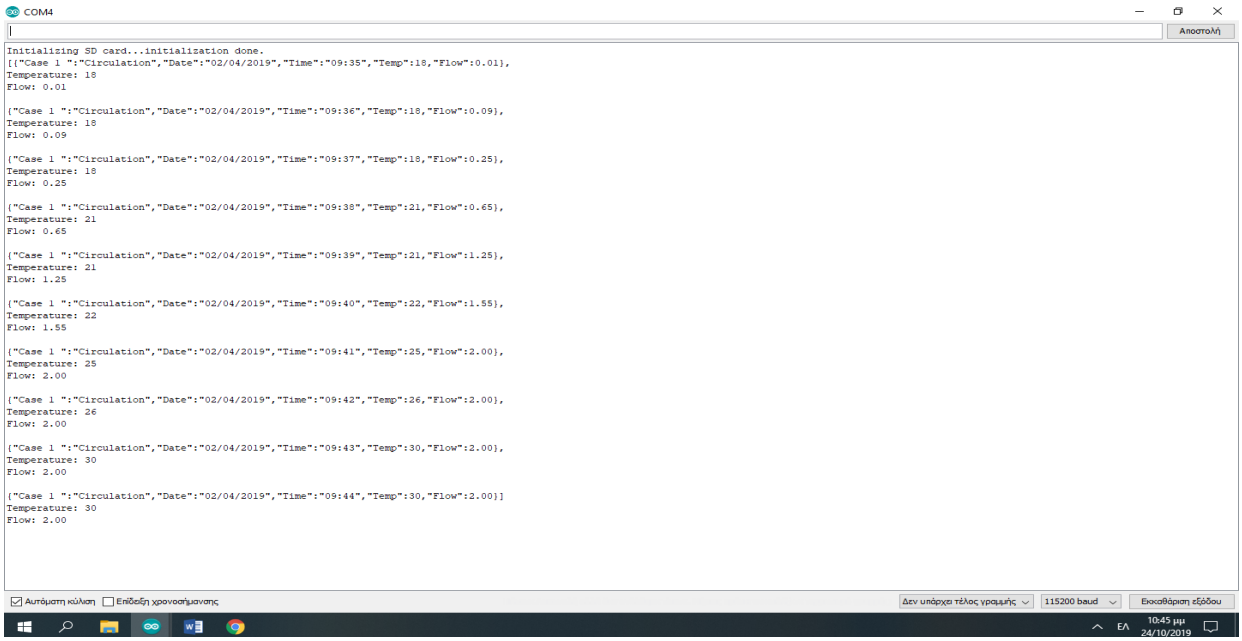
Πρόβλημα Nº 1: Η SDlibrary που χρησιμοποιούμε για να διαβάσουμε και να γράψουμε αρχεία στην κάρτα μνήμης, μπορεί να διαβάζει ονόματα αρχείων 8.3. Αυτό σημαίνει ότι το όνομα του αρχείου δεν μπορεί να αποτελείται από περισσότερα των 8 στοιχεία και ο τύπος του αρχείου περισσότερα από 3 στοιχεία. Στην πράξη αυτό σημαίνει ότι το αρχείο 1april.**txt** το διαβάζει κανονικά και το αρχείο 1april.**json** δεν μπορεί να το διαβάσει.

Λύση 1: Μελετάμε άλλες βιβλιοθήκες που θα μπορούσαμε με αυτές να διαβάσουμε αρχεία .json.
Λύση 2: Μετατρέπουμε χειροκίνητα τα αρχεία .json σε .txt, πριν τα αποθηκεύσουμε στην κάρτα μνήμης.

Επιλέξαμε τη **Λύση 2** διότι:

- Άλλες βιβλιοθήκες που βρήκαμε (sdfat, sdfat – adafruit fork) είναι σε beta έκδοση και έχουν σφάλματα σε εσωτερικές τους βιβλιοθήκες που δεν μπορούμε να διορθώσουμε.
- Μετά από έρευνα σε διάφορα fora καταλάβαμε ότι το Arduino δεν έχει πολλές δυνατότητες να επεξεργάζεται Strings.

Με τον κώδικα που παραθέτουμε στο Παράρτημα 1 – Ανάγνωση Δεδομένων από την κάρτα SD, μπορούμε να πάρουμε τις τιμές–μετρήσεις, από το αρχείο της κάρτας SD. Στην εικόνα 22 βλέπουμε τα αποτελέσματα της ανάγνωσης δεδομένων στο serial monitor του Arduino IDE.



Εικόνα 22 - PrtSc του Serial Monitor: Ανάγνωση και εμφάνιση τιμών

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Το αρχείο που διαβάζει ο κώδικας είναι .txt και περιέχει συγκεκριμένα μετρήσεις Θερμοκρασίας και Ροής. Γι' αυτό είναι σύντομος και επικεντρωμένος σε αυτά τα δύο είδη μετρήσεων.

Διευκρίνηση

Οι τιμές που φαίνονται είναι σε αριθμητική μορφή και όχι χαρακτήρες (char/strings), άρα μπορούν να συμμετέχουν σε αριθμητικές πράξεις.

Πρόβλημα Nº 2: Ταυτόχρονη λειτουργία SD card και επεξεργαστή w5100 wiznet του Ethernet shield. Επειδή και οι δύο περιφερειακές συσκευές μοιράζονται το **S**erial **P**eripheral **I**nterface bus του Ethernet shield, δεν μπορούν να είναι ενεργές ταυτόχρονα.

Η λύση που προτείνεται είναι να απενεργοποιείται κάθε φορά η συσκευή που δεν χρησιμοποιείται. Για να απενεργοποιηθεί η **SD** κάρτα πρέπει να τεθεί σε υψηλό (**High**) ο ψηφιακός ακροδέκτης **4** του Arduino. Για να απενεργοποιηθεί το κύκλωμα **Ethernet** (W5100) πρέπει να τεθεί σε υψηλό (**High**) ο ψηφιακός ακροδέκτης **10** του Arduino. (Ηλ.Πηγή: κεφ.5[6])

Αυτή η λύση δεν βοηθάει καθόλου τη δομή και τις απαιτήσεις του προγράμματος, αφού πρέπει να υπάρχει μια μόνιμη και σταθερή σύνδεση στο διαδίκτυο έτσι ώστε οι τιμές που αναγιγνώσκονται να στέλνονται αμέσως στον broker. Δεν πρέπει δηλαδή να σταματάει η μία συσκευή για να λειτουργεί η άλλη. Παρόλα αυτά δοκίμασα και την Ethernet2.h βιβλιοθήκη μήπως λύνει αυτό το πρόβλημα, αλλά **δεν** το κατάφερε, αν και μειώθηκε η χρήση μνήμης 95% → 88%. Συνεχίζοντας την αναζήτηση λύσης και σε διάφορα fora, η λύση που προτείνουν είναι να συνδεθεί το Arduino στο διαδίκτυο ασύρματα με κάποιο wifi module όπως το Esp8266.

Πρόβλημα Nº 3: Χωρητικότητα μνήμης Arduino. Ολοκληρώνοντας την ανάπτυξη του κώδικα διαπίστωσα ότι:

«*Το σχέδιο χρησιμοποιεί 33270 bytes (103%) του χώρου αποθήκευσης του προγράμματος. Το μέγιστο είναι 32256 bytes.*
Οι καθολικές μεταβλητές χρησιμοποιούν 1993 bytes (97%) δυναμικής μνήμης, αφήνοντας 55 bytes για τοπικές μεταβλητές. Το μέγιστο είναι 2048 bytes.
Πολύ μεγάλο σχέδιο, δες συμβουλές στο http://www.arduino.cc/en/Guide/Troubleshooting#size για την μείωση του.
Σφάλμα μεταγλώττισης για την πλακέτα Arduino/Genuino Uno.»

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Για να μειώσουμε τον όγκο του προγράμματος σύμφωνα με τις συμβουλές που δίνονται στον σύνδεσμο: https://www.arduino.cc/en/Guide/Troubleshooting#size μπορούμε να κάνουμε τα εξής:

1. Να αλλάξουμε τις μεταβλητές κινητής υποδιαστολής (floating point) σε ακέραιες (Integer).
2. Να σβήσουμε τυχόν βιβλιοθήκες που δεν χρησιμοποιούνται στο πρόγραμμα (#include…)
3. Να μικρύνουμε το πρόγραμμα.

Δράσεις:

1. Άλλαξα όσες μεταβλητές ήταν δυνατόν από double σε integer.
2. Έλεγξα τις βιβλιοθήκες και όλες όσες υπάρχουν είναι χρήσιμες και απαραίτητες.
3. Μίκρυνα τον κώδικα, θέτοντας μεγάλα κομμάτια του σε σχόλια.
4. Μίκρυνα περαιτέρω τον κώδικα, αλλάζοντας και τη μορφή του αρχείου που διαβάζει και δεν διορθώθηκε κάτι σημαντικό (95% → 94%).

Αποτέλεσμα:

«*Το σχέδιο χρησιμοποιεί 30706 bytes (95%) του χώρου αποθήκευσης του προγράμματος. Το μέγιστο είναι 32256 bytes.*

Οι καθολικές μεταβλητές χρησιμοποιούν 1827 bytes (89%) δυναμικής μνήμης, αφήνοντας 221 bytes για τοπικές μεταβλητές. Το μέγιστο είναι 2048 bytes.

Λίγη διαθέσιμη μνήμη, μπορεί να προκύψουν προβλήματα ευστάθειας.»

Κρίνοντας από όσα εμφανίζονται στο serial monitor, όταν τρέχει το πρόγραμμα, όντως δεν αποδίδει σταθερά αποτελέσματα.

Η **επόμενη λύση** που σκεφτήκαμε ήταν να «σπάσουμε» τον κώδικα σε δύο κομμάτια:

1. Να διαβάζει το αρχείο και να στέλνει τις τιμές στον broker, μειώνοντας περισσότερο τον κώδικα, ώστε να διαβάζει μία «τοποθεσία» χαρακτήρων και όχι δύο όπως ο αρχικός κώδικας, αλλάζοντας παράλληλα και τη μορφή του αρχείου (καταργώντας την πρώτη αγκύλη, στην πρώτη γραμμή). Τα αποτελέσματα ήταν όμοια με την προηγούμενη προσπάθεια μείωσης του κώδικα.
2. Να περιμένει τις εισερχόμενες εντολές από την πλατφόρμα Node-Red. Πρόβλημα χώρου μνήμης δεν αντιμετωπίσαμε σε αυτό το κομμάτι.

Αυτή η λύση βέβαια δεν είναι πραγματική λύση, διότι το Arduino Uno τρέχει σειριακά τον κώδικα άρα δύο ξεχωριστοί κώδικες είναι αδύνατο να τρέχουν παράλληλα, ώστε να παραχθεί το προσδοκώμενο αποτέλεσμα.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Τελικό συμπέρασμα

Στην εργασία αυτή θελήσαμε να μελετήσουμε τον σταθμό μετρήσεων και ελέγχου της Festo και να τον συνδέσουμε στο περιβάλλον IoT. Θέλαμε δηλαδή οι μετρήσεις να εξαχθούν στο διαδίκτυο και οι εντολές ελέγχου από το διαδίκτυο να φτάσουν στον σταθμό της Festo. Καταλήξαμε σε προσομοίωση αυτής της διαδικασίας εξαιτίας του περιορισμού που διαθέτει ο σταθμός και δεν μπορεί να επικοινωνήσει με άλλες συσκευές.

Για τον λόγο αυτό έπρεπε να στήσουμε ένα middleware. Το Arduino Uno λοιπόν έπρεπε να προγραμματιστεί κατάλληλα ώστε να λαμβάνει τις τιμές, να τις ελέγχει και να τις στέλνει στο διαδίκτυο. Καθώς επίσης και να δέχεται τα εισερχόμενα μηνύματα των εντολών ελέγχου και να τα προωθεί στο σύστημα της Festo.

Η σχεδίαση και υλοποίηση της πλατφόρμας Node-Red έπρεπε να αναπαριστά τον σταθμό ελέγχου της Festo, έτσι ώστε ο χειριστής της να αντιλαμβάνεται άμεσα και κατανοητά τις μετρήσεις και να δίνει τις απαραίτητες εντολές ελέγχου χωρίς καθυστέρηση.

Η μελέτη του υλικού, των βιβλιοθηκών και των συνδεσμολογιών αυτών κατέδειξε πόσο δύσκολο είναι να προγραμματίσουμε ένα middleware, όπως το Arduino για να κάνει περισσότερες της μίας εργασίες. Οι τεχνολογικοί περιορισμοί του Arduino Uno και των περιφερειακών συσκευών του καθώς και η υποστήριξη του υλικού μέσω των βιβλιοθηκών του δεν βοηθούν στην ολοκλήρωση του παρούσας εργασίας. Δυστυχώς το Arduino Uno δεν είναι το κατάλληλο υλικό για την υποστήριξη του κώδικα που σχεδιάσαμε και υλοποιήσαμε.

Σε μια μελλοντική εργασία θα μπορούσαμε να δοκιμάσουμε ισχυρότερα υλικά μέσα όπως το Arduino Mega ή το Raspberry και ενδεχομένως να εντάσσαμε και κάποιο RTOS, έτσι ώστε να γίνει περισσότερο αποδοτική και άμεση η ανταπόκριση του συστήματος.

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Βιβλιογραφία

[1] Alexandros Preventis, Kostas Stravoskoufos, Stelios Sotiriadis and Euripides G. M. Petrakis, "IoT-A and FIWARE: Bridging the Barriers between the Cloud and IoT Systems Design and Implementation", Chania - Greece 2016

[2] Aymen Abdullah Alsaffar, Hung Phuoc Pham, Choong-Seon Hong, Eui-Nam Huh, and Mohammad Aazam, "An Architecture of IoT Service Delegation and Resource Allocation Based on Collaboration between Fog and Cloud Computing", 2016

[3] R A Atmoko, R Riantini, and M K Hasin, "IoT real time data acquisition using MQTT protocol", Indonesia 2017

[4] P. Ferrari, E. Sisinni, D. Brandão, M. Rocha, "Evaluation of communication latency in Industrial IoT Applications", 978-1-5090-5679-8/17/$31.00 ©2017 IEEE

[5] Goiuri Peralta, Markel Iglesias-Urkia, Marc Barcelo, Raul Gomez, Adrian Moran and Josu Bilbao, "Fog Computing Based Efficient IoT Scheme for the Industry 4.0", Spain

[6] Ghyzlane Cherradi, Adil el Bouziri, Azedine Boulmakoul, "Smart Data Collection Based on IoT Protocols", Conference Paper, December 2016

[7] Dipa Soni, Ashwin Makwana, "A survey on MQTT: a protocol of internet of things(IoT)", India 2017

[8] Veeramanikandan M, Suresh Sankaranarayanan, "Publish/Subscribe Broker based Architecture for Fog Computing", International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017)

[9] Jurgen Helmich, ADIRO - MPS® PA Compact Workstation Manual, Esslingen Germany, 12/2008

[10] Nandor Verba, Kuo-Ming Chao, Jacek Lewandowski, Nazaraf Shah, Anne James, Feng Tian, "Modelling Industry 4.0 based Fog Computing environments for Application analysis and Deployment", September 4, 2018

[11] Jakub Pizoń, Jerzy Lipski, "Perspectives for Fog Computing in Manufacturing", *Applied Computer Science*, vol. 12, no. 3, pp. 37–46, September 2016

[12] Pérez D, Alarcón F, Boza A, "Industry 4.0: A classification scheme", International Joint Conference - CIO-ICIEOM-IIE-AIM (IJC 2016), San Sebastián, Spain, July 13-15, 2016

[13] Andreja Rojko, "Industry 4.0 Concept: Background and Overview", *International Journal of Interactive Mobile Technologies*, Vol 11, No 5 (2017)

[14] Dragan Vuksanović, Jelena Ugarak, Davor Korčok, "INDUSTRY 4.0: the future concepts and new visions of factory of the future development", Conference Paper, January 2016 DOI: 10.15308/Sinteza-2016-293-298

[15] Daniel Happ, Niels Karowski, Thomas Menzel, Vlado Handziski, Adam Wolisz, "Meeting IoT Platform Requirements with Open Pub/Sub Solutions", July 14, 2016, available at http://link.springer.com

[16] M. Sruthi, B. R. Kavitha, "A Survey On Iot Platform", *IJSRME ISSN: 2455 – 5630 Volume I, Issue I*, 2016

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Ηλεκτρονικές Πηγές

Κεφάλαιο 2:

[1] https://www.gslab.com/blog-post/how-manufacturing-should-leverage-industry-4-0/

[2] https://www.headland.com.au/industry-4-0-cheat-sheet-industry4-0-cloud-computing-smart-factory-internet-of-things/

[3] https://au.pcmag.com/networking-communications-software/29902/what-is-cloud-computing

[4] https://leanbi.ch/en/blog/iot-and-predictive-analytics-fog-and-edge-computing-for-industries-versus-cloud-19-1-2018/

[5] https://www.researchgate.net/figure/Fog-computing-in-Industrial-IoT_fig3_326359269

[6] https://thenewstack.io/mqtt-protocol-iot/

[7] https://www.iotforall.com/what-is-an-iot-platform/

Κεφάλαιο 3:

[1] https://www.festo-didactic.com/int-en/learning-systems/process-automation/compact-workstation/mps-pa-compact-workstation-with-level,flow-rate,pressure-and-temperature-controlled-systems.htm

[2] https://unitronicsplc.com/vision-series-vision570/#1449515771593-63d974b6-05ac1451602511853

[3] https://grobotronics.com/rs232-shield-v2.html

[4] https://grobotronics.com/sparkfun-can-bus-shield.html%20

[5] https://www.arduino.cc/en/Reference/SD

[6] https://shop.dakboard.com/products/pre-loaded-micro-sd-card

[7] https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/

Κεφάλαιο 5:

[1] http://henrysbench.capnfatz.com/henrys-bench/arduino-projects-tips-and-more/esp8266-esp-01-pin-outs-and-schematics/

[2] https://components101.com/wireless/esp8266-pinout-configuration-features-datasheet

[3] http://www.teomaragakis.com/hardware/electronics/how-to-connect-an-esp8266-to-an-arduino-uno/

[4] https://room-15.github.io/blog/2015/03/26/esp8266-at-command-reference/#at-commands

[5] https://www.arduino.cc/en/Tutorial/HelloWorld

[6] https://blog.startingelectronics.com/disabling-the-ethernet-chip-and-sd-card-on-an-arduino-ethernet-shield/

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Παράρτημα 1 - Κώδικας Arduino

Ανάγνωση Δεδομένων από την κάρτα SD

```
#include <SPI.h>
#include <SD.h>

SDFile myFile;

String TempStr, s1, s11, s2, s22, s1Val, s11Val, s2Val, s22Val;
int ValTemp;
double ValFlow, ValPress, ValLev1;

void ReadValues()
{
  myFile = SD.open("April/2april.txt");
  if (myFile)
  {
    while (myFile.available())  // read from the file until there's nothing else in it
    {
      TempStr = myFile.readStringUntil('\n');  //krataei oli ti grammi
      Serial.println(TempStr);

      s1 = TempStr.substring(62,66);
      s11 = TempStr.substring(61,65);
      s2 = TempStr.substring(72,76);
      s22 = TempStr.substring(71,75);

      if(s1 == "Temp")
      {
        s1Val = TempStr.substring(68,70);
        ValTemp = s1Val.toInt();
        Serial.print("Temperature: ");
        Serial.println(ValTemp);

        if(s2 == "Flow")
        {
          s2Val = TempStr.substring(78,82);
          ValFlow = s2Val.toDouble();
          Serial.print("Flow: ");
          Serial.println(ValFlow);
          Serial.println();
        }
      }//if s1=Temp
      else if(s11 == "Temp")
      {
        s11Val = TempStr.substring(67,69);
        ValTemp = s11Val.toInt();
        Serial.print("Temperature: ");
        Serial.println(ValTemp);

        if(s22 == "Flow")
        {
          s22Val = TempStr.substring(77,81);
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
        ValFlow = s22Val.toDouble();
        Serial.print("Flow: ");
        Serial.println(ValFlow);
        Serial.println();
      }
    }//else if s11=Temp

}//while

  myFile.close();  // close the file
 }//if
 else
 {
  Serial.println("error opening 1april.txt");
 }
}//ReadValues()

void setup()
{
 Serial.begin(115200);
 Serial.print("Initializing SD card...");

 if (!SD.begin(10))
 {
  Serial.println("initialization failed!");
  while (1);
 }
 Serial.println("initialization done.");

 ReadValues();
}

void loop()
{

}
```

Βιβλιοθήκες και μεταβλητές

```
#include <SD.h>
#include <SPI.h>
#include <Ethernet.h>
#include <PubSubClient.h>
#include <LiquidCrystal.h>

SDFile myFile;
LiquidCrystal lcd(9, 8, 7, 6, 5, 4);

String TempStr, s1, s11, s2, s22, s1Val, s11Val, s2Val, s22Val;
double ValFlow, ValPress, ValTemp , ValLev1, ValLev2;

byte mac[] = {  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };   // mac address Arduino
IPAddress ip(192, 168, 1, 119);                         //ip Arduino
//IPAddress server(37,187,106,16);    //test.mosquitto.org
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
IPAddress server(52,58,252,138);     //broker.hivemq.com

EthernetClient ethClient;
PubSubClient client(ethClient);
```

Σύνδεση και επανασύνδεση με τον Broker

```
void reconnect()  // Loop until we're reconnected
{
  while (!client.connected())
  {
    Serial.print("Attempting MQTT connection...");
    if (client.connect("arduinoClient"))            // Attempt to connect
    {
      Serial.println("connected");
      Serial.println();
    }//if
    else
    {
      Serial.print("failed, rc= ");
      Serial.print(client.state());
      Serial.println(" try again in 2 seconds");
      Serial.println();
      delay(2000);                              // Wait 2 seconds before retrying
    }  //else
  }  //while
} //reconnect
```

Ανάγνωση, Έλεγχος και Αποστολή Δεδομένων στον Broker

```
void ReadSendValues()
{
  myFile = SD.open("April/2april.txt");
  char buffer[14];

  if (myFile)
  {
    while (myFile.available())  // read from the file until there's nothing else in it
    {
      TempStr = myFile.readStringUntil('\n');  //krataei oli ti grammi
      Serial.println(TempStr);

      s1 = TempStr.substring(62,66);
      s11 = TempStr.substring(61,65);
      s2 = TempStr.substring(72,76);
      s22 = TempStr.substring(71,75);
//-------------------------Prwti Grammi Arxeiou---------------------------------------------
      if(s1 == "Temp")
      {
        s1Val = TempStr.substring(68,72);
        ValTemp = s1Val.toDouble();
        if (ValTemp < 0 || ValTemp > 100) //fog computing
        {
          Serial.println("Error Temperature Sensor!");
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
}
else
{
  dtostrf(ValTemp, 5, 2, buffer);
  client.publish("FestoLar/Temp", buffer);
}//else

if(s2 == "Flow")
{
  s2Val = TempStr.substring(78,82);
  ValFlow = s2Val.toDouble();
  if (ValFlow < 0 || ValFlow > 100) //fog computing
  {
    Serial.println("Error Flow Sensor!");
  }
  else
  {
    dtostrf(ValFlow, 5, 2, buffer);
    client.publish("FestoLar/Flow", buffer);
  }//else
}//if

else if(s2 == "Lev1")
{
  s2Val = TempStr.substring(78,82);
  if(s2Val == "LSH1")
  {
     ValLev1 = 80;
     dtostrf(ValLev1, 5, 2, buffer);
     client.publish("FestoLar/Tank1", buffer);
  }
  else if (s2Val == "LSL1")
  {
    ValLev1 = 20;
    dtostrf(ValLev1, 5, 2, buffer);
    client.publish("FestoLar/Tank1", buffer);
  }
  else
  {
    Serial.println("Error on Level Sensors of Tank 1."); //fog computing
  }
}//else if

else if(s2 == "Lev2")
{
  s2Val = TempStr.substring(78,82);
  if(s2Val == "LSH2")
  {
     ValLev2 = 80;
     dtostrf(ValLev2, 5, 2, buffer);
     client.publish("FestoLar/Tank2", buffer);
  }
  else if (s2Val == "LSL2")
  {
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
      ValLev2 = 20;
      dtostrf(ValLev2, 5, 2, buffer);
      client.publish("FestoLar/Tank2", buffer);
    }
    else
    {
      Serial.println("Error on Level Sensors of Tank 2.");  //fog computing
    }
  }//else if

  else if(s2 == "Pres")
  {
    s2Val = TempStr.substring(78,82);
    ValPress = s2Val.toDouble();
    if (ValPress < 0 || ValPress > 5) //fog computing
    {
      Serial.println("Error Pressure Sensor!");
    }
    else
    {
      dtostrf(ValPress, 5, 2, buffer);
      client.publish("FestoLar/Press", buffer);
    }//else
  }//else if s2=pres
}//if s1=Temp

else if(s1 == "Flow")
{
  s1Val = TempStr.substring(68,72);
  ValFlow = s1Val.toDouble();
  if (ValFlow < 0 || ValFlow > 100) //fog computing
  {
    Serial.println("Error Flow Sensor!");
  }
  else
  {
    dtostrf(ValFlow, 5, 2, buffer);
    client.publish("FestoLar/Flow", buffer);
  }//else

  if(s2 == "Temp")
  {
    s2Val = TempStr.substring(78,82);
    ValTemp = s2Val.toDouble();
    if (ValTemp < 0 || ValTemp > 100) //fog computing
    {
      Serial.println("Error Temperature Sensor!");
    }
    else
    {
      dtostrf(ValTemp, 5, 2, buffer);
      client.publish("FestoLar/Temp", buffer);
    }//else
  }//if
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
else if(s2 == "Lev1")
{
  s2Val = TempStr.substring(78,82);
  if(s2Val == "LSH1")
  {
     ValLev1 = 80;
     dtostrf(ValLev1, 5, 2, buffer);
     client.publish("FestoLar/Tank1", buffer);
  }
  else if (s2Val == "LSL1")
  {
    ValLev1 = 20;
    dtostrf(ValLev1, 5, 2, buffer);
    client.publish("FestoLar/Tank1", buffer);
  }
  else
  {
    Serial.println("Error on Level Sensors of Tank 1."); //fog computing
  }
}//else if

else if(s2 == "Lev2")
{
  s2Val = TempStr.substring(78,82);
  if(s2Val == "LSH2")
  {
     ValLev2 = 80;
     dtostrf(ValLev2, 5, 2, buffer);
     client.publish("FestoLar/Tank2", buffer);
  }
  else if (s2Val == "LSL2")
  {
    ValLev2 = 20;
    dtostrf(ValLev2, 5, 2, buffer);
    client.publish("FestoLar/Tank2", buffer);
  }
  else
  {
    Serial.println("Error on Level Sensors of Tank 2.");  //fog computing
  }
}//else if

else if(s2 == "Pres")
{
  s2Val = TempStr.substring(78,82);
  ValPress = s2Val.toDouble();
  if (ValPress < 0 || ValPress > 5) //fog computing
  {
    Serial.println("Error Pressure Sensor!");
  }
  else
  {
    dtostrf(ValPress, 5, 2, buffer);
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
      client.publish("FestoLar/Press", buffer);
    }//else
  }
}//else if s1=Flow

else if(s1 == "Lev1")
{
  s1Val = TempStr.substring(68,72);
  if(s1Val == "LSH1")
  {
    ValLev1 = 80;
    dtostrf(ValLev1, 5, 2, buffer);
    client.publish("FestoLar/Tank1", buffer);
  }
  else if (s1Val == "LSL1")
  {
    ValLev1 = 20;
    dtostrf(ValLev1, 5, 2, buffer);
    client.publish("FestoLar/Tank1", buffer);
  }
  else
  {
      Serial.println("Error on Level Sensors of Tank 1."); //fog computing
  }

  if(s2 == "Temp")
  {
    s2Val = TempStr.substring(78,82);
    ValTemp = s2Val.toDouble();
    if (ValTemp < 0 || ValTemp > 100) //fog computing
    {
      Serial.println("Error Temperature Sensor!");
    }
    else
    {
      dtostrf(ValTemp, 5, 2, buffer);
      client.publish("FestoLar/Temp", buffer);
    }//else
  }

  else if(s2 == "Flow")
  {
    s2Val = TempStr.substring(78,82);
    ValFlow = s2Val.toDouble();
    if (ValFlow < 0 || ValFlow > 100) //fog computing
    {
      Serial.println("Error Flow Sensor!");
    }
    else
    {
      dtostrf(ValFlow, 5, 2, buffer);
      client.publish("FestoLar/Flow", buffer);
    }//else
  }
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
  else if(s2 == "Lev2")
  {
    s2Val = TempStr.substring(78,82);
    if(s2Val == "LSH2")
    {
       ValLev2 = 80;
       dtostrf(ValLev2, 5, 2, buffer);
       client.publish("FestoLar/Tank2", buffer);
    }
    else if (s2Val == "LSL2")
    {
      ValLev2 = 20;
      dtostrf(ValLev2, 5, 2, buffer);
      client.publish("FestoLar/Tank2", buffer);
    }
    else
    {
      Serial.println("Error on Level Sensors of Tank 2.");  //fog computing
    }
  }

  else if(s2 == "Pres")
  {
    s2Val = TempStr.substring(78,82);
    ValPress = s2Val.toDouble();
    if (ValPress < 0 || ValPress > 5) //fog computing
    {
      Serial.println("Error Pressure Sensor!");
    }
    else
    {
      dtostrf(ValPress, 5, 2, buffer);
      client.publish("FestoLar/Press", buffer);
    }//else
  }
}//else if s1=Lev1

else if(s1 == "Lev2")
{
  s1Val = TempStr.substring(68,72);
  if(s1Val == "LSH2")
  {
    ValLev2 = 80;
    dtostrf(ValLev2, 5, 2, buffer);
    client.publish("FestoLar/Tank2", buffer);
  }
  else if (s1Val == "LSL2")
  {
    ValLev2 = 20;
    dtostrf(ValLev2, 5, 2, buffer);
    client.publish("FestoLar/Tank2", buffer);
  }
  else
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
{
   Serial.println("Error on Level Sensors of Tank 2."); //fog computing
}

if(s2 == "Temp")
{
  s2Val = TempStr.substring(78,82);
  ValTemp = s2Val.toDouble();
  if (ValTemp < 0 || ValTemp > 100) //fog computing
  {
    Serial.println("Error Temperature Sensor!");
  }
  else
  {
    dtostrf(ValTemp, 5, 2, buffer);
    client.publish("FestoLar/Temp", buffer);
  }//else
}

else if(s2 == "Flow")
{
  s2Val = TempStr.substring(78,82);
  ValFlow = s2Val.toDouble();
  if (ValFlow < 0 || ValFlow > 100) //fog computing
  {
    Serial.println("Error Flow Sensor!");
  }
  else
  {
    dtostrf(ValFlow, 5, 2, buffer);
    client.publish("FestoLar/Flow", buffer);
  }//else
}

else if(s2 == "Lev1")
{
  s2Val = TempStr.substring(78,82);
  if(s2Val == "LSH1")
  {
     ValLev1 = 80;
     dtostrf(ValLev1, 5, 2, buffer);
     client.publish("FestoLar/Tank1", buffer);
  }
  else if (s2Val == "LSL1")
  {
    ValLev1 = 20;
    dtostrf(ValLev1, 5, 2, buffer);
    client.publish("FestoLar/Tank1", buffer);
  }
  else
  {
    Serial.println("Error on Level Sensors of Tank 1.");  //fog computing
  }
}
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
  else if(s2 == "Pres")
  {
    s2Val = TempStr.substring(78,82);
    ValPress = s2Val.toDouble();
    if (ValPress < 0 || ValPress > 5) //fog computing
    {
      Serial.println("Error Pressure Sensor!");
    }
    else
    {
      dtostrf(ValPress, 5, 2, buffer);
      client.publish("FestoLar/Press", buffer);
    }//else
  }
}//else if s1=Lev2

else if(s1 == "Pres")
{
  s1Val = TempStr.substring(68,72);
  ValPress = s1Val.toDouble();
  if (ValPress < 0 || ValPress > 5) //fog computing
  {
    Serial.println("Error Pressure Sensor!");
  }
  else
  {
    dtostrf(ValPress, 5, 2, buffer);
    client.publish("FestoLar/Press", buffer);
  }//else

  if(s2 == "Temp")
  {
    s2Val = TempStr.substring(78,82);
    ValTemp = s2Val.toDouble();
    if (ValTemp < 0 || ValTemp > 100) //fog computing
    {
      Serial.println("Error Temperature Sensor!");
    }
    else
    {
      dtostrf(ValTemp, 5, 2, buffer);
      client.publish("FestoLar/Temp", buffer);
    }//else
  }

  else if(s2 == "Flow")
  {
    s2Val = TempStr.substring(78,82);
    ValFlow = s2Val.toDouble();
    if (ValFlow < 0 || ValFlow > 100) //fog computing
    {
      Serial.println("Error Flow Sensor!");
    }
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
        else
        {
          dtostrf(ValFlow, 5, 2, buffer);
          client.publish("FestoLar/Flow", buffer);
        }//else
      }

      else if(s2 == "Lev1")
      {
        s2Val = TempStr.substring(78,82);
        if(s2Val == "LSH1")
        {
           ValLev1 = 80;
           dtostrf(ValLev1, 5, 2, buffer);
           client.publish("FestoLar/Tank1", buffer);
        }
        else if (s2Val == "LSL1")
        {
          ValLev1 = 20;
          dtostrf(ValLev1, 5, 2, buffer);
          client.publish("FestoLar/Tank1", buffer);
        }
        else
        {
          Serial.println("Error on Level Sensors of Tank 1."); //fog computing
        }
      }//else if

      else if(s2 == "Lev2")
      {
        s2Val = TempStr.substring(78,82);
        if(s2Val == "LSH2")
        {
           ValLev2 = 80;
           dtostrf(ValLev2, 5, 2, buffer);
           client.publish("FestoLar/Tank2", buffer);
        }
        else if (s2Val == "LSL2")
        {
          ValLev2 = 20;
          dtostrf(ValLev2, 5, 2, buffer);
          client.publish("FestoLar/Tank2", buffer);
        }
        else
        {
          Serial.println("Error on Level Sensors of Tank 2.");  //fog computing
        }
      }//else if
    }//else if s1=Pres
//--------------------------------Ypoloipes Grammes Arxeiou-------------------------------------------
    else if(s11 == "Temp")
    {
      s11Val = TempStr.substring(67,71);
      ValTemp = s11Val.toDouble();
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
if (ValTemp < 0 || ValTemp > 100) //fog computing
{
  Serial.println("Error Temperature Sensor!");
}
else
{
  dtostrf(ValTemp, 5, 2, buffer);
  client.publish("FestoLar/Temp", buffer);
}//else

if(s22 == "Flow")
{
  s22Val = TempStr.substring(77,81);
  ValFlow = s22Val.toDouble();
  if (ValFlow < 0 || ValFlow > 100) //fog computing
  {
    Serial.println("Error Flow Sensor!");
  }
  else
  {
    dtostrf(ValFlow, 5, 2, buffer);
    client.publish("FestoLar/Flow", buffer);
  }//else
}

else if(s22 == "Lev1")
{
  s22Val = TempStr.substring(77,81);
  if(s22Val == "LSH1")
  {
     ValLev1 = 80;
     dtostrf(ValLev1, 5, 2, buffer);
     client.publish("FestoLar/Tank1", buffer);
  }
  else if (s2Val == "LSL1")
  {
    ValLev1 = 20;
    dtostrf(ValLev1, 5, 2, buffer);
    client.publish("FestoLar/Tank1", buffer);
  }
  else
  {
    Serial.println("Error on Level Sensors of Tank 1."); //fog computing
  }
}

else if(s22 == "Lev2")
{
  s22Val = TempStr.substring(77,81);
  if(s22Val == "LSH2")
  {
     ValLev2 = 80;
     dtostrf(ValLev2, 5, 2, buffer);
     client.publish("FestoLar/Tank2", buffer);
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
    }
    else if (s22Val == "LSL2")
    {
      ValLev2 = 20;
      dtostrf(ValLev2, 5, 2, buffer);
      client.publish("FestoLar/Tank2", buffer);
    }
    else
    {
      Serial.println("Error on Level Sensors of Tank 2."); //fog computing
    }
  }

  else if(s22 == "Pres")
  {
    s22Val = TempStr.substring(77,81);
    ValPress = s22Val.toDouble();
    if (ValPress < 0 || ValPress > 5) //fog computing
    {
      Serial.println("Error Pressure Sensor!");
    }
    else
    {
      dtostrf(ValPress, 5, 2, buffer);
      client.publish("FestoLar/Press", buffer);
    }//else
  }//else if s22=press
}//else if s11=Temp

else if(s11 == "Flow")
{
  s11Val = TempStr.substring(67,71);
  ValFlow = s11Val.toDouble();
  if (ValFlow < 0 || ValFlow > 100) //fog computing
  {
    Serial.println("Error Flow Sensor!");
  }
  else
  {
    dtostrf(ValFlow, 5, 2, buffer);
    client.publish("FestoLar/Flow", buffer);
  }//else

  if(s22 == "Temp")
  {
    s22Val = TempStr.substring(77,81);
    ValTemp = s22Val.toDouble();
    if (ValTemp < 0 || ValTemp > 100) //fog computing
    {
      Serial.println("Error Temperature Sensor!");
    }
    else
    {
      dtostrf(ValTemp, 5, 2, buffer);
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
    client.publish("FestoLar/Temp", buffer);
  }//else
}
else if(s22 == "Lev1")
{
  s22Val = TempStr.substring(77,81);
  if(s22Val == "LSH1")
  {
     ValLev1 = 80;
     dtostrf(ValLev1, 5, 2, buffer);
     client.publish("FestoLar/Tank1", buffer);
  }
  else if (s2Val == "LSL1")
  {
    ValLev1 = 20;
    dtostrf(ValLev1, 5, 2, buffer);
    client.publish("FestoLar/Tank1", buffer);
  }
  else
  {
    Serial.println("Error on Level Sensors of Tank 1."); //fog computing
  }
}

else if(s22 == "Lev2")
{
  s22Val = TempStr.substring(77,81);
  if(s22Val == "LSH2")
  {
     ValLev2 = 80;
     dtostrf(ValLev2, 5, 2, buffer);
     client.publish("FestoLar/Tank2", buffer);
  }
  else if (s22Val == "LSL2")
  {
    ValLev2 = 20;
    dtostrf(ValLev2, 5, 2, buffer);
    client.publish("FestoLar/Tank2", buffer);
  }
  else
  {
    Serial.println("Error on Level Sensors of Tank 2."); //fog computing
  }
}

else if(s22 == "Pres")
{
  s22Val = TempStr.substring(77,81);
  ValPress = s22Val.toDouble();
  if (ValPress < 0 || ValPress > 5) //fog computing
  {
    Serial.println("Error Pressure Sensor!");
  }
  else
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
    {
      dtostrf(ValPress, 5, 2, buffer);
      client.publish("FestoLar/Press", buffer);
    }//else
  }
}//else if s11=Flow

else if(s11 == "Lev1")
{
  s11Val = TempStr.substring(67,71);
  if(s11Val == "LSH1")
  {
    ValLev1 = 80;
    dtostrf(ValLev1, 5, 2, buffer);
    client.publish("FestoLar/Tank1", buffer);
  }
  else if (s11Val == "LSL1")
  {
    ValLev1 = 20;
    dtostrf(ValLev1, 5, 2, buffer);
    client.publish("FestoLar/Tank1", buffer);
  }
  else
  {
      Serial.println("Error on Level Sensors of Tank 1."); //fog computing
  }

  if(s22 == "Temp")
  {
    s22Val = TempStr.substring(77,81);
    ValTemp = s22Val.toDouble();
    if (ValTemp < 0 || ValTemp > 100) //fog computing
    {
      Serial.println("Error Temperature Sensor!");
    }
    else
    {
      dtostrf(ValTemp, 5, 2, buffer);
      client.publish("FestoLar/Temp", buffer);
    }//else
  }

  else if(s22 == "Flow")
  {
    s22Val = TempStr.substring(77,81);
    ValFlow = s22Val.toDouble();
    if (ValFlow < 0 || ValFlow > 100) //fog computing
    {
      Serial.println("Error Flow Sensor!");
    }
    else
    {
      dtostrf(ValFlow, 5, 2, buffer);
      client.publish("FestoLar/Flow", buffer);
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
    }//else
  }

  else if(s22 == "Lev2")
  {
    s22Val = TempStr.substring(77,81);
    if(s22Val == "LSH2")
    {
       ValLev2 = 80;
       dtostrf(ValLev2, 5, 2, buffer);
       client.publish("FestoLar/Tank2", buffer);
    }
    else if (s22Val == "LSL2")
    {
      ValLev2 = 20;
      dtostrf(ValLev2, 5, 2, buffer);
      client.publish("FestoLar/Tank2", buffer);
    }
    else
    {
      Serial.println("Error on Level Sensors of Tank 2."); //fog computing
    }
  }

  else if(s22 == "Pres")
  {
    s22Val = TempStr.substring(77,81);
    ValPress = s22Val.toDouble();
    if (ValPress < 0 || ValPress > 5) //fog computing
    {
      Serial.println("Error Pressure Sensor!");
    }
    else
    {
      dtostrf(ValPress, 5, 2, buffer);
      client.publish("FestoLar/Press", buffer);
    }//else
  }
}//else if s11=Lev1

else if(s11 == "Lev2")
{
  s11Val = TempStr.substring(67,71);
  if(s11Val == "LSH2")
  {
    ValLev2 = 80;
    dtostrf(ValLev2, 5, 2, buffer);
    client.publish("FestoLar/Tank2", buffer);
  }
  else if (s1Val == "LSL2")
  {
    ValLev2 = 20;
    dtostrf(ValLev2, 5, 2, buffer);
    client.publish("FestoLar/Tank2", buffer);
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
}
else
{
   Serial.println("Error on Level Sensors of Tank 2."); //fog computing
}

if(s22 == "Temp")
{
  s22Val = TempStr.substring(77,81);
  ValTemp = s22Val.toDouble();
  if (ValTemp < 0 || ValTemp > 100) //fog computing
  {
    Serial.println("Error Temperature Sensor!");
  }
  else
  {
    dtostrf(ValTemp, 5, 2, buffer);
    client.publish("FestoLar/Temp", buffer);
  }//else
}

else if(s22 == "Flow")
{
  s22Val = TempStr.substring(77,81);
  ValFlow = s22Val.toDouble();
  if (ValFlow < 0 || ValFlow > 100) //fog computing
  {
    Serial.println("Error Flow Sensor!");
  }
  else
  {
    dtostrf(ValFlow, 5, 2, buffer);
    client.publish("FestoLar/Flow", buffer);
  }//else
}

else if(s22 == "Lev1")
{
  s22Val = TempStr.substring(77,81);
  if(s22Val == "LSH1")
  {
     ValLev1 = 80;
     dtostrf(ValLev1, 5, 2, buffer);
     client.publish("FestoLar/Tank1", buffer);
  }
  else if (s22Val == "LSL1")
  {
    ValLev1 = 20;
    dtostrf(ValLev1, 5, 2, buffer);
    client.publish("FestoLar/Tank1", buffer);
  }
  else
  {
    Serial.println("Error on Level Sensors of Tank 1.");  //fog computing
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
    }
  }

  else if(s22 == "Pres")
  {
    s22Val = TempStr.substring(77,81);
    ValPress = s22Val.toDouble();
    if (ValPress < 0 || ValPress > 5) //fog computing
    {
      Serial.println("Error Pressure Sensor!");
    }
    else
    {
      dtostrf(ValPress, 5, 2, buffer);
      client.publish("FestoLar/Press", buffer);
    }//else
  }
}//else if s1=Lev2

else if(s11 == "Pres")
{
  s11Val = TempStr.substring(67,71);
  ValPress = s11Val.toDouble();
  if (ValPress < 0 || ValPress > 5) //fog computing
  {
    Serial.println("Error Pressure Sensor!");
  }
  else
  {
    dtostrf(ValPress, 5, 2, buffer);
    client.publish("FestoLar/Press", buffer);
  }//else

  if(s22 == "Temp")
  {
    s22Val = TempStr.substring(77,81);
    ValTemp = s22Val.toDouble();
    if (ValTemp < 0 || ValTemp > 100) //fog computing
    {
      Serial.println("Error Temperature Sensor!");
    }
    else
    {
      dtostrf(ValTemp, 5, 2, buffer);
      client.publish("FestoLar/Temp", buffer);
    }//else
  }

  else if(s22 == "Flow")
  {
    s22Val = TempStr.substring(77,81);
    ValFlow = s22Val.toDouble();
    if (ValFlow < 0 || ValFlow > 100) //fog computing
    {
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
      Serial.println("Error Flow Sensor!");
    }
    else
    {
      dtostrf(ValFlow, 5, 2, buffer);
      client.publish("FestoLar/Flow", buffer);
    }//else
  }

  else if(s22 == "Lev1")
  {
    s22Val = TempStr.substring(77,81);
    if(s22Val == "LSH1")
    {
       ValLev1 = 80;
       dtostrf(ValLev1, 5, 2, buffer);
       client.publish("FestoLar/Tank1", buffer);
    }
    else if (s22Val == "LSL1")
    {
      ValLev1 = 20;
      dtostrf(ValLev1, 5, 2, buffer);
      client.publish("FestoLar/Tank1", buffer);
    }
    else
    {
      Serial.println("Error on Level Sensors of Tank 1.");  //fog computing
    }
  }

  else if(s22 == "Lev2")
  {
    s22Val = TempStr.substring(77,81);
    if(s22Val == "LSH2")
    {
       ValLev2 = 80;
       dtostrf(ValLev2, 5, 2, buffer);
       client.publish("FestoLar/Tank2", buffer);
    }
    else if (s22Val == "LSL2")
    {
      ValLev2 = 20;
      dtostrf(ValLev2, 5, 2, buffer);
      client.publish("FestoLar/Tank2", buffer);
    }
    else
    {
      Serial.println("Error on Level Sensors of Tank 2."); //fog computing
    }
  }

 }//else if s11=Pres
 delay(5000);
}//while
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
    myFile.close();  // close the file
  }//if
  else
  {
    Serial.println("error opening 1april.txt");
  }
}//ReadSendValues
```

Ανάγνωση Μηνυμάτων από τον Broker – CallBack

```
void Callback(char* topic, byte* payload, unsigned int length)
{                          // When a message arrives from the sensor for a topic we are subscribed to
  int i = 0;
  char message_buff[5];     // this buffers our incoming messages so we can do something on
certain commands
  String msgString;

  for (i = 0; i < length; i++)
  {
    message_buff[i] = payload[i];
  }//for
  message_buff[i]= '\0';

  msgString = String(message_buff);    //Convert message from buffer into string

  lcd.setCursor(0,0);
  lcd.print("Topic: "+String(topic));
  lcd.setCursor(0,1);
  lcd.print("Message: "+msgString);

/*Serial.println("Message arrived from topic: " + String(topic));
  Serial.println("Length: " + String(length,DEC));
  Serial.println( "Payload: " + msgString);

  if (topic == Valve && msgString == "ON")     //if Valve is ON show on LCD/PLC
  {
    lcd.setCursor(0,0);
    lcd.print("Valve:  ON");
    delay(5000);
  }//if
  else if (topic == Valve && msgString == "OFF")  //if Valve is OFF show on LCD/PLC
  {
    lcd.setCursor(0,0);
    lcd.print("Valve:  OFF");
    delay(5000);
  }//else if
  if (topic == Heat && msgString == "ON")     //if Heat is ON show on LCD/PLC
  {
    lcd.setCursor(0,0);
    lcd.print("Heat:  ON");
    delay(5000);
  }//if
  else if (topic == Heat && msgString == "OFF")  //if Heat is OFF show on LCD/PLC
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
  {
    lcd.setCursor(0,0);
    lcd.print("Heat:  OFF");
    delay(5000);
  }//else if
  if (topic == Pres && msgString == "ON")     //if Pressure is ON show on LCD/PLC
  {
    lcd.setCursor(0,0);
    lcd.print("Pressure:  ON");
    delay(5000);
  }//if
  else if (topic == Pres && msgString == "OFF")  //if Pressure is OFF show on LCD/PLC
  {
    lcd.setCursor(0,0);
    lcd.print("Pressure:  OFF");
    delay(5000);
  }//else if
  else
  {
    lcd.setCursor(0,0);
    lcd.print("Undefined mes/ge");
    lcd.setCursor(0,1);
    lcd.print("from Broker.");

    delay(5000);
  }//else  */
  for (i = 0; i < length; i++)  //clear the buffer
    {
      message_buff[i] = '\0';
    }//for
}//callback
```

Setup

```
  Serial.begin(115200);
  lcd.begin(16, 2);

  client.setServer("52,58,252,138", 1883);
  client.setCallback(Callback);
  Ethernet.begin(mac);
  Serial.println(Ethernet.localIP());

  Serial.print("Initializing SD card...");
  if (!SD.begin(10))
  {
    Serial.println("initialization failed!");
    while (1);
  }
  Serial.println("initialization done.");
  ReadSendValues();
```

Loop

```
if (!client.connected())
  {
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
  reconnect();
  client.subscribe("FestoLar/Valve");
  client.subscribe("FestoLar/Heat");
  client.subscribe("FestoLar/Pres");
}//if
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Παράρτημα 2 - Κώδικας Node-Red


```
[
    {
        "id": "66fe8ec0.17bed",
        "type": "tab",
        "label": "Flow 1",
        "disabled": false,
        "info": ""
    },
    {
        "id": "337c4eff.f17f02",
        "type": "ui_base",
        "theme": {
            "name": "theme-dark",
            "lightTheme": {
                "default": "#0094CE",
                "baseColor": "#ff0000",
                "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-
Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
                "edited": true,
                "reset": false
            },
            "darkTheme": {
                "default": "#097479",
                "baseColor": "#ff0000",
                "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-
Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
                "edited": true,
                "reset": false
            },
            "customTheme": {
                "name": "Untitled Theme 1",
                "default": "#4B7930",
                "baseColor": "#4B7930",
                "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-
Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
            },
            "themeState": {
                "base-color": {
                    "default": "#097479",
                    "value": "#ff0000",
                    "edited": true
                },
                "page-titlebar-backgroundColor": {
                    "value": "#ff0000",
                    "edited": false
                },
                "page-backgroundColor": {
                    "value": "#111111",
                    "edited": false
                },
                "page-sidebar-backgroundColor": {
                    "value": "#ffffff",
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

"edited": false
},
"group-textColor": {
"value": "#ff4d4d",
"edited": false
},
"group-borderColor": {
"value": "#555555",
"edited": false
},
"group-backgroundColor": {
"value": "#333333",
"edited": false
},
"widget-textColor": {
"value": "#eeeeee",
"edited": false
},
"widget-backgroundColor": {
"value": "#ff0000",
"edited": false
},
"widget-borderColor": {
"value": "#333333",
"edited": false
},
"base-font": {
"value": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
}
}
},
"site": {
"name": "Larissa Festo System",
"hideToolbar": "false",
"allowSwipe": "false",
"allowTempTheme": "true",
"dateFormat": "DD/MM/YYYY",
"sizes": {
"sx": 48,
"sy": 48,
"gx": 6,
"gy": 6,
"cx": 6,
"cy": 6,
"px": 0,
"py": 0
}
}
},
{
"id": "58f6d71f.d1f2e8",
"type": "mqtt-broker",
"z": "",

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

"name": "",
"broker": "broker.hivemq.com",
"port": "1883",
"clientid": "",
"usetls": false,
"compatmode": true,
"keepalive": "60",
"cleansession": true,
"birthTopic": "",
"birthQos": "0",
"birthPayload": "",
"closeTopic": "",
"closeQos": "0",
"closePayload": "",
"willTopic": "",
"willQos": "0",
"willPayload": ""
},
{
"id": "3ef6de4.1372a22",
"type": "ui_tab",
"z": "",
"name": "Festo_Larissa",
"icon": "dashboard"
},
{
"id": "90c5c14a.3a27e",
"type": "ui_group",
"z": "",
"name": "Switch Heat",
"tab": "3ef6de4.1372a22",
"order": 7,
"disp": true,
"width": "8",
"collapse": false
},
{
"id": "e774d8c1.db94e8",
"type": "ui_group",
"z": "",
"name": "Switch Pressure",
"tab": "3ef6de4.1372a22",
"order": 8,
"disp": true,
"width": "8",
"collapse": false
},
{
"id": "f63559f3.f24598",
"type": "ui_group",
"z": "",
"name": "Valve 1",
"tab": "3ef6de4.1372a22",
"order": 9,

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

"disp": true,
"width": "8",
"collapse": false
},
{
"id": "c5a67e18.9eacd",
"type": "ui_group",
"z": "",
"name": "Flow",
"tab": "3ef6de4.1372a22",
"order": 3,
"disp": true,
"width": "5",
"collapse": false
},
{
"id": "fcd771fe.20b48",
"type": "ui_group",
"z": "",
"name": "Secondary Tank",
"tab": "3ef6de4.1372a22",
"order": 2,
"disp": true,
"width": "5",
"collapse": false
},
{
"id": "fb6e95d3.6804c8",
"type": "ui_group",
"z": "",
"name": "Switch Pump",
"tab": "3ef6de4.1372a22",
"order": 6,
"disp": true,
"width": "8",
"collapse": false
},
{
"id": "9fc9a995.ddc238",
"type": "ui_group",
"z": "",
"name": "Temperature",
"tab": "3ef6de4.1372a22",
"order": 4,
"disp": true,
"width": "5",
"collapse": false
},
{
"id": "3836cbe0.97c894",
"type": "ui_group",
"z": "",
"name": "Pressure",
"tab": "3ef6de4.1372a22",

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

"order": 5,
"disp": true,
"width": "5",
"collapse": false
},
{
"id": "b8f03d27.8f69a",
"type": "ui_group",
"z": "",
"name": "Valve 2",
"tab": "3ef6de4.1372a22",
"order": 10,
"disp": true,
"width": "8",
"collapse": false
},
{
"id": "b687dd3b.bcf62",
"type": "ui_group",
"z": "",
"name": "Valve 3",
"tab": "3ef6de4.1372a22",
"order": 11,
"disp": true,
"width": "8",
"collapse": false
},
{
"id": "53fcc163.cbc29",
"type": "ui_group",
"z": "",
"name": "Valve 4",
"tab": "3ef6de4.1372a22",
"order": 12,
"disp": true,
"width": "8",
"collapse": false
},
{
"id": "1e53abdd.954834",
"type": "ui_group",
"z": "",
"name": "Valve 5",
"tab": "3ef6de4.1372a22",
"order": 13,
"disp": true,
"width": "8",
"collapse": false
},
{
"id": "69942e3c.fb3b2",
"type": "ui_group",
"z": "",
"name": "Valve 6",

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

"tab": "3ef6de4.1372a22",
"order": 14,
"disp": true,
"width": "8",
"collapse": false
},
{
"id": "79a2eba3.db2544",
"type": "ui_group",
"z": "",
"name": "Main Tank",
"tab": "3ef6de4.1372a22",
"order": 1,
"disp": true,
"width": "5",
"collapse": false
},
{
"id": "5d32fb0d.f2ecc4",
"type": "mqtt in",
"z": "66fe8ec0.17bed",
"name": "",
"topic": "FestoLar/Tank2",
"qos": "1",
"broker": "58f6d71f.d1f2e8",
"x": 100,
"y": 120,
"wires": [
[
"7b449147.328c5",
"c0e84cbd.0d2fb"
]
]
},
{
"id": "b94b7b0c.759f48",
"type": "mqtt out",
"z": "66fe8ec0.17bed",
"name": "",
"topic": "FestoLar/Pump",
"qos": "1",
"retain": "",
"broker": "58f6d71f.d1f2e8",
"x": 680,
"y": 40,
"wires": []
},
{
"id": "7b449147.328c5",
"type": "ui_gauge",
"z": "66fe8ec0.17bed",
"name": "Sec Tank",
"group": "fcd771fe.20b48",
"order": 1,

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
    "width": "5",
    "height": "5",
    "gtype": "wave",
    "title": "Level 2",
    "label": "cm",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "x": 280,
    "y": 120,
    "wires": []
},
{
    "id": "7887e47f.5e7d4c",
    "type": "mqtt in",
    "z": "66fe8ec0.17bed",
    "name": "",
    "topic": "FestoLar/Tank1",
    "qos": "1",
    "broker": "58f6d71f.d1f2e8",
    "x": 100,
    "y": 40,
    "wires": [
        [
            "24a2e7f3.0ea7e8",
            "fb6c1881.00c888"
        ]
    ]
},
{
    "id": "24a2e7f3.0ea7e8",
    "type": "ui_gauge",
    "z": "66fe8ec0.17bed",
    "name": "Main Tank",
    "group": "79a2eba3.db2544",
    "order": 0,
    "width": "5",
    "height": "5",
    "gtype": "wave",
    "title": "Level 1",
    "label": "cm",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
        "#00b500",
        "#e6e600",
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "x": 290,
    "y": 40,
    "wires": []
  },
  {
    "id": "b0306c20.4c9a8",
    "type": "ui_switch",
    "z": "66fe8ec0.17bed",
    "name": "",
    "label": "Pump",
    "group": "fb6e95d3.6804c8",
    "order": 1,
    "width": "8",
    "height": "1",
    "passthru": true,
    "decouple": "false",
    "topic": "FestoLar/Pump",
    "style": "",
    "onvalue": "ON",
    "onvalueType": "str",
    "onicon": "",
    "oncolor": "",
    "offvalue": "OFF",
    "offvalueType": "str",
    "officon": "",
    "offcolor": "",
    "x": 490,
    "y": 40,
    "wires": [
        [
            "b94b7b0c.759f48"
        ]
    ]
  },
  {
    "id": "b78c3c6b.0c114",
    "type": "mqtt in",
    "z": "66fe8ec0.17bed",
    "name": "",
    "topic": "FestoLar/Flow",
    "qos": "1",
    "broker": "58f6d71f.d1f2e8",
    "x": 90,
    "y": 200,
    "wires": [
        [
            "33c55c15.d105b4",
            "bdfbd1c0.0616d"
        ]
    ]
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

},
{
"id": "83d4918b.389cf",
"type": "ui_switch",
"z": "66fe8ec0.17bed",
"name": "",
"label": "Heat",
"group": "90c5c14a.3a27e",
"order": 1,
"width": "8",
"height": "1",
"passthru": true,
"decouple": "false",
"topic": "FestoLar/Heat",
"style": "",
"onvalue": "ON",
"onvalueType": "str",
"onicon": "",
"oncolor": "",
"offvalue": "OFF",
"offvalueType": "str",
"officon": "",
"offcolor": "",
"x": 490,
"y": 100,
"wires": [
[
"6d580f9c.79eef"
]
]
},
{
"id": "6d580f9c.79eef",
"type": "mqtt out",
"z": "66fe8ec0.17bed",
"name": "",
"topic": "FestoLar/Heat",
"qos": "1",
"retain": "",
"broker": "58f6d71f.d1f2e8",
"x": 680,
"y": 100,
"wires": []
},
{
"id": "ffe3c0cb.c08b5",
"type": "mqtt out",
"z": "66fe8ec0.17bed",
"name": "",
"topic": "FestoLar/Pres",
"qos": "1",
"retain": "",
"broker": "58f6d71f.d1f2e8",
"x": 680,

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

"y": 160,
"wires": []
},
{
"id": "544a9413.f30fac",
"type": "ui_switch",
"z": "66fe8ec0.17bed",
"name": "",
"label": "Pressure",
"group": "e774d8c1.db94e8",
"order": 1,
"width": "8",
"height": "1",
"passthru": true,
"decouple": "false",
"topic": "FestoLar/Pres",
"style": "",
"onvalue": "ON",
"onvalueType": "str",
"onicon": "",
"oncolor": "",
"offvalue": "OFF",
"offvalueType": "str",
"officon": "",
"offcolor": "",
"x": 500,
"y": 160,
"wires": [
[
"ffe3c0cb.c08b5"
]
]
},
{
"id": "39e4c121.e68efe",
"type": "mqtt in",
"z": "66fe8ec0.17bed",
"name": "",
"topic": "FestoLar/Temp",
"qos": "1",
"broker": "58f6d71f.d1f2e8",
"x": 100,
"y": 280,
"wires": [
[
"9e51632e.a93b2",
"4f886ef3.147b8"
]
]
},
{
"id": "79d680d0.51b",
"type": "mqtt in",
"z": "66fe8ec0.17bed",

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
    "name": "",
    "topic": "FestoLar/Press",
    "qos": "1",
    "broker": "58f6d71f.d1f2e8",
    "x": 100,
    "y": 360,
    "wires": [
        [
            "ea596850.dd0fd8",
            "9541d2c9.3f9f6"
        ]
    ]
},
{
    "id": "9e51632e.a93b2",
    "type": "ui_gauge",
    "z": "66fe8ec0.17bed",
    "name": "",
    "group": "9fc9a995.ddc238",
    "order": 0,
    "width": "5",
    "height": "5",
    "gtype": "gage",
    "title": "",
    "label": "C",
    "format": "{{msg.payload}}",
    "min": 0,
    "max": "95",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "45",
    "seg2": "65",
    "x": 270,
    "y": 280,
    "wires": []
},
{
    "id": "33c55c15.d105b4",
    "type": "ui_gauge",
    "z": "66fe8ec0.17bed",
    "name": "",
    "group": "c5a67e18.9eacd",
    "order": 0,
    "width": "5",
    "height": "5",
    "gtype": "donut",
    "title": "",
    "label": "cm3/sec",
    "format": "{{msg.payload}}",
    "min": 0,
    "max": "50",
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

```
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "20",
    "seg2": "35",
    "x": 270,
    "y": 200,
    "wires": []
},
{
    "id": "ea596850.dd0fd8",
    "type": "ui_gauge",
    "z": "66fe8ec0.17bed",
    "name": "",
    "group": "3836cbe0.97c894",
    "order": 0,
    "width": "5",
    "height": "5",
    "gtype": "gage",
    "title": "",
    "label": "psi",
    "format": "{{msg.payload}}",
    "min": 0,
    "max": "4.9",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "1.9",
    "seg2": "3.5",
    "x": 270,
    "y": 360,
    "wires": []
},
{
    "id": "12a77241.40297e",
    "type": "ui_switch",
    "z": "66fe8ec0.17bed",
    "name": "",
    "label": "Valve 1",
    "group": "f63559f3.f24598",
    "order": 0,
    "width": "8",
    "height": "1",
    "passthru": true,
    "decouple": "false",
    "topic": "FestoLar/Valve1",
    "style": "",
    "onvalue": "ON",
    "onvalueType": "str",
    "onicon": "",
```


Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

"oncolor": "",
"offvalue": "OFF",
"offvalueType": "str",
"officon": "",
"offcolor": "",
"x": 500,
"y": 220,
"wires": [
[
"58139989.ca72f8"
]
]
},
{
"id": "74d83f08.3b1e",
"type": "ui_switch",
"z": "66fe8ec0.17bed",
"name": "",
"label": "Valve 2",
"group": "b8f03d27.8f69a",
"order": 0,
"width": "8",
"height": "1",
"passthru": true,
"decouple": "false",
"topic": "FestoLar/Valve2",
"style": "",
"onvalue": "ON",
"onvalueType": "str",
"onicon": "",
"oncolor": "",
"offvalue": "OFF",
"offvalueType": "str",
"officon": "",
"offcolor": "",
"x": 500,
"y": 280,
"wires": [
[
"3c6cc4f.5413b3c"
]
]
},
{
"id": "a756b382.ac451",
"type": "ui_switch",
"z": "66fe8ec0.17bed",
"name": "",
"label": "Valve 3",
"group": "b687dd3b.bcf62",
"order": 0,
"width": "8",
"height": "1",
"passthru": true,

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

"decouple": "false",
"topic": "FestoLar/Valve3",
"style": "",
"onvalue": "ON",
"onvalueType": "str",
"onicon": "",
"oncolor": "",
"offvalue": "OFF",
"offvalueType": "str",
"officon": "",
"offcolor": "",
"x": 500,
"y": 340,
"wires": [
[
"280fc4e8.8d69cc"
]
]
},
{
"id": "541cdf0f.d057e",
"type": "ui_switch",
"z": "66fe8ec0.17bed",
"name": "",
"label": "Valve 4",
"group": "53fcc163.cbc29",
"order": 0,
"width": "8",
"height": "1",
"passthru": true,
"decouple": "false",
"topic": "FestoLar/Valve4",
"style": "",
"onvalue": "ON",
"onvalueType": "str",
"onicon": "",
"oncolor": "",
"offvalue": "OFF",
"offvalueType": "str",
"officon": "",
"offcolor": "",
"x": 500,
"y": 400,
"wires": [
[
"499f751f.d64f0c"
]
]
},
{
"id": "8b0f2e5a.5999f",
"type": "ui_switch",
"z": "66fe8ec0.17bed",
"name": "",

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

"label": "Valve 5",
"group": "1e53abdd.954834",
"order": 0,
"width": "8",
"height": "1",
"passthru": true,
"decouple": "false",
"topic": "FestoLar/Valve5",
"style": "",
"onvalue": "ON",
"onvalueType": "str",
"onicon": "",
"oncolor": "",
"offvalue": "OFF",
"offvalueType": "str",
"officon": "",
"offcolor": "",
"x": 500,
"y": 460,
"wires": [
[
"49a6f3dd.043a3c"
]
]
},
{
"id": "ef5a752f.79c7c8",
"type": "ui_switch",
"z": "66fe8ec0.17bed",
"name": "",
"label": "Valve 6",
"group": "69942e3c.fb3b2",
"order": 0,
"width": "8",
"height": "1",
"passthru": true,
"decouple": "false",
"topic": "FestoLar/Valve6",
"style": "",
"onvalue": "ON",
"onvalueType": "str",
"onicon": "",
"oncolor": "",
"offvalue": "OFF",
"offvalueType": "str",
"officon": "",
"offcolor": "",
"x": 500,
"y": 520,
"wires": [
[
"937ac13.131e04"
]
]

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

},
{
"id": "58139989.ca72f8",
"type": "mqtt out",
"z": "66fe8ec0.17bed",
"name": "",
"topic": "FestoLar/Valve1",
"qos": "1",
"retain": "",
"broker": "58f6d71f.d1f2e8",
"x": 680,
"y": 220,
"wires": []
},
{
"id": "3c6cc4f.5413b3c",
"type": "mqtt out",
"z": "66fe8ec0.17bed",
"name": "",
"topic": "FestoLar/Valve2",
"qos": "1",
"retain": "",
"broker": "58f6d71f.d1f2e8",
"x": 680,
"y": 280,
"wires": []
},
{
"id": "280fc4e8.8d69cc",
"type": "mqtt out",
"z": "66fe8ec0.17bed",
"name": "",
"topic": "FestoLar/Valve3",
"qos": "1",
"retain": "",
"broker": "58f6d71f.d1f2e8",
"x": 680,
"y": 340,
"wires": []
},
{
"id": "499f751f.d64f0c",
"type": "mqtt out",
"z": "66fe8ec0.17bed",
"name": "",
"topic": "FestoLar/Valve4",
"qos": "1",
"retain": "",
"broker": "58f6d71f.d1f2e8",
"x": 680,
"y": 400,
"wires": []
},
{

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

"id": "49a6f3dd.043a3c",
"type": "mqtt out",
"z": "66fe8ec0.17bed",
"name": "",
"topic": "FestoLar/Valve5",
"qos": "1",
"retain": "",
"broker": "58f6d71f.d1f2e8",
"x": 680,
"y": 460,
"wires": []
},
{
"id": "937ac13.131e04",
"type": "mqtt out",
"z": "66fe8ec0.17bed",
"name": "",
"topic": "FestoLar/Valve6",
"qos": "1",
"retain": "",
"broker": "58f6d71f.d1f2e8",
"x": 680,
"y": 520,
"wires": []
},
{
"id": "fb6c1881.00c888",
"type": "ui_text",
"z": "66fe8ec0.17bed",
"group": "79a2eba3.db2544",
"order": 0,
"width": "5",
"height": "1",
"name": "",
"label": "ΠΡΟΣΟΧΗ: ",
"format": "{{msg.payload}}",
"layout": "row-spread",
"x": 290,
"y": 80,
"wires": []
},
{
"id": "c0e84cbd.0d2fb",
"type": "ui_text",
"z": "66fe8ec0.17bed",
"group": "fcd771fe.20b48",
"order": 0,
"width": "5",
"height": "1",
"name": "",
"label": "ΠΡΟΣΟΧΗ:",
"format": "{{msg.payload}}",
"layout": "row-spread",
"x": 290,

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

"y": 160,
"wires": []
},
{
"id": "4f886ef3.147b8",
"type": "ui_text",
"z": "66fe8ec0.17bed",
"group": "9fc9a995.ddc238",
"order": 0,
"width": "5",
"height": "1",
"name": "",
"label": "ΠΡΟΣΟΧΗ:",
"format": "{{msg.payload}}",
"layout": "row-spread",
"x": 290,
"y": 320,
"wires": []
},
{
"id": "9541d2c9.3f9f6",
"type": "ui_text",
"z": "66fe8ec0.17bed",
"group": "3836cbe0.97c894",
"order": 0,
"width": "5",
"height": "1",
"name": "",
"label": "ΠΡΟΣΟΧΗ:",
"format": "{{msg.payload}}",
"layout": "row-spread",
"x": 290,
"y": 400,
"wires": []
},
{
"id": "bdfbd1c0.0616d",
"type": "ui_text",
"z": "66fe8ec0.17bed",
"group": "c5a67e18.9eacd",
"order": 0,
"width": "5",
"height": "1",
"name": "",
"label": "ΠΡΟΣΟΧΗ:",
"format": "{{msg.payload}}",
"layout": "row-spread",
"x": 290,
"y": 240,
"wires": []
}
]

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212

Παράρτημα 3 - Συνοπτικός πίνακας εντολών του ESP

ESP8266 AT Command Set

Function	*AT Command*	*Response*
Working	AT	OK
Restart	AT+RST	OK [System Ready, Vendor:www.ai-thinker.com]
Firmware version	AT+GMR	AT+GMR 0018000902 OK
List Access Points	AT+CWLAP	AT+CWLAP +CWLAP:(4,"RochefortSurLac",38,"70:62:b8:6f:6d:58",1) +CWLAP:(4,"LiliPad2.4",-83,"f8:7b:8c:1e:7c:6d",1) OK
Join Access Point	AT+CWJAP? AT+CWJAP="SSID","Password"	Query AT+CWJAP? +CWJAP:"RochefortSurLac" OK
Quit Access Point	AT+CWQAP=? AT+CWQAP	Query OK
Get IP Address	AT+CIFSR	AT+CIFSR 192.168.0.105 OK
Set Parameters of Access Point	AT+ CWSAP? AT+ CWSAP= <ssid>,<pwd>,<chl>, <ecn>	Query ssid, pwd, chl = channel, ecn = encryption
WiFi Mode	AT+CWMODE? AT+CWMODE=1 AT+CWMODE=2 AT+CWMODE=3	Query STA AP BOTH
Set up TCP or UDP connection	AT+CIPSTART=? (CIPMUX=0) AT+CIPSTART = <type>,<addr>,<port> (CIPMUX=1) AT+CIPSTART= <id><type>,<addr>, <port>	Query id = 0-4, type = TCP/UDP, addr = IP address, port= port
TCP/UDP Connections	AT+ CIPMUX? AT+ CIPMUX=0 AT+ CIPMUX=1	Query Single Multiple
Check join devices' IP	AT+CWLIF	
TCP/IP Connection Status	AT+CIPSTATUS	AT+CIPSTATUS? no this fun
Send TCP/IP data	(CIPMUX=0) AT+CIPSEND=<length>; (CIPMUX=1) AT+CIPSEND= <id>,<length>	
Close TCP / UDP connection	AT+CIPCLOSE=<id> or AT+CIPCLOSE	
Set as server	AT+ CIPSERVER= <mode>[,<port>]	mode 0 to close server mode; mode 1 to open; port = port
Set the server timeout	AT+CIPSTO? AT+CIPSTO=<time>	Query <time>0~28800 in seconds
Baud Rate*	AT+CIOBAUD? Supported: 9600, 19200, 38400, 74880, 115200, 230400, 460800, 921600	Query AT+CIOBAUD? +CIOBAUD:9600 OK
Check IP address	AT+CIFSR	AT+CIFSR 192.168.0.106 OK
Firmware Upgrade (from Cloud)	AT+CIUPDATE	1. +CIPUPDATE:1 found server 2. +CIPUPDATE:2 connect server 3. +CIPUPDATE:3 got edition 4. +CIPUPDATE:4 start update
Received data	+IPD	(CIPMUX=0): + IPD, <len>: (CIPMUX=1): + IPD, <id>, <len>: <data>
Watchdog Enable*	AT+CSYSWDTENABLE	Watchdog, auto restart when program errors occur: enable
Watchdog Disable*	AT+CSYSWDTDISABLE	Watchdog, auto restart when program errors occur: disable

Από το https://4tronix.co.uk/picobot2/ESP8266ATCommandsSet.pdf

Institutional Repository - Library & Information Centre - University of Thessaly
15/10/2023 05:24:20 EEST - 167.114.118.212