



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Έξυπνη Κυψέλη Μελισσών

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

ΓΕΩΡΓΙΑΚΑΚΗ ΙΩΑΝΝΗ

(ΑΕΜ: 2964)

Επιβλέπων : Δημήτρης Βέργαδος
Καθηγητής

Καστοριά Νοέμβριος - 2023



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Έξυπνη Κυψέλη Μελισσών

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

ΓΕΩΡΓΙΑΚΑΚΗ ΙΩΑΝΝΗ

(ΑΕΜ: 2964)

Επιβλέπων : Δημήτρης Βέργαδος
Αναπληρωτής Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 3/11/2023

Δόσης Μιχαήλ

Νικολάου Σπυρίδων

Βέργαδος Δημήτριος

Καθηγητής

Λέκτορας

Αναπληρωτής Καθηγητής

Καστοριά Νόεμβριος - 2023

Copyright ©2023 ΓΕΩΡΓΙΑΚΑΚΗΣ ΙΩΑΝΝΗΣ

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Δυτικής Μακεδονίας.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένεια μου, **Αντώνη, Μαρία** και **Κατερίνα** για την αδιάκοπη στήριξη τους σε όλα τα χρόνια της μαθητικής και φοιτητικής μου πορείας καθώς και την κοπέλα μου **Κέλλυ** για όλη την βοήθεια και τη στήριξη που μου παρείχε στα φοιτητικά μου χρόνια. Επιπλέον, θα ήθελα να ευχαριστήσω την άλλη μου οικογένεια, την παρέα μου, τον **Μήτσο, Δημήτρη, Πασχάλη** και **Μάριο**, που ήταν κοντά μου σε καλές και κακές στιγμές, που ενώ δεν το γνωρίζουν, είναι και αυτοί ο λόγος που βρίσκομαι εδώ.

Τέλος θα ήθελα να αφιερώσω την πτυχιακή αυτή, στον Παππού μου Παναγιώτη και την Γεωργίτσα. Στον Παππού μου που δεν πρόλαβε να διαβάσει αυτή την πτυχιακή για να καταλάβει επιτέλους τι σπούδασα, και την Γεωργία που δεν πρόλαβε να γράψει τη δική της ιστορία στον ακαδημαϊκό αθλητικό κόσμο.

Σας ευχαριστώ όλους.

Abstract

Beehive monitoring lies at the confluence of nature and technology, revealing a symbiotic relationship essential for modern agriculture and biodiversity conservation. This paper delves into the intricacies of utilizing advanced technological tools for the systematic monitoring of bee hives. Beginning with a detailed exploration of the ecological importance of bees, the narrative evolves to highlight the importance of modern monitoring systems. The research highlights the integration of the ESP32 microcontroller with state-of-the-art sensors, the effectiveness of real-time data communication protocols such as MQTT, and the profound impact of such systems on real-world beekeeping practices. Bridging theoretical knowledge with practical application, this thesis presents a holistic view of the current landscape of beehive monitoring. The findings clarify that with accurate data collection and analysis, technological interventions can drastically improve the understanding and conservation of these vital pollinators, thereby enhancing their pivotal role in the global ecology and economy.

Λέξεις Κλειδιά: *IoT, ESP32, NodeRed, MQTT, Beehive Monitoring, Grafana, Proxmox, Virtualization, Docker*

Περίληψη

Η παρακολούθηση των κυψελών βρίσκεται στη συμβολή της φύσης και της τεχνολογίας, αποκαλύπτοντας μια συμβιωτική σχέση απαραίτητη για τη σύγχρονη γεωργία και τη διατήρηση της βιοποικιλότητας. Η παρούσα εργασία εμβαθύνει στις περιπλοκές της αξιοποίησης προηγμένων τεχνολογικών εργαλείων για τη συστηματική παρακολούθηση των κυψελών μελισσών. Ξεκινώντας με μια λεπτομερή διερεύνηση της οικολογικής σημασίας των μελισσών, η αφήγηση εξελίσσεται για να υπογραμμίσει τη σημασία των σύγχρονων συστημάτων παρακολούθησης. Η έρευνα αναδεικνύει την ενσωμάτωση του μικροελεγκτή ESP32 με αισθητήρες τελευταίας τεχνολογίας, την αποτελεσματικότητα πρωτοκόλλων επικοινωνίας δεδομένων σε πραγματικό χρόνο, όπως το MQTT, και τον βαθύ αντίκτυπο τέτοιων συστημάτων στις πρακτικές μελισσοκομίας του πραγματικού κόσμου. Γεφυρώνοντας τη θεωρητική γνώση με την πρακτική εφαρμογή, η παρούσα εργασία παρουσιάζει μια ολιστική άποψη του σημερινού τοπίου της παρακολούθησης των κυψελών. Τα ευρήματα διευκρινίζουν ότι με την ακριβή συλλογή και ανάλυση δεδομένων, οι τεχνολογικές παρεμβάσεις μπορούν να βελτιώσουν δραστικά την κατανόηση και τη διατήρηση αυτών των ζωτικών επικονιαστών, ενισχύοντας έτσι τον κομβικό τους ρόλο στην παγκόσμια οικολογία και οικονομία.

Λέξεις-κλειδιά: *IoT, ESP32, NodeRed, MQTT, Beehive Monitoring, Grafana, Proxmox, Virtualization, Docker*

Πίνακας Περιεχομένων

Εισαγωγή.....	1
1. Σημασία των Κυψελών.....	2
1.1 Ιστορικό και σημασία της παρακολούθησης των κυψελών	2
1.2 Η ανάγκη για μια ολοκληρωμένη λύση παρακολούθησης.....	2
1.3 Case Studies:	2
1.4 Πεδίο εφαρμογής της διατριβής.....	3
2. Infrastructure & Hosting.....	4
2.1 Proxmox: Virtualization	4
2.1.1 Ρόλος στο σύστημα παρακολούθησης κυψέλης	4
2.2 Docker: Το σύγχρονο πρότυπο για Containerized εφαρμογές.....	5
2.3 Εξασφαλίζοντας την ασφάλεια μέσω Cloudflare Tunnels.....	5
3. Tech Stack - Επιλογή, χρήση και ανάλυση	6
3.1 Η ανάγκη για το ESP32.....	6
3.1.1 Ανταγωνιστικά εργαλεία.....	6
3.1.2 Γιατί επιλέχθηκε το ESP32.....	6
3.1.3 Χαρακτηριστικά και δυνατότητες.....	7
3.1.4 Χρήση στο έργο.....	7
3.2 Πρωτόκολλο MQTT.....	7
3.2.1 Η ανάγκη για το MQTT	7
3.2.2 Ανταγωνιστικά εργαλεία.....	7
3.2.3 Γιατί επιλέχθηκε το MQTT.....	8
3.2.4 Χαρακτηριστικά και δυνατότητες.....	8
3.2.5 Χρήση στο έργο.....	8
3.3 Πλατφόρμα Docker (συγκεκριμένα Node-RED, InfluxDB2 και Grafana)	9
3.3.1 Η ανάγκη για το Docker.....	9
3.3.2 Ανταγωνιστικά εργαλεία.....	9
3.3.3 Γιατί επιλέχθηκε το Docker	9
3.3.4 Χαρακτηριστικά και δυνατότητες.....	9
3.3.5 Χρήση στο έργο.....	10
4. Μικροελεγκτής ESP32	10
4.1 Ενσωμάτωση με μικροελεγκτή ESP32	11
4.2 Αξιοποίηση δεδομένων από τον ενισχυτή κυψελών φορτίου HX711.....	11

4.3	Διαπιστώσεις σε πραγματικό χρόνο από τον αισθητήρα DHT11	11
4.4	Έλεγχος κυψέλης με μαγνητικό αισθητήρα	12
4.5	Τροφοδοσία του συστήματος με την μπαταρία 18650	13
5.	Μετάδοση δεδομένων & MQTT.....	13
5.1	Η σημασία του MQTT στην επικοινωνία δεδομένων σε πραγματικό χρόνο ..	13
5.1.1	Εξέλιξη των πρωτοκόλλων μετάδοσης δεδομένων	14
5.1.2	MQTT: Το βασικό πρωτόκολλο για το IoT	14
5.1.3	Χαρακτηριστικά που κάνουν το MQTT ευνοϊκό για την παρακολούθηση της κυψέλης	14
5.2	Ρύθμιση και διαμόρφωση του MQTT Broker.....	14
5.2.1	Ο ρόλος ενός MQTT Broker	14
5.2.2	Κριτήρια για την επιλογή ενός αποδοτικού MQTT Broker	15
5.2.3	Πρακτικά βήματα για τη διαμόρφωση του διαμεσολαβητή για βέλτιστη απόδοση	15
6.	Node-RED μέσα στο Docker	15
6.1	Εισαγωγή και εγκατάσταση του Node-RED.....	15
6.2	Επεξεργασία δεδομένων, ενορχήστρωση και δρομολόγηση στο InfluxDB2...	16
7.	InfluxDB2 μέσα στο Docker	17
7.1	Δυνατότητες και διαμόρφωση του InfluxDB2.....	17
7.1.1	Χαρακτηριστικά που διαφοροποιούν το InfluxDB2	17
7.1.2	Διαμόρφωση του InfluxDB2 στο Docker.....	17
7.2	Στρατηγικές αποθήκευσης δεδομένων και ενσωμάτωση με την Grafana	18
7.2.1	Buckets και διατήρηση δεδομένων	18
7.2.2	Ενσωμάτωση με την Grafana	18
8.	Grafana μέσα στο Docker.....	18
8.1	Ρύθμιση της Grafana και των βασικών χαρακτηριστικών της	19
8.1.1	Ρύθμιση της Grafana και των βασικών χαρακτηριστικών της	19
8.1.2	Ανάπτυξη της Grafana σε Docker	19
8.2	Δημιουργία ταμπλό για αποτελεσματική παρακολούθηση κυψέλης.....	19
8.2.1	Gauges	19
8.2.2	Charts	20
8.2.3	Statistics	21
8.2.4	GeoMap.....	21
8.2.5	Tables	22

9.	Ολοκλήρωση συστήματος, ασφάλεια και δοκιμές	23
9.1	Ενσωμάτωση όλων των στοιχείων σε ασφαλές περιβάλλον	23
9.1.1	Η διαδικασία ολοκλήρωσης.....	23
9.1.2	Προκλήσεις και λύσεις	23
9.2	Ασφάλεια στην ενσωμάτωση	25
9.2.1	Κατανόηση των Cloudflare Tunnels	25
9.2.2	Εφαρμογή Cloudflare tunnels για την παρακολούθηση των κυψελών... ..	25
9.3	Μεθοδολογίες δοκιμών, αποτελέσματα και ανάλυση επιδόσεων	25
9.3.1	Επιλογή των κατάλληλων μεθοδολογιών δοκιμών	25
9.3.2	Αποτελέσματα και ανάλυση	25
	Συμπεράσματα.....	25
	Βελτιώσεις	26
	Βιβλιογραφία	27
	Παράρτημα Κώδικα	28

Λίστα Εικόνων

Εικόνα 1 ProxMox UI.....	4
Εικόνα 2 Docker Architecture	5
Εικόνα 3 Cloudflare Tunnels	6
Εικόνα 4 ESP32.....	10
Εικόνα 5 ESP32 - HX711	11
Εικόνα 6 ESP32 - DHT11.....	12
Εικόνα 7 Manget Sensor	12
Εικόνα 8 18650 Battery.....	13
Εικόνα 9 MQTT Broker	14
Εικόνα 10 NodeRed UI	16
Εικόνα 11 InfluxDB2.....	17
Εικόνα 12 Grafana UI	18
Εικόνα 13 Grafana - Gauge	20
Εικόνα 14 Grafana - Chart.....	20
Εικόνα 15 Grafana - Graph.....	20
Εικόνα 16 Grafana - Weight Stat.....	21
Εικόνα 17 Grafana - State Stat.....	21
Εικόνα 18 Grafana GeoMap.....	22
Εικόνα 19 Grafana - Table.....	22
Εικόνα 20 Docker Compose Script.....	23
Εικόνα 21 HX711 Platform.....	24
Εικόνα 22 Beehive with Platform.....	24
Εικόνα 23 Grafana - Dashboard	26
Εικόνα 24 Thesis Circuit	34
Εικόνα 25 Thesis Schematic.....	34

Εισαγωγή

Στη σημερινή ταχέως εξελισσόμενη ψηφιακή εποχή, η σύνθεση της τεχνολογίας και της φύσης αναδεικνύεται σε πρωταρχικό μέλημα. Η παρακολούθηση της κυψέλης χρησιμεύει ως απόδειξη αυτής της ένωσης, τονίζοντας τον καθοριστικό ρόλο που διαδραματίζει η τεχνολογία στην κατανόηση και τη διατήρηση του περιβάλλοντός μας. Το παρόν έγγραφο εμβαθύνει στους μηχανισμούς, τη σημασία και τις επιπτώσεις της χρήσης προηγμένων τεχνολογικών εργαλείων στην ευαίσθητη τέχνη της παρακολούθησης των κυψελών μελισσών.

Δομή της εργασίας:

Το χειρόγραφο είναι σχολαστικά δομημένο για να προσφέρει στους αναγνώστες μια ολοκληρωμένη κατανόηση της παρακολούθησης των κυψελών στον σύγχρονο κόσμο. Ξεκινά με ένα θεμελιώδες υπόβαθρο, παρουσιάζοντας την παγκόσμια σημασία των μελισσών, και προοδευτικά περιηγείται στις περιπλοκές της τεχνολογίας, της διαχείρισης δεδομένων και των εφαρμογών στον πραγματικό κόσμο. Μέχρι το αποκορύφωμα της εργασίας, οι αναγνώστες θα έχουν διατρέξει όλο το φάσμα της παρακολούθησης κυψελών, από τη σύλληψή της έως τις απτές επιπτώσεις της στην κοινωνία και το περιβάλλον.

Δομή των κεφαλαίων:

- 1. Ιστορικό και σημασία της παρακολούθησης των κυψελών:** Αυτό το κεφάλαιο υπογραμμίζει τον ζωτικό ρόλο που διαδραματίζουν οι μέλισσες στο οικοσύστημά μας, προσφέροντας ένα ισχυρό επιχειρήμα για την ανάγκη περίπλοκης παρακολούθησης.
- 2. Ανάλυση του Infrastructure και του μικροελεγκτή που έχουν χρησιμοποιηθεί για την παρακολούθηση των κυψελών:** ESP32 και την ενσωμάτωσή του με διάφορους αισθητήρες.
- 3. Τεχνικές συλλογής δεδομένων:** Αυτή η ενότητα αναλύει την απρόσκοπτη αλληλεπίδραση μεταξύ των στοιχείων υλικού για την ακριβή καταγραφή των μετρικών της κυψέλης.
- 4. Μετάδοση και διαχείριση δεδομένων:** MQTT, αναδεικνύει την αποτελεσματικότητα στην επικοινωνία, αποθήκευση και ανάλυση δεδομένων σε πραγματικό χρόνο.
- 5. Εφαρμογές και επιπτώσεις στον πραγματικό κόσμο:** Αυτό το κεφάλαιο, γεφυρώνοντας τη θεωρία με την πράξη, διευκρινίζει πώς οι τεχνολογικές εξελίξεις του συστήματός μας παρακολούθησης επηρεάζουν τη μελισσοκομία στην πράξη.

Μέχρι την ολοκλήρωση του παρόντος εγγράφου, αναμένεται ότι οι αναγνώστες θα κατανοήσουν όχι μόνο τη βαθιά σημασία της παρακολούθησης των κυψελών, αλλά και τη λεπτή τεχνολογική εννοχρήστρωση που την τροφοδοτεί. Μέσω αυτής της έκθεσης, το έγγραφο φιλοδοξεί να εμπνεύσει περισσότερες τέτοιες διασταυρώσεις της τεχνολογίας και της φύσης, προωθώντας ένα μέλλον όπου θα συνυπάρχουν για μια πιο αρμονική συνύπαρξη.

1. Σημασία των Κυψελών

1.1 Ιστορικό και σημασία της παρακολούθησης των κυψελών

Σημασία των μελισσών στο οικοσύστημά μας: Οι μέλισσες είναι κομβικής σημασίας στο οικολογικό μας σύστημα. Πέρα από την παραγωγή μελιού, ο ρόλος τους στην επικονίαση επηρεάζει άμεσα την επιτυχία διαφόρων καλλιεργειών, εξασφαλίζοντας τόσο την επισιτιστική ασφάλεια όσο και την οικολογική ποικιλομορφία. Η μείωσή τους θα είχε ως αποτέλεσμα εκτεταμένες επιπτώσεις σε όλα τα οικοσυστήματα, επηρεάζοντας τόσο τα φυτά όσο και τα ζώα.

Μέλισσες και γεωργία: Πέρα από τα λουλούδια, οι μέλισσες διαδραματίζουν κρίσιμο ρόλο στη γονιμοποίηση πολλών φρούτων, λαχανικών και ξηρών καρπών που αποτελούν μέρος της καθημερινής μας διατροφής. Η μείωση της δραστηριότητας των μελισσών μπορεί να οδηγήσει σε μείωση των αποδόσεων των καλλιεργειών, η οποία με τη σειρά της επηρεάζει τις οικονομίες και τον παγκόσμιο εφοδιασμό σε τρόφιμα.

Προκλήσεις στη μελισσοκομία: Οι σύγχρονοι μελισσοκόμοι παλεύουν με διάφορες προκλήσεις, από παράσιτα όπως το άκαρι *Varroa* μέχρι τις περιβαλλοντικές αλλαγές και τις επιπτώσεις των φυτοφαρμάκων. Η παρακολούθηση της υγείας και της δραστηριότητας των μελισσών είναι απαραίτητη όχι μόνο για την παραγωγή μελιού, αλλά και για τη διατήρηση αυτών των πολύτιμων εντόμων.

Επιπτώσεις της παραμέλησης: Χωρίς αποτελεσματική παρακολούθηση, ζητήματα όπως οι εστίες ασθενειών ή μια ξαφνική πτώση της δραστηριότητας των μελισσών μπορεί να περάσουν απαρατήρητα, οδηγώντας σε πλήρη κατάρρευση των κυψελών. Τα περιστατικά αυτά μπορεί να έχουν σημαντικές επιπτώσεις στα τοπικά οικοσυστήματα και τις οικονομίες.

1.2 Η ανάγκη για μια ολοκληρωμένη λύση παρακολούθησης

Περιορισμοί των παραδοσιακών πρακτικών: Ενώ οι παραδοσιακές μέθοδοι παρακολούθησης κυψελών, όπως οι χειροκίνητες επιθεωρήσεις, έχουν τα πλεονεκτήματά τους, συχνά στερούνται της ακρίβειας και της αμεσότητας που απαιτείται στα σημερινά δυναμικά περιβάλλοντα. Επιπλέον, οι μέθοδοι αυτές μπορεί να είναι επεμβατικές, προκαλώντας αδικαιολόγητο στρες στις μέλισσες.

1.3 Case Studies:

Τα αμυγδαλοτροφεία της Καλιφόρνιας:

Ιστορικό: Η κεντρική κοιλάδα της Καλιφόρνια είναι το κέντρο της παραγωγής αμυγδάλων, συμβάλλοντας σε πάνω από το 80% των αμυγδάλων παγκοσμίως. Κάθε χρόνο, αυτό απαιτεί τις προσπάθειες επικονίασης δεσκατομμυρίων μελισσών.

Η πρόκληση: Σε μια εποχή, πολλοί μελισσοκόμοι ανέφεραν υψηλότερο από το συνηθισμένο ποσοστό θνησιμότητας μεταξύ των αποικιών τους. Οι παραδοσιακές πρακτικές παρακολούθησης δεν ήταν σε θέση να εντοπίσουν έγκαιρα την ακριβή αιτία.

Αποτελέσματα: Λόγω της καθυστερημένης δράσης, η θνησιμότητα είχε ως αποτέλεσμα τη μείωση του αριθμού των μελισσών που ήταν διαθέσιμες για επικονίαση, με αποτέλεσμα την

αισθητή πτώση της απόδοσης αμυγδάλων για εκείνη τη χρονιά. Ο οικονομικός αντίκτυπος έγινε αισθητός όχι μόνο από τους μελισσοκόμους αλλά και από τους αμυγδαλοπαραγωγούς.

Μάθημα: Αν υπήρχε ένα ολοκληρωμένο, τεχνολογικά καθοδηγούμενο σύστημα παρακολούθησης, οι ανωμαλίες στην υγεία των κυψελών θα μπορούσαν να είχαν εντοπιστεί νωρίτερα, αποτρέποντας ενδεχομένως τέτοιες απώλειες μελισσών μεγάλης κλίμακας.

Το ξέσπασμα του ευρωπαϊκού ακάρεως Varroa:

Ιστορικό: Το άκαρι Varroa, ένα παράσιτο που είναι γνωστό ότι προσβάλλει αποικίες μελισσών, ήταν ένα σποραδικό ζήτημα σε μέρη της Ευρώπης. Οι μελισσοκόμοι συνήθως διαχειρίζονταν αυτά τα παράσιτα μέσω περιοδικών χειροκίνητων επιθεωρήσεων και θεραπειών.

Η πρόκληση: Κατά τη διάρκεια μιας συγκεκριμένης περιόδου, παρατηρήθηκε ταχύτερη από το κανονικό εξάπλωση της προσβολής από το άκαρεο σε πολλά μελισσοκομεία. Οι παραδοσιακές ρουτίνες επιθεώρησης δεν ήταν σε θέση να παρακολουθήσουν την ταχέως κλιμακούμενη κατάσταση.

Αποτελέσματα: Πολλές αποικίες αποδεκατίστηκαν από το άκαρι Varroa πριν μπορέσουν να αναπτυχθούν αποτελεσματικά αντίμετρα. Η μειωμένη δραστηριότητα των μελισσών επηρέασε αρνητικά την τοπική χλωρίδα, ιδίως τα αγριολούλουδα, και επηρέασε την παραγωγή μελιού.

Μάθημα: Ένα πιο συνεχές και σε πραγματικό χρόνο σύστημα παρακολούθησης θα μπορούσε να είχε εντοπίσει την ταχεία εξάπλωση των ακάρεων σε πρώιμο στάδιο, επιτρέποντας στους μελισσοκόμους να λάβουν γρήγορα προληπτικά μέτρα.

Οφέλη των σύγχρονων λύσεων: Οι σύγχρονες, τεχνολογικά εμπλουτισμένες λύσεις παρακολούθησης προσφέρουν ακρίβεια, άμεση ανατροφοδότηση και ελάχιστη επεμβατικότητα, παρέχοντας στους μελισσοκόμους βαθύτερη γνώση της υγείας των κυψελών και εξασφαλίζοντας τη βέλτιστη παραγωγικότητα. Τα συστήματα αυτά μπορούν να ανιχνεύσουν αλλαγές στις αποχρώσεις, δίνοντας στους μελισσοκόμους τη δυνατότητα να λαμβάνουν προληπτικά μέτρα. Ενώ οι παραδοσιακές μέθοδοι παρακολούθησης κυψελών, όπως οι χειροκίνητες επιθεωρήσεις, έχουν τα πλεονεκτήματά τους, συχνά στερούνται της ακρίβειας και της αμεσότητας που απαιτείται στα σημερινά δυναμικά περιβάλλοντα. Επιπλέον, οι μέθοδοι αυτές μπορεί να είναι επεμβατικές, προκαλώντας αδικαιολόγητο στρες στις μέλισσες.

1.4 Πεδίο εφαρμογής της διατριβής

Η παρούσα διατριβή προσπαθεί να διερευνήσει το σχεδιασμό, την εφαρμογή και την αποτελεσματικότητα ενός ολοκληρωμένου συστήματος παρακολούθησης κυψελών που αξιοποιεί τις τεχνολογίες αυτές. Μέσα από τα κεφάλαια, θα διερευνήσουμε την ενσωμάτωση των διαφόρων τεχνολογιών σε ένα συνεκτικό σύστημα, θα συζητήσουμε τις πιθανές προκλήσεις και λύσεις κατά την εφαρμογή του και θα αξιολογήσουμε την εφαρμοσιμότητα και την αποτελεσματικότητά του σε πραγματικές συνθήκες.

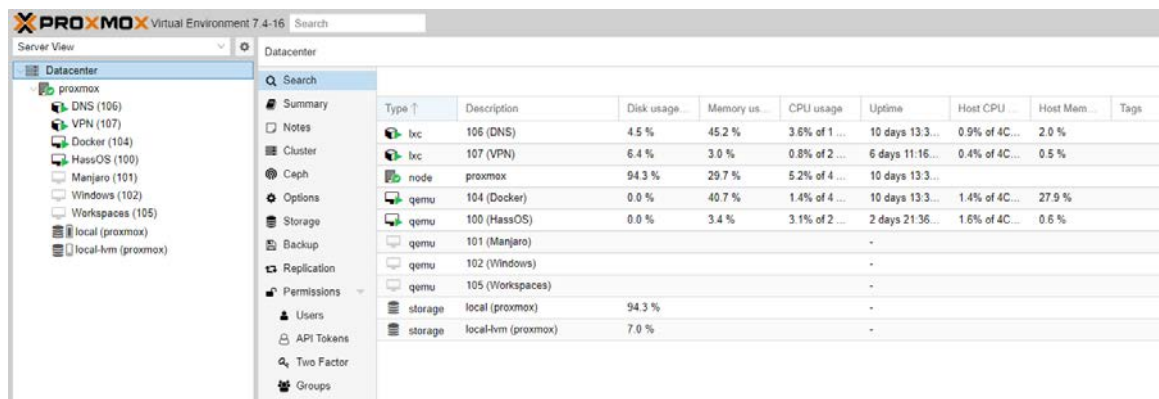
2. Infrastructure & Hosting

Στον σημερινό ψηφιακό κόσμο, οι θεμελιώδεις τεχνολογίες και οι στρατηγικές φιλοξενίας που χρησιμοποιούνται σε κάθε έργο καθορίζουν την ανθεκτικότητα, την αποτελεσματικότητα και την προσαρμοστικότητα του. Το παρόν κεφάλαιο επιδιώκει να εμβαθύνει στις μεθοδολογίες Infrastructure και Hosting που υιοθετήθηκαν για το σύστημα παρακολούθησης των κυψελών.

2.1 Proxmox: Virtualization

Στην καρδιά της υποδομής φιλοξενίας μας βρίσκεται το Proxmox, μια λύση διαχείρισης Virtual Hosts, ανοιχτού κώδικα. Το Proxmox διευκολύνει τη διαχείριση Virtual Machines, Containers, ακόμη και αποθηκευτικών μέσων, καθιστώντας το ένα ολοκληρωμένο εργαλείο στο οπλοστάσιό μας. Γεννημένο από την ανάγκη για κλιμακούμενους υπολογιστές υψηλής απόδοσης, το Proxmox έχει γίνει γρήγορα αγαπημένο σε πολλούς κλάδους, και η εφαρμογή μας δεν αποτελεί εξαίρεση.

Οι εγγενείς δυνατότητες του Proxmox να εκτελεί πολλαπλά απομονωμένα συστήματα σε έναν μόνο φυσικό διακομιστή ευθυγραμμίζονται απόλυτα με τις ανάγκες μας, εξασφαλίζοντας ότι τα συστήματα παρακολούθησης είναι τόσο καταναμημένα όσο και βελτιστοποιημένα. Πέρα από τις βασικές του λειτουργίες, το Proxmox ξεχωρίζει για την ευκολία χρήσης του. Είτε πρόκειται για τη διαισθητική διεπαφή ιστού είτε για το ισχυρό API, η ενσωμάτωση και η διαχείριση των συστημάτων παρακολούθησης ήταν παιχνιδάκι[5].



Type	Description	Disk usage	Memory us...	CPU usage	Uptime	Host CPU	Host Mem	Tags
lxc	106 (DNS)	4.5 %	45.2 %	3.6% of 1 ...	10 days 13.3...	0.9% of 4C...	2.0 %	
lxc	107 (VPN)	6.4 %	3.0 %	0.8% of 2 ...	6 days 11.16...	0.4% of 4C...	0.5 %	
node	proxmox	94.3 %	29.7 %	5.2% of 4 ...	10 days 13.3...			
qemu	104 (Docker)	0.0 %	40.7 %	1.4% of 4 ...	10 days 13.3...	1.4% of 4C...	27.9 %	
qemu	100 (HassOS)	0.0 %	3.4 %	3.1% of 2 ...	2 days 21.36...	1.6% of 4C...	0.6 %	
qemu	101 (Manjaro)	-	-	-	-	-	-	
qemu	102 (Windows)	-	-	-	-	-	-	
qemu	105 (Workspaces)	-	-	-	-	-	-	
storage	local (proxmox)	94.3 %	-	-	-	-	-	
storage	local-lvm (proxmox)	7.0 %	-	-	-	-	-	

Εικόνα 1 Proxmox UI

Πηγή: Thesis Infrastructure

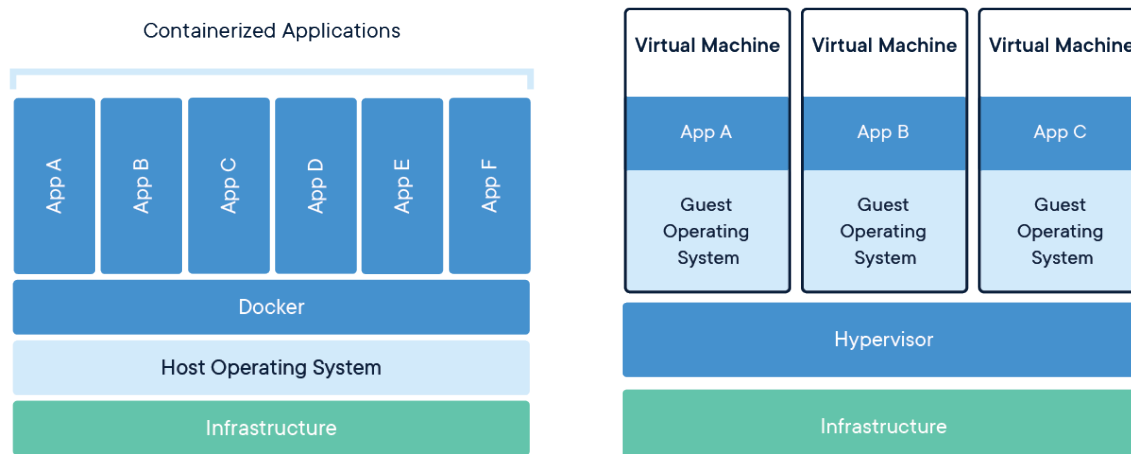
2.1.1 Ρόλος στο σύστημα παρακολούθησης κυψέλης

Για το σύστημα παρακολούθησης κυψέλης, το Proxmox διευκολύνει την απρόσκοπτη διαχείριση και κατανομή πόρων σε διάφορα στοιχεία όπως το Node-RED, το Grafana και το InfluxDB 2, το καθένα τοποθετημένο μέσα στα κοντέινερ και τα VM του. Αυτή η διάταξη εξασφαλίζει την απομονωμένη, ασφαλή και αποτελεσματική λειτουργία κάθε στοιχείου,

υποστηριζόμενη από τις ισχυρές και ευέλικτες δυνατότητες εικονικοποίησης (Virtualization) του Proxmox.

2.2 Docker: Το σύγχρονο πρότυπο για Containerized εφαρμογές

Καθώς η λύση μας απαιτούσε ένα συνεπές περιβάλλον σε διάφορα στάδια, από την ανάπτυξη έως την παραγωγή, ο ρόλος του Docker έγινε υψίστης σημασίας. Το Docker προσφέρει μια έξυπνη λύση σε ένα πανάρχαιο πρόβλημα - τη διασφάλιση ότι το λογισμικό τρέχει το ίδιο, ανεξάρτητα από το πού αναπτύσσεται. Πακετάροντας την εφαρμογή παρακολούθησης της κυψέλης μας σε κοντέινερ, το Docker διασφαλίζει ότι όλες οι εξαρτήσεις, οι βιβλιοθήκες και τα δυαδικά αρχεία είναι συνεπή, μετριάζοντας τυχόν απρόβλεπτα προβλήματα ανάπτυξης. Η ομορφιά του Docker έγκειται στην απλότητά του. Τα Containers, ελαφριά και ευέλικτα, παρέχουν ένα περιβάλλον όσο το δυνατόν πιο κοντά στην παραγωγή, διασφαλίζοντας ότι οι προγραμματιστές και οι μηχανικοί μας εργάζονται πάντα σε ένα ρεαλιστικό περιβάλλον. Αυτό όχι μόνο επιταχύνει τον αγωγό ανάπτυξής μας, αλλά μειώνει επίσης τις περιπτώσεις απρόβλεπτων σφαλμάτων ή προβλημάτων.



Εικόνα 2 Docker Architecture

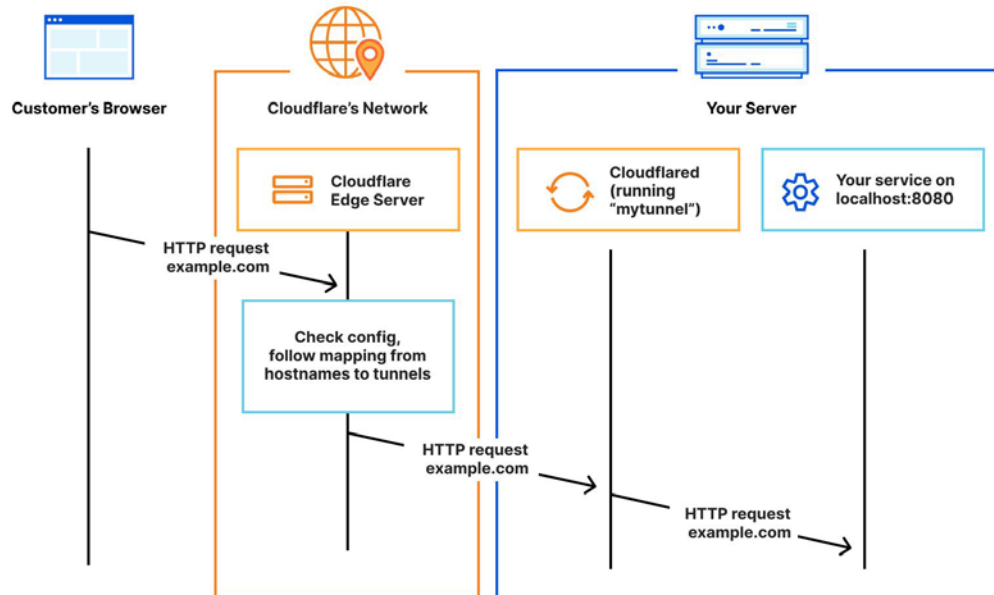
Πηγή: <http://www.docker.com>

2.3 Εξασφαλίζοντας την ασφάλεια μέσω Cloudflare Tunnels

Στην εποχή των απειλών στον κυβερνοχώρο και των παραβιάσεων δεδομένων, η διασφάλιση της ακεραιότητας και της ασφάλειας των διαδικτυακών συστημάτων μας δεν είναι απλώς σημαντική, αλλά επιτακτική ανάγκη. Εισάγοντας τα Cloudflare-Tunnels, την επιλεγμένη μέθοδο για την ασφαλή έκθεση των διαδικτυακών υπηρεσιών μας. Τα Cloudflare Tunnels λειτουργούν ως προστατευτικοί αγωγοί, διασφαλίζοντας ότι οποιαδήποτε δεδομένα κινούνται μεταξύ των διακομιστών μας και των τελικών χρηστών είναι κρυπτογραφημένα και ασφαλή. Πέρα από την ασφάλεια, η Cloudflare προσφέρει απaráμιλλη απόδοση[2].

Αξιοποιώντας το τεράστιο παγκόσμιο δίκτυο της Cloudflare, η λύση παρακολούθησης κυψέλης μας δεν είναι μόνο ασφαλής αλλά και εξαιρετικά γρήγορη, διασφαλίζοντας ότι οι χρήστες λαμβάνουν δεδομένα σε πραγματικό χρόνο χωρίς καμία καθυστέρηση. Η δημιουργία και

διαχείριση των Cloudflare tunnels είναι επίσης μια απόδειξη της προσέγγισης της Cloudflare που επικεντρώνεται στον χρήστη, όπου η ισχυρή ασφάλεια δεν έχει ως κόστος την πολυπλοκότητα.



Εικόνα 3 Cloudflare Tunnels

Πηγή: <http://www.cloudflare.com>

3. Tech Stack - Επιλογή, χρήση και ανάλυση

3.1 Η ανάγκη για το ESP32

Στον τεράστιο τομέα του IoT, το βασικό στοιχείο που γεφυρώνει το χάσμα μεταξύ του φυσικού και του ψηφιακού κόσμου είναι ο Μικροελεγκτής. Ο μικροελεγκτής ESP32, με τις ισχυρές του δυνατότητες, δεν χρησιμεύει απλώς ως συλλέκτης δεδομένων, αλλά ως εγκέφαλος πίσω από το σύστημα παρακολούθησης της κυψέλης, διασφαλίζοντας ότι αποκτώνται και μεταδίδονται αποτελεσματικά ακριβείς πληροφορίες σε πραγματικό χρόνο σχετικά με την κατάσταση της κυψέλης [3].

3.1.1 Ανταγωνιστικά εργαλεία

Η αγορά μικροελεγκτών προσφέρει πληθώρα επιλογών, καθεμία από τις οποίες διαθέτει τα μοναδικά της χαρακτηριστικά. Μερικοί από τους αξιοσημείωτους ανταγωνιστές του ESP32 περιλαμβάνουν το Arduino, το Raspberry Pi και το STM32. Αυτοί οι μικροελεγκτές, ενώ είναι εξαιρετικοί από τη θέση τους, ανταποκρίνονται σε διαφορετικές ανάγκες και απαιτήσεις έργων.

3.1.2 Γιατί επιλέχθηκε το ESP32

Η επιλογή του σωστού μικροελεγκτή μπορεί να επηρεάσει σημαντικά την επιτυχία του έργου. Μεταξύ των ομοειδών του, ο ESP32 προσφέρει ξεχωριστά πλεονεκτήματα. Η χαμηλή του κατανάλωση ενέργειας το καθιστά κατάλληλο για μακροχρόνιες εφαρμογές χωρίς συχνές αλλαγές μπαταρίας. Επιπλέον, οι ενσωματωμένες δυνατότητες Wi-Fi και Bluetooth που διαθέτει εξαλείφουν την ανάγκη για εξωτερικές μονάδες, απλοποιώντας τη σχεδίαση. Τέλος, η αποδοτικότητά του εξασφαλίζει ότι το έργο παραμένει φιλικό προς τον προϋπολογισμό χωρίς συμβιβασμούς στην ποιότητα ή τη λειτουργικότητα.

3.1.3 Χαρακτηριστικά και δυνατότητες

Το ESP32 δεν είναι απλώς ένας Μικροελεγκτής, είναι μια μονάδα παραγωγής ενέργειας με πολλά χαρακτηριστικά. Η στέγαση ενός διπύρηνου επεξεργαστή σημαίνει ότι μπορεί να χειριστεί με ευκολία πολλαπλές εργασίες. Οι εκτεταμένες ακίδες εισόδου/εξόδου γενικού σκοπού (GPIO) προσφέρουν ευελιξία στη σύνδεση διαφόρων αισθητήρων. Επιπλέον, η λειτουργία βαθιάς αναστολής λειτουργίας του εξασφαλίζει ελάχιστη κατανάλωση ενέργειας κατά τη διάρκεια ανενεργών περιόδων. Με δυνατότητες επέκτασης της μνήμης με τη χρήση εξωτερικών καρτών και τη δυνατότητα ενσωμάτωσης με διάφορες πλατφόρμες και πρωτόκολλα, οι δυνατότητες με το ESP32 είναι σχεδόν απεριόριστες. [3]

3.1.4 Χρήση στο έργο

Στο πλαίσιο του συστήματος παρακολούθησης της κυψέλης, το ESP32 έπαιξε καθοριστικό ρόλο. Προγραμματίστηκε για να συλλέγει περιοδικά δεδομένα από την κυψέλη, όπως μετρήσεις βάρους από τον αισθητήρα HX711. Οι δυνατότητές του Wi-Fi εξασφάλιζαν ότι τα δεδομένα αυτά μεταδίδονταν με συνέπεια στον διακομιστή μέσω του πρωτοκόλλου MQTT. Επιπλέον, η ενεργειακή του απόδοση αποδείχθηκε ζωτικής σημασίας για την αντιμετώπιση προκλήσεων όπως η αποφόρτιση της μπαταρίας. Με τη βελτιστοποίηση των ρυθμίσεών του και την αξιοποίηση της λειτουργίας **βαθιάς αναστολής λειτουργίας (Deep Sleep)**[3], εξασφαλίστηκε η μακροζωία του συστήματος.

3.2 Πρωτόκολλο MQTT

3.2.1 Η ανάγκη για το MQTT

Το MQTT, ακρωνύμιο των Message Queuing Telemetry Transport, γεννήθηκε από την ανάγκη για ένα ελαφρύ και αποτελεσματικό πρωτόκολλο επικοινωνίας. Σε περιβάλλοντα όπου οι συσκευές συχνά περιορίζονται από την ισχύ και την αξιοπιστία του δικτύου, όπως τα μελίσσια σε απομακρυσμένες τοποθεσίες, υπάρχει ουσιαστική ανάγκη για αποτελεσματικούς μηχανισμούς μετάδοσης δεδομένων. Ο σχεδιασμός του MQTT είναι προσαρμοσμένος ώστε να διασφαλίζει ελάχιστη κατανάλωση εύρους ζώνης δικτύου και μειωμένη χρήση πόρων της συσκευής, καθιστώντας το ιδανικό για τη μετάδοση δεδομένων από μικροελεγκτές, όπως ο ESP32, σε διακομιστές.

3.2.2 Ανταγωνιστικά εργαλεία

Ενώ το MQTT ξεχωρίζει για πολλές εφαρμογές IoT, δεν είναι χωρίς ανταγωνισμό. Το CoAP ή Constrained Application Protocol (Πρωτόκολλο περιορισμένης εφαρμογής) είναι ένας άλλος υποψήφιος που έχει σχεδιαστεί ειδικά για περιορισμένες συσκευές και εφαρμογές χαμηλής κατανάλωσης ενέργειας. Το πλεονέκτημά του έγκειται στο μοντέλο αλληλεπίδρασης που

μοιάζει με το HTTP. Το AMQP, από την άλλη πλευρά, σημαίνει Advanced Message Queuing Protocol και είναι πιο κατάλληλο για ισχυρή δρομολόγηση και ουρά μηνυμάτων. Αξίζει να σημειωθεί ότι το AMQP προσφέρει πιο προηγμένα χαρακτηριστικά, τα οποία, αν και ισχυρά, το καθιστούν επίσης βαρύτερο από το MQTT.

3.2.3 Γιατί επιλέχθηκε το MQTT

Υπάρχον διάφοροι λόγοι που κάνουν το MQTT να πλεονεκτεί σε σχέση με τα υπόλοιπα:

- **Αποτελεσματικότητα:** Η ελαφριά φύση του MQTT εξασφαλίζει ελάχιστη επιβάρυνση δεδομένων, ζωτικής σημασίας για εφαρμογές πραγματικού χρόνου.
- **Μοντέλο δημοσίευσης-συνδρομής (Publish-Subscribe):** Αυτό το μοντέλο αποφεύγει την ανάγκη για συνεχή αναζήτηση. Οι συσκευές μπορούν να "εγγραφούν" σε θέματα και να λαμβάνουν ενημερώσεις μόλις αυτές "δημοσιευτούν", εξασφαλίζοντας ενημερώσεις σε πραγματικό χρόνο.
- **Επεκτασιμότητα:** Το MQTT μπορεί να χειριστεί μεταδόσεις δεδομένων χωρίς σημαντική επιβάρυνση.

Αυτά τα χαρακτηριστικά, σε συνδυασμό με τις χαμηλές απαιτήσεις του πρωτοκόλλου σε εύρος ζώνης, κατέστησαν το MQTT την καλύτερη επιλογή για τις ανάγκες μεταφοράς δεδομένων σε πραγματικό χρόνο του έργου.

3.2.4 Χαρακτηριστικά και δυνατότητες

Το MQTT είναι κάτι περισσότερο από ένα απλό πρωτόκολλο ανταλλαγής μηνυμάτων. Είναι γεμάτο με χαρακτηριστικά που ενισχύουν τη χρηστικότητά του:

- **Επίπεδα ποιότητας υπηρεσιών (QoS):** Το MQTT προσφέρει τρία επίπεδα QoS για την κάλυψη διαφορετικών απαιτήσεων παράδοσης - από "μήνυμα που παραδίδεται το πολύ μία φορά" έως "μήνυμα που παραδίδεται ακριβώς μία φορά".
- **Τελευταία βούληση και διαθήκη (Last Will and Testament):** Θα αποσταλεί αν ο διακομιστής εντοπίσει αποσύνδεση που δεν είναι καθαρή.
- **Διατηρημένα μηνύματα:** εξασφαλίζοντας ότι οι νέοι συνδρομητές θα λαμβάνουν αμέσως τα τελευταία δεδομένα κατά την εγγραφή τους.

3.2.5 Χρήση στο έργο

Το MQTT ενσωματώθηκε απρόσκοπτα στο έργο παρακολούθησης των κυψελών. Ο Μικροελεγκτής ESP32 ρυθμίστηκε ώστε να συλλέγει δεδομένα και στη συνέχεια να τα "δημοσιεύει" σε συγκεκριμένα θέματα. Από την άλλη πλευρά, ο διακομιστής, εξοπλισμένος με τα κατάλληλα εργαλεία οπτικοποίησης και βάσης δεδομένων, "εγγράφηκε" σε αυτά τα θέματα. Τη στιγμή που τα δεδομένα δημοσιεύονταν, προωθούνταν αμέσως στον διακομιστή, επιτρέποντας την οπτικοποίηση και ανάλυση δεδομένων σε πραγματικό χρόνο. Αυτή η συνέργεια μεταξύ του MQTT, του ESP32 και των εργαλείων οπτικοποίησης ήταν καθοριστική για τη διασφάλιση της επιτυχίας του έργου.

3.3 Πλατφόρμα Docker (συγκεκριμένα Node-RED, InfluxDB2 και Grafana)

3.3.1 Η ανάγκη για το Docker

Στο εξελισσόμενο τοπίο της ανάπτυξης και διαχείρισης λογισμικού, η ανάγκη διατήρησης της συνέπειας σε διαφορετικά περιβάλλοντα είναι υψίστης σημασίας. Εδώ είναι που μπαίνει στη μάχη το Docker. Ως πλατφόρμα, το Docker επιτρέπει στους προγραμματιστές να δημιουργούν, να αναπτύσσουν και να εκτελούν εφαρμογές μέσα σε Containers. Αυτά τα κοντέινερ ενθυλακώνουν μια εφαρμογή με όλες τις εξαρτήσεις της, διασφαλίζοντας ότι εκτελείται ομοιόμορφα, ανεξάρτητα από το περιβάλλον ανάπτυξης. Αυτή η ομοιομορφία είναι ιδιαίτερα κρίσιμη σε έργα όπως η παρακολούθηση της κυψέλης, όπου η συνέπεια στην επεξεργασία και οπτικοποίηση δεδομένων είναι κρίσιμη.

3.3.2 Ανταγωνιστικά εργαλεία

Το Docker είναι αναμφισβήτητα ο πρωτοπόρος στον κόσμο της διαχείρισης εμπορευματοκιβωτίων, αλλά δεν είναι μόνο του. Το Kubernetes, για παράδειγμα, υπερέρχει στην ενορχήστρωση εμπορευματοκιβωτίων και είναι ιδιαίτερα χρήσιμο όταν διαχειρίζεται εφαρμογές πολλαπλών εμπορευματοκιβωτίων σε κλίμακα. Το Rodman, από την άλλη πλευρά, είναι παρόμοιο με το Docker, αλλά δεν απαιτεί δαίμονα για να εκτελεστεί και μπορεί να χρησιμοποιηθεί από χρήστες που δεν είναι χρήστες root. Και τα δύο προσφέρουν μοναδικά χαρακτηριστικά που ανταποκρίνονται σε συγκεκριμένες ανάγκες.

3.3.3 Γιατί επιλέχθηκε το Docker

Αρκετοί επιτακτικοί λόγοι έκαναν το Docker το εργαλείο μεταφοράς εμπορευματοκιβωτίων της επιλογής για αυτό το έργο:

- **Ευκολία χρήσης:** Η απλή διεπαφή γραμμής εντολών του Docker και τα φιλικά προς το χρήστη αρχεία Docker κάνουν τη διαδικασία δημιουργίας και διαχείρισης των εμπορευματοκιβωτίων απλή.
- **Κοινοτική υποστήριξη:** Το Docker διαθέτει μια τεράστια και ενεργή κοινότητα. Αυτό εξασφαλίζει πληθώρα διαθέσιμων πόρων, είτε πρόκειται για σεμινάρια, είτε για φόρουμ, είτε για έτοιμες εικόνες Docker.
- **Συμβατότητα πολλαπλών πλατφορμών:** από έναν τοπικό φορητό υπολογιστή έως έναν διακομιστή cloud, εξασφαλίζοντας ευελιξία στην ανάπτυξη.

3.3.4 Χαρακτηριστικά και δυνατότητες

Το Docker προσφέρει ένα πλήθος χαρακτηριστικών που αναβαθμίζουν τη χρησιμότητά του:

- **Ενορχήστρωση Containers:** Docker Compose διευκολύνει τη διαχείριση και ρύθμιση εφαρμογών Docker με πολλά Containers.

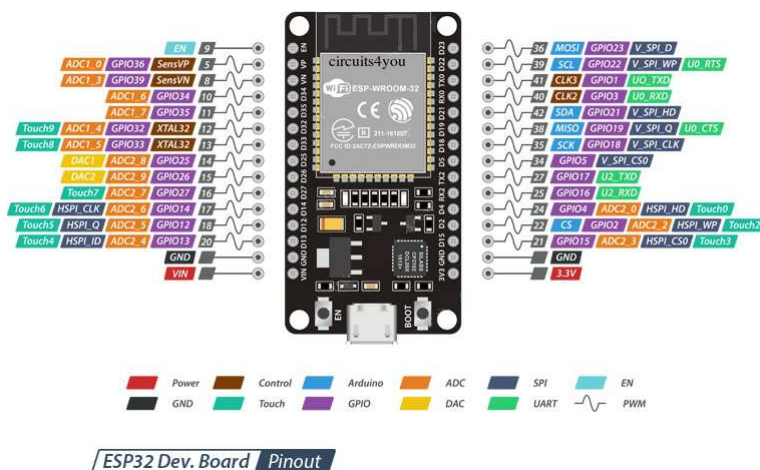
- **Απομόνωση:** Κάθε Container Docker εκτελείται απομονωμένα, διασφαλίζοντας ότι το περιβάλλον της εφαρμογής παραμένει συνεπές ανεξάρτητα από το πού αναπτύσσεται το Container.
- **Έλεγχος εκδόσεων:** επιτρέποντας στους προγραμματιστές να επανέλθουν σε προηγούμενες εκδόσεις αν χρειαστεί.
- **Συμβατότητα:** Docker μπορεί να τρέξει σε μια ποικιλία λειτουργικών συστημάτων, από Linux έως Windows, χωρίς προβλήματα.

3.3.5 Χρήση στο έργο

Για το έργο παρακολούθησης της κυψέλης, το Docker έπαιξε καθοριστικό ρόλο. Πλατφόρμες όπως η Node-RED, η InfluxDB2 και η Grafana χρησιμοποιήθηκαν σε Containers με τη χρήση του Docker. Αυτό εξασφάλιζε ότι κάθε μία από αυτές τις πλατφόρμες εκτελούσε με συνέπεια, ανεξάρτητα από τον τόπο ανάπτυξής τους. Επιπλέον, η ικανότητα του Docker να βελτιστοποιεί την ανάπτυξη σήμαινε ότι η κλιμάκωση του έργου, η προσθήκη περισσότερων κυψελών ή η ενσωμάτωση περισσότερων εργαλείων έγινε μια προσπάθεια χωρίς προβλήματα. Τα συνδυασμένα πλεονεκτήματα του Docker και των επιλεγμένων πλατφορμών εξασφάλισαν μια ομαλή και αποτελεσματική εμπειρία παρακολούθησης.

4. Μικροελεγκτής ESP32

Επομένως, η υγεία των κυψελών είναι υψίστης σημασίας, όχι μόνο για τις μέλισσες, αλλά και για το ευρύτερο οικολογικό σύστημα και την ανθρώπινη επιβίωση. Το προηγμένο σύστημα παρακολούθησης των κυψελών μας διαδραματίζει κρίσιμο ρόλο σε αυτό, και η επιτυχία του εξαρτάται από διάφορα αναπόσπαστα στοιχεία[3].



Εικόνα 4 ESP32

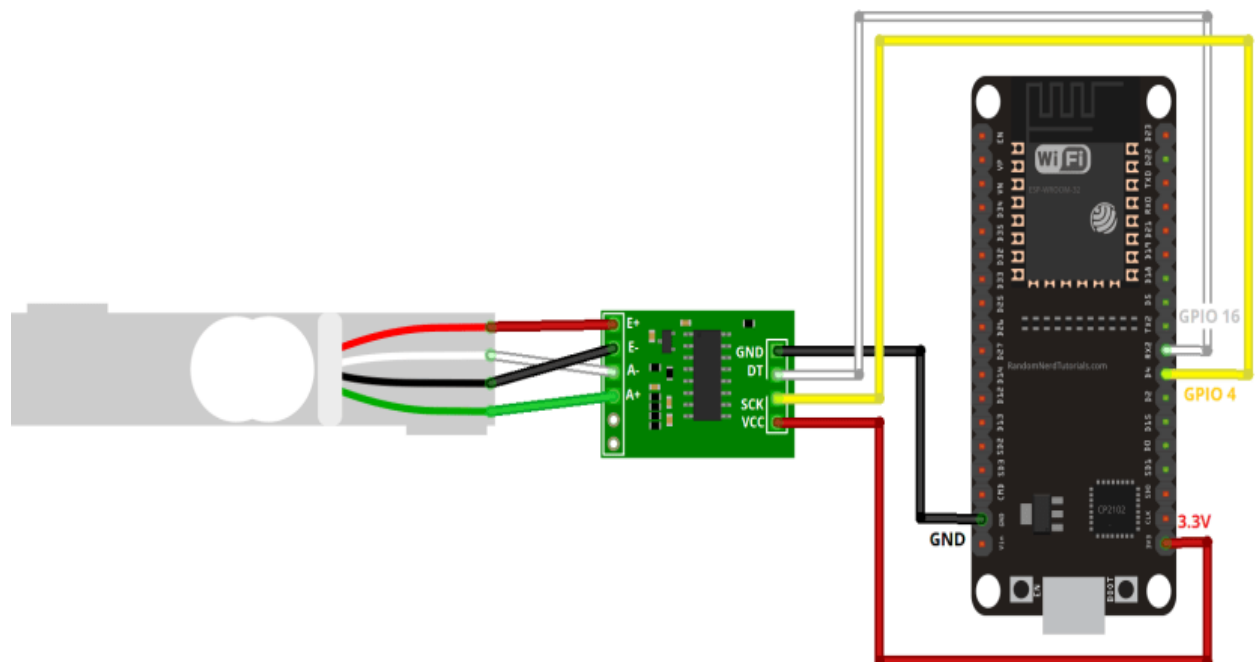
Πηγή: <https://circuits4you.com/>

4.1 Ενσωμάτωση με μικροελεγκτή ESP32

Ο μικροελεγκτής ESP32 είναι η καρδιά του συστήματος παρακολούθησης, που λειτουργεί ως κεντρική μονάδα συλλογής και επεξεργασίας δεδομένων. Η ευελιξία του, σε συνδυασμό με τη χαμηλή κατανάλωση ενέργειας και την πληθώρα ενσωματωμένων λειτουργιών, τον καθιστά ιδανική επιλογή για μια τόσο κρίσιμη εφαρμογή. Διασυνδέεται με μια ποικιλία αισθητήρων, συλλέγοντας ζωτικά δεδομένα όπως η θερμοκρασία, η υγρασία και το βάρος. Επιπλέον, η ενσωματωμένη μονάδα Wi-Fi διευκολύνει τη μετάδοση δεδομένων σε πραγματικό χρόνο, διασφαλίζοντας ότι τυχόν διακυμάνσεις ή ανωμαλίες θα αναφέρονται αμέσως.

4.2 Αξιοποίηση δεδομένων από τον ενισχυτή κυψελών φορτίου HX711

Το βάρος της κυψέλης χρησιμεύει ως κρίσιμος δείκτης της υγείας και της δραστηριότητάς της. Ο ενισχυτής κυψέλης φορτίου HX711, σε συνδυασμό με μια κυψέλη φορτίου ακριβείας (Load Cell), μας επιτρέπει να καταγράφουμε αυτές τις μετρήσεις βάρους με άψογη ακρίβεια. Αυτό όχι μόνο παρέχει πληροφορίες σχετικά με τους ρυθμούς παραγωγής μελιού, αλλά δίνει επίσης έγκαιρες προειδοποιήσεις για πιθανά προβλήματα, όπως η μείωση του πληθυσμού των μελισσών ή πιθανές εισβολές σε κυψέλες. Επιπλέον, η κατανόηση αυτών των δυναμικών βάρους επιτρέπει στους μελισσοκόμους να προσαρμόζουν τις παρεμβάσεις τους, εξασφαλίζοντας τη βέλτιστη παραγωγικότητα και υγεία των κυψελών.



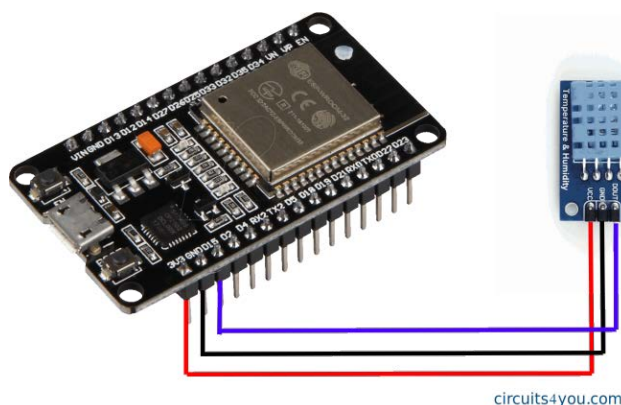
Εικόνα 5 ESP32 - HX711

Πηγή: <http://www.circuits4you.com>

4.3 Διαπιστώσεις σε πραγματικό χρόνο από τον αισθητήρα DHT11

Η θερμοκρασία και η υγρασία είναι δύο στοιχειώδεις παράγοντες που επηρεάζουν σημαντικά την υγεία των κυψελών. Το DHT11 είναι ένας χαμηλού κόστους αλλά εξαιρετικά αποτελεσματικός αισθητήρας που μετρά αυτές τις παραμέτρους. Με την ενσωμάτωσή του με

το ESP32, μπορούμε να παρακολουθούμε συνεχώς τις εσωτερικές συνθήκες της κυψέλης. Αυτό είναι ζωτικής σημασίας, καθώς οι μέλισσες είναι ευαίσθητες στο περιβάλλον τους. Οι ξαφνικές αυξήσεις ή μειώσεις της θερμοκρασίας μπορούν να επηρεάσουν αρνητικά τη συμπεριφορά των μελισσών, ακόμη και την επιβίωσή τους. Έχοντας δεδομένα σε πραγματικό χρόνο για αυτές τις παραμέτρους, μπορούν να γίνουν γρήγορες παρεμβάσεις, αποτρέποντας ενδεχομένως τυχόν μεγάλες κρίσεις[4].



Εικόνα 6 ESP32 - DHT11

Πηγή: <http://www.cicruits4you.com>

4.4 Έλεγχος κυψέλης με μαγνητικό αισθητήρα

Η ενσωμάτωση ενός μαγνητικού αισθητήρα αποτελεί σημαντική πρόοδο στα μέτρα ασφαλείας για τις μελισσοκομικές επιχειρήσεις. Αυτός ο αισθητήρας, όταν τοποθετείται σε μια κυψέλη, μπορεί να ανιχνεύσει μη εξουσιοδοτημένη πρόσβαση, λειτουργώντας αποτελεσματικά ως σύστημα έγκαιρης προειδοποίησης. Λειτουργεί με βάση την αρχή της διακοπής του μαγνητικού πεδίου- όταν παραβιάζεται η είσοδος της κυψέλης, το κύκλωμα διακόπτεται, ενεργοποιώντας μια ειδοποίηση. Αυτό όχι μόνο συμβάλλει στην προστασία των πολύτιμων μελισσοσμηνών από βανδαλισμούς και κλοπές, αλλά εξασφαλίζει επίσης ότι ο μελισσοκόμος ενημερώνεται αμέσως για τυχόν κινδύνους για την ακεραιότητα της κυψέλης. Με την αξιοποίηση αυτής της τεχνολογίας, οι μελισσοκόμοι μπορούν να διατηρούν μια μη παρεμβατική, αλλά άγρυπνη επιτήρηση των κυψελών τους, εξασφαλίζοντας την ασφάλεια και την προστασία των μελισσών και του ενδιαιτήματός τους.



Εικόνα 7 Manget Sensor

Πηγή: Thesis

4.5 Τροφοδοσία του συστήματος με την μπαταρία 18650

Μια αποδοτική και μακράς διάρκειας πηγή ενέργειας είναι ζωτικής σημασίας για την αδιάλειπτη λειτουργία του συστήματός μας. Η μπαταρία 18650, μια φημισμένη επαναφορτιζόμενη παραλλαγή ιόντων λιθίου, αναδεικνύεται ως η προφανής επιλογή. Η ενεργειακή πυκνότητά της εξασφαλίζει ότι το σύστημα παρακολούθησης παραμένει λειτουργικό για μεγάλα χρονικά διαστήματα, μειώνοντας τη συχνότητα των χειροκίνητων επεμβάσεων από τους μελισσοκόμους. Επιπλέον, οι λειτουργίες βελτιστοποίησης ισχύος του ESP32, όταν συνδυάζονται με αυτή την μπαταρία, εξασφαλίζουν μέγιστη διάρκεια λειτουργίας, ένα χαρακτηριστικό που αποδεικνύεται ανεκτίμητο σε απομακρυσμένες τοποθεσίες.



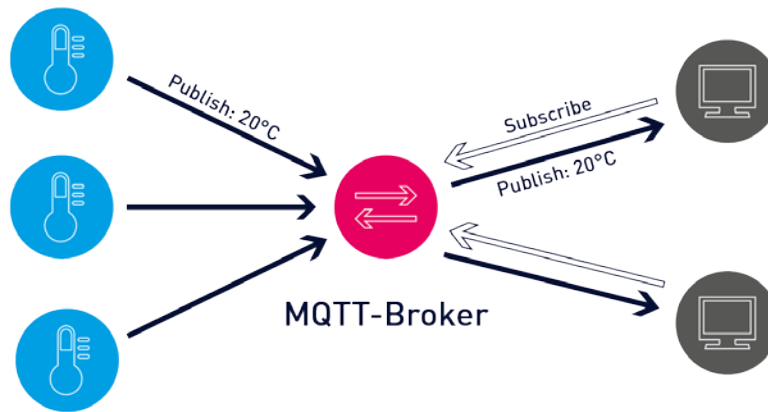
Εικόνα 8 18650 Battery

Πηγή: <https://www.18650batterystore.com/>

5. Μετάδοση δεδομένων & MQTT

5.1 Η σημασία του MQTT στην επικοινωνία δεδομένων σε πραγματικό χρόνο

Η ψηφιακή εποχή, η οποία χαρακτηρίζεται από μια ακόρεστη ανάγκη για ανταλλαγή δεδομένων, έχει οδηγήσει στην ανάπτυξη πολυάριθμων πρωτοκόλλων επικοινωνίας. Τα πρωτόκολλα αυτά είναι προσαρμοσμένα για την αντιμετώπιση συγκεκριμένων αναγκών, που κυμαίνονται από τη μεταφορά δεδομένων υψηλής ταχύτητας για τη ροή βίντεο έως την αργή αλλά σταθερή μεταφορά διαγνωστικών δεδομένων από αισθητήρες. Μεταξύ αυτών, το MQTT ξεχωρίζει ως το πρωτόκολλο για την επικοινωνία δεδομένων σε πραγματικό χρόνο, ιδίως σε σενάρια που απαιτούν ελαφριές ανταλλαγές μέσω πιθανώς ασταθών δικτύων.



Εικόνα 9 MQTT Broker

Πηγή: <http://www.www.paessler.com>

5.1.1 Εξέλιξη των πρωτοκόλλων μετάδοσης δεδομένων

Τα σύγχρονα συστήματα παρακολούθησης, ιδίως σε σενάρια πραγματικού χρόνου, απαιτούν τη χρήση αποτελεσματικών πρωτοκόλλων μετάδοσης δεδομένων. Με την πάροδο των ετών, τα πρωτόκολλα έχουν εξελιχθεί από τη βασική σειριακή επικοινωνία σε σύνθετες ασύρματες λύσεις, καλύπτοντας τις απαιτήσεις υψηλότερης ταχύτητας και αξιοπιστίας.

5.1.2 MQTT: Το βασικό πρωτόκολλο για το IoT

Το MQTT (Message Queuing Telemetry Transport) είναι ένα ελαφρύ πρωτόκολλο ανταλλαγής μηνυμάτων βελτιστοποιημένο για δίκτυα χαμηλού εύρους ζώνης και υψηλής καθυστέρησης, καθιστώντας το ιδανικό για εφαρμογές Internet of Things (IoT). Ο σχεδιασμός του αποσκοπεί κυρίως στη διασφάλιση της αποδοτικής μετάδοσης δεδομένων σε σενάρια όπου η αξιοπιστία του δικτύου μπορεί να αποτελεί ένα πρόβλημα.

5.1.3 Χαρακτηριστικά που κάνουν το MQTT ευνοϊκό για την παρακολούθηση της κυψέλης

Με την ικανότητά του να προσφέρει επικοινωνία σε πραγματικό χρόνο, χαμηλή κατανάλωση ενέργειας και εξασφαλισμένη παράδοση μηνυμάτων, το MQTT γίνεται προφανής επιλογή για εφαρμογές όπως η παρακολούθηση κυψέλης. Επιπλέον, τα επίπεδα QoS (Quality of Service) διασφαλίζουν ότι κρίσιμα δεδομένα, όπως μια ξαφνική αύξηση της θερμοκρασίας, δεν θα χαθούν ποτέ.

5.2 Ρύθμιση και διαμόρφωση του MQTT Broker

Ο διαμεσολαβητής (Broker) βρίσκεται στην καρδιά του MQTT, διαχειριζόμενος τα μηνύματα μεταξύ των συσκευών αποστολέα και παραλήπτη.

5.2.1 Ο ρόλος ενός MQTT Broker

Στην ορολογία του MQTT, ο **Broker** είναι ένας διακομιστής που μεσολαβεί στις ανταλλαγές μηνυμάτων μεταξύ συσκευών. Λαμβάνει μηνύματα, τα επεξεργάζεται, καθορίζει τους

κατάλληλους συνδρομητές και στη συνέχεια αποστέλλει το μήνυμα σε αυτούς τους συνδρομητές, εξασφαλίζοντας την απρόσκοπτη επικοινωνία σε ένα δίκτυο.

5.2.2 Κριτήρια για την επιλογή ενός αποδοτικού MQTT Broker

Για ένα σύστημα τόσο ζωτικής σημασίας όσο η παρακολούθηση κυψέλης, είναι ζωτικής σημασίας να επιλέξετε έναν μεσίτη που εγγυάται χρόνο λειτουργίας, χειρίζεται τεράστιο αριθμό πελατών και διασφαλίζει την ασφάλεια των δεδομένων. Παράγοντες όπως η επεκτασιμότητα, η ευρωστία και η υποστήριξη διαφόρων επιπέδων QoS είναι καθοριστικής σημασίας για την επιλογή αυτή.

5.2.3 Πρακτικά βήματα για τη διαμόρφωση του διαμεσολαβητή για βέλτιστη απόδοση

Η διαμόρφωση ενός Broker MQTT δεν αφορά απλώς την εγκατάσταση, αλλά τη λεπτομερή ρύθμισή του για τη συγκεκριμένη εφαρμογή. Αυτό περιλαμβάνει τη ρύθμιση των κατάλληλων δομών θεμάτων (topics), τη διασφάλιση της ασφάλειας με κρυπτογράφηση SSL/TLS και την ενεργοποίηση της καταγραφής για την αντιμετώπιση προβλημάτων και την ανάλυση επιδόσεων.

6. Node-RED μέσα στο Docker

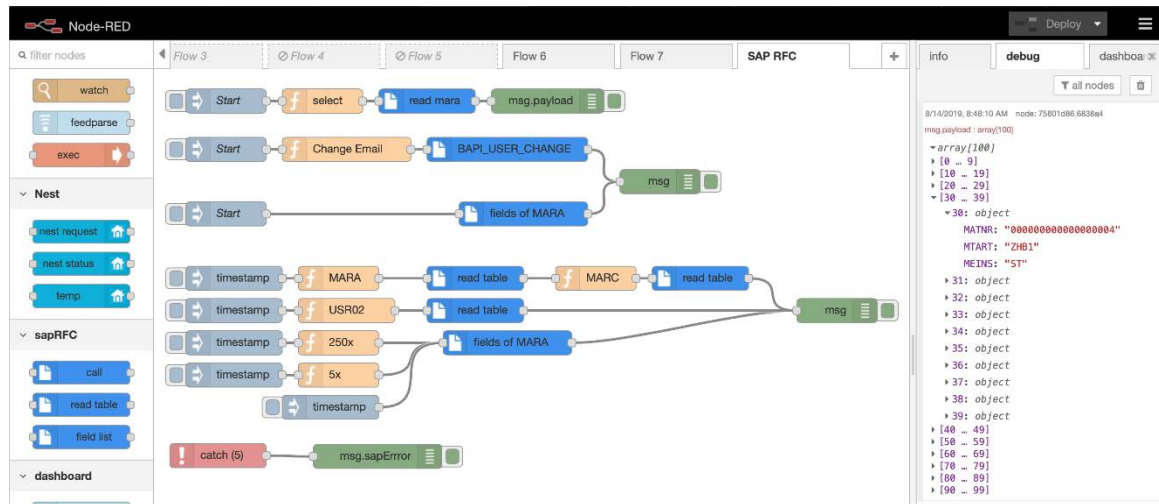
Σε μια εποχή που κυριαρχείται από την αλληλεπίδραση των δεδομένων σε διάφορες συσκευές και πλατφόρμες, η σημασία της διαχείρισης της ροής δεδομένων δεν μπορεί να υποτιμηθεί. Το Node-RED, ένα εργαλείο ανάπτυξης βασισμένο στη ροή δεδομένων, επιτρέπει στους χρήστες να συνδέουν συσκευές, API και διαδικτυακές υπηρεσίες με νέους τρόπους. Ωστόσο, με την έλευση του containerization και την πανταχού παρούσα χρήση του Docker, η φιλοξενία τέτοιων εργαλείων έχει πάρει μια νέα κατεύθυνση, μεγιστοποιώντας την αποδοτικότητα και την επεκτασιμότητα.

6.1 Εισαγωγή και εγκατάσταση του Node-RED

Το γραφικό περιβάλλον του Node-RED απλοποιεί τη διαδικασία δημιουργίας, κατανόησης και διαχείρισης ροών εργασίας. Δημιουργήθηκε από την ομάδα Emerging Technology της IBM και βασίζεται στο Node.js, γεγονός που το καθιστά εγγενώς ελαφρύ και ικανό να εκτελείται σε συσκευές χαμηλής κατανάλωσης ενέργειας, μια κρίσιμη πτυχή για εφαρμογές όπως η παρακολούθηση κυψέλης.

Γιατί Node-RED; Η δύναμη του Node-RED έγκειται στο σύστημα "κόμβων". Κάθε κόμβος έχει έναν συγκεκριμένο σκοπό, από την ανάγνωση δεδομένων από έναν αισθητήρα έως την πραγματοποίηση μιας κλήσης API ή ακόμη και τη μετατροπή δεδομένων. Με την απλή σύνδεση αυτών των κόμβων με λογικό τρόπο, μπορούν να επιτευχθούν πολύπλοκες εργασίες χωρίς την ανάγκη περίπλοκης κωδικοποίησης. Αυτή η φιλικότητα προς τον χρήστη διασφαλίζει ότι ακόμη και όσοι έχουν περιορισμένες γνώσεις κωδικοποίησης μπορούν να δημιουργήσουν και να διαχειριστούν ροές δεδομένων κυψέλης.

Η εγκατάσταση του Node-RED είναι πανεύκολη, ειδικά στο πλαίσιο του Docker. Η ενθουσία που παρέχει το Docker εξασφαλίζει ότι το Node-RED εκτελείται σε ένα απομονωμένο περιβάλλον, αποτρέποντας πιθανές συγκρούσεις με άλλο λογισμικό και απλοποιώντας τον έλεγχο εκδόσεων.



Εικόνα 10 NodeRed UI

Πηγή: <http://www.nodered.org>

6.2 Επεξεργασία δεδομένων, ενορχήστρωση και δρομολόγηση στο InfluxDB2

Μόλις δημιουργηθούν οι ροές δεδομένων στο Node-RED, το επόμενο βήμα είναι η επεξεργασία των δεδομένων. Εδώ, τα ακατέργαστα δεδομένα από τους αισθητήρες μπορούν να μετασχηματιστούν, να αναλυθούν και να προετοιμαστούν για αποθήκευση ή περαιτέρω χρήση.

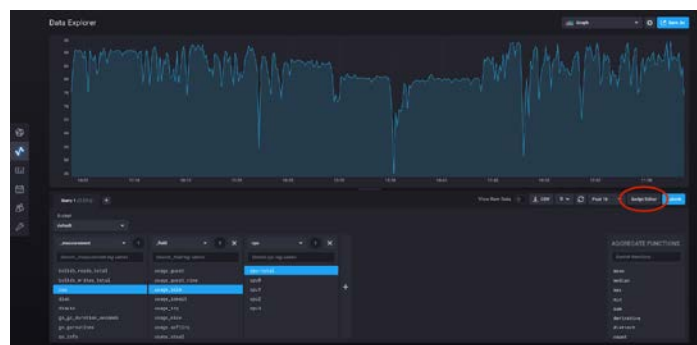
- **Μετασχηματισμός δεδομένων:** Δεδομένης της ποικιλόμορφης φύσης των δεδομένων από την παρακολούθηση των κυψελών, ενδέχεται να απαιτείται μετατροπή τους σε τυποποιημένη μορφή ή μονάδα. Οι κόμβοι λειτουργιών του Node-RED μπορούν να χρησιμοποιηθούν για την απρόσκοπτη εκτέλεση αυτών των λειτουργιών.
- **Ενορχήστρωση:** Πρόκειται επίσης για ενορχήστρωση. Ανάλογα με ορισμένες συνθήκες, μπορούν να ενεργοποιηθούν διαφορετικές ροές. Για παράδειγμα, εάν η θερμοκρασία στο εσωτερικό μιας κυψέλης υπερβεί ένα συγκεκριμένο όριο, μπορεί να ξεκινήσει μια ροή για να ειδοποιηθεί ο μελισσοκόμος.
- **Δρομολόγηση στην InfluxDB2:** Μετά την επεξεργασία, τα δεδομένα πρέπει να αποθηκευτούν για μελλοντική ανάλυση ή παρουσίαση σε πραγματικό χρόνο. Οι κόμβοι της Node-RED που έχουν σχεδιαστεί για βάσεις δεδομένων καθιστούν αυτή την εργασία απλή. Στην περίπτωση του InfluxDB2, μιας βάσης δεδομένων χρονοσειρών, τα δεδομένα από το Node-RED μπορούν να δρομολογηθούν αποτελεσματικά, διασφαλίζοντας τη διατήρηση της χρονικής φύσης των δεδομένων.

Συμπερασματικά, το Node-RED στο πλαίσιο του Docker παρέχει μια βέλτιστη πλατφόρμα για τη διαχείριση της ροής δεδομένων στην παρακολούθηση κυψέλης. Συνδυάζει τη δύναμη του

διαισθητικού σχεδιασμού του Node-RED με την αποτελεσματικότητα του Docker, προσφέροντας μια ισχυρή λύση για τις σύγχρονες προκλήσεις της μελισσοκομίας.

7. InfluxDB2 μέσα στο Docker

Στο χώρο των βάσεων δεδομένων χρονοσειρών, η InfluxDB2 ξεχωρίζει για τις υψηλές επιδόσεις της, την εύκολη γλώσσα ερωτημάτων και τη συμβατότητά της με εργαλεία απεικόνισης δεδομένων σε πραγματικό χρόνο, όπως η Grafana. Ειδικά για εφαρμογές όπως η παρακολούθηση κυψέλης, η δυνατότητα παρακολούθησης των μετρήσεων σε βάθος χρόνου είναι απαραίτητη και η InfluxDB2 παρέχει τα απαραίτητα εργαλεία για την επίτευξη αυτού του στόχου. Η εγκατάστασή του μέσα σε ένα δοχείο Docker απλοποιεί την εγκατάστασή του, διασφαλίζει την επεκτασιμότητα και προσφέρει περιβαλλοντική συνέπεια .



Εικόνα 11 InfluxDB2

Πηγή: <https://docs.influxdata.com/influxdb/v2/>

7.1 Δυνατότητες και διαμόρφωση του InfluxDB2

Η InfluxDB2, που αναπτύχθηκε από την InfluxData, έχει σχεδιαστεί ρητά για δεδομένα χρονοσειρών, καθιστώντας την ιδανική επιλογή για εφαρμογές IoT. Διευκολύνει ταχύτερες δυνατότητες ανάγνωσης και εγγραφής, βελτιστοποιημένες για σύνολα δεδομένων υψηλής καρδινικότητας. Με την InfluxQL, τη φιλική προς το χρήστη γλώσσα ερωτημάτων της, η εξαγωγή ουσιαστικών πληροφοριών γίνεται πιο απλή υπόθεση.

7.1.1 Χαρακτηριστικά που διαφοροποιούν το InfluxDB2

Ενώ η παραδοσιακή εγκατάσταση μπορεί να απαιτεί ρυθμίσεις ειδικά για το λειτουργικό σύστημα, το Docker προσφέρει μια πιο απλή προσέγγιση. Αντλώντας την εικόνα του InfluxDB2 από το Docker Hub και δημιουργώντας ένα δοχείο, μπορείτε να ξεκινήσετε τη λειτουργία μέσα σε λίγα λεπτά. Αυτή η ενότητα θα σας καθοδηγήσει με τα απαραίτητα βήματα και τις εντολές του Docker για να το πετύχετε αυτό.

7.1.2 Διαμόρφωση του InfluxDB2 στο Docker

Ενώ η παραδοσιακή εγκατάσταση μπορεί να απαιτεί ρυθμίσεις ειδικά για το λειτουργικό σύστημα, το Docker προσφέρει μια πιο απλή προσέγγιση. Αντλώντας την εικόνα του InfluxDB2 από το Docker Hub και δημιουργώντας ένα δοχείο (container), μπορείτε να ξεκινήσετε τη λειτουργία μέσα σε λίγα λεπτά. [6].

7.2 Στρατηγικές αποθήκευσης δεδομένων και ενσωμάτωση με την Grafana

Η InfluxDB2, ενώ είναι ισχυρή από μόνη της, λάμπει περισσότερο όταν ενσωματώνεται με εργαλεία οπτικοποίησης. Το Grafana είναι ένα τέτοιο εργαλείο που συνδυάζεται τέλεια με αυτό. Αλλά πριν από την οπτικοποίηση, η κατανόηση του μηχανισμού αποθήκευσης δεδομένων είναι απαραίτητη.

7.2.1 Buckets και διατήρηση δεδομένων

Σε αντίθεση με τις παραδοσιακές βάσεις δεδομένων, η InfluxDB2 χρησιμοποιεί "κάδους" για την αποθήκευση δεδομένων χρονοσειρών. Κάθε κάδος αντιστοιχεί σε ένα σύνολο δεδομένων, με την πολιτική διατήρησής του. Αυτό σημαίνει ότι μπορείτε να ορίσετε πόσο καιρό θα πρέπει να διατηρούνται τα δεδομένα σε έναν συγκεκριμένο κάδο[6].

7.2.2 Ενσωμάτωση με την Grafana

Η Grafana, μια πλατφόρμα ανοικτού κώδικα για την παρακολούθηση και την παρατηρησιμότητα, είναι ο τέλειος συνεργάτης για το InfluxDB2. Με το ποικίλο σύνολο πρόσθετων και τα διαισθητικά εργαλεία σχεδιασμού ταμπλό (Dashboard), η οπτικοποίηση δεδομένων χρονοσειρών γίνεται εύκολα.

8. Grafana μέσα στο Docker

Η Grafana έχει συγκεντρώσει σημαντική προσοχή στον κόσμο της οπτικοποίησης δεδομένων, ιδίως για την ικανότητά της να αναπαριστά μετρήσεις χρονοσειρών σε κατανοητές μορφές. Δεν πρόκειται μόνο για διαγράμματα- πρόκειται για την αφήγηση μιας ιστορίας με δεδομένα, κάνοντας τις ανωμαλίες ανιχνεύσιμες και τα μοτίβα διακριτά [1]. Η ενσωμάτωση του Grafana στο Docker εξασφαλίζει ένα συνεπές περιβάλλον, γρήγορη ανάπτυξη και επεκτασιμότητα, κάτι που είναι απαραίτητο για την παρακολούθηση κυψέλης σε πραγματικό χρόνο.



Εικόνα 12 Grafana UI

Πηγή: <https://www.grafana.com/>

8.1 Ρύθμιση της Grafana και των βασικών χαρακτηριστικών της

Η Grafana, ως εκ τούτου, είναι μια πλατφόρμα ανοικτού κώδικα που επιτρέπει στους χρήστες να μετατρέπουν τα συλλεγμένα δεδομένα σε όμορφα και ουσιαστικά ταμπλό. Με την ευέλικτη αρχιτεκτονική των plugins της, η Grafana υποστηρίζει ένα ευρύ φάσμα πηγών δεδομένων, συμπεριλαμβανομένου του InfluxDB2, καθιστώντας την ιδιαίτερα κατάλληλη για την περίπτωση χρήσης μας.

8.1.1 Ρύθμιση της Grafana και των βασικών χαρακτηριστικών της

- **Ποικίλες επιλογές οπτικοποίησης:** Η Grafana προσφέρει μια πληθώρα τύπων οπτικοποίησης προσαρμοσμένων για διαφορετικά είδη αφηγήσεων δεδομένων.
- **Μηχανισμός ειδοποίησης:** Στην παρακολούθηση σε πραγματικό χρόνο, η λήψη έγκαιρων ειδοποιήσεων μπορεί να είναι η διαφορά μεταξύ ενός μικρού προβλήματος και μιας καταστροφής. Το ενσωματωμένο σύστημα ειδοποιήσεων της Grafana ειδοποιεί τους χρήστες για τυχόν ανωμαλίες ή παραβιάσεις ορίων.
- **Εκτεταμένη αρχιτεκτονική πρόσθετων:** επιτρέποντας την ενσωμάτωση με αμέτρητες πηγές δεδομένων και την επέκταση των δυνατοτήτων απεικόνισης.

8.1.2 Ανάπτυξη της Grafana σε Docker

Όπως και η InfluxDB2, η Grafana μπορεί να εγκατασταθεί γρήγορα και αποτελεσματικά σε περιβάλλον Docker. Αξιοποιώντας τα Docker Containers, οι χρήστες μπορούν να διασφαλίσουν ότι η Grafana εκτελείται σε production περιβάλλον, απαλλαγμένο από ασυνέπειες που σχετίζονται με το σύστημα.

8.2 Δημιουργία ταμπλό για αποτελεσματική παρακολούθηση κυψέλης

Μόλις η Grafana τεθεί σε λειτουργία, το επόμενο βήμα είναι η αξιοποίηση των δυνατοτήτων της για την οπτικοποίηση των μετρήσεων της κυψέλης με τον πιο διορατικό τρόπο. Η δημιουργία ταμπλό (Dashboards) δεν είναι απλώς κατατοπιστική αλλά και διαδραστική.

8.2.1 Gauges

Τα Gauges είναι απαραίτητα για την παρακολούθηση δεδομένων σε πραγματικό χρόνο σε ένα ταμπλό Grafana. Αναπαριστούν οπτικά μετρικές όπως τα επίπεδα θερμοκρασίας και υγρασίας μέσα σε μια κυψέλη, χρησιμοποιώντας μια βελόνα για να δείξουν την τρέχουσα κατάσταση σε σχέση με μια χρωματιστή κλίμακα που μπορεί να αναδείξει διαφορετικά κατώτατα όρια. Στην συγκεκριμένη υλοποίηση τα όρια (thresholds) τοποθετήθηκαν από 6 βαθμούς κελσίου.



Εικόνα 13 Grafana - Gauge

Πηγή: Thesis

8.2.2 Charts

Τα Charts είναι ευέλικτα εργαλεία στο Grafana που απεικονίζουν σημεία δεδομένων σε βάθος χρόνου, επιτρέποντας στους μελισσοκόμους να παρακολουθούν τάσεις και μοτίβα. Τα διαγράμματα χρονοσειρών, για παράδειγμα, μπορούν να παρουσιάσουν το επίπεδο δραστηριότητας των μελισσών ή τα ποσοστά παραγωγής μελιού, παρέχοντας πληροφορίες για την υγεία της κυψέλης.



Εικόνα 14 Grafana - Chart

Πηγή: Thesis



Εικόνα 15 Grafana - Graph

Πηγή: Thesis

8.2.3 Statistics

Τα Statistics, είναι οι πίνακες στατιστικών, όπου στα ταμπλό της Grafana προσφέρουν μια γρήγορη σύνοψη των βασικών μετρήσεων. Μπορούν να εμφανίσουν το μέσο ρυθμό ανάπτυξης του πληθυσμού μιας κυψέλης ή το συνολικό μέλι που συλλέγεται, προσφέροντας στους μελισσοκόμους μια στιγμιαία εικόνα διαφόρων αριθμητικών σημείων δεδομένων με μια ματιά.



Εικόνα 16 Grafana - Weight Stat

Πηγή: Thesis



Εικόνα 17 Grafana - State Stat

Πηγή: Thesis

8.2.4 GeoMap

Ο πίνακας GeoMap είναι χρήσιμος για μελισσοκόμους με πολλές κυψέλες σε διαφορετικές τοποθεσίες. Μπορεί να χαρτογραφήσει τα δεδομένα GPS κάθε κυψέλης, να απεικονίσει τις περιβαλλοντικές συνθήκες ή να παρακολουθήσει τα μοτίβα μετακίνησης των μελισσών, βοηθώντας στην κατανόηση των γεωγραφικών επιπτώσεων στην παραγωγικότητα των κυψελών. Στην συγκεκριμένη υλοποίηση δεν περιέχεται GPS αλλά θα χρησιμοποιηθεί για τυχόν αναβαθμίσεις.

9. Ολοκλήρωση συστήματος, ασφάλεια και δοκιμές

Καθώς οι τεχνολογίες εξελίσσονται και τα συστήματα παρακολούθησης γίνονται όλο και πιο εξελιγμένα, είναι ζωτικής σημασίας να διασφαλιστεί ότι όλα τα στοιχεία ενός συστήματος παρακολούθησης κυψελών λειτουργούν σε συνδυασμό. Αυτό το κεφάλαιο εξετάζει την ενσωμάτωση αυτών των επιμέρους στοιχείων, την ασφάλεια των δεδομένων και την εφαρμογή μεθοδολογιών δοκιμών για την εξασφάλιση της βέλτιστης απόδοσης.

9.1 Ενσωμάτωση όλων των στοιχείων σε ασφαλές περιβάλλον

Η ενσωμάτωση πολλαπλών τεχνολογιών, όπως μικροελεγκτές, βάσεις δεδομένων και εργαλεία οπτικοποίησης, απαιτεί προσεκτικό σχεδιασμό και υλοποίηση. Ο στόχος είναι να εξασφαλιστεί η απρόσκοπτη επικοινωνία μεταξύ κάθε συστατικού, επιτρέποντας έτσι την αρμονική λειτουργία ολόκληρου του συστήματος.

9.1.1 Η διαδικασία ολοκλήρωσης

Σύνδεση ESP32 και MQTT: Το ESP32, που είναι ο κύριος συλλέκτης δεδομένων, χρειαζόταν μια αδιάλειπτη σύνδεση με το MQTT. Η σύνδεση αυτή εξασφάλιζε την ταχεία μετάδοση των δεδομένων της κυψέλης χωρίς καμία απώλεια. Η Βιβλιοθήκη που χρησιμοποιήθηκε για την συγκεκριμένη υλοποίηση είναι η **PubSubClient**, η οποία είναι από τις πιο διαδεδομένες βιβλιοθήκες που είναι συμβατή με την πληθώρα μικροελεγκτών (ESP32, Arduino, Raspberry)

Ροή δεδομένων από το MQTT στις πλατφόρμες Docker: Τα δεδομένα από το ESP32 που μεταδίδονταν μέσω MQTT κατευθύνονταν στις πλατφόρμες Docker. Το **Node-RED ενήργησε ως διαμεσολαβητής**, επεξεργάστηκε τα δεδομένα αυτά και τα ανακατεύθυνε στο **InfluxDB2** για αποθήκευση. Η Grafana είχε στη συνέχεια πρόσβαση σε αυτά τα αποθηκευμένα δεδομένα, οπτικοποιώντας τα για εύκολη ερμηνεία.

Διασύνδεση υπηρεσιών Docker: Δεδομένων των πολλαπλών υπηρεσιών εντός του Docker, ένα αρχείο **Docker-Compose** ήταν καθοριστικό για την υπαγόρευση των αλληλεπιδράσεων μεταξύ των Containers. Αυτό επέτρεψε σε πλατφόρμες όπως η Node-RED, η InfluxDB2 και η Grafana να μοιράζονται και να έχουν πρόσβαση σε δεδομένα απρόσκοπτα.

```

1 version: "3"
2
3 services:
4   influxdb:
5     image: influxdb:2.0
6     container_name: influxdb
7     ports:
8       - "8086:8086"
9     volumes:
10      - influxdb-data:/var/lib/influxdb2
11     environment:
12       - DOCKER_INFLUXDB_INIT_MODE=setup
13       - DOCKER_INFLUXDB_INIT_USERNAME=bee
14       - DOCKER_INFLUXDB_INIT_PASSWORD=bee
15       - DOCKER_INFLUXDB_INIT_ORG=bee
16       - DOCKER_INFLUXDB_INIT_BUCKET=bee
17
18   grafana:
19     image: grafana/grafana:latest
20     container_name: grafana
21     ports:
22       - "3000:3000"
23     volumes:
24       - grafana-data:/var/lib/grafana
25     depends_on:

```

Εικόνα 20 Docker Compose Script

Πηγή: Thesis

9.1.2 Προκλήσεις και λύσεις

Αποφόρτιση της μπαταρίας του ESP32: Ένα σημαντικό πρόβλημα που αντιμετωπίστηκε ήταν η ταχεία αποστράγγιση της μπαταρίας του ESP32. Αυτό απαιτούσε συχνές αντικαταστάσεις της μπαταρίας, διακόπτοντας τη συνεχή παρακολούθηση.

Προβλήματα δικτύου: Υπήρχαν περιπτώσεις όπου το σύστημα αντιμετώπιζε διακοπές λειτουργίας του δικτύου λόγω προβλημάτων του παρόχου υπηρεσιών διαδικτύου. Αυτό δημιούργησε προκλήσεις στη μετάδοση και ανάκτηση δεδομένων σε πραγματικό χρόνο.

Βαθμονόμηση του HX711: Ο αισθητήρας HX711 έδωσε αρχικά λανθασμένες μετρήσεις. Μόνο μετά το σχεδιασμό μιας μοναδικής πλατφόρμας για τον HX711 κάτω από την κυψέλη, οι ακριβείς μετρήσεις έγιναν σταθερές.



Εικόνα 21 HX711 Platform



Εικόνα 22 Beehive with Platform

Πηγή: Thesis

9.2 Ασφάλεια στην ενσωμάτωση

Η ασφάλεια των δεδομένων είναι υψίστης σημασίας, πολύ περισσότερο όταν τα συστήματα είναι προσβάσιμα εξωτερικά. Με τις σήραγγες Cloudflare, η διασφάλιση της ασφάλειας των δεδομένων συνδυάστηκε με την αποτελεσματική ανάκτηση δεδομένων.

9.2.1 Κατανόηση των Cloudflare Tunnels

Τα Cloudflare Tunnels λειτουργούν ως προστατευτικά στρώματα, διασφαλίζοντας ότι τα δεδομένα που μεταδίδονται μέσω δικτύων κρυπτογραφούνται και θωρακίζονται από πιθανές παραβιάσεις. Δίνουν προτεραιότητα στην ασφάλεια των δεδομένων, ενώ παράλληλα βελτιστοποιούν την ταχύτητα μετάδοσής τους[2].

9.2.2 Εφαρμογή Cloudflare tunnels για την παρακολούθηση των κυψελών

Για το σύστημα παρακολούθησης της κυψέλης μας, ενσωματώθηκαν τα Cloudflare Tunnels για να διασφαλιστεί η ασφαλής πρόσβαση σε κάθε κομμάτι δεδομένων, είτε αυτά αποθηκεύονται στο InfluxDB2 είτε απεικονίζονται στο Grafana. Αυτό σήμαινε ότι οποιαδήποτε εξωτερική προσπάθεια πρόσβασης στα δεδομένα έπρεπε να περάσει από αυτές τις ασφαλείς σήραγγες, μειώνοντας αποτελεσματικά τον κίνδυνο μη εξουσιοδοτημένης πρόσβασης.

9.3 Μεθοδολογίες δοκιμών, αποτελέσματα και ανάλυση επιδόσεων

Μετά την ενσωμάτωση όλων των στοιχείων, ήταν απαραίτητη η αυστηρή δοκιμή για να διασφαλιστεί η λειτουργικότητα και η αποδοτικότητα του συστήματος.

9.3.1 Επιλογή των κατάλληλων μεθοδολογιών δοκιμών

Διερευνήθηκαν διάφορες μεθοδολογίες. Από τη δοκιμή μεμονωμένων συστατικών μονάδων έως τις ολοκληρωμένες δοκιμές ολοκλήρωσης που έλεγχαν τη συνολική απόδοση του συστήματος, ο στόχος ήταν να εντοπιστούν πιθανά αδύναμα σημεία.

9.3.2 Αποτελέσματα και ανάλυση

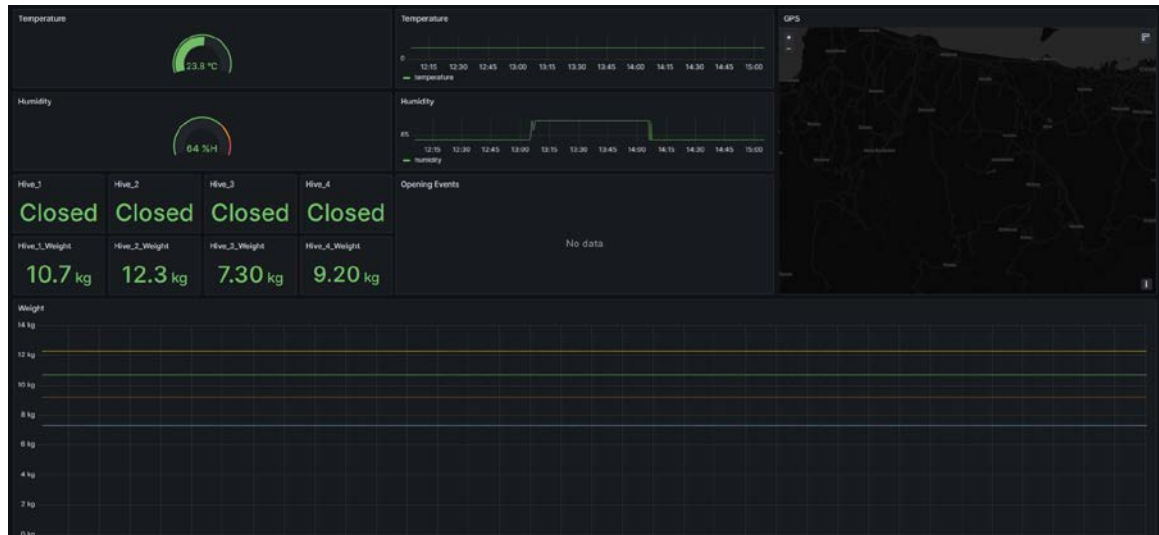
Μετά την ολοκλήρωση των δοκιμών, τα αποτελέσματα συγκεντρώθηκαν και αναλύθηκαν. Η ανάλυση αυτή παρείχε πληροφορίες σχετικά με την ευρωστία του συστήματος και υπέδειξε περιοχές που απαιτούν περαιτέρω βελτίωση ή τροποποίηση. Συγκεκριμένα ζητήματα, όπως η **αποφόρτιση** της μπαταρίας του ESP32 ή οι **ανακριβείς ενδείξεις** του HX711, διορθώθηκαν κατά τη διάρκεια αυτής της φάσης, εξασφαλίζοντας την αξιοπιστία του συστήματος σε πραγματικές συνθήκες.

Συμπεράσματα

Κατά τη διάρκεια αυτής της έρευνας και της υλοποίησης του συστήματος, σημειώθηκαν σημαντικές πρόοδοι στον τομέα της παρακολούθησης των κυψελών. Ένα πρωταρχικό συμπέρασμα που εξάγεται είναι η βελτιωμένη ορατότητα της απόδοσης και της υγείας των κυψελών μας. Με την ενσωμάτωση διαφόρων τεχνολογιών, από τον μικροελεγκτή ESP32 έως πλατφόρμες Docker όπως η Grafana, όχι μόνο εκσυγχρονίσαμε τη συλλογή δεδομένων αλλά και

διευκολύνουμε την ερμηνεία τους. Αυτό μας παρείχε πληροφορίες σε πραγματικό χρόνο για τα μελίσσια μας, επιτρέποντας προληπτικά μέτρα για να διασφαλίσουμε την ευημερία τους.

Η ικανότητα του συστήματος να επισημαίνει τις αποκλίσεις, είτε αυτές προκύπταν από ζητήματα υλικού, όπως η αποφόρτιση της μπαταρίας του ESP32, είτε από προκλήσεις βαθμονόμησης με τον αισθητήρα ΗΧ711, υπογράμμισε περαιτέρω την αξία του. Αντιμετωπίζοντας αυτές τις προκλήσεις κατά μέτωπο, βελτιώσαμε τη ρύθμιση παρακολούθησης, διασφαλίζοντας ότι ήταν τόσο ισχυρή όσο και αξιόπιστη.



Εικόνα 23 Grafana - Dashboard

Πηγή: Thesis

Βελτιώσεις

Ενώ το τρέχον σύστημα έχει αποδειχθεί εξαιρετικά αποτελεσματικό, υπάρχει πάντα περιθώριο για βελτίωση. Οι πιθανές αναβαθμίσεις και βελτιώσεις για το μέλλον περιλαμβάνουν:

- **Ενσωμάτωση GPS:** Με την ενσωμάτωση μιας μονάδας GPS, καθίσταται εφικτή η παρακολούθηση της θέσης κάθε κυψέλης. Αυτό μπορεί να είναι ζωτικής σημασίας για μεγαλύτερα μελισσοκομεία ή για τον εντοπισμό κυψελών σε μεταναστευτικές μελισσοκομικές πρακτικές.
- **Εφαρμογή μικροφώνου:** Η ενσωμάτωση ενός μικροφώνου μπορεί να προσφέρει πολύτιμες πληροφορίες για την ακουστική της κυψέλης. Τα ηχητικά μοτίβα μέσα σε μια κυψέλη μπορούν να αποκαλύψουν πολλά για την υγεία της, τα επίπεδα δραστηριότητάς της, ακόμη και για πιθανές απειλές.
- **Ενσωμάτωση κάμερας στην Grafana:** Μια ζωντανή τροφοδοσία από τις κυψέλες μπορεί να είναι ανεκτίμητη. Με την εφαρμογή μιας κάμερας και την ενσωμάτωση της τροφοδοσίας της στην Grafana, είναι δυνατή η οπτική παρακολούθηση της κυψέλης σε πραγματικό χρόνο, συμπληρώνοντας τα δεδομένα που ήδη συλλέγονται.

- **Προηγμένη ανάλυση και προληπτική συντήρηση:** Καθώς συλλέγονται περισσότερα δεδομένα, υπάρχει η δυνατότητα αξιοποίησης προηγμένων αναλύσεων για την πρόβλεψη πιθανών προβλημάτων πριν αυτά γίνουν κρίσιμα. Οι αλγόριθμοι μηχανικής μάθησης μπορούν να εκπαιδευτούν για να αναγνωρίζουν μοτίβα που μπορεί να υποδεικνύουν φθίνουσα υγεία της κυψέλης ή άλλες προκλήσεις.
- **Δυνατότητες απομακρυσμένου ελέγχου:** Εκτός από την παρακολούθηση, οι μελλοντικές επαναλήψεις του συστήματος θα μπορούσαν να επιτρέψουν απομακρυσμένες παρεμβάσεις. Για παράδειγμα, εάν μια συγκεκριμένη παράμετρος πέσει εκτός ενός καθορισμένου εύρους, θα μπορούσαν να ενεργοποιηθούν αυτοματοποιημένα μέτρα για τη διόρθωση της κατάστασης.

Εν κατακλείδι, τα βήματα που έγιναν στην παρούσα έρευνα άνοιξαν το δρόμο για πιο εξελιγμένες πρακτικές διαχείρισης των κυψελών με βάση τα δεδομένα. Και καθώς η τεχνολογία συνεχίζει να εξελίσσεται, οι δυνατότητες περαιτέρω βελτίωσης και βελτίωσης των συστημάτων παρακολούθησης των κυψελών παραμένουν τεράστιες.

Βιβλιογραφία

[1] "Grafana," Grafana Labs. [Online]. Available: <https://grafana.com/docs/> [Accessed: Nov. 3, 2023].

[2] "Cloudflare," Cloudflare, Inc. [Online]. Available: <https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/> [Accessed: Nov. 3, 2023].

[3] "Espressif Systems," Espressif Systems. [Online]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/> [Accessed: Nov. 3, 2023].

[4] A. Spiess, YouTube. Available: <https://www.youtube.com/@AndreasSpiess> [Accessed: Nov. 3, 2023].

[5] "Proxmox," Proxmox Server Solutions GmbH. [Online]. Available: <https://pve.proxmox.com/pve-docs/pve-admin-guide.pdf> [Accessed: Nov. 3, 2023].

[6] "InfluxDB," InfluxData. [Online]. Available: <https://docs.influxdata.com/influxdb/v2/>
[Accessed: Nov. 3, 2023].

Παράρτημα Κώδικα

ESP32

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <SimpleDHT.h>
#include <HX711.h>

// Wi-Fi Configuration
const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";

// MQTT Broker Configuration
const char* mqtt_server = "broker.hivemq.com"; // HiveMQ public broker
const int mqtt_port = 1883;
WiFiClient espClient;
PubSubClient client(espClient);

// DHT11 Configuration
const int DHT11_PIN = 2;
SimpleDHT11 dht11;

// HX711 Configuration
const int DT_PIN = 4;
const int SCK_PIN = 5;
HX711 scale;

// Magnet Sensor Configuration
const int MAGNET_PIN = 15; // Change to the pin you have the magnet sensor connected to
bool magnet_state = HIGH;
```

```
bool last_magnet_state = HIGH;
unsigned long lastDebounceTime = 0; // the last time the output pin was toggled
unsigned long debounceDelay = 50; // the debounce time; increase if the output flickers

void setup() {
  Serial.begin(115200);
  pinMode(MAGNET_PIN, INPUT_PULLUP); // Initialize magnet sensor as input with internal pull-up

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  // Connect to MQTT Broker
  client.setServer(mqtt_server, mqtt_port);
  while (!client.connected()) {
    if (client.connect("ESP32Client")) {
      Serial.println("Connected to MQTT Broker!");
    } else {
      Serial.print("Failed MQTT connection, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }

  // Initialize HX711
  scale.begin(DT_PIN, SCK_PIN);
  scale.set_scale();
}
```



```
scale.tare();

// Send an initial state for the magnet sensor
last_magnet_state = digitalRead(MAGNET_PIN);
client.publish("home/bee/magnet", last_magnet_state ? "Closed" : "Opened");
}

void loop() {
// Check the magnet sensor for any state change
int reading = digitalRead(MAGNET_PIN);

// If the switch changed, due to noise or pressing:
if (reading != last_magnet_state) {
    lastDebounceTime = millis();
}

if ((millis() - lastDebounceTime) > debounceDelay) {
// Whatever the reading is at, it's been there for longer than the debounce
// delay, so take it as the actual current state:

// If the magnet state has changed:
if (reading != magnet_state) {
    magnet_state = reading;

// Only publish the state if it changes
client.publish("home/bee/magnet", magnet_state ? "Closed" : "Opened");

// Change the sleep mode based on the magnet state
if (magnet_state) {
// If the magnet is closed, go to sleep for 5 hours
esp_sleep_enable_timer_wakeup(5 * 60 * 60 * 1000000ULL);
esp_deep_sleep_start();
```

```
}  
}  
}  
  
// Save the reading. Next time through the loop, it'll be the lastButtonState:  
last_magnet_state = reading;  
  
// Publish temperature and humidity  
byte temperature = 0;  
byte humidity = 0;  
if (dht11.read(DHT11_PIN, &temperature, &humidity, NULL) == 0) {  
  char tempMsg[50];  
  sprintf(tempMsg, "Temperature: %dC, Humidity: %d%%",  
(int)temperature, (int)humidity);  
  client.publish("home/bee/temperature", tempMsg);  
} else {  
  client.publish("home/bee/error", "Failed to read from DHT11 sensor");  
}  
  
// Publish weight  
float weight = scale.get_units(5);  
  
}
```

NodeRed Value Parsing (Temperature)

```
var newMsg = { payload: {} };  
if (msg.topic === "beehive_moulete/dht11/temperature") {  
  newMsg.payload.temperature = parseFloat(msg.payload);  
}  
return newMsg;
```

NodeRed Value Parsing (Humidity)

```
var newMsg = { payload: {} };
```

```
if (msg.topic === "beehive_moulete/dht11/humidity") {  
    newMsg.payload.humidity = parseFloat(msg.payload);  
}  
return newMsg;
```

Grafana Flux Queries: Το Bucket στο influxdb έχει ονομαστεί bee

HX711 Graph

```
from(bucket: "bee")  
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)  
  |> filter(fn: (r) => r["_measurement"] == "scale")
```

Temperature Graph

```
from(bucket: "bee")  
  |> range(start: v.timeRangeStart, stop:v.timeRangeStop)  
  |> filter(fn: (r) => r["_measurement"] == "sensor_data" and r["_field"] == "temperature")  
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)  
  |> yield(name: "mean")
```

Humidity Graph

```
from(bucket: "bee")  
  |> range(start: v.timeRangeStart, stop:v.timeRangeStop)  
  |> filter(fn: (r) => r["_measurement"] == "sensor_data" and r["_field"] == "humidity")  
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)  
  |> yield(name: "mean")
```

Docker Compose Script

```
version: '3'  
  
services:  
  influxdb:
```

image: influxdb:2.0

container_name: influxdb

ports:

- "8086:8086"

volumes:

- influxdb-data:/var/lib/influxdb2

environment:

- DOCKER_INFLUXDB_INIT_MODE=setup

- DOCKER_INFLUXDB_INIT_USERNAME=""

- DOCKER_INFLUXDB_INIT_PASSWORD=""

- DOCKER_INFLUXDB_INIT_ORG=bee

- DOCKER_INFLUXDB_INIT_BUCKET=bee

grafana:

image: grafana/grafana:latest

container_name: grafana

ports:

- "3000:3000"

volumes:

- grafana-data:/var/lib/grafana

depends_on:

- influxdb

nodered:

image: nodered/node-red:latest

container_name: nodered

ports:

- "1880:1880"

volumes:

- nodered-data:/data

environment:

- TZ=Europe/Athens

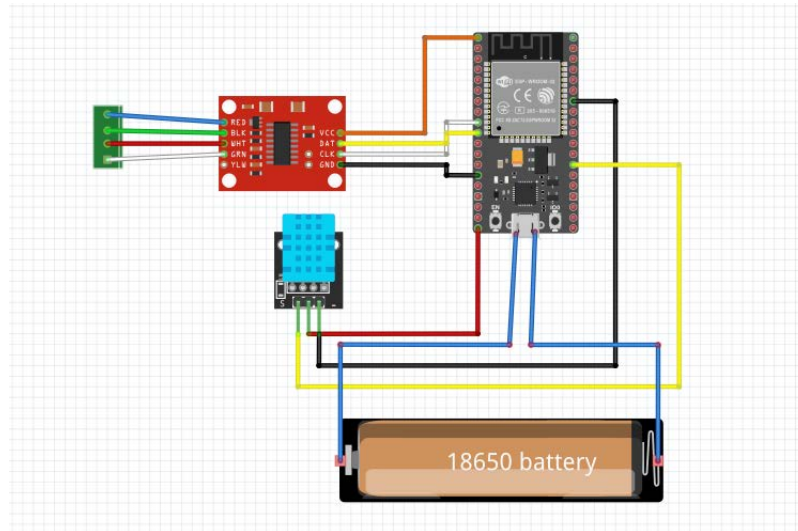
volumes:

influxdb-data:

grafana-data:

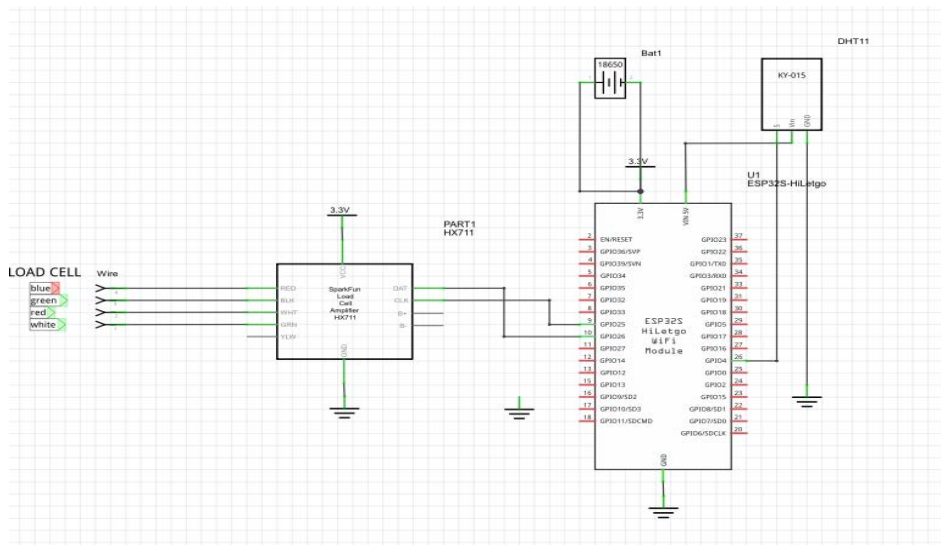
nodered-data:

Κύκλωμα



Εικόνα 24 Thesis Circuit

Πηγή: Thesis



Εικόνα 25 Thesis Schematic

Πηγή: Thesis