



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ανάπτυξη Συστήματος IoT για Δασική Προστασία

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Κωνσταντίνου Λαβού

Επιβλέπων : Επίκουρος καθηγητής, Κωνσταντίνος Μαλιάτσος

Μέλη εξεταστικής επιτροπής: Καθηγητής Δημοσθένης Βουγιούκας, Επίκουρος Καθηγητής Δημήτριος Σκούτας

Σάμος, Ιούνιος 2024

Πρόλογος και ευχαριστίες

Η παρούσα διπλωματική εργασία υλοποιήθηκε στα πλαίσια ολοκλήρωσης του Προγράμματος Προπτυχιακών Σπουδών (ΠΠΣ) του τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων, της πολυτεχνικής σχολής, του Πανεπιστημίου Αιγαίου. Σε αυτό το σημείο, θα ήθελα να ευχαριστήσω συγκεκριμένα, τον επιβλέποντα καθηγητή της διπλωματικής εργασίας, επίκουρο καθηγητή, Κωνσταντίνο Μαλιάτσο, για τη συνεχή υποστήριξη που παρείχε, την άψογη συνεργασία, την επικοινωνία και την εποικοδομητική καθοδήγηση στην περάτωση της συγκεκριμένης εργασίας. Ακόμη, δε θα μπορούσα να παραλείψω, την ευγνωμοσύνη μου, στην οικογένειά μου για τη συνεχή στήριξη και τη συμπαράσταση που μου παρείχαν μέχρι την ολοκλήρωση των σπουδών μου.

© [2024]

του

Κωνσταντίνου Λαβού

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Πίνακας περιεχομένων

1	Εισαγωγή	1
1.1	Συστήματα IoT για περιβαλλοντολογική παρακολούθηση.....	1
1.2	Αντικείμενο της διπλωματικής	2
1.2.1	Προκλήσεις στην έγκαιρη και αξιόπιστη ανίχνευση φωτιάς σε δασικές περιοχές.....	2
1.2.2	Τεχνικές προτεινόμενες λύσεις στη κατεύθυνση της ανίχνευσης; φωτιάς σε δασικές περιοχές	2
1.3	Δομή της διπλωματικής	3
2	Διαδικτύου των Αντικειμένων (Internet of Things - IoT).....	4
2.1	Ορισμός Internet of Things.....	4
2.2	Αρχιτεκτονικές Internet of Things.....	5
2.2.1	3 Επιπέδων (3-Layers).....	5
2.2.2	5 Επιπέδων (5-Layers).....	6
2.2.3	Γενικευμένες Αρχιτεκτονικές	7
2.3	Ασύρματες Τεχνολογίες Επικοινωνιών για IoT εφαρμογές.....	9
2.3.1	Μεγάλης Εμβέλειας (Long-Range).....	10
2.3.2	Μικρής Εμβέλειας (Short-Range).....	11
2.4	Πρωτόκολλα Επιπέδου Εφαρμογής.....	13
2.4.1	HTTP.....	13
2.4.2	MQTT.....	14
2.4.3	CoAP.....	15
3	Συσκευές και Εξαρτήματα υλοποίησης της IoT συσκευής.....	16
3.1	Αισθητήρες (Sensors)	16
3.1.1	DHT11 αισθητήρας υγρασίας θερμοκρασίας.....	16
3.1.2	Αισθητήρας οξυγόνου (O2).....	17
3.1.3	Αισθητήρας διοξειδίου του άνθρακα (CO2)	18
3.1.4	Αισθητήρας διοξειδίου του αζώτο.....	18
3.2	Προγραμματισμός IoT συσκευής μέσω Waspnote board.....	20
3.3	LoRa Gateway.....	29
4	Ανάλυση Λογισμικών και εφαρμογών	31
4.1	Βάση Δεδομένων Εφαρμογών	32
4.2	MQTT App	33
4.3	Web Service Monitoring.....	35
4.4	Σενάρια Εκτέλεσης Εφαρμογής Web Monitoring	39
5	Συμπεράσματα, Προκλήσεις και Μελλοντικές Κατευθύνσεις.....	47

5.1	Συμπεράσματα	47
5.2	Υπάρχουσα Προβλήματα-Προκλήσεις	47
5.3	Μελλοντική Εργασία	48
Βιβλιογραφία.....		49

Λίστα Σχημάτων

Σχήμα 1: Πεδίο του ΔτΠ	5
Σχήμα 2: Αρχιτεκτονική 3 και 5 επιπέδων ΔτΠ	7
Σχήμα 3: Γενικευμένες αρχιτεκτονικές ΔτΠ (Fog, Cloud, Edge)	9
Σχήμα 4: DHT11 αισθητήρας θερμοκρασίας, υγρασίας	16
Σχήμα 5 Αισθητήρας οξυγόνου, Oxygen LF02-A4	17
Σχήμα 6: Αισθητήρας διοξειδίου του άνθρακα - CO ₂	18
Σχήμα 7 Αισθητήρας διοξειδίου του αζώτου NO ₂	19
Σχήμα 8 : IoT συσκευή.....	20
Σχήμα 9 : Βιβλιοθήκες εφαρμογής.....	21
Σχήμα 10 Ρυθμίσεις socket, port και παραμέτρων συσκευής	21
Σχήμα 11 Εγγραφή δρομολογητή στο TNN.....	29
Σχήμα 12 Συμπλήρωση φόρμας του Gateway στο TTN	30
Σχήμα 13 Αρχιτεκτονική του προτεινόμενου συστήματος - Επίπεδα	32
Σχήμα 14 Αρχιτεκτονική εφαρμογής MQTT App	34
Σχήμα 15 Εκτέλεση MQTT App εφαρμογής	35
Σχήμα 16 Αρχιτεκτονική εφαρμογής Environmental Monitoring	37
Σχήμα 17 Σελίδα εγγραφή χρήστη	37
Σχήμα 18 Σελίδα login	38
Σχήμα 19 Σελίδα κεντρικού μενού	38
Σχήμα 20 Διάγραμμα CO ₂	39
Σχήμα 21 Διάγραμμα O ₂	40
Σχήμα 22 Διάγραμμα Υγρασίας	40
Σχήμα 23 Διάγραμμα NO ₂	41
Σχήμα 24 Διαγράμματα real-time CO ₂ και O ₂	42
Σχήμα 25 Διαγράμματα real-time Υγρασίας και NO ₂	43
Σχήμα 26 Menu αναζήτησης	44
Σχήμα 27 Διαγράμματα CO ₂ , O ₂ σε συγκεκριμένο χρονικό διάστημα.....	45
Σχήμα 28 Διαγράμματα Υγρασίας και NO ₂ σε συγκεκριμένο χρονικό διάστημα.....	46

Λίστα Πινάκων

Πίνακας 1 Τεχνικές παράμετροι πρωτοκόλλων HTTP, MWTT, CoAP	15
Πίνακας 2 Τεχνικές προδιαγραφές αισθητήρα οξυγόνου.....	17
Πίνακας 3: Τεχνικές προδιαγραφές αισθητήρα CO2	18
Πίνακας 4: Τεχνικές προδιαγραφές αισθητήρα διοξειδίου του αζώτου	19
Πίνακας 5: Πίνακας ΒΔ 'Users'	33
Πίνακας 6: Πίνακας ΒΔ 'Measurements'	33
Πίνακας 7:Εξαρτήσεις Python εφαρμογής MQTT App.....	33
Πίνακας 8: HTTP URLs	35
Πίνακας 9: Εξαρτήσεις εφαρμογής Environmental Monitoring.....	36

Ακρωνύμια

3GPP	3 rd Generation Partnership Project
AMQP	Advanced Message Queuing Protocol
BLE	Bluetooth Low Energy
CoAP	Constrained Application Protocol
CO ₂	Carbon Dioxide
CSS	Chirp Spread Spectrum
DTLS	Datagram Transport Layer Security
FDMA	Frequency Division Multiple Access
GFSK	Gaussian Frequency Shift Keying
GHz	Gigahertz
GW	Gateway
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronic Engineering
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
ISM	Industrial, Scientific, Medical
ISO	International Organization for Standardization
JS	JavaScript
LoRa	Long-Range
LoRaWAN	LoRA Wide Area Network
LPWA	Low Power Wide Area
LTE	Long-Term Evolution
LTE-M	Long-Term Evolution Machine Type Communication
MAC	Medium Access Control
MHz	Megahertz
MQTT	Message Queuing Telemetry Transport
NB-IoT	Narrowband IoT
NFC	Near Field Communication
NO ₂	Nitrogen Dioxide
QoS	Quality of Service
REST	Representational State Transfer
RFID	Radio Frequency Identification

RSSI	Received Signal Strength Indicator
SNR	Signal-to-Noise Ratio
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TLS	Transport Layer Security
TTN	The Things Network
UDP	User Datagram Protocol
URL	Uniform Resource Locator
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity
XMPP	Extensible Messaging and Presence Protocol

Περίληψη

Αποτελεί γεγονός ότι, τα έξυπνα συστήματα παρακολούθησης Internet of Things (IoT), έχουν ενσωματωθεί σε πάρα πολλά περιβάλλοντα όπως τα οικιακά, τα βιομηχανικά, τα αστικά, τα γεωργικά, ακόμη και τα δασικά. Συνεπώς, η αξιοποίηση τους αποτελεί μονόδρομος αφού έτσι, μπορεί να υπάρξει μια σύγχρονη υποδομή παρακολούθησης, επεξεργασίας δεδομένων και ενημέρωσης των κατάλληλων ανθρώπων για τα παραγόμενα αποτελέσματα. Ειδικά, στις δασικές εκτάσεις και περιοχές, όπου φαινόμενα μόλυνσης του οικοσυστήματος, πυρκαγιές, ή πλημμύρες μπορεί να συμβούν και να προκαλέσουν σημαντική καταστροφή στη φύση αλλά και στη παραγωγική αλυσίδα. Για να αντιμετωπιστεί ειδικότερα το φαινόμενο των πυρκαγιών σε δασικές περιοχές, στα πλαίσια της παρούσας διπλωματικής εργασίας, θα επικεντρωθούμε στο σχεδιασμό και την υλοποίηση ενός συστήματος Διαδικτύου των Πραγμάτων (IoT) για την παρακολούθηση του περιβάλλοντος σε πραγματικό χρόνο στο νησί της Σάμου, Ελλάδα. Ο απώτερος στόχος είναι η υλοποίηση, ανάπτυξη και λειτουργία ενός πλήρους βιώσιμου IoT συστήματος για την παρακολούθηση μιας περιοχής του ελληνικού νησιού με σκοπό την ανίχνευση πυρκαγιών σε δασικές εκτάσεις. Τα συστατικά που χρησιμοποιήθηκαν για αυτό το σκοπό περιλαμβάνουν την πλακέτα ανάπτυξης, τους αισθητήρες περιβαλλοντολογικών δεδομένων, τις ασύρματες τεχνολογίες επικοινωνίας, τις υπηρεσίες αποθήκευσης IoT, τις οντότητες λογισμικού σε Python και μια υπηρεσία ιστού, ενοποιημένες σε μια ενιαία υπηρεσία για τον τελικό χρήστη.

Στο πλαίσιο αυτό, το σύστημα μας χρησιμοποιεί μια πλακέτα Libelium εξοπλισμένη με αισθητήρες για τη μέτρηση της υγρασίας, του διοξειδίου του αζώτου (NO₂), του οξυγόνου (O₂) και του διοξειδίου του άνθρακα (CO₂) σε σταθερή τοποθεσία. Τα δεδομένα που συλλέγονται από αυτούς τους αισθητήρες μεταδίδονται μέσω τεχνολογίας LoRa σε μια πύλη (gateway) Raspberry Pi LoRa (GW). Αυτή η πύλη, συνδεδεμένη στο Διαδίκτυο, προωθεί τα δεδομένα στην υπηρεσία IoT του The Things Network (TTN) για αποθήκευση και περαιτέρω επεξεργασία.

Σχετικά με τις οντότητες λογισμικού, έχουν αναπτυχθεί δύο εφαρμογές σε Python για τη διαχείριση και ανάλυση των δεδομένων των αισθητήρων. Η πρώτη εφαρμογή χρησιμοποιεί το πρωτόκολλο MQTT για την ανάκτηση δεδομένων από το TTN και την αποθήκευσή τους σε μια βάση δεδομένων PostgreSQL, διασφαλίζοντας αξιόπιστη και κλιμακούμενη διαχείριση δεδομένων. Η δεύτερη εφαρμογή είναι μια υπηρεσία ιστού που παρέχει δυνατότητες οπτικοποίησης δεδομένων και παρακολούθησης σε πραγματικό χρόνο. Αυτή η υπηρεσία διαβάζει δεδομένα από τη βάση δεδομένων, PostgreSQL, και προσφέρει στους χρήστες διαδραστικά γραφήματα και ζωντανή ροή δεδομένων μέσω μιας διαδικτυακής διεπαφής.

Τελικά, το προτεινόμενο IoT σύστημα υλοποιεί και υποστηρίζει ειδοποιήσεις βάσει ορίων τιμών, ενημερώνοντας τους χρήστες όταν οι περιβαλλοντικές παράμετροι υπερβαίνουν τα προκαθορισμένα όρια. Γίνεται αντιληπτό ότι, με την ενσωμάτωση αυτών των τεχνολογιών, η προτεινόμενη λύση διευκολύνει την ολοκληρωμένη και σε πραγματικό χρόνο παρακολούθηση του περιβάλλοντος, παρέχοντας πολύτιμες πληροφορίες και υποστηρίζοντας τις διαδικασίες λήψης αποφάσεων για τη διαχείριση του περιβάλλοντος στο νησί της Σάμου. Επιπλέον, η προτεινόμενη λύση θα μπορούσε να βοηθήσει τους πυροσβέστες στην κατεύθυνση της έγκαιρης ανίχνευσης πυρκαγιών, ειδοποιώντας τους σε καταστάσεις έκτακτης ανάγκης.

Λέξεις Κλειδιά: *Διαδίκτυο των Αντικειμένων (ΔτΑ), Ανίχνευση Φωτιάς, Υγρασία, Θερμοκρασία, Πλακέτες ανάπτυξης, Διαδικτυακές Υπηρεσίες, Διαδικτυακή υπηρεσία*

Abstract

Smart monitoring systems using the Internet of Things (IoT) are increasingly being integrated into various settings, including homes, industries, cities, agriculture, and forests. Utilizing these systems is essential for creating modern infrastructure that allows for efficient monitoring, data processing, and timely notification of relevant parties. This is especially important in forested areas where pollution, fires, or floods can cause significant damage to the environment and the local economy.

This thesis focuses on addressing the problem of forest fires monitoring by designing and implementing an IoT system for real-time environmental monitoring on Samos Island, Greece. The goal is to create a sustainable IoT solution for monitoring specific areas of the island to detect fires early. The system includes a development board with sensors, wireless communication technologies, IoT data storage services, Python software applications, and a web service, all integrated into a unified platform for users.

Our system uses a Libelium board with sensors that measure humidity, nitrogen dioxide (NO₂), oxygen (O₂), and carbon dioxide (CO₂) levels at a fixed location. The collected data is transmitted via LoRa technology to a Raspberry Pi LoRa gateway. This gateway, connected to the Internet, sends the data to The Things Network (TTN) for storage and further analysis.

We developed two Python applications to manage and analyze the sensor data. The first application uses the MQTT protocol to retrieve data from TTN and store it in a PostgreSQL database, ensuring reliable and scalable data management. The second application is a web service that provides real-time data visualization and monitoring. This service reads data from the PostgreSQL database and presents it through interactive plots and live data streams on a web interface.

The proposed IoT system also supports threshold-based alerts, notifying users when environmental parameters exceed predefined limits. By integrating these technologies, our solution offers comprehensive and real-time environmental monitoring, providing valuable insights and supporting decision-making processes for managing the environment on Samos Island. Additionally, this system could help firefighters by providing early fire detection alerts in emergency situations.

Keywords: *Internet of Things (IoT), Fire Detection, LoRa, Arduino, Raspberry Pi, Python, Flask, Web Service, Humidity*

1

Εισαγωγή

1.1 Συστήματα IoT για περιβαλλοντολογική παρακολούθηση

Η παρακολούθηση του περιβάλλοντος μέσω των Έξυπνων Συστημάτων Διαδικτύου των Πραγμάτων (Internet of Things-IoT) έχει εξελιχθεί σημαντικά τα τελευταία χρόνια, προσφέροντας νέες δυνατότητες για τη συλλογή και ανάλυση δεδομένων που αφορούν το περιβάλλον. Η χρήση IoT λύσεων σε περιβαλλοντικές παρακολουθήσεις επιτρέπει την πραγματικού χρόνου παρακολούθηση και αξιολόγηση των περιβαλλοντικών παραμέτρων, όπως η υγρασία, η θερμοκρασία, η σύγκριση των διοξειδίων του άζωτου, οξυγόνου και άνθρακα σε περιβάλλοντα όπως οι δασικές εκτάσεις.

Τεχνολογικά, οι πρώτες IoT λύσεις χρησιμοποιούσαν συχνά ενσύρματους αισθητήρες και συστήματα επικοινωνίας που ήταν περιορισμένα στην εμβέλεια και τη διαθεσιμότητα δεδομένων. Σήμερα, οι τεχνολογίες LoRa (Long Range) και MQTT (Message Queuing Telemetry Transport) επιτρέπουν την ασύρματη μετάδοση δεδομένων σε μεγάλες αποστάσεις και την άμεση σύνδεση σε πλατφόρμες cloud για αποθήκευση και ανάλυση. Αυτές οι τεχνολογικές πρόοδοι επιτρέπουν την πραγματικού χρόνου παρακολούθηση και αντίδραση σε περιβαλλοντικούς κινδύνους όπως οι πυρκαγιές.

Ιδιαίτερα στη σημερινή εποχή, η αξιοποίηση του IoT για την παρακολούθηση του περιβάλλοντος είναι κρίσιμη λόγω της αυξημένης συχνότητας και έντασης των φυσικών καταστροφών όπως οι πυρκαγιές, οι πλημμύρες και οι κλιματικές αλλαγές. Η πρόληψη και η άμεση αντίδραση σε αυτούς τους κινδύνους μπορεί να σώσει ζωές και περιουσίες, ενώ η ακριβής γνώση των συνθηκών στο περιβάλλον ενισχύει τις προσπάθειες για τη διατήρηση της βιοποικιλότητας και την αειφόρο διαχείριση των φυσικών πόρων.

Η ανίχνευση φωτιάς, ιδιαίτερα, είναι ένα κρίσιμο σημείο, καθώς οι πυρκαγιές μπορούν να εξαπλωθούν με γρήγορο ρυθμό και να προκαλέσουν μαζική καταστροφή στο περιβάλλον και τις κοινότητες διαβίωσης. Η έγκαιρη ανίχνευση και ειδοποίηση των αρμόδιων αρχών μπορεί να βοηθήσει στην ταχεία και αποτελεσματική αντιμετώπιση του φαινομένου, προστατεύοντας έτσι το φυσικό και κοινωνικό περιβάλλον.

1.2 Αντικείμενο της διπλωματικής

1.2.1 Προκλήσεις στην έγκαιρη και αξιόπιστη ανίχνευση φωτιάς σε δασικές περιοχές

Στο υποκεφάλαιο αυτό, αναλύουμε τις βασικές προκλήσεις που προκύπτουν στην προσπάθεια να προσφέρουμε έγκαιρη και αξιόπιστη ανίχνευση φωτιάς σε δασικές περιοχές. Αναφέρουμε τις τεχνολογικές και λειτουργικές δυσκολίες που πρέπει να αντιμετωπίσουμε:

Ένας από τους βασικούς περιορισμούς είναι η ανάγκη για αισθητήρες με υψηλή ακρίβεια και ευαισθησία για την ανίχνευση φωτιάς, οι οποίοι πρέπει να λειτουργούν αξιόπιστα σε ακραίες και απρόβλεπτες κλιματικές συνθήκες. Αναφέρουμε επίσης τη σημασία της αποτελεσματικής επικοινωνίας και μετάδοσης δεδομένων από τους αισθητήρες στα κεντρικά συστήματα παρακολούθησης, καθώς και την ανάγκη για υψηλή συχνότητα δειγματοληψίας για έγκαιρη αντίδραση.

1.2.2 Τεχνικές προτεινόμενες λύσεις στη κατεύθυνση της ανίχνευσης; φωτιάς σε δασικές περιοχές

Στο υποκεφάλαιο αυτό, προτείνουμε τεχνικές λύσεις που μπορούν να βελτιώσουν την ανίχνευση φωτιάς σε δασικές περιοχές. Συγκεκριμένα, αναφέρουμε ότι οι τεχνικές προτεινόμενες λύσεις περιλαμβάνουν τη χρήση προηγμένων αλγορίθμων ανίχνευσης βασισμένων σε εικόνες και δεδομένα αισθητήρων, τα οποία επιτρέπουν την ανίχνευση φωτιάς σε πρώιμα στάδια. Επιπρόσθετα, επισημαίνουμε την ανάγκη ανάπτυξης ισχυρών μοντέλων πρόβλεψης κινδύνου και τον προσδιορισμό πιθανών θέσεων φωτιάς, καθώς και την εφαρμογή αυτοματοποιημένων συστημάτων ειδοποίησης και αντίδρασης βασισμένων στα δεδομένα από τους αισθητήρες. Τέλος, εξηγούμε τη χρήση ασύρματων τεχνολογιών για την αξιόπιστη μετάδοση δεδομένων από απομακρυσμένες περιοχές, βελτιώνοντας έτσι την ανίχνευση και την αντίδραση σε φωτιές στις δασικές περιοχές.

Αυτές οι λεπτομέρειες παρέχουν ένα σαφές πλαίσιο για την προσέγγιση και τη μεθοδολογία που θα ακολουθηθούν στην περαιτέρω ανάπτυξη και υλοποίηση της διπλωματικής εργασίας.

1.3 Δομή της διπλωματικής

Το 2ο κεφάλαιο, περιγράφει το απαραίτητο θεωρητικό υπόβαθρο για το οικοσύστημα του IoT, που περιλαμβάνει τον ορισμό του, τις τεχνολογίες επικοινωνιών, τις αρχιτεκτονικές του και τα πρωτόκολλα εφαρμογής. Στο κεφάλαιο 3, γίνεται μια περιγραφή των αισθητήρων και των στοιχείων που συλλέγουν, της πλακέτας υλικού, του κώδικα που εκτελεί η πλακέτα ώστε να συλλέξει και να στείλει τα δεδομένα στο επόμενο κόμβο και τέλος, του Long-Range (LoRa) κόμβου (Gateway-GW). Το 4ο κεφάλαιο, περιγράφει το σχεδιασμό, την υλοποίηση και τα αποτελέσματα των λογισμικών εφαρμογών που αναπτύχθηκαν με γνώμονα τη παροχή υπηρεσιών προς τον χρήστη. Τελικά, στο 5ο κεφάλαιο, γίνεται μια περίληψη του χώρου, αποτυπώνονται τα συμπεράσματα της τεχνικής λύσης, και παρουσιάζονται μελλοντικές προοπτικές του χώρου.

2

Διαδικτύου των Αντικειμένων (Internet of Things - IoT)

2.1 Ορισμός Internet of Things

Το 1999, ο Kevin Ashton της εταιρείας Procter & Gamble πρωτοανέφερε τον όρο Διαδίκτυο των Πραγμάτων (Internet of Things IoT). Το IoT, με πολύ απλά λόγια, αποτελεί ένα δίκτυο αντικειμένων τα οποία είναι εφοδιασμένα με αισθητήρες, λογισμικό και άλλες ανάλογες τεχνολογίες, με βασικό σκοπό τη σύνδεσή τους, και την ανταλλαγή δεδομένων με άλλες συσκευές και συστήματα μέσω του διαδικτύου [1]. Αυτά τα αντικείμενα, γνωστά και έως έξυπνα αντικείμενα (smart things) εκτείνονται από απλές οικιακές συσκευές έως πολύπλοκες βιομηχανικές συσκευές.

Πρόκειται ουσιαστικά για ένα οικοσύστημα όπου άνθρωποι, αντικείμενα/συσκευές και το διαδίκτυο αλληλοεπιδρούν και συνεργάζονται μεταξύ τους, δημιουργώντας το IoT, όπως απεικονίζεται στο Σχήμα 1.



Σχήμα 1: Πεδίο του ΔτΠ

Ωστόσο από το 1970 έως και σήμερα έχουν δοθεί πολλοί ορισμοί του IoT. Σε αυτή τη διατριβή επιλέξαμε να ακολουθήσουμε τον ορισμό του IoT που προστέθηκε στο λεξικό oxford τον Αύγουστο του 2013: «Μία προτεινόμενη ανάπτυξη του διαδικτύου, στην οποία κάθε αντικείμενο έχει συνδεσιμότητα στο διαδίκτυο επιτρέποντάς του να στείλει και να λάβει δεδομένα» [2].

2.2 Αρχιτεκτονικές Internet of Things

Σήμερα, έχουν αναδειχθεί αρκετές αρχιτεκτονικές του IoT, ανάλογα με τις δυνατότητες δικτύωσης, υπηρεσιών, εφαρμογών, αλλά δεν υπάρχει μια ενιαία καθορισμένη αρχιτεκτονική για όλα τα πιθανά σενάρια. Έχουν προταθεί οι αρχιτεκτονικές των 3 επιπέδων όπου είναι και γενικά αποδεκτή και των 5 επιπέδων [3].

2.2.1 3 Επιπέδων (3-Layers)

Στην αρχιτεκτονική 3 επιπέδων, τα 3 επίπεδα είναι, της εφαρμογής, του δικτύου και της αίσθησης.

- **Perception-Sensing Layer**: Η κύρια εργασία σε αυτό το επίπεδο είναι η αντίληψη των ιδιοτήτων των πραγμάτων που είναι μέρος του IoT συστήματος. Δηλαδή, η αίσθηση των οντοτήτων που το IoT σύστημα προσπαθεί να καταλάβει. Αυτή η διεργασία βασίζεται σε μια σειρά από τεχνολογίες αίσθησης όπως το RFID, NFC, GPS και άλλες [3]. Επίσης, η πληροφορία που έχει μια φυσική διάσταση, πρέπει να μετατραπεί σε ψηφιακή για να είναι έτοιμη για επεξεργασία και μετάδοση. Όμως, μερικές οντότητες μπορεί να μη μπορούν άμεσα να γίνουν αντιληπτές από τις πλατφόρμες IoT, και συνεπώς, διάφορα μικρο-υλικά (microchips) χρειάζεται να ενσωματωθούν σε αυτές τις οντότητες για να προσδώσουν

δυνατότητες sensing. Παράδειγμα, τέτοιων μικρο-υλικών είναι οι νανο-τεχνολογίες που επιφορτίζουν μικρά αντικείμενα με επεξεργαστική ισχύ.

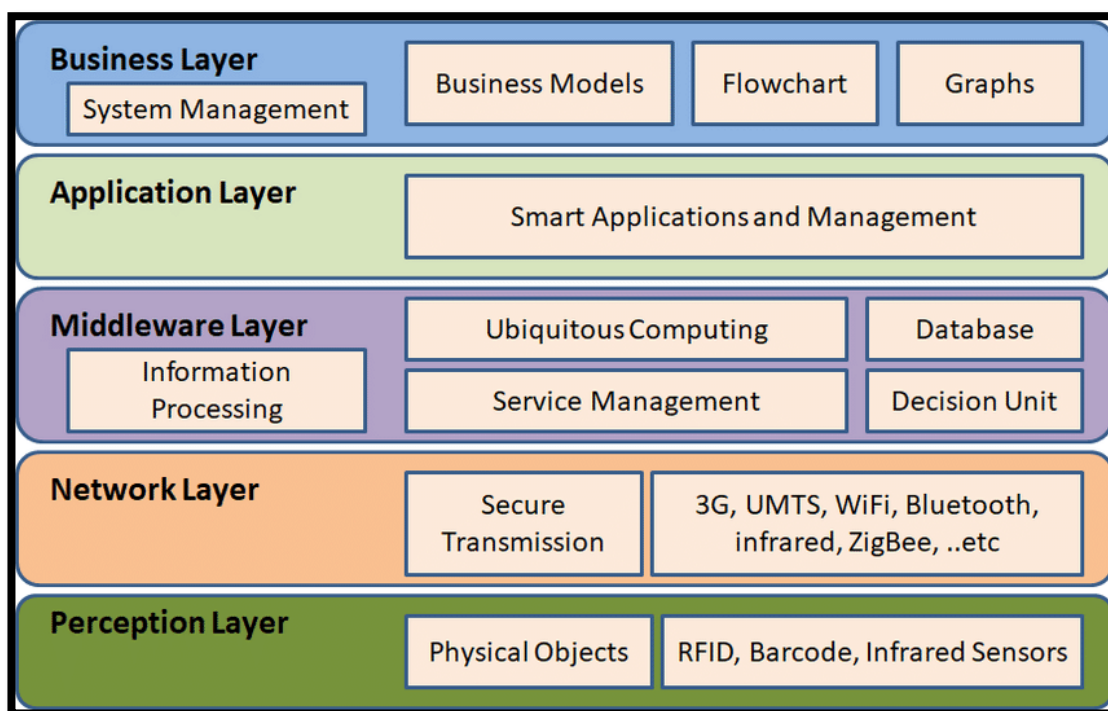
- **Network Layer:** Το επίπεδο δικτύου είναι επιφορτισμένο για την επεξεργασία των δεδομένων που λαμβάνει από το επίπεδο Sensing. Ακόμη, έχει το ρόλο της αποστολής των δεδομένων στο επίπεδο εφαρμογής αξιοποιώντας μια πληθώρα τεχνολογιών επικοινωνίας, είτε ενσύρματες (LAN, Ethernet), είτε ασύρματες (Wi-Fi, 3/4/5G, Bluetooth). Συνήθως, η δυνατότητα δικτύωσης πρέπει να είναι τέτοια ώστε να εξυπηρετεί μεταφορές μεγάλων ποσοτήτων δεδομένων, καθιστώντας έτσι ωφέλιμο το ενδιάμεσο επίπεδο (middleware), όπως το cloud computing [4].
- **Application Layer:** Το τελευταίο επίπεδο, είναι της εφαρμογής. Δέχεται τα δεδομένα από το επίπεδο δικτύου και δρα ως το εμπρόσθιο μέρος (front end interface) της συνολικής IoT αρχιτεκτονικής προσφέροντας τη τελική υπηρεσία. Επιπλέον, αυτό το επίπεδο παρέχει τα αναγκαία εκείνα εργαλεία για τους προγραμματιστές να αναπτύξουν τη δική τους λογική για την IoT εφαρμογή τους ή το σύστημα τους. Τα 3 επίπεδα της αρχιτεκτονικής 3 layers, μπορούν να απεικονισθούν στο Σχήμα 2 [1].

2.2.2 Αρχιτεκτονική 5 Επιπέδων (5-Layers)

Με τη πάροδο του χρόνου, και με τις τεχνολογικές αναβαθμίσεις, δύο επιπλέον επίπεδα προστέθηκαν, το επίπεδο επεξεργασίας/μεσισμικού (processing/middleware layer) και το επίπεδο επιχειρηματικού σεναρίου (business layer). Τα 5 επίπεδα της αρχιτεκτονικής 5 layers, μπορούν να απεικονισθούν στο Σχήμα 2 [5] [6].

- **Processing/Middleware Layer:** Σε αυτό το επίπεδο το οποίο έχει δύο ονόματα, επεξεργασίας και μεσισμικού, λαμβάνουν μέρος οι λειτουργίες αποθήκευσης, ανάλυσης, και επεξεργασίας μεγάλης ποσότητας δεδομένων που έρχονται από το δίκτυο. Σχετικές τεχνολογίες που μπορούν να αξιοποιηθούν εδώ, είναι η υπολογιστική νέφους (cloud computing), οι βάσεις δεδομένων (databases) και μονάδες επεξεργασίας μεγάλου όγκου δεδομένων (big data) [1].
- **Business Layer:** Το τελευταίο επίπεδο στην αρχιτεκτονική 5 επιπέδων, έχει το ρόλο του ελέγχου ολόκληρου του IoT συστήματος, συμπεριλαμβανομένων των εφαρμογών, των επιχειρηματικών μοντέλων και της ασφάλειας. Η επιτυχία του IoT συστήματος δε καθορίζεται μόνο από τις χρησιμοποιούμενες τεχνολογίες, αλλά και στο πως αυτές οι τεχνολογίες παραδίδονται και προσφέρονται στους τελικούς χρήστες, γεγονός που καθιστά

αυτό το επίπεδο αναγκαίο. Ως αποτέλεσμα, μπορεί να ληφθεί υπόψη κάθε διάγραμμα ροής, γράφος, οπτικοποίηση που καθορίζει πως το IoT σύστημα μπορεί να βελτιωθεί [7].



Σχήμα 2: Αρχιτεκτονική 3 και 5 επιπέδων ΔτΠ

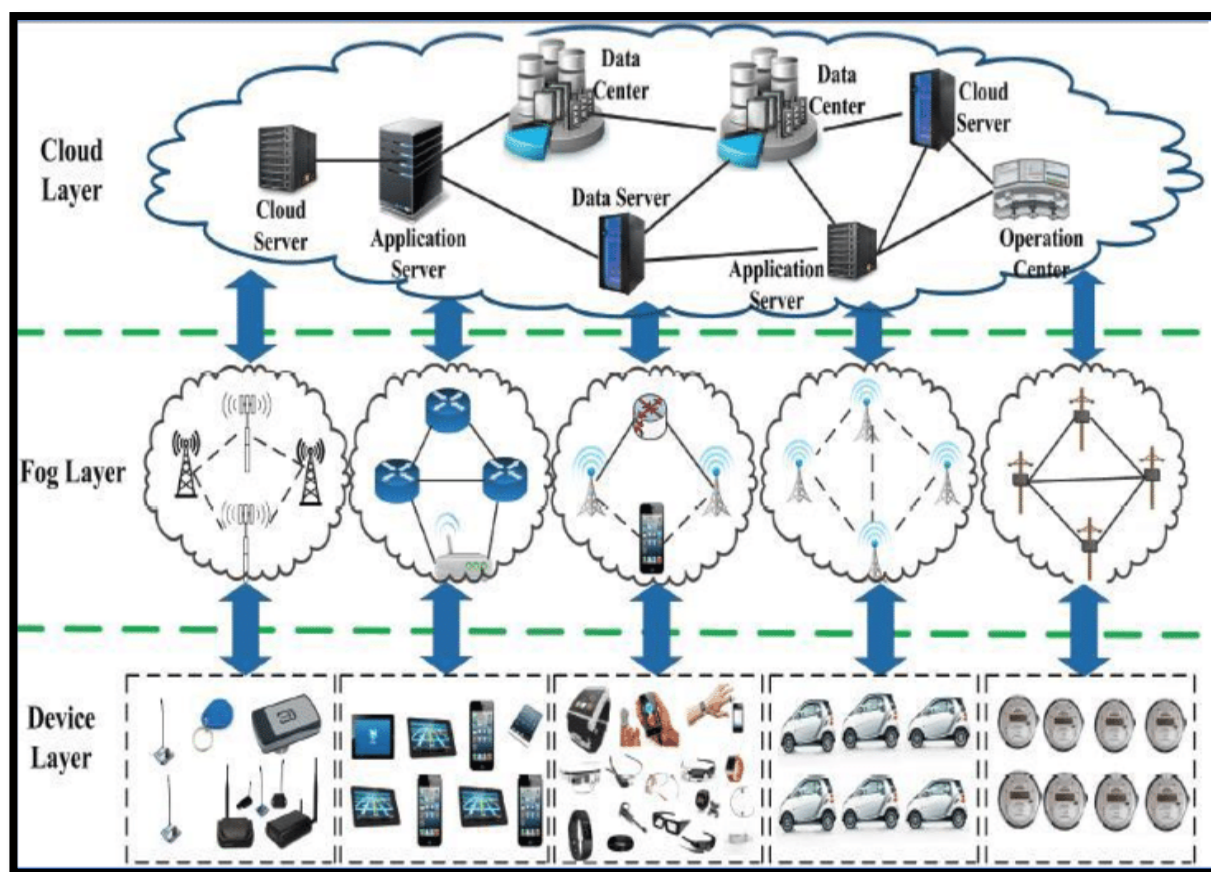
2.2.3 Γενικευμένες Αρχιτεκτονικές

Ενώ, οι αρχιτεκτονικές τριών και πέντε στρωμάτων εκφράζουν τη πλήρη στοίβα του IoT, υπάρχουν επίσης οι γενικευμένες αρχιτεκτονικές που επικεντρώνονται κυρίως στην επεξεργασία πληροφορίας, στο πως πραγματοποιείται, στο πως μπορεί να αποκεντρωθεί, να γίνει πιο γρήγορα αγνοώντας μερικώς τα υπόλοιπα επίπεδα. Οι ερευνητές στο χώρο του IoT, έχουν οδηγηθεί σε 3 γνωστές αρχιτεκτονικές, την αρχιτεκτονική με βάση το cloud, το fog, και την edge αρχιτεκτονική όπου η κάθε μια έχει τα δικά της πλεονεκτήματα [1], [5], [6]. Στο Σχήμα 3, απεικονίζονται οι διάφορες αρχιτεκτονικές.

- **Cloud Αρχιτεκτονική:** Σε αυτό το σενάριο, γίνεται χρήση κατανεμημένων κόμβων επεξεργασίας μακριά από τις IoT συσκευές. Οι κόμβοι που βρίσκονται μέσα στο νέφος, φέρουν μεγάλη ισχύ επεξεργασίας, αποθήκευσης καθώς διαθέτουν αρκετούς υπολογιστικούς πόρους για να διεκπεραιώσουν οποιαδήποτε υπηρεσία IoT. Αν χρειασθεί να κλιμακωθεί η ισχύς, τότε αυτή η αρχιτεκτονική μπορεί να προσφέρει περισσότερα ικανοποιώντας έτσι τις μεταβαλλόμενες απαιτήσεις που υπάρχουν σε ένα IoT περιβάλλον [3], [7], [8]. Το μειονέκτημα όμως, εμφανίζεται στην αύξηση της καθυστέρησης λόγω των πολλών δικτύων που παρεμβάλλονται μεταξύ IoT και cloud points. Ακόμη, η μεταφορά

τεράστιου όγκου δεδομένων μπορεί να εγείρει μεγάλη κατανάλωση του διαθέσιμου εύρους ζώνης (bandwidth) και εν γένη κόστους.

- **Fog Architecture:** Η αρχιτεκτονική ομίχλης, επεκτείνει το μοντέλο νέφους στα άκρα του δικτύου, πιο κοντά στις IoT συσκευές, εισάγοντας κόμβους fog, οι οποίοι παρέχουν πόρους επεξεργασίας, αποθήκευσης και δικτύωσης. Τα εμφανή πλεονεκτήματα είναι η μείωση της καθυστέρησης αφού η επεξεργασία γίνεται πιο κοντά στο επίπεδο του IoT, μειώνοντας τα δεδομένα που απαιτείται να μεταφερθούν στο νέφος. Επίσης, η εγγύηση για την ασφάλεια μπορεί να μεγιστοποιηθεί, αφού οι διαδικασίες επεξεργασίας των δεδομένων γίνονται τοπικά χωρίς να ταξιδέψουν πιο μακριά. Από την άλλη πλευρά, η διαχείριση των fog κόμβων είναι σαφώς πιο πολύπλοκη διότι η κατανομή των κόμβων στο δίκτυο απαιτεί εργαλεία τύπου orchestration [7], [8], [9].
- **Edge Architecture:** Η τελευταία αρχιτεκτονική, edge, ουσιαστικά, έρχεται πιο κοντά από την αρχιτεκτονική fog, επιφορτίζοντας τις ίδιες τις συσκευές IoT με ρόλους επεξεργασίας δεδομένων. Έτσι, η απόσταση μεταξύ παραγωγής και επεξεργασίας δεδομένων μπορεί πλέον υπό συνθήκες να μηδενισθεί. Η καθυστέρηση και η ασφάλεια μπορούν σε αυτό το σενάριο να εγγυηθούν και να μεγιστοποιηθούν με το κόστος όμως της μερικής ανεπάρκειας δυνατοτήτων των συσκευών. Επιπρόσθετα, η διαχείριση των edge συσκευών είναι σύνθετη αλλά και η δυναμική αύξηση τους ανάλογα με τις ανάγκες των εφαρμογών IoT [10].



Σχήμα 3: Γενικευμένες αρχιτεκτονικές ΔτΠ (Fog, Cloud, Edge)

2.3 Ασύρματες Τεχνολογίες Επικοινωνιών για IoT εφαρμογές

Τα ασύρματα δίκτυα μικρής και μεγάλης εμβέλειας παίζουν καθοριστικό ρόλο στην σχεδίαση και υλοποίηση των IoT εφαρμογών. Συγκεκριμένα, τα ασύρματα δίκτυα μεγάλης εμβέλειας, όπως το Long Range Wide Area Network (LoRaWAN) και το Narrowband Internet of Things (NB-IoT), επιτρέπουν τη διασύνδεση συσκευών σε μεγάλες αποστάσεις, εξασφαλίζοντας χαμηλή κατανάλωση ενέργειας και αξιοπιστία στη μεταφορά δεδομένων [2], [4], [10]. Αυτά τα δίκτυα είναι ιδανικά για εφαρμογές που αφορούν έξυπνες πόλεις και αγροτική παρακολούθηση, όπου οι συσκευές μπορεί να βρίσκονται διάσπαρτες σε μεγάλες περιοχές. Αντίθετα, τα ασύρματα δίκτυα μικρής εμβέλειας, όπως το Bluetooth και το Zigbee, είναι κατάλληλα για εφαρμογές που απαιτούν γρήγορη και αποδοτική επικοινωνία σε μικρές αποστάσεις, όπως είναι για παράδειγμα τα έξυπνα σπίτια και ο βιομηχανικός αυτοματισμός. Η επιλογή του κατάλληλου δικτύου εξαρτάται από τις συγκεκριμένες απαιτήσεις της εκάστοτε IoT εφαρμογής, όπως η εμβέλεια, η κατανάλωση ενέργειας και η πυκνότητα των συσκευών.

2.3.1 Μεγάλης Εμβέλειας (Long-Range)

- **Κυψελωτά δίκτυα:** Τα κυψελωτά δίκτυα παίζουν σημαντικό ρόλο στην υποστήριξη των εφαρμογών του Διαδικτύου των Πραγμάτων (IoT), προσφέροντας αξιόπιστη και ευέλικτη συνδεσιμότητα σε διάφορα περιβάλλοντα. Σε αυτά τα δίκτυα, οι περιοχές κάλυψης χωρίζονται σε «κυψέλες» (cells) και κάθε κυψέλη εξυπηρετείται από τον δικό της σταθμό βάσης. Επίσης, οι κυψέλες μπορούν να διακριθούν σε μεγακυψέλες (megacells), μακροκυψέλες (macrocells), μικροκυψέλες (microcells) και πικοκυψέλες (picocells), ανάλογα με την ακτίνα κάλυψής τους. Στην εξέλιξη των κυψελωτών δικτύων, οι τεχνολογίες όπως το NB-IoT και το LTE-M αξιοποιούν τις υποδομές των δικτύων κινητής τηλεφωνίας για την παροχή αποδοτικών λύσεων σε IoT εφαρμογές. Το Narrowband Internet of Things (NB-IoT) είναι μια τεχνολογία που συμπεριλήφθηκε στο 3GPP Release 13, σχεδιασμένη για την υποστήριξη εφαρμογών IoT με χαμηλές απαιτήσεις σε ταχύτητα μετάδοσης δεδομένων και χαμηλή κατανάλωση ενέργειας. Το NB-IoT χρησιμοποιεί εύρος ζώνης 180 kHz, το οποίο επιτρέπει τη μείωση της πολυπλοκότητας των συσκευών, με αντίστοιχη μείωση στη μέγιστη ταχύτητα μετάδοσης δεδομένων (περίπου 50 kb/s για την ανερχόμενη ζεύξη και 30 kb/s για την κατερχόμενη ζεύξη). Αντίθετα, το LTE-M (Long Term Evolution for Machines) χρησιμοποιεί ένα υποσύνολο του προτύπου LTE, όπως για παράδειγμα η αρχιτεκτονική δικτύου και οι τεχνικές πολυπλεξίας παρέχοντας υψηλότερο ρυθμό δεδομένων και δυνατότητα μετάδοσης φωνής σε σύγκριση με το NB-IoT. Ωστόσο, για να επιτύχει αυτές τις δυνατότητες, το LTE-M απαιτεί και μεγαλύτερο εύρος ζώνης, το οποίο είναι περίπου 1.4 MHz [1], [11].
- **LoRa:** Τα ασύρματα δίκτυα LoRa είναι ειδικά σχεδιασμένα για να υποστηρίζουν συσκευές χαμηλής κατανάλωσης ισχύος και ασύρματες ζεύξεις μεγάλης εμβέλειας. Οι συσκευές που λειτουργούν σε δίκτυα LoRa μπορούν να είναι ενεργειακά αυτόνομες, με τη μπαταρία τους να διαρκεί έως και δέκα χρόνια. Το LoRa λειτουργεί σε ραδιοσυχνότητες κάτω του 1 GHz, χωρίς να απαιτείται άδεια για τη χρήση των συχνοτήτων. Ο ρυθμός μετάδοσης δεδομένων του LoRa κυμαίνεται από 0.3 kbps έως 50 kbps, ενώ το εύρος ζώνης των ραδιοδιαύλων μπορεί να είναι 125 kHz, 250 kHz ή 500 kHz, ανάλογα με την περιοχή ή την κατανομή συχνοτήτων. Η εμβέλεια του LoRa κυμαίνεται στα 2 με 5 km σε αστικά περιβάλλοντα και έως 15 km σε προαστιακά, ανάλογα το περιβάλλον. Τεχνικά, το LoRa χρησιμοποιεί διαμόρφωση διασποράς φάσματος μέσω σήματος chirp (Chirp Spread Spectrum – CSS). Στο σήμα chirp, η συχνότητα αυξάνεται ή μειώνεται χρονικά, ενώ η συχνότητα του εκπεμπόμενου σήματος αλλάζει σύμφωνα με μια ακολουθία κώδικα διασποράς. Αυτό

επιτρέπει στο LoRa να χρησιμοποιεί εύρος ζώνης που είναι πολύ μεγαλύτερο από το πραγματικό εύρος ζώνης της πληροφορίας που μεταφέρει, με ταυτόχρονη μείωση της φασματικής πυκνότητας ισχύος για σταθερή συνολική ισχύ [12].

- **SigFox:** Η τεχνολογία SigFox είναι μία από τις πιο δημοφιλείς τεχνολογίες Low Power Wide Area (LPWA), μαζί με τη LoRaWAN, και χρησιμοποιείται ευρέως για την ασύρματη επικοινωνία μεταξύ συσκευών στο πλαίσιο του Διαδικτύου των Πραγμάτων. Λειτουργεί στις συχνότητες 868 MHz στην Ευρώπη και 915 MHz στη Βόρεια Αμερική, οι οποίες ανήκουν στη ζώνη συχνοτήτων Industrial, Scientific, and Medical (ISM). Η εμβέλεια της τεχνολογίας αυτής είναι αρκετά ικανοποιητική, καθώς φτάνει τα 3 με 10 χιλιόμετρα σε αστικά περιβάλλοντα και τα 30 με 50 χιλιόμετρα σε αγροτικές περιοχές. Το σήμα της SigFox μεταδίδεται σε μια εξαιρετικά στενή ζώνη συχνοτήτων της τάξης των 100 Hz, κάτι που επιτρέπει στις συσκευές λήψης να έχουν πολύ υψηλή ευαισθησία και συνεπώς να επιτυγχάνεται μεγάλη εμβέλεια. Συγκεκριμένα, η τεχνολογία SigFox χρησιμοποιεί 192 KHz από το φάσμα συχνοτήτων για τη μετάδοση των δεδομένων. Κάθε μήνυμα που στέλνει έχει εύρος μόνο 100 Hz, κάτι που βοηθά στην αποδοτικότητα της μετάδοσης. Χάρη στην τυχαία πρόσβαση (random access), εξασφαλίζεται υψηλή ποιότητα υπηρεσίας, με τα δεδομένα να μεταδίδονται με ρυθμό περίπου 100 bps. Κάθε μήνυμα μπορεί να έχει μέγιστο μέγεθος 12 bytes και κάθε συσκευή μπορεί να στέλνει μέχρι 14 μηνύματα την ημέρα. Τέλος, στα δίκτυα SigFox υπάρχουν δύο βασικά επίπεδα. Το πρώτο είναι το επίπεδο δικτυακού εξοπλισμού (network equipments layer), που περιλαμβάνει τους σταθμούς βάσης και τις κεραίες για τη λήψη και μεταφορά των σημάτων, και το δεύτερο είναι το επίπεδο συστημάτων υποστήριξης (support systems layer), που επεξεργάζεται τα μηνύματα και τα αποστέλλει στο σύστημα πελατών (customer system), επιτρέποντας την αλληλεπίδραση των συσκευών [6], [12], [13].

2.3.2 Μικρής Εμβέλειας (Short-Range)

- **ZigBee:** Το ZigBee, το οποίο αναπτύχθηκε από την ZigBee Alliance, αποτελεί ένα πρότυπο χαμηλού κόστους και πολύ χαμηλής κατανάλωσης ενέργειας για αμφίδρομη και ασύρματη επικοινωνία. Βασίζεται στην τεχνολογία του IEEE 802.15.4, επεκτείνοντας τη στοίβα πρωτοκόλλων του με τα επίπεδα δικτύου και εφαρμογών, ενώ αξιοποιεί τις υπηρεσίες του φυσικού επιπέδου και του MAC υπο-επιπέδου του IEEE 802.15.4. Ένα από τα βασικά χαρακτηριστικά του ZigBee είναι ο χαμηλός ρυθμός μετάδοσης δεδομένων και η δυνατότητα υποστήριξης έως και 254 συσκευών σε τοπολογία αστέρα. Το δίκτυο ZigBee επιτρέπει πολλαπλή πρόσβαση, προσφέροντας ίση πρόσβαση σε όλες τις συσκευές στο μέσο

επικοινωνίας. Υποστηρίζονται δύο τύποι μηχανισμών πολλαπλής πρόσβασης: ο πρώτος αφορά τη λειτουργία beacon, όπου οι συσκευές εκπέμπουν μόνο σε προκαθορισμένες χρονοθυρίδες, και ο δεύτερος τη λειτουργία non-beacon, στην οποία οι συσκευές μπορούν να εκπέμπουν οποιαδήποτε στιγμή που το κανάλι είναι ελεύθερο. Κάθε συσκευή συμβατή με το ZigBee είναι επίσης συμβατή με το πρότυπο IEEE 802.15.4 και λειτουργεί στις ζώνες συχνοτήτων 868 MHz, 915 MHz και 2.4 GHz, με μέγιστο ρυθμό μετάδοσης δεδομένων 250 kbps. Οι συσκευές ZigBee παραμένουν το μεγαλύτερο μέρος του χρόνου σε κατάσταση αδράνειας (sleep mode), παρουσιάζοντας πολύ μικρή κατανάλωση ενέργειας. Αν και ο χαμηλός ρυθμός μετάδοσης δεδομένων σημαίνει ότι το ZigBee δεν είναι κατάλληλο για εφαρμογές που απαιτούν υψηλές ταχύτητες, όπως ασύρματη σύνδεση στο διαδίκτυο ή ασύρματα hands-free με ποιότητα φωνής αντίστοιχη του CD, είναι η βέλτιστη επιλογή για εφαρμογές που απαιτούν απλή ανταλλαγή εντολών ή συγκέντρωση δεδομένων από αισθητήρες, όπως θερμοκρασίας και υγρασίας [1], [14], [14].

- **NFC:** Η Ασύρματη Επικοινωνία Κοντινού πεδίου (Near Field Communication - NFC) είναι μια τεχνολογία ασύρματης επικοινωνίας που βασίζεται στο πρότυπο ISO/IEC 13157 και επιτρέπει την επικοινωνία μεταξύ συσκευών που βρίσκονται σε πολύ μικρή απόσταση, συνήθως έως 10 εκατοστά. Χρησιμοποιείται κυρίως σε κινητά τηλέφωνα και φορητές συσκευές, προσφέροντας δυνατότητες όπως ανέπαφες χρηματικές συναλλαγές (contactless payments). Η τεχνολογία NFC αποτελεί μια εξέλιξη της τεχνολογίας RFID, συνδυάζοντας την αποθήκευση και ανάγνωση δεδομένων στην ίδια συσκευή και βασίζεται στο φαινόμενο της επαγωγής μέσω μαγνητικού πεδίου ανάμεσα στις κεραίες των συσκευών. Η συχνότητα διαμόρφωσης των σημάτων που εκπέμπονται από το NFC είναι 13.56 MHz, ενώ οι ρυθμοί μεταφοράς των δεδομένων μπορούν να φτάσουν τα 420 kbps. Το NFC μπορεί να λειτουργεί σε τρεις διαφορετικές καταστάσεις: Read/Write, όπου μια συσκευή είναι ενεργή και η άλλη παθητική, επιτρέποντας τη μετάδοση και λήψη δεδομένων, Card emulation, που επιτρέπει στις NFC συσκευές να συμπεριφέρονται σαν έξυπνη κάρτα, και Peer-to-peer, όπου δύο συσκευές μπορούν να ανταλλάσσουν δεδομένα. Ωστόσο, κατά την ανταλλαγή των δεδομένων μέσω NFC δεν χρησιμοποιείται κάποια μέθοδος κρυπτογράφησης, με αποτέλεσμα το κανάλι μετάδοσης να είναι διαθέσιμο στο ευρύτερο περιβάλλον. Αυτό δημιουργεί ευπάθειες, καθώς κακόβουλοι χρήστες, γνωστοί ως ωτακουστές (eavesdroppers), μπορούν να υποκλέψουν ή να αλλάξουν τα μεταδιδόμενα δεδομένα, καθιστώντας σημαντική την ανάγκη για μέτρα ασφάλειας κατά τη χρήση της τεχνολογίας αυτής [4], [10].

- **Bluetooth Low Energy (BLE):** Το Bluetooth Low Energy (BLE) ή Bluetooth Smart είναι ένα πρότυπο που έχει προταθεί ειδικά για φορητές συσκευές και εφαρμογές IoT, χαρακτηριζόμενο από χαμηλή κατανάλωση ενέργειας και για συγκεκριμένους τεχνολογικούς στόχους. Παρόλο που δανείζεται το όνομα της τεχνολογίας Bluetooth και μερικά από τα τεχνολογικά της στοιχεία, αποτελεί ξεχωριστή τεχνολογία με εστίαση σε απλές συνδέσεις και μικρές ανταλλαγές δεδομένων, διατηρώντας τις συσκευές ανενεργές για μεγάλα χρονικά διαστήματα. Χρησιμοποιείται στη ζώνη συχνοτήτων ISM των 2.4 GHz, με πρόσβαση στο κανάλι να γίνεται μέσω πολλαπλής πρόσβασης διαίρεσης συχνότητας (FDMA) ή χρόνου (TDMA). Ο ρυθμός μετάδοσης είναι 1 Mbps, χρησιμοποιώντας διαμόρφωση GFSK (Gaussian Frequency Shift Modulation), με εμβέλεια που φτάνει έως 100 μέτρα και μέγιστη ισχύς μετάδοσης 10 mW [1].

2.4 Πρωτόκολλα Επιπέδου Εφαρμογής

Το επίπεδο εφαρμογής συνήθως είναι το τελευταίο επίπεδο και λειτουργεί ως μια διεπαφή αλληλεπίδρασης μεταξύ του τελικού χρήστη και των υπολοίπων επιπέδων. Πρωτόκολλα που εκτελούνται σε αυτό το επίπεδο είναι τα Hypertext Transfer Protocol (HTTP), το Message Queuing Telemetry Transport (MQTT), το Constrained Application Protocol (CoAP), το Extensible Messaging Presence Protocol (XMPP) και το Advanced Message Queuing Protocol (AMQP) καθώς και άλλα όχι τόσο ευρέως διαδεδομένα [1], [4], [14]. Όμως, η χρήση του πρωτοκόλλου που θα χρησιμοποιηθεί σε ένα IoT περιβάλλον εξαρτάται από πολλούς παράγοντες, όπως το πλήθος των χρηστών που υπάρχουν, η απαίτηση για γρήγορο ή χαμηλό ρυθμό μετάδοσης δεδομένων, η ανάγκη για ασφαλή μετάδοση δεδομένων και άλλα. Οι τεχνικές παράμετροι των τριών πρωτοκόλλων είναι αποτυπωμένες στο Πίνακα 1.

2.4.1 HTTP

Το HTTP, αποτελεί το πιο γνωστό πρότυπο στο world wide web (www) και η προτυποποίηση του πραγματοποιήθηκε από τον οργανισμό Internet Engineering Task Force (IETF). Το μοντέλο λειτουργίας του είναι βασισμένο στο μοντέλο client-server, όπου ο client αιτείται για σελίδες τύπου Hypertext Markup Language (HTML) από τον server και ο server επίσης πρέπει να απαντήσει με σελίδα HTML. Βασίζεται στο πρωτόκολλο Transmission Control Protocol (TCP) στο επίπεδο μεταφοράς όπου και γίνεται χρήση υποδοχής (sockets) για τη μεταφορά δεδομένων. Συνήθως, η έναρξη σύνδεσης ανάμεσα στον client και τον server, ξεκινά από τον client, μέσω αρχικοποίησης ενός socket στη πόρτα 80 και αφού η σύνδεση εγκαθιδρυθεί, ο server αποδέχεται το αίτημα του

client και ξεκινά η υπηρεσία. Στο τερματισμό της σύνδεσης, το πρωτόκολλο TCP τερματίζει τη σύνδεση ανάμεσα σε client και server, και καθαρίζει όλη τη μνήμη που δεσμεύθηκε για το σκοπό αυτό [4], [10], [15].

Αναφορικά με τις μεθόδους που διαθέτει το πρότυπο HTTP, οι κυριότερες μέθοδοι είναι οι GET, PUT, POST, DELETE. Η μέθοδος GET απεικονίζει μια σελίδα ιστού κατόπιν αιτήσεως του χρήστη. Οι μέθοδοι PUT και POST χρησιμοποιούνται για τροποποίηση των πόρων του server και η μέθοδος DELETE για διαγραφή πόρων που δεν χρειάζονται πλέον. Επιπρόσθετα, με βάση το TCP, υπάρχουν 2 είδη HTTP συνδέσεων, το πρότυπο non-persistent (HTTP/1.0) και το persistent (HTTP/1.1). Η κύρια διαφορά τους εντοπίζεται στον αριθμό των TCP συνδέσεων που απαιτείται για να μεταδοθεί το Uniform Resource Locator (URL) μιας σελίδας ιστού και των αντικειμένων της.

Αξίζει να σημειωθεί ότι με βάση το HTTP, η τεχνολογία Representational State Transfer (REST) είναι μια γλώσσα ανεξάρτητη του λειτουργικού συστήματος που εκτελεί η συσκευή IoT και χρησιμοποιείται για να συνδέσει πολλές συσκευές IoT μαζί. Είναι stateless, ακολουθεί το μοντέλο client-server και εκτελείται πάνω από το HTTP πρότυπο.

2.4.2 MQTT

Το MQTT πρωτόκολλο είναι μια τεχνολογία διασύνδεσης ανάμεσα σε μηχανές (machine-to-machine, M2M) κατάλληλη για δίκτυα χαμηλής ισχύος. Ακολουθεί το μοντέλο του client-server και υλοποιεί το παράδειγμα του publish/subscribe messaging. Η υλοποίηση του βασίζεται στο πρωτόκολλο TCP/IP, είναι lightweight, ανοιχτού τύπου και εύκολο να υιοθετηθεί από τις συσκευές IoT [4]. Συνήθως, ο client χαρακτηρίζεται ως Publisher, ο server ως Broker, καθώς υπάρχουν και οι ρόλοι του Subscriber και των Topics. Συνεπώς, με βάση το προηγούμενο παράδειγμα, πρέπει να υπάρχει ένας κεντρικός MQTT Broker ώστε να διαχειρίζεται και να δρομολογεί τα δεδομένα ανάμεσα στους MQTT κόμβους ή Subscribers. Οι Publishers είναι συσκευές που παράγουν δεδομένα και συνδέονται σε έναν Broker για να προωθήσουν τα δεδομένα τους. Οι Subscribers μπορεί να είναι είτε συσκευές είτε εφαρμογές που λογικά συνδέονται σε έναν client που ενδιαφέρεται για κάποια δεδομένα και συνήθως συνδέονται δικτυακά με κάποιον Broker για να λάβουν την απαιτούμενη πληροφορία. Τα δεδομένα που φυλάσσουν οι Brokers είναι αποθηκευμένα σε Topics, δηλαδή ουρές μηνυμάτων που υποστηρίζουν το publish/subscribe pattern.

Αναφορικά με τις μεθόδους μηνυμάτων που χρησιμοποιεί το πρωτόκολλο, υποστηρίζει 14 μεθόδους. Οι κυριότερες μέθοδοι που ο τελικός χρήστης πρέπει να χρησιμοποιεί είναι οι αιτήσεις σύνδεσης προς τον server, και είναι οι CONNECT, DISCONNECT, SUBSCRIBE, UNSUBSCRIBE, PUBLISH καθώς οι υπόλοιπες είναι κυρίως για εσωτερικούς μηχανισμούς και ροές μηνυμάτων.

Αξίζει να σημειωθεί ότι, το MQTT υποστηρίζει διάφορα επίπεδα Quality of Service (QoS), και συγκεκριμένα 3 επίπεδα πάνω από το TCP. Το QoS level 0, όπου τα μηνύματα παραδίδονται σε σειρά, το QoS level 1, όπου τα μηνύματα πρέπει να γίνονται acknowledged. Τελικά, το QoS level 2, είναι το πιο υψηλό, ασφαλή αλλά και πιο αργό επίπεδο.

2.4.3 CoAP

Το πρωτόκολλο CoAP, είναι ένα πρωτόκολλο που βασίζεται στο μοντέλο client-server, παρόμοια με το HTTP. Έχει σχεδιαστεί για συσκευές με περιορισμούς στην επεξεργαστική ισχύ, στο δίκτυο αλλά και στη διαθέσιμη ισχύ κατανάλωσης. Έτσι, το CoAP, αντιμετωπίζει τις αδυναμίες του HTTP. Το πρωτόκολλο που βασίζεται είναι το User Datagram Protocol (UDP) για ασύγχρονη μετάδοση και όχι στο TCP. Επιπλέον, οι συσκευές που είναι συνδεδεμένες με το CoAP μπορούν να ανακαλύψουν η μια την άλλη με στόχο να ανταλλάξουν δεδομένα. Ακόμη, ένας client μπορεί να ειδοποιηθεί μόνο όταν συγκεκριμένες δράσεις γίνονται. Όσο αφορά τους τύπους μηνυμάτων, υποστηρίζονται τέσσερις, οι Confirmable, Non-Confirmable, Acknowledgement και Reset.

Οι μέθοδοι αιτημάτων του πρωτοκόλλου είναι παρόμοιοι με του HTTP και είναι οι GET, POST, PUT και DELETE. Τέλος, ο μηχανισμός ασφάλειας που έχει αναπτυχθεί για το CoAP είναι ο Datagram Transport Layer Security (DTLS) παρέχοντας την αναδιάταξη με σειρά των πακέτων και κρυπτογράφηση των δεδομένων [9], [14].

Πίνακας 1 Τεχνικές παράμετροι πρωτοκόλλων HTTP, MQTT, CoAP

	HTTP	MQTT	CoAP
Model	Request/Reply	Pub/Sub	Request/Reply
Architecture	P2P	Brokered	P2P
Interoperability	Yes	Partial	Yes
QoS	TCP-based	3	Confirmable, No-Confirmable
Transport Layer Protocol	TCP	TCP	UDP
Subscription Control	N/A	Hierarchical Matching	Multicast addressing messages
Dynamic Discovery	No	No	Yes
Security	SSL/TLS	SSL	DTLS

3

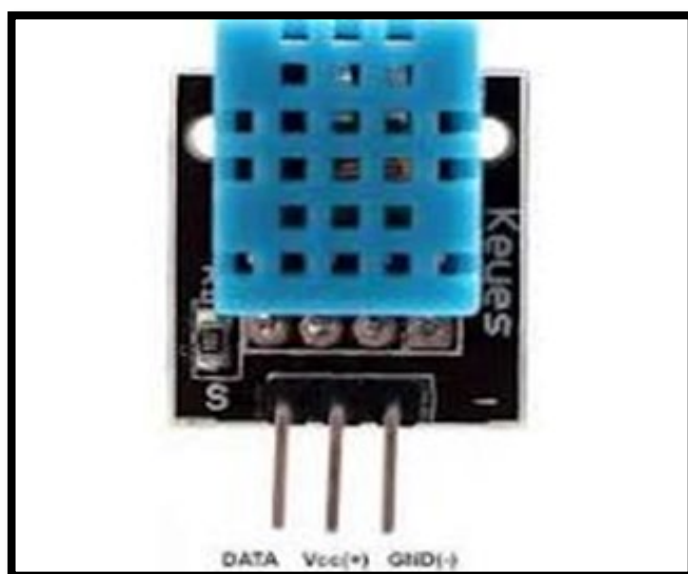
Συσκευές και Εξαρτήματα υλοποίησης της IoT συσκευής.

3.1 Αισθητήρες (Sensors)

3.1.1 DHT11 αισθητήρας υγρασίας θερμοκρασίας

Ο αισθητήρας DHT11 (Σχήμα 4), είναι ένας ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας κατάλληλος για διάφορες εφαρμογές, όπως συστήματα εξαερισμού και κλιματισμού, συστήματα ελέγχου θερμοκρασίας, έξυπνα συστήματα γεωργίας, μετεωρολογικούς σταθμούς κ.α. Συγκεκριμένα, εκπέμπει ένα ψηφιακό σήμα στον ακροδέκτη δεδομένων, εξαλείφοντας την ανάγκη για αναλογικές ακίδες εισόδου και καθιστώντας τον εύκολο στη σύνδεση με μικροελεγκτές Arduino, waspmote, esp32 και άλλες πολλές πλακέτες. Ο DHT11 διαθέτει 3 ακίδες:

1. VCC (3.3V): Τροφοδοσία του αισθητήρα με 3.3V.
2. GND: Γείωση του αισθητήρα.
3. DATA: Ψηφιακή έξοδος δεδομένων.



Σχήμα 4: DHT11 αισθητήρας θερμοκρασίας, υγρασίας

Επιπλέον θεωρείται είναι ένας από τους πιο οικονομικούς ψηφιακούς αισθητήρες θερμοκρασίας και υγρασίας στην αγορά καθώς καταναλώνει ελάχιστη ενέργεια, καθιστώντας τον ιδανικό για εφαρμογές που λειτουργούν με μπαταρία.

3.1.2 Αισθητήρας οξυγόνου (O₂)

Η Libelium προσφέρει προηγμένους αισθητήρες οξυγόνου ως μέρος των πλατφορμών Smart Water και Smart Water Xtreme. Αυτοί οι αισθητήρες είναι ειδικά σχεδιασμένοι για ένα ευρύ φάσμα εφαρμογών παρακολούθησης δασικών πυρκαγιών, ατμοσφαιρικής ρύπανσης και άλλων πολλών. Ο αισθητήρας που χρησιμοποιήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας απεικονίζεται στο Σχήμα 5, και τα τεχνικά χαρακτηριστικά του αναγράφονται στον Πίνακα 2.



Σχήμα 5 Αισθητήρας οξυγόνου, Oxygen LF02-A4

Πίνακας 2 Τεχνικές προδιαγραφές αισθητήρα οξυγόνου

Χαρακτηριστικά Απόδοσης	
Χρόνος Απόκρισης (T90)	≤ 17 δευτερόλεπτα
Ευαισθησία	80-130 μ A @ 20.9% O ₂
Ακρίβεια	όσο καλή όσο $\pm 0.1\%$ (ιδανικές συνθήκες)
Εύρος Θερμοκρασίας	-30 °C έως 50 °C
Λειτουργική Υγρασία	5 έως 95% RH χωρίς συμπύκνωση
Αναμενόμενη Διάρκεια Λειτουργίας	2 χρόνια: 85% της αρχικής απόδοσης

3.1.3 Αισθητήρας διοξειδίου του άνθρακα (CO₂)

Ο αισθητήρας CO-A4, όπως απεικονίζεται στο Σχήμα 6, είναι ένας εξειδικευμένος αισθητήρας για την ανίχνευση και μέτρηση του Διοξειδίου του άνθρακα (CO₂) σε αέρια μορφή. Συγκεκριμένα, αυτός ο αισθητήρας είναι ιδανικός για εφαρμογές όπου απαιτείται ακριβής και αξιόπιστη ανίχνευση του διοξειδίου του άνθρακα, όπως σε συστήματα ασφαλείας, ελέγχου περιβαλλοντικών συνθηκών καθώς και παρακολούθησης βιομηχανικών διαδικασιών. Με το γρήγορο χρόνο απόκρισης και την υψηλή ακρίβεια του, ο CO-A4 εξασφαλίζει την ανίχνευση ακόμη και μικρών συγκεντρώσεων CO₂, προσφέροντας σημαντική ασφάλεια και αξιοπιστία. Τέλος τα τεχνικά χαρακτηριστικά του αισθητήρα 4-CO-500 αναγράφονται στο Πίνακα 3.



Σχήμα 6: Αισθητήρας διοξειδίου του άνθρακα - CO₂

Πίνακας 3: Τεχνικές προδιαγραφές αισθητήρα CO₂

Χαρακτηριστικά Απόδοσης	
Χρόνος Απόκρισης (T90)	≤ 60 δευτερόλεπτα
Εύρος	0 έως 5000 ppm
Ακρίβεια	±1 ppm σε ιδανικές συνθήκες
Εύρος Θερμοκρασίας	-40 °C έως 60 °C
Λειτουργική Υγρασία	0 έως 95% RH χωρίς συμπύκνωση
Αναμενόμενη Διάρκεια Λειτουργίας	5 χρόνια

3.1.4 Αισθητήρας διοξειδίου του αζώτο

Ο αισθητήρας NO₂-A43F, όπως απεικονίζεται στο Σχήμα 7, της Alphasense, κατασκευασμένος στο Ηνωμένο Βασίλειο, είναι ένας ηλεκτροχημικός αισθητήρας σχεδιασμένος για την ανίχνευση

διοξειδίου του αζώτου (NO_2). Με ευαισθησία από -175 έως -450 nA/ppm στα 2 ppm NO_2 και χρόνο απόκρισης μικρότερο από 60 δευτερόλεπτα, προσφέρει γρήγορη και ακριβή ανίχνευση. Αυτός ο αισθητήρας είναι ιδανικός για συστήματα ανίχνευσης πυρκαγιάς και ελέγχου ποιότητας αέρα, παρέχοντας έγκαιρη ειδοποίηση για την παρουσία τοξικών αερίων και συμβάλλοντας στη βελτίωση της ασφάλειας σε βιομηχανικά περιβάλλοντα. Τα χαρακτηριστικά του απεικονίζονται στο Πίνακα 4.



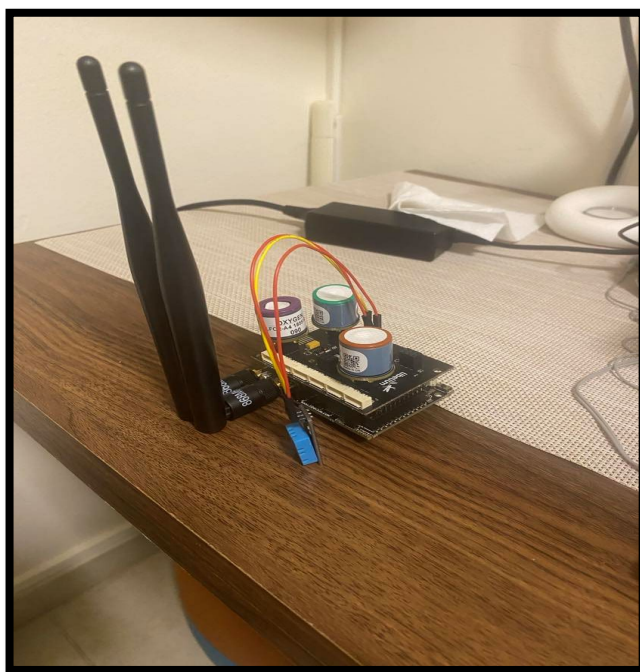
Σχήμα 7 Αισθητήρας διοξειδίου του αζώτου NO_2

Πίνακας 4: Τεχνικές προδιαγραφές αισθητήρα διοξειδίου του αζώτου

Χαρακτηριστικά Απόδοσης	
Χρόνος Απόκρισης (T90)	≤ 60 δευτερόλεπτα
Εύρος	20 ppm
Ατμοσφαιρική πίεση	<60 δευτερόλεπτα από 0 έως 2ppm NO_2
Εύρος Θερμοκρασίας	-30 °C έως 40 °C
Λειτουργική Υγρασία	15 έως 85% RH
Αναμενόμενη Διάρκεια Λειτουργίας	5 χρόνια

3.2 Προγραμματισμός IoT συσκευής μέσω Wasmote board

Σε αυτή την υποενότητα παρέχεται μια συνοπτική περιγραφή του προγραμματισμού της IoT συσκευής για τη λήψη μετρήσεων από τους αισθητήρες που περιεγράφηκαν προηγουμένως και την αποστολή τους στο The Things Network μέσω του δικτύου LoRaWAN. Για την υλοποίηση της IoT συσκευής, τόσο σε επίπεδο υλικού (hardware) όσο και λογισμικού (software), χρησιμοποιήθηκε η ανοιχτού κώδικα πλατφόρμα Wasmote της Libelium. Η πλατφόρμα αυτή, η οποία χρησιμοποιείται από προγραμματιστές που ασχολούνται με IoT εφαρμογές, είναι τροποποιήσιμη, ισχυρή και ευέλικτη. Επιτρέπει τη σύνδεση οποιουδήποτε αισθητήρα με οποιαδήποτε πλατφόρμα νέφους (cloud platform) χρησιμοποιώντας οποιαδήποτε τεχνολογία ασύρματης επικοινωνίας. Το Σχήμα 8, απεικονίζει το πλήρες υλικό, συμπεριλαμβανομένων των αισθητήρων, της πλακέτας και του LoRa Module.



Σχήμα 8 : IoT συσκευή

Το Σχήμα 9, παρουσιάζει τις βιβλιοθήκες που χρησιμοποιήθηκαν για την επικοινωνία του IoT Board με τους αισθητήρες και το δίκτυο LoRaWAN, ενώ στο Σχήμα 10, απεικονίζονται οι ορισμοί των socket και οι μοναδικοί κωδικοί για την επικοινωνία με το δίκτυο LoRaWAN.

```
#include <WaspFrame.h>
#include <WaspLoRaWAN.h>
#include <WaspSensorGas_Pro.h>

// socket to use
////////////////////////////////////
uint8_t socket = SOCKET0;
////////////////////////////////////

// Device parameters for Back-End registration
////////////////////////////////////
char DEVICE_EUI[] = "*****";
char APP_EUI[] = "*****";
char APP_KEY[] = "*****";
////////////////////////////////////

// Define port to use in Back-End: from 1 to 223
uint8_t PORT = 1;

// variable
uint8_t error;
uint8_t error_config = 0;
```

Σχήμα 9 : Βιβλιοθήκες εφαρμογής

```
// O2 Sensor must be connected in SOCKET_1
O2SensorClass O2Sensor(SOCKET_1);
// CO Sensor must be connected physically in SOCKET_4
COSensorClass COSensor;
// NO2 Sensor must be connected physically in SOCKET_3
NO2SensorClass NO2Sensor;

// Percentage values of Oxygen
#define POINT1_PERCENTAGE 0.0
#define POINT2_PERCENTAGE 5.0

// Calibration Voltage Obtained during calibration process (in mV)
#define POINT1_VOLTAGE 0.35
#define POINT2_VOLTAGE 2.0

// Concentrations used in calibration process CO
#define POINT1_PPM_CO 100.0 // <--- Ro value at this concentration
#define POINT2_PPM_CO 300.0 //
#define POINT3_PPM_CO 1000.0 //

// Calibration resistances obtained during calibration process CO
#define POINT1_RES_CO 230.30 // <-- Ro Resistance at 100 ppm. Necessary value.
#define POINT2_RES_CO 40.665 //
#define POINT3_RES_CO 20.300 //

// Concentrations used in calibration process
#define POINT1_PPM_NO2 10.0 // <-- Normal concentration in air
#define POINT2_PPM_NO2 50.0
#define POINT3_PPM_NO2 100.0

// Calibration voltages obtained during calibration process (in KOHMs)
#define POINT1_RES_NO2 45.25 // <-- Rs at normal concentration in air
#define POINT2_RES_NO2 25.50
```

Σχήμα 10 Ρυθμίσεις socket, port και παραμέτρων συσκευής

Παρακάτω απεικονίζεται ο κώδικας της setup() συνάρτησης ή οποία ρυθμίζει τους αισθητήρες και την επικοινωνία με το δίκτυο LoRaWAN, ενώ ελέγχει για τυχόν σφάλματα και εκτυπώνει κατάλληλα μηνύματα στην οθόνη USB.

```
void setup()
{
  USB.ON();
  USB.println(F("Sensor reading..."));

  O2Sensor.setCalibrationPoints(voltages, concentrations);
  COSensor.setCalibrationPoints(resValuesCO, concentrationsCO, numPoints);
  NO2Sensor.setCalibrationPoints(voltagesNO2, concentrationsNO2, numPoints);

  Gases.ON();
  O2Sensor.ON();
  COSensor.ON();
  NO2Sensor.ON();

  error = LoRaWAN.ON(socket);
  if ( error == 0 ) USB.println(F("1. Switch ON OK"));
  else { USB.print(F("1. Switch ON error = ")); USB.println(error, DEC);
error_config = 1; }

  error = LoRaWAN.setDataRate(3);
  if ( error == 0 ) USB.println(F("2. Data rate set OK"));
  else { USB.print(F("2. Data rate set error= ")); USB.println(error, DEC);
error_config = 2; }

  error = LoRaWAN.setDeviceEUI(DEVICE_EUI);
  if ( error == 0 ) USB.println(F("3. Device EUI set OK"));
  else { USB.print(F("3. Device EUI set error = ")); USB.println(error, DEC);
error_config = 3; }

  error = LoRaWAN.setAppEUI(APP_EUI);
  if ( error == 0 ) USB.println(F("4. Application EUI set OK"));
  else { USB.print(F("4. Application EUI set error = ")); USB.println(error, DEC);
error_config = 4; }

  error = LoRaWAN.setAppKey(APP_KEY);
  if ( error == 0 ) USB.println(F("5. Application Key set OK"));
  else { USB.print(F("5. Application Key set error = ")); USB.println(error, DEC);
error_config = 5; }
```

```
error = LoRaWAN.joinOTAA();
if ( error == 0 ) USB.println(F("6. Join network OK"));
else { USB.print(F("6. Join network error = ")); USB.println(error, DEC);
error_config = 6; }

error = LoRaWAN.saveConfig();
if ( error == 0 ) USB.println(F("7. Save configuration OK"));
else { USB.print(F("7. Save configuration error = ")); USB.println(error, DEC);
error_config = 7; }

error = LoRaWAN.OFF(socket);
if ( error == 0 ) USB.println(F("8. Switch OFF OK"));
else { USB.print(F("8. Switch OFF error = ")); USB.println(error, DEC);
error_config = 8; }

if (error_config == 0) {
    USB.println(F("\n-----
--"));
    USB.println(F("Module configured"));
    USB.println(F("After joining through OTAA, the module and the network exchanged
"));
    USB.println(F("the Network Session Key and the Application Session Key which
"));
    USB.println(F("are needed to perform communications. After that, 'ABP mode' is
used"));
    USB.println(F("to join the network and send messages after powering on the
module"));
    USB.println(F("-----
\n"));
    USB.println();
}
else {
    USB.println(F("\n-----
--"));
    USB.println(F("Module not configured"));
    USB.println(F("Check OTTA parameters and restart the code."));
    USB.println(F("If you continue executing the code, frames will not be sent."));
    USB.println(F("\n-----
--"));
}
}
```

SetUp Συνάρτηση

Στην συνέχεια δίνεται ο κώδικας της loop() συνάρτησης, η οποία αναλαμβάνει να συλλέξει και να στείλει τις μετρήσεις των αισθητήρων στο δίκτυο LoRaWAN σε τακτά χρονικά διαστήματα. Αρχικά, διαβάζει τις τάσεις και τις συγκεντρώσεις από τους αισθητήρες οξυγόνου (O2), μονοξειδίου του άνθρακα (CO) και διοξειδίου του αζώτου (NO2), καθώς και την υγρασία. Οι μετρήσεις αυτές αποθηκεύονται σε κατάλληλες μεταβλητές και προβάλλονται στην οθόνη μέσω USB. Στη συνέχεια, η συσκευή ενεργοποιεί το LoRaWAN, εντάσσεται στο δίκτυο μέσω της μεθόδου ABP και ετοιμάζει τα δεδομένα σε μορφή byte για αποστολή. Τα δεδομένα αποστέλλονται στο προκαθορισμένο κανάλι επικοινωνίας και ελέγχεται η επιτυχία της αποστολής. Τέλος, η συσκευή απενεργοποιεί το LoRaWAN και περιμένει για τρία δευτερόλεπτα πριν επαναλάβει τη διαδικασία.

```
void loop()
{
  //sensor loop start
  // O2 Sensor does not need power supply
  float O2Vol = O2Sensor.readVoltage();

  float COVol = COSensor.readVoltage();    // Voltage value of the sensor
  float CORes = COSensor.readResistance();  // Resistance of the sensor
  float COPPM = COSensor.readConcentration(); // PPM value of CO

  // PPM value of NO2
  float NO2Vol = NO2Sensor.readVoltage();    // Voltage value of the sensor
  float NO2Res = NO2Sensor.readResistance(); // Resistance of the sensor
  float NO2PPM = NO2Sensor.readConcentration(); // PPM value of NO2

  USB.print(F("O2 concentration Estimated: "));
  USB.print(O2Vol);
  USB.print(F(" mV | "));

  delay(100);

  // Read the concentration value in %
  float O2Val = O2Sensor.readConcentration();

  USB.print(F(" O2 concentration Estimated: "));
  USB.print(O2Val);
  USB.println(F(" %"));
```

```
// Print of the results
USB.print(F("CO Sensor Voltage: "));
USB.print(COVol);
USB.print(F(" mV |"));

// Print of the results
USB.print(F(" CO Sensor Resistance: "));
USB.print(CORes);
USB.print(F(" Ohms |"));

    USB.print(F(" CO concentration Estimated: "));
USB.print(COPPM);
USB.println(F(" ppm"));

// Print of the results
USB.print(F("NO2 Sensor Voltage: "));
USB.print(NO2Vol);
USB.print(F(" V |"));

// Print of the results
USB.print(F(" NO2 Sensor Resistance: "));
USB.print(NO2Res);
USB.print(F(" Ohms |"));

delay(100);
// read Humidity sensor connected to ANALOG7

int humidity = Utils.readHumidity();
// Print of the results
USB.print(F(" Humidity: "));
USB.print(humidity);
////////////////////////////////////
// 1. Switch on
////////////////////////////////////

error = LoRaWAN.ON(socket);

// Check status
if ( error == 0 )
{
    USB.println(F("1. Switch ON OK"));
}
```

```
else
{
    USB.print(F("1. Switch ON error = "));
    USB.println(error, DEC);
}

////////////////////////////////////
// 2. Join network
////////////////////////////////////

error = LoRaWAN.joinABP();

// Check status
if ( error == 0 )
{
    USB.println(F("2. Join network OK"));

    ///////////////////////////////////

    // Multiply by 100 to preserve up to two decimal places and cast it to uint32_t
    uint32_t sendO2Int = static_cast<uint32_t>(O2Val * 100.0);
    uint32_t sendCOInt = static_cast<uint32_t>(COPPM * 100.0);
    uint32_t sendNO2Int = static_cast<uint32_t>(NO2PPM * 100.0);
    uint32_t sendHumidityInt = static_cast<uint32_t>(humidity * 100.0);
    USB.println(F("\nsendO2Int: "));
    USB.print((sendO2Int));
    USB.println(F("\nsendCOInt: "));
    USB.print((sendCOInt));
    USB.println(F("\nsendNO2Int: "));
    USB.print((sendNO2Int));
    USB.println(F("\n"));
    USB.println(F("\n sendHumidityInt: "));
    USB.print((sendHumidityInt));

    byte data[16]; // the data we shall sent to the cloud. It can only 8 bits per cell

    // Store the unsigned 32 bit integer values in the byte array
    // How it works, it shift the value by multiples of 8 in order to isolate the byte we want
    //Then, the result is bitwise AND with 0xFF ensures that only the lowest 8 bits are retained
    //The resulting value is stored in data
```

```
data[0] = (sendO2Int >> 24) & 0xFF; //The bits being shifted and stored are the 8 most significant
bits, ranging from bit 31 to bit 24.
data[1] = (sendO2Int >> 16) & 0xFF; //The bits being shifted and stored are the next 8 bits, ranging
from bit 23 to bit 16.
data[2] = (sendO2Int >> 8) & 0xFF; //The bits being shifted and stored are the next 8 bits, ranging
from bit 15 to bit 8.
data[3] = sendO2Int & 0xFF; //The bits being stored directly are the 8 least significant bits, ranging
from bit 7 to bit 0.

data[4] = (sendCOInt >> 24) & 0xFF; //The bits being shifted and stored are the 8 most significant
bits, ranging from bit 31 to bit 24.
data[5] = (sendCOInt >> 16) & 0xFF; //The bits being shifted and stored are the next 8 bits,
ranging from bit 23 to bit 16.
data[6] = (sendCOInt >> 8) & 0xFF; //The bits being shifted and stored are the next 8 bits, ranging
from bit 15 to bit 8.
data[7] = sendCOInt & 0xFF; //The bits being stored directly are the 8 least significant bits,
ranging from bit 7 to bit 0.

data[8] = (sendNO2Int >> 24) & 0xFF; //The bits being shifted and stored are the 8 most
significant bits, ranging from bit 31 to bit 24.
data[9] = (sendNO2Int >> 16) & 0xFF; //The bits being shifted and stored are the next 8 bits,
ranging from bit 23 to bit 16.
data[10] = (sendNO2Int >> 8) & 0xFF; //The bits being shifted and stored are the next 8 bits,
ranging from bit 15 to bit 8.
data[11] = sendNO2Int & 0xFF; //The bits being stored directly are the 8 least significant bits,
ranging from bit 7 to bit 0.

data[12] = (sendHumidityInt >> 24) & 0xFF; //The bits being shifted and stored are the 8 most
significant bits, ranging from bit 31 to bit 24.
data[13] = (sendHumidityInt >> 16) & 0xFF; //The bits being shifted and stored are the next 8
bits, ranging from bit 23 to bit 16.
data[14] = (sendHumidityInt >> 8) & 0xFF; //The bits being shifted and stored are the next 8 bits,
ranging from bit 15 to bit 8.
data[15] = sendHumidityInt & 0xFF; //The bits being stored directly are the 8 least significant
bits, ranging from bit 7 to bit 0.

error = LoRaWAN.sendConfirmed(PORT, data, 16); //sends data to the things network

// Error messages:
/*
'6' : Module hasn't joined a network
'5' : Sending error
```

```
'4' : Error with data length
'2' : Module didn't response
'1' : Module communication error
*/
// Check status
if ( error == 0 )
{
  USB.println(F("3. Send Confirmed packet OK"));
  if (LoRaWAN._dataReceived == true)
  {
    USB.print(F("  There's data on port number "));
    USB.print(LoRaWAN._port, DEC);
    USB.print(F(".\r\n  Data: "));
    USB.println(LoRaWAN._data);
  }
}
else
{
  USB.print(F("3. Send Confirmed packet error = "));
  USB.println(error, DEC);
}
}
else
{
  USB.print(F("2. Join network error = "));
  USB.println(error, DEC);
}

////////////////////////////////////
// 4. Switch off
////////////////////////////////////

error = LoRaWAN.OFF(socket);

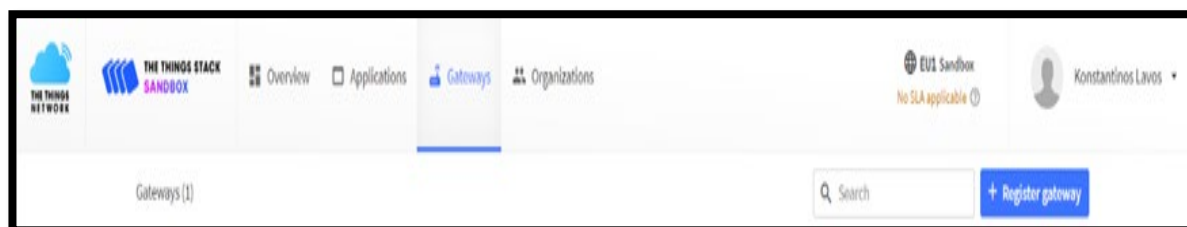
// Check status
if ( error == 0 )
{
  USB.println(F("4. Switch OFF OK"));
}
else
{
  USB.print(F("4. Switch OFF error = "));
```

```
USB.println(error, DEC);  
USB.println();  
}  
delay(3000);  
}
```

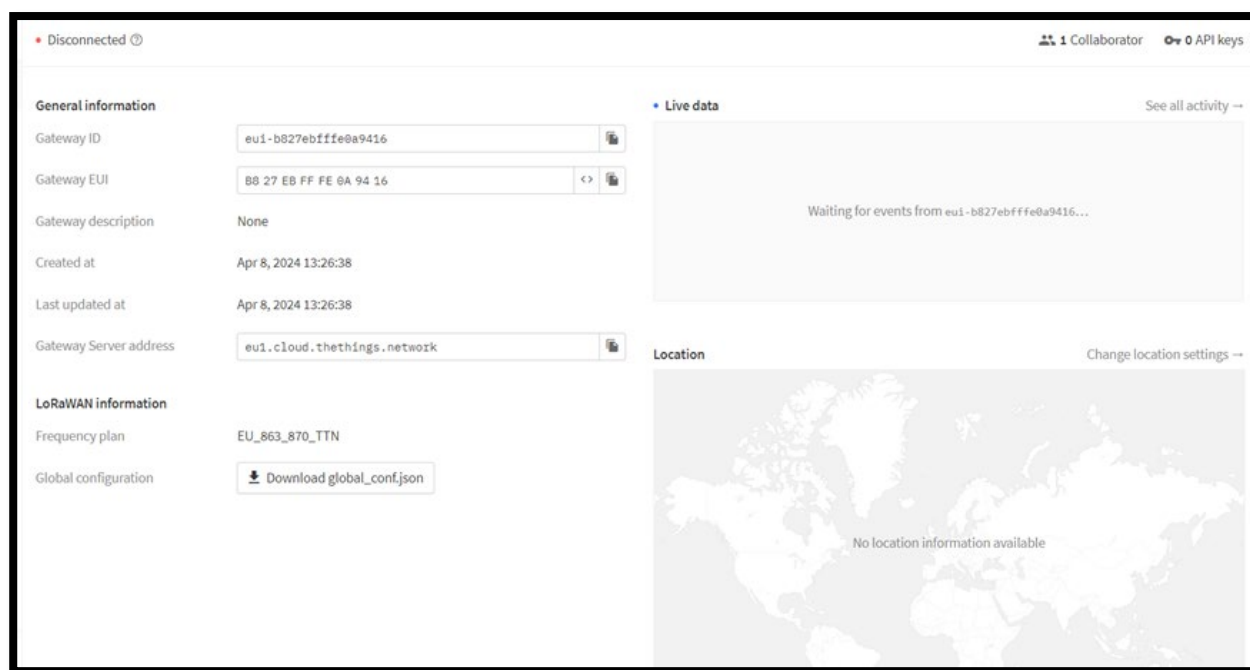
Loop Συνάρτηση

3.3 LoRa Gateway

Η συσκευή που δρα ως LoRa Gateway, αποτελεί το φυσικό κουτί ή περίβλημα που φιλοξενεί το υλικό και το λογισμικό για την εκτέλεση βασικών εργασιών που συνδέουν συσκευές IoT στο cloud. Συγκεκριμένα, η διαδικασία ξεκινά από τις IoT συσκευές (π.χ. αισθητήρες) οι οποίες συλλέγουν και μεταδίδουν περιοδικά τα δεδομένα τους ασύρματα χρησιμοποιώντας τη διαμόρφωση LoRa. Έπειτα, ο LoRa Gateway αποδιαμορφώνει τα λαμβανόμενα σήματα προκειμένου να εξάγει τα δεδομένα. Αφού τα λάβει, ενσωματώνει το timestamp, και αλλά μεταδιδόμενα όπως το RSSI (δείκτης ισχύος ληφθέντος σήματος), το SNR (αναλογία σήματος προς θόρυβο) και το μοναδικό του αναγνωριστικό, σε μια τυπική μορφή πακέτου IP και τα προωθεί σε κάποιον server στον cloud όπως για παράδειγμα είναι το The Things Network μέσω του backhaul (π.χ. Ethernet, Wi-Fi, 4G, 5G). Για την προώθηση των δεδομένων στο The Things Networks (TNN) απαιτείται η εγγραφή του δρομολογητή στην πλατφόρμα τους όπως απεικονίζεται στο Σχήμα 11, και στο Σχήμα 12, αντίστοιχα.



Σχήμα 11 Εγγραφή δρομολογητή στο TNN



Σχήμα 12 Συμπλήρωση φόρμας του Gateway στο TTN

Ο μέγιστος αριθμός αισθητήρων ή συσκευών που μπορεί να υποστηρίξει ένας gateway εξαρτάται από διάφορους παράγοντες, όπως ο ρυθμός δεδομένων και το μέγεθος του πακέτου, αγγίζοντας έως και εκατοντάδες ή και χιλιάδες συσκευές ανά gateway. Η χρήση ενός LoRa Gateway προσφέρει πλήθος πλεονεκτημάτων στις εφαρμογές IoT, επιτρέποντας την αποδοτική και αξιόπιστη διασύνδεση συσκευών, ενώ παράλληλα διευκολύνει τη διαχείριση και την ανάλυση των δεδομένων που συλλέγονται. Η επεκτασιμότητα του συστήματος, σε συνδυασμό με την κάλυψη μεγάλων αποστάσεων και την χαμηλή κατανάλωση ενέργειας των συσκευών LoRa, καθιστούν τους LoRa Gateways απαραίτητα στοιχεία σε σύγχρονα δίκτυα IoT.

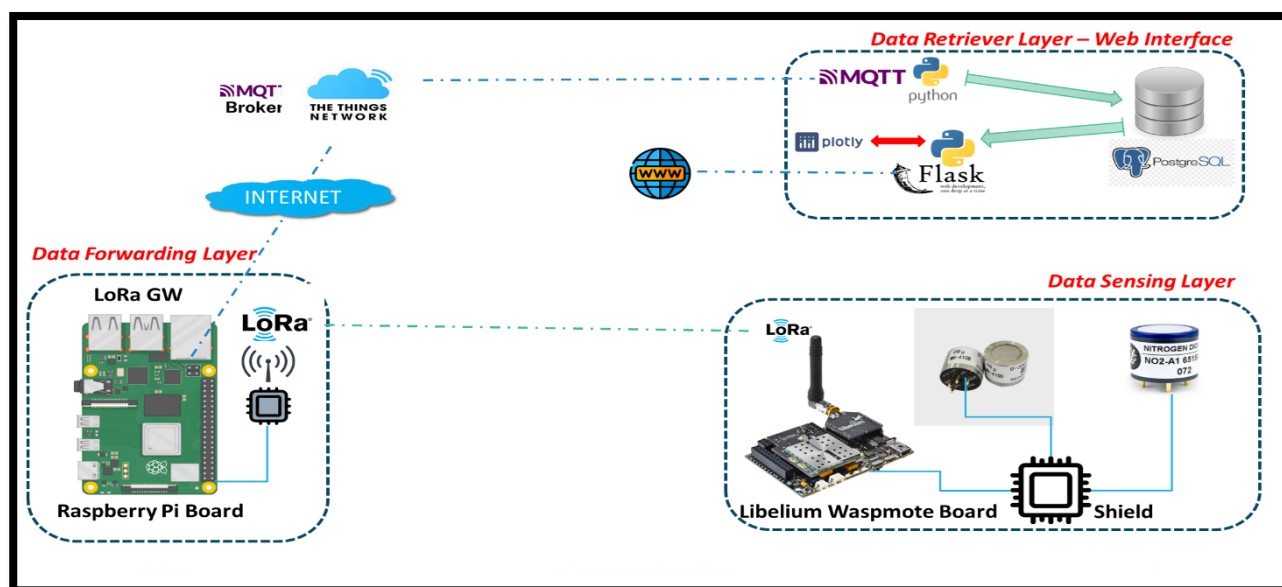
4

Ανάλυση Λογισμικών και εφαρμογών

Στο παρόν κεφάλαιο, θα παρουσιασθεί και θα αναλυθεί η υπηρεσία ανίχνευσης φωτιάς σε δασικές περιοχές και ειδικότερα, τα δομικά της στοιχεία που αφορούν τις εφαρμογές λογισμικού, τα οποία αποτελούνται από 2 λογισμικές διεργασίες (software processes). Τα υπόλοιπα δομικά στοιχεία του συστήματος μαζί με τις διαθέσιμες τεχνολογίες και frameworks που χρησιμοποιήθηκαν για την τελική υλοποίηση του προτεινόμενου συστήματος, εκκινώντας από τη πλακέτα Libelium, τους αισθητήρες, το LoRa GW, την υπηρεσία TNN και τέλος, τις δύο εφαρμογές λογισμικού. Η αρχιτεκτονική του συστήματος καθώς και τα επίπεδα της, απεικονίζονται στο Σχήμα 13.

Αναφορικά, με τις δύο εφαρμογές, η μια εφαρμογή ονομάζεται ‘MQTT App’ και εκτελεί το ρόλο της αίτησης δεδομένων από το TNN επίπεδο και της αποθήκευσης τους σε τοπική βάση δεδομένων ώστε να μπορούν να ανακτηθούν. Η δεύτερη εφαρμογή με το όνομα ‘Web Service Monitoring’, έχει ως στόχο τη παρουσίαση στο τελικό χρήστη ενός συνόλου ιστο-σελίδων με πληροφορίες για τα δεδομένα που συλλέγονται με μεθόδους οπτικοποίησης αλλά και στο να εμφανίσει τα όρια στα οποία τα δεδομένα αποτελούν μια πραγματική φωτιά στο χώρο ώστε ο χρήστης να ενημερωθεί και να προβεί σε σειρά κατάλληλων ενεργειών.

Ειδικότερα, οι τεχνολογίες για την εφαρμογή ‘MQTT App’, περιλαμβάνουν τη γλώσσα προγραμματισμού Python, τη χρήση βιβλιοθήκης του πρωτοκόλλου εφαρμογής, MQTT, και την εγκατάσταση και εκτέλεση βάσης δεδομένων, PostgreSQL, με κατάλληλους πίνακες. Ενώ, η εφαρμογή ‘Web Service Monitoring’, δρα ως μια υπηρεσία ιστού (web service), όπου ο χρήστης μέσω δικτύου μπορεί να συνδεθεί και να αιτηθεί διάφορες ιστοσελίδες. Η γλώσσα προγραμματισμού για την ανάπτυξη της εφαρμογής αυτής, είναι η Python, οι μέθοδοι δημιουργίας δια δραστικών διαγραμμάτων βασίζονται στο Python framework, Plotly, η ανάπτυξη των ιστοσελίδων βασίστηκε στις γλώσσες προγραμματισμού, JavaScript (JS) και HTML, ενώ τέλος, η βάση δεδομένων από την οποία αντλεί τα δεδομένα είναι τεχνολογίας PostgreSQL. Η βάση δεδομένων αποτελεί κοινή υλοποίηση και λειτουργία στο ίδιο υπολογιστικό περιβάλλον για λόγους ευκολίας, οπότε και η παραμετροποίηση της πραγματοποιήθηκε μια φορά.



Σχήμα 13 Αρχιτεκτονική του προτεινόμενου συστήματος - Επίπεδα

4.1 Βάση Δεδομένων Εφαρμογών

Για να μπορούν να αποθηκεύονται τα δεδομένα που λαμβάνονται από την υπηρεσία TNN μέσω της εφαρμογής 'MQTT APP', πρέπει να γίνει εγκατάσταση μιας ΒΔ. Η ΒΔ που έχει χρησιμοποιηθεί στο προτεινόμενο σύστημα, είναι η PostgreSQL, μια πολύ γνωστή τεχνολογία ΒΔ, η οποία διαθέτει υποστήριξη και για γεω-χωρικά (geospatial data) δεδομένα αλλά και για σχεσιακά δεδομένα (relational). Οι πίνακες που έχουν σχεδιασθεί και υλοποιηθεί για να υποστηριχθούν οι 2 εφαρμογές, είναι ο πίνακας 'Users' και ο πίνακας 'Measurements'. Αναλυτικά, η δομή των πινάκων, οι τύποι δεδομένων, τα πρωτεύον κλειδιά, μπορούν να απεικονισθούν στους παρακάτω πίνακες.

Ο πίνακας 'Users' ουσιαστικά αποθηκεύει τις πληροφορίες που απαιτούνται για κάθε χρήστη που ενδιαφέρεται να χρησιμοποιήσει την εφαρμογή. Τα πεδία του πίνακα, είναι ένα μοναδικό id, το email και ο κωδικός. Συνεπώς, όταν ο χρήστης πρόκειται να κάνει τις ενέργειες login, register, τότε πρέπει να γίνει ο απαραίτητος έλεγχος στο πίνακα αυτό. Η δομή του πίνακα φαίνεται στο Πίνακα 5. Αντίστοιχα, ο πίνακας 'Measurements', αποθηκεύει τις μετρήσεις που έρχονται από την υπηρεσία TNN ως records. Κάθε εγγραφή (record), αποτελείται από ένα μοναδικό timestamp της μέτρησης (χρονοσφραγίδα), την υγρασία, το διοξείδιο του άνθρακα, το άζωτο και το οξείδιο. Η δομή του πίνακα φαίνεται στο Πίνακα 6.

Οι ρυθμίσεις της ΒΔ, αφορούν τη πόρτα (5432 default), τον host (localhost), το όνομα της βάσης (postgres default), τον χρήστη της βάσης (postgres default) και έναν κωδικό που έχει ορισθεί.

Πίνακας 5: Πίνακας ΒΔ 'Users'

Users		
user_id (column 1)	user_email (column 2)	Password (column 3)
double precision	double precision	double precision

Πίνακας 6: Πίνακας ΒΔ 'Measurements'

Measurements				
timestamp (column 1)	co2 (column 2)	humidity (column 3)	o2 (column 4)	no2 (column 5)
Timestamp with zone	double precision	double precision	double precision	double precision

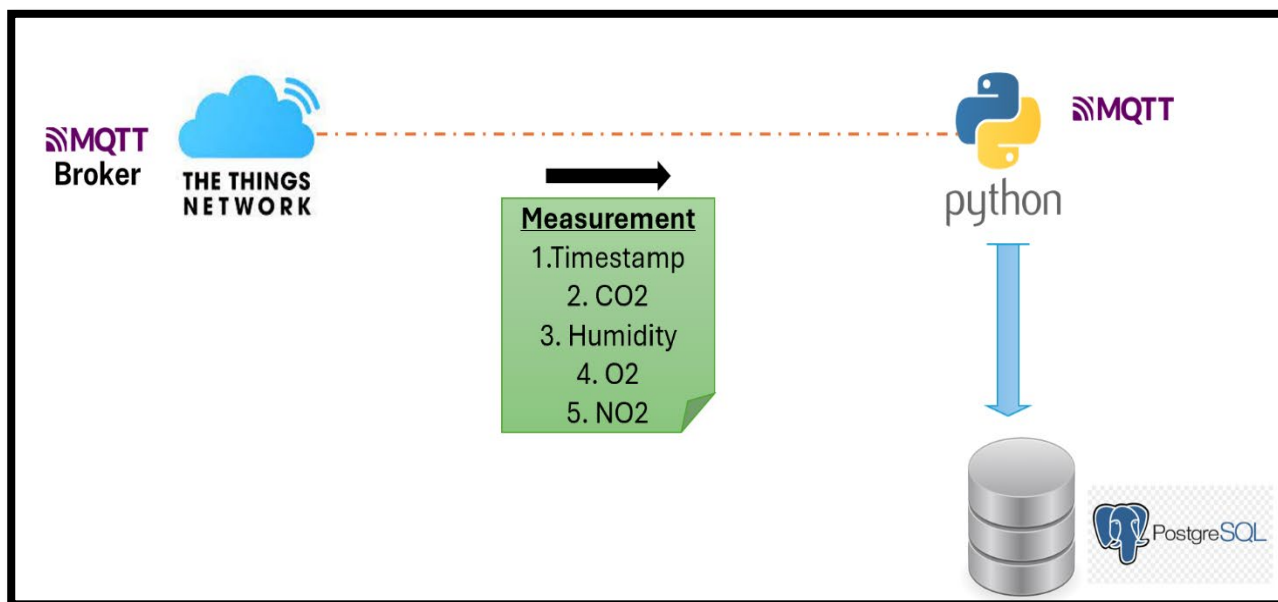
4.2 MQTT App

Η εφαρμογή αυτή, προσπαθεί να αιτηθεί τα δεδομένα από την υπηρεσία The Things Network. Αφού κάνει εγγραφή (subscribe) στο συγκεκριμένο topic, τότε η εφαρμογή περιμένει ατέρμονα για μηνύματα. Όταν ένα μήνυμα έρθει (από το Libelium board), μια ειδική μέθοδος callback ενεργοποιείται, και η εφαρμογή διαβάζει τα δεδομένα (υγρασία, διοξείδιο του άνθρακα, άζωτο και οξείδιο μαζί με τη χρονοσφραγίδα) και ακολούθως τα αποθηκεύει στη βάση δεδομένων σε κατάλληλο πίνακα 'Measurements'. Η κάθε μέτρηση όπως παρατηρείται στο Σχήμα 15 ενέχει τη χρονοσφραγίδα τη στιγμή της μέτρησης, και τις τέσσερες υπόλοιπες τιμές. Στο πλαίσιο των τεχνικών απαιτήσεων της εφαρμογής, η έκδοση Python διερμηνέα που απαιτείται, είναι v3.10 και υψηλότερη καθώς μια λίστα από modules για να εκτελεσθεί σωστά η εφαρμογή. Η εφαρμογή απαιτεί συγκεκριμένα Python modules για να παρέχει τη σωστή λειτουργικότητα. Τα modules με τις ακριβείς εκδόσεις τους απεικονίζονται στο Πίνακα 7.

Πίνακας 7: Εξαρτήσεις Python εφαρμογής MQTT App

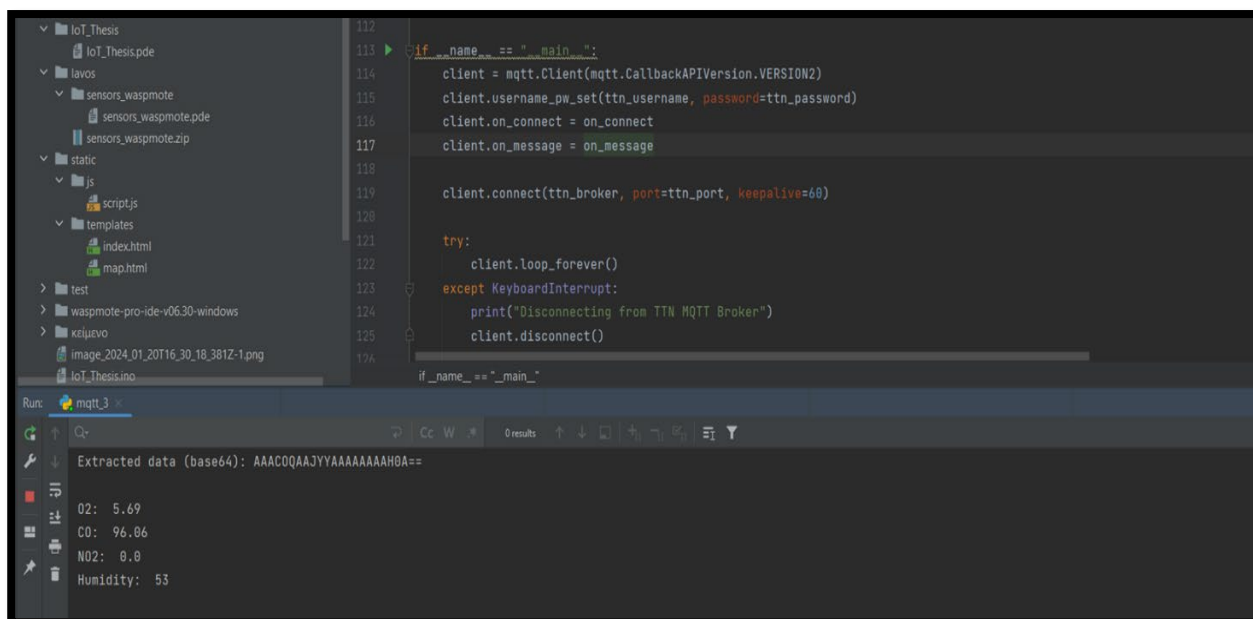
Python Modules Requirements	
certifi	2024.2.2
charset-normalizer	3.3.2
Idna	3.7
paho-mqtt	2.1.0
psycpg2	2.9.9
Requests	2.31.0
Urllib3	2.2.1

Το Σχήμα 14, απεικονίζει την αρχιτεκτονική της εφαρμογής. Η ροή δεδομένων ξεκινά από την υπηρεσία TNN, τα δεδομένα μέσω MQTT περνάνε στην εφαρμογή Python όπου εκεί με τη δημιουργία σύνδεσης με τη ΒΔ PostgreSQL, τα δεδομένα εισάγονται ως μια εγγραφή στο πίνακα ‘Measurements’.



Σχήμα 14 Αρχιτεκτονική εφαρμογής MQTT App

Στο Σχήμα 15 , παρατηρείται το αποτέλεσμα της εκτέλεσης της εφαρμογής. Το περιβάλλον είναι του προγράμματος, PyCharm Community Edition. Η εφαρμογή, λαμβάνει τις μετρικές όπως φαίνεται για τη συγκεκριμένη χρονική στιγμή.



Σχήμα 15 Εκτέλεση MQTT App εφαρμογής

4.3 Web Service Monitoring

Η εφαρμογή αυτή, προσπαθεί να ανακτήσει τα δεδομένα από τη ΒΔ, και να οπτικοποιήσει με βάση το framework Python Plotly ώστε να τα προσφέρει στο χρήστη προς ενημέρωση. Στη πραγματικότητα, η εφαρμογή αυτή δρα ως μια υπηρεσία ιστού (web service) όπου είναι προσβάσιμη μέσω του δικτύου και προσφέρει στο χρήστη ένα σύνολο από ιστοσελίδες. Ο χρήστης αφού πρώτα πραγματοποιήσει επιτυχώς την ενέργεια του login, έπειτα μπορεί να αιτηθεί διάφορες διαδικτυακές υπηρεσίες προς την εφαρμογή αυτή. Οι κυριότερες λειτουργίες της είναι σε μορφή HTTP URL όπως απεικονίζεται στο Πίνακα 8.

Πίνακας 8: HTTP URLs

HTTP-based services	
'/	Login Σελίδα

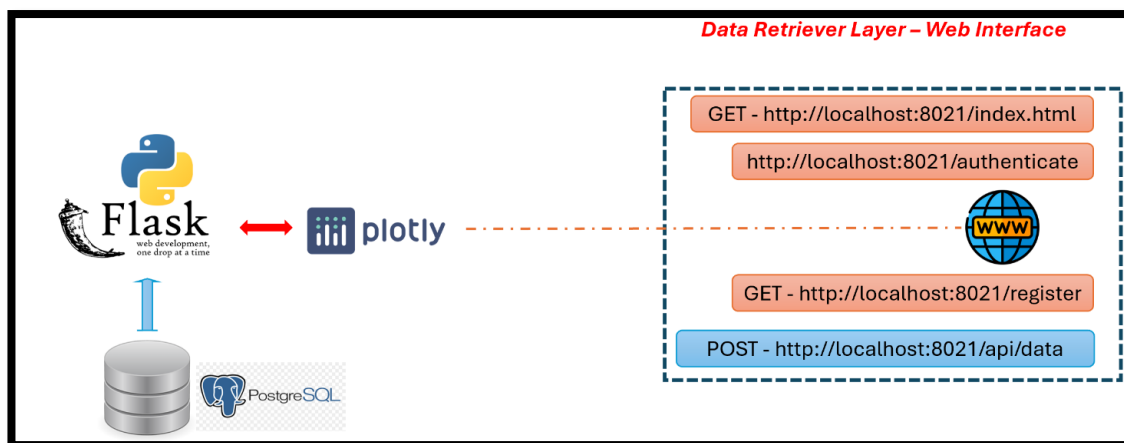
‘/menu’	Εμφάνιση κεντρικού μενού
‘/register’	Σελίδα Εγγραφής
‘/information’	Σελίδα πληροφοριών
‘/reset_pass_user’	Σελίδα reset κωδικού
‘/measurements’	Σελίδα με Plot για κάθε μετρική (current – 1 week before)
‘/streaming’	Σελίδα με live streaming plots για κάθε μετρική
‘/search’	Αναζήτηση μετρικών με βάση χρονικό διάστημα
‘\logout’	Αποσύνδεση από την εφαρμογή.

Στο πλαίσιο των τεχνικών απαιτήσεων της εφαρμογής, η έκδοση Python διερμηνέα που απαιτείται, είναι v3.12 και υψηλότερη καθώς μια λίστα από modules για να εκτελεσθεί σωστά η εφαρμογή. Η εφαρμογή απαιτεί συγκεκριμένα Python modules για να παρέχει τη σωστή λειτουργικότητα. Τα modules με τις ακριβείς εκδόσεις τους απεικονίζονται στο Πίνακα 9.

Πίνακας 9: Εξαρτήσεις εφαρμογής Environmental Monitoring

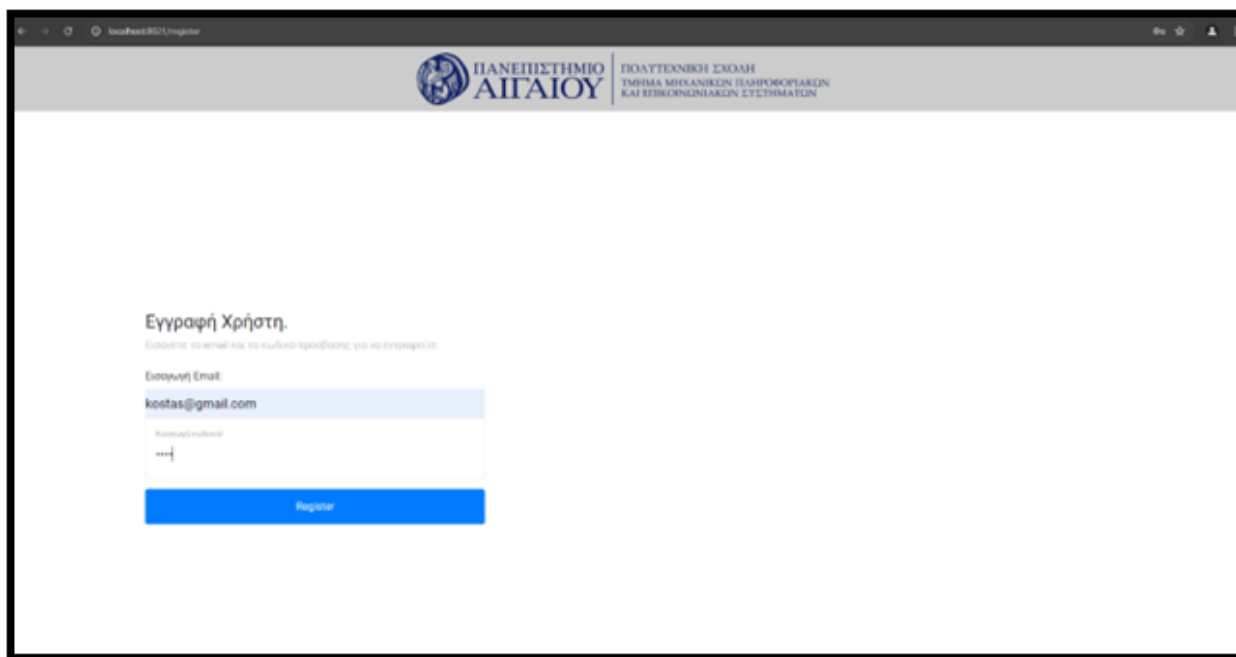
Python Modules Requirements	
Pandas	2.2.0
plotly	5.19.0
psycopg2	2.9.9
PyYAML	6.0.1
Flask	3.0.2
numpy	1.26.4

Στο Σχήμα 16, απεικονίζεται η αρχιτεκτονική της εφαρμογής. Το αριστερό μέρος, περιλαμβάνει την εφαρμογή ως web service, το διάβασμα από το πίνακα ‘Measurements’. Στο δεξιά μέρος της εικόνας, βρίσκεται ο χρήστης μέσω του προγράμματος περιήγησης του. Εκεί, με διάφορα HTTP αιτήματα, αιτείται την υπηρεσία που ενδιαφέρεται προς την εφαρμογή.



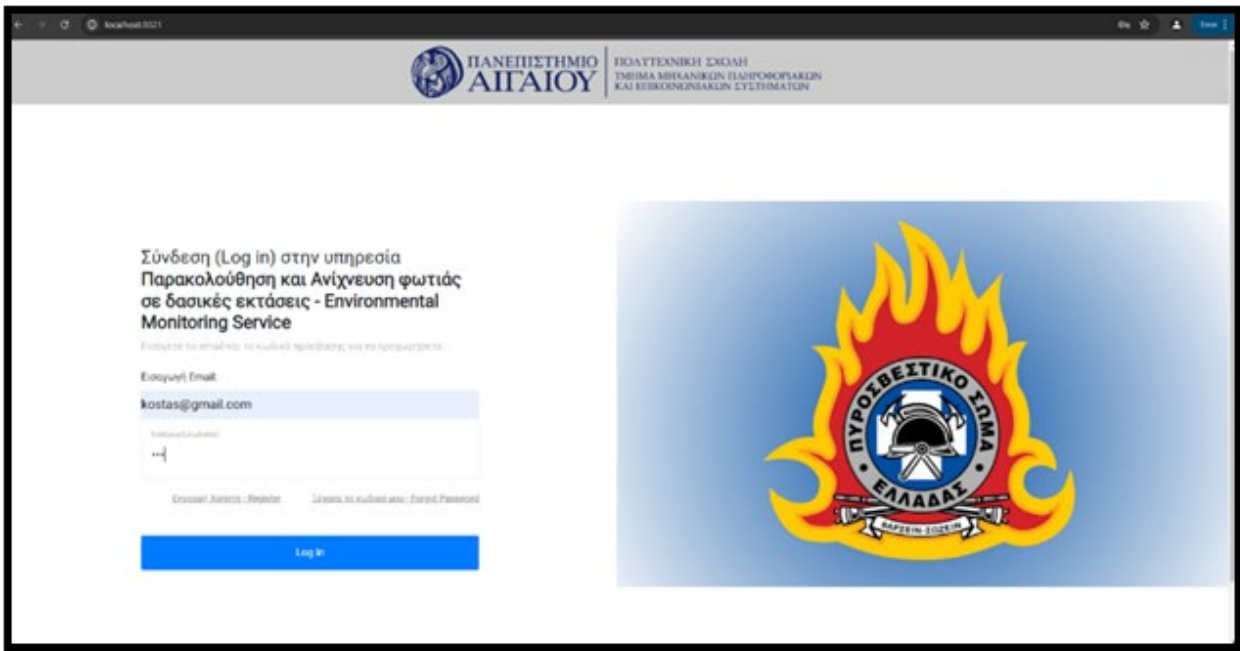
Σχήμα 16 Αρχιτεκτονική εφαρμογής Environmental Monitoring

Στην συνέχεια ακολουθούν μερικές οθόνες εκτέλεσης της εφαρμογής. Αρχικά, η πρώτη λειτουργία είναι του register. Ο χρήστης αφού εισάγει τα στοιχεία του (email, κωδικό), μπορεί να πατήσει το μπλε κουμπί της εγγραφής. Η διαδικασία, φαίνεται στο Σχήμα 17.



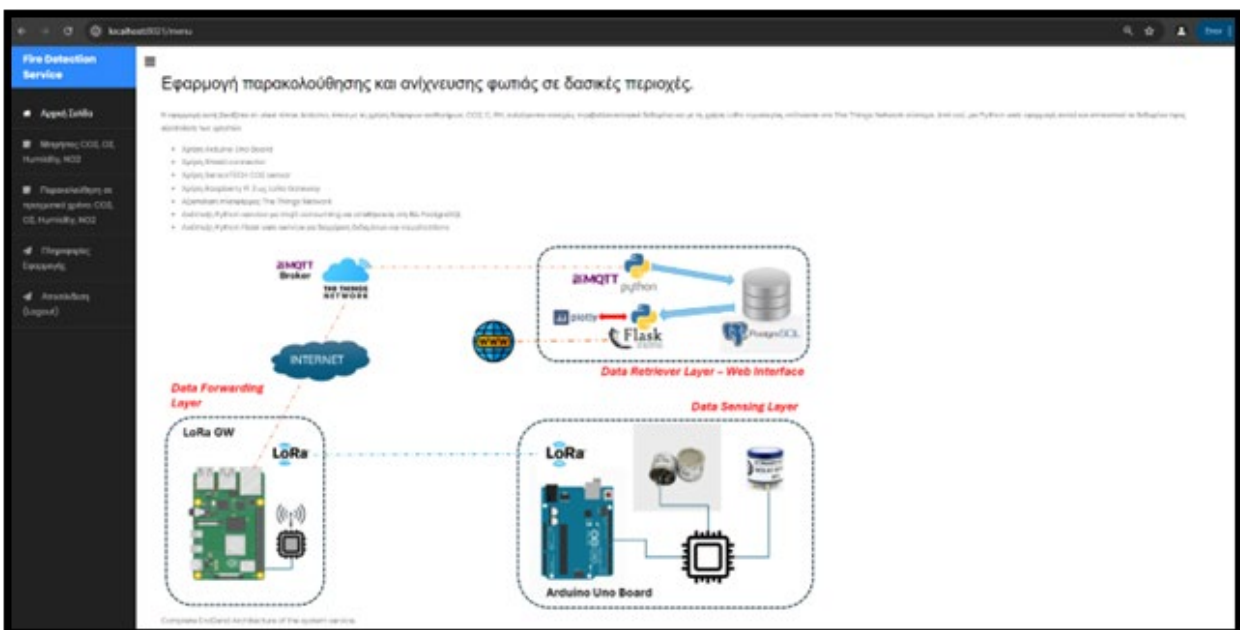
Σχήμα 17 Σελίδα εγγραφή χρήστη

Ακολούθως, μεταβαίνει στην σελίδα αυθεντικοποίησης, Login, όπως απεικονίζεται στο Σχήμα 18. Εδώ, θα εισάγει τα στοιχεία που είχε στην εγγραφή ώστε να μπορεί να δει τα διαγράμματα και να χρησιμοποιήσει τη πλατφόρμα.



Σχήμα 18 Σελίδα login

Έπειτα, θα εισέλθει στο κεντρικό μενού της εφαρμογής όπου μπορεί να επιλέξει τη λειτουργία που επιθυμεί, όπως απεικονίζεται στο Σχήμα 19.



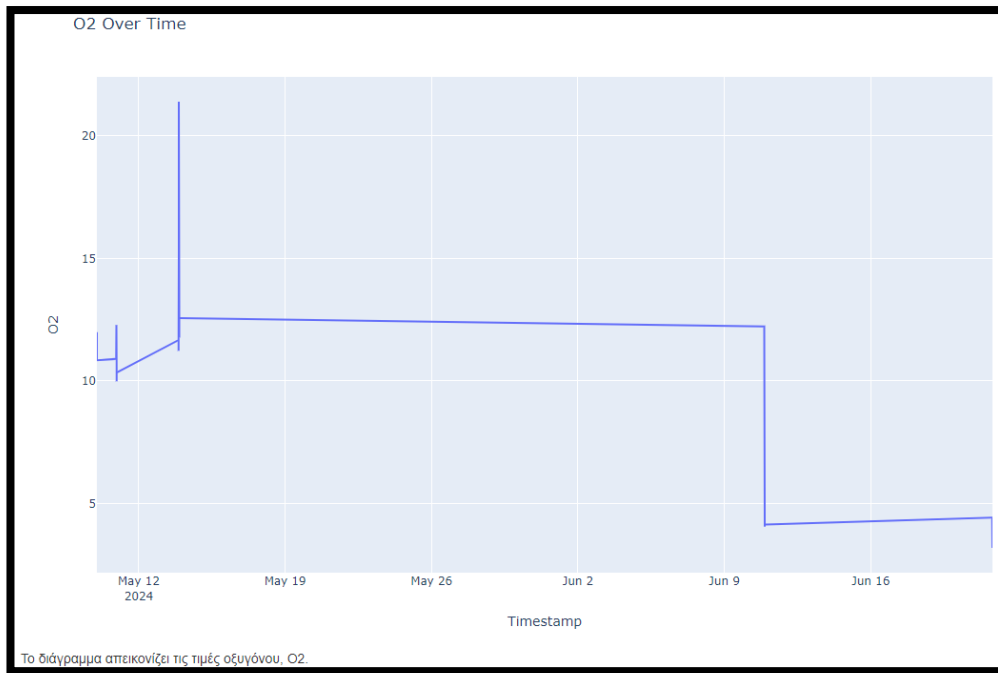
Σχήμα 19 Σελίδα κεντρικού μενού

4.4 Σενάρια Εκτέλεσης Εφαρμογής Web Monitoring

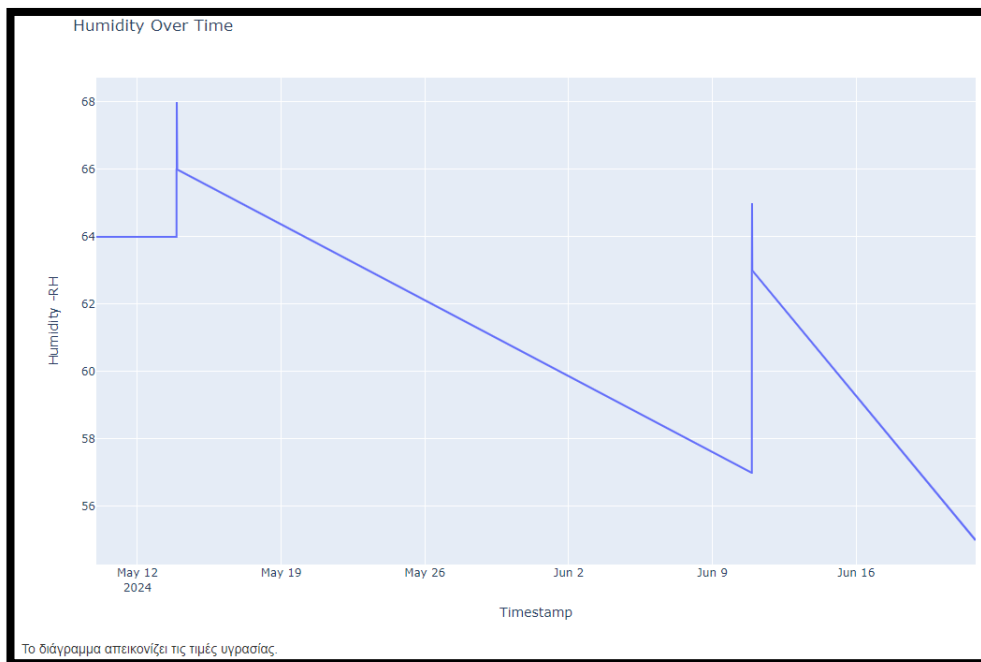
Οι μέθοδοι, ‘αναζήτηση’, εμφάνιση διαγραμμάτων και live streaming είναι μια από τις λειτουργίες που παρέχει η εφαρμογή σε περίπτωση που ο χρήστης επιθυμεί να δει μετρικές σε ένα συγκεκριμένο χρονικό διάστημα του παρελθόντος ή να δει ζωντανά τη μεταβολή των τιμών. Με την επιλογή ‘Μετρήσεις’, θα δει τα διαγράμματα των τεσσάρων μετρικών από τη τρέχουσα στιγμή μέχρι πριν ένα μήνα περίπου πίσω. Συγκεκριμένα, το Σχήμα 20 δείχνει με βάση το χρόνο τις τιμές του διοξειδίου του άνθρακα. Το Σχήμα 21 δείχνει με βάση το χρόνο τις τιμές του οξυγόνου, ενώ το Σχήμα 22, απεικονίζει τη μεταβολή της υγρασίας στο χρόνο. Το Σχήμα 23 δείχνει το άζωτο στο χρόνο.



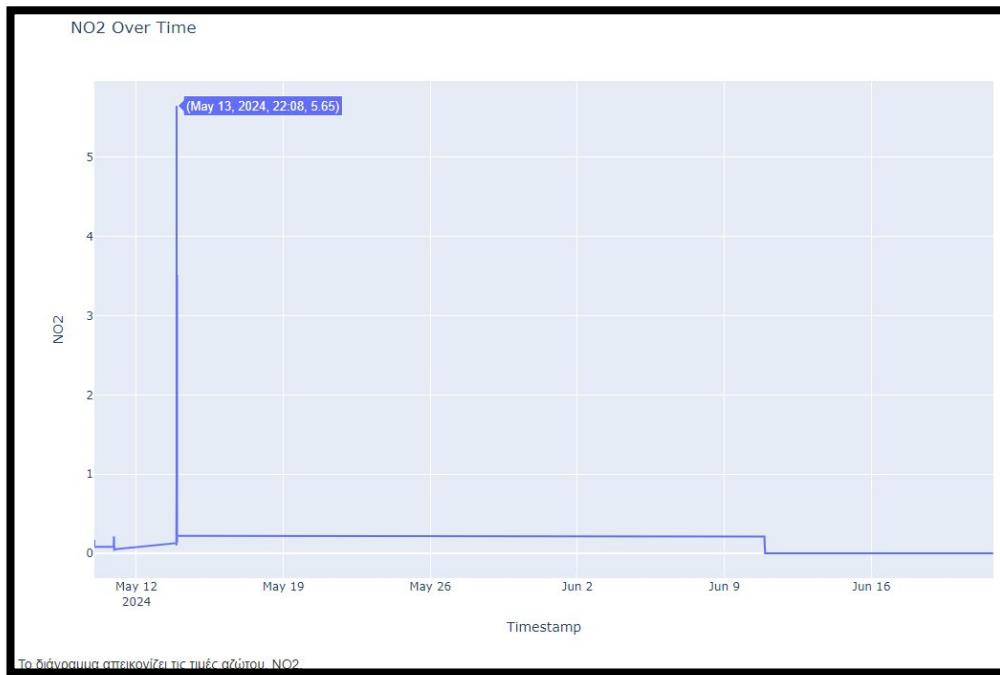
Σχήμα 20 Διάγραμμα CO2



Σχήμα 21 Διάγραμμα O2

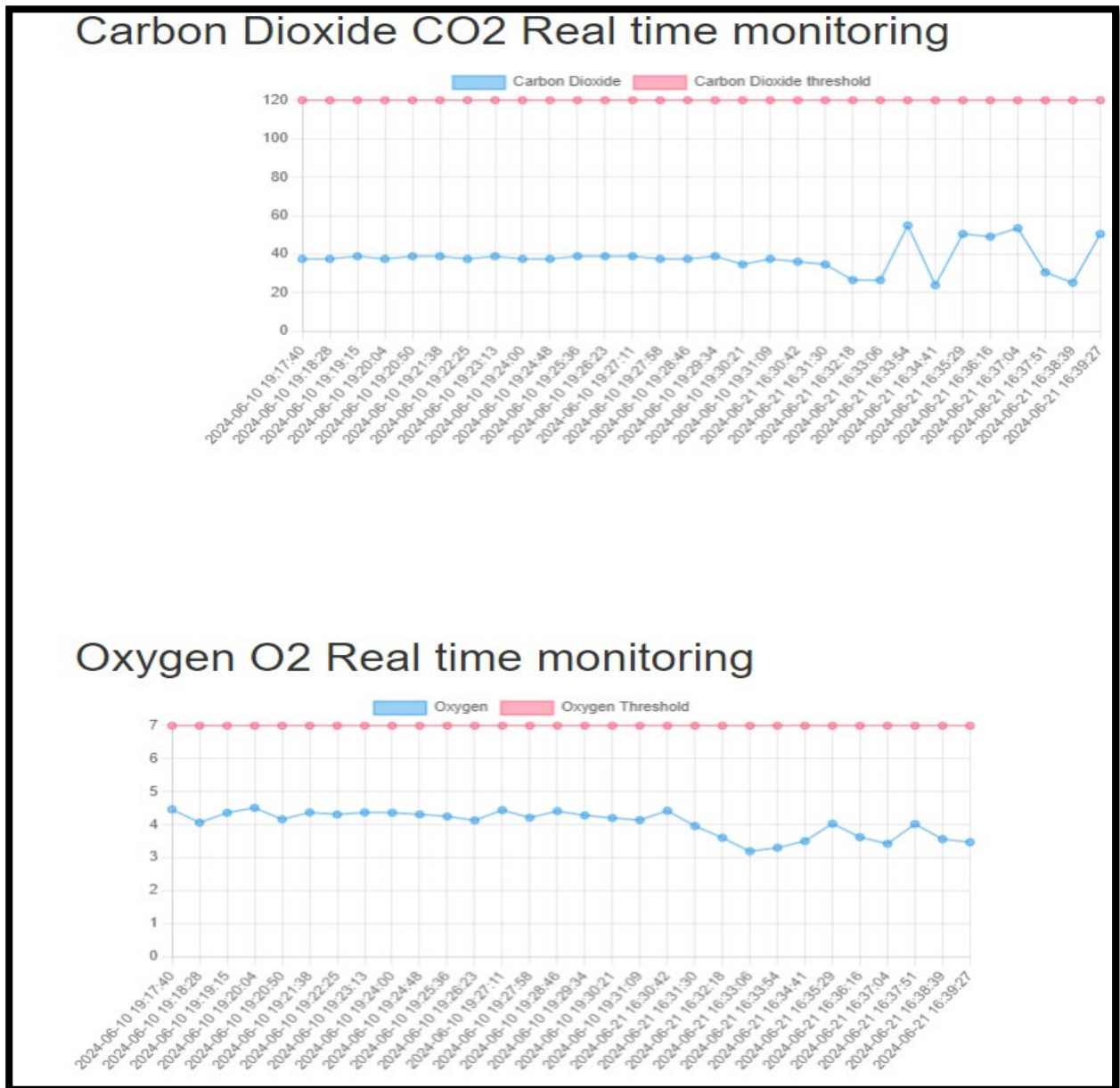


Σχήμα 22 Διάγραμμα Υγρασίας

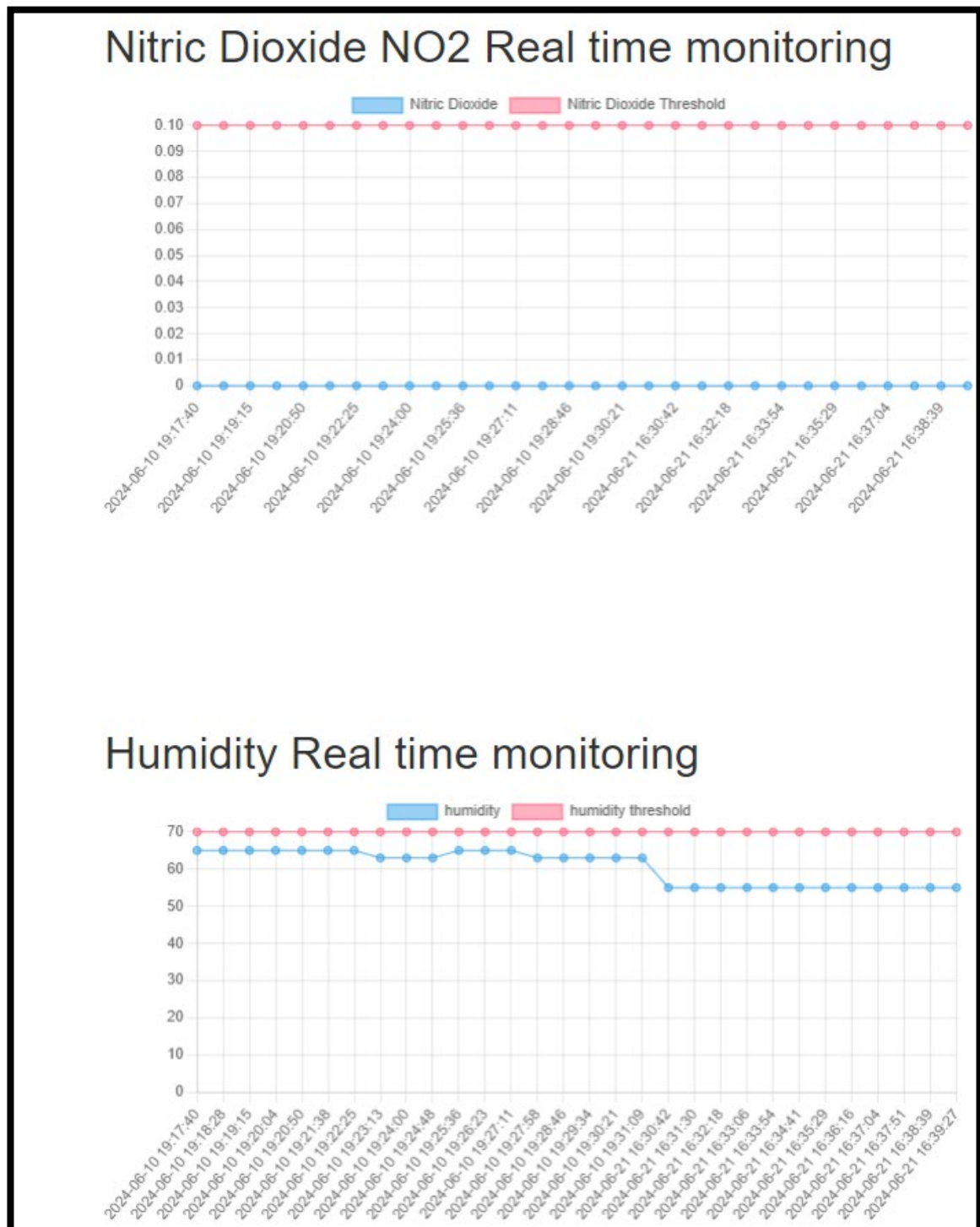


Σχήμα 23 Διάγραμμα NO2

Με την επιλογή ‘Παρακολούθηση σε πραγματικό χρόνο’, θα δει τα διαγράμματα των τεσσάρων μετρικών κάθε 2 δευτερόλεπτα με δυναμική ανανέωση. Τα διαγράμματα του πραγματικού χρόνου απεικονίζονται στο Σχήμα 24 και στο Σχήμα 25 αντίστοιχα. Κάθε δύο δευτερόλεπτα, οι τιμές ανανεώνονται δυναμικά και απεικονίζονται στα διαγράμματα πάνω στη μπλε καμπύλη.

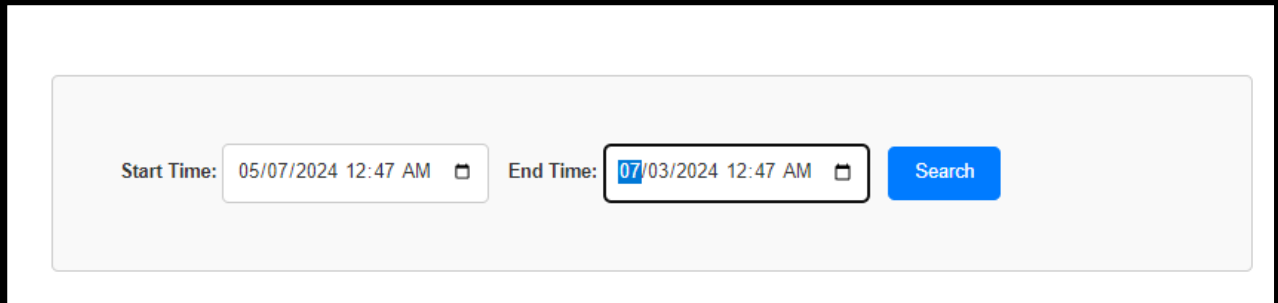


Σχήμα 24 Διαγράμματα real-time CO2 και O2



Σχήμα 25 Διαγράμματα real-time Υγρασίας και NO2

Με την επιλογή ‘Αναζήτηση’ σε συγκεκριμένο χρονικό διάστημα, ο χρήστης δίνει την ημερομηνία έναρξης και λήξης, και σε εκείνο το διάστημα, οι μετρικές εμφανίζονται. Τα δύο πεδία της χρονικής στιγμής έναρξης και χρονικής στιγμής λήξης, και η επιλογή των ημερομηνιών, απεικονίζονται στο Σχήμα 26. Για το συγκεκριμένο χρονικό διάστημα στο Σχήμα 27 φαίνονται οι τιμές του οξυγόνου και διοξειδίου του άνθρακα ενώ στο Σχήμα 28, οι τιμές υγρασίας και αζώτου παρατηρούνται.

The image shows a web interface for searching data. It features two input fields for time selection. The first field is labeled 'Start Time:' and contains the text '05/07/2024 12:47 AM' followed by a calendar icon. The second field is labeled 'End Time:' and contains the text '07/03/2024 12:47 AM' followed by a calendar icon. To the right of these fields is a blue button with the text 'Search' in white. The entire interface is enclosed in a light gray rounded rectangle, which is itself within a larger black-bordered frame.

Σχήμα 26 Menu αναζήτησης



Σχήμα 27 Διαγράμματα CO2, O2 σε συγκεκριμένο χρονικό διάστημα



Σχήμα 28 Διαγράμματα Υγρασίας και NO2 σε συγκεκριμένο χρονικό διάστημα,

5

Συμπεράσματα, Προκλήσεις και Μελλοντικές Κατευθύνσεις

5.1 Συμπεράσματα

Είναι δεδομένο ότι, τα υπάρχουσα IoT συστήματα που δραστηριοποιούνται στο χώρο της ανίχνευσης περιβαλλοντολογικών δεδομένων, όπως η ανίχνευση φωτιάς, πλημμύρας, συστήματα προειδοποίησης έκτακτων γεω-φυσικών φαινομένων, γίνονται ολοένα και πιο αναγκαία σε ένα περιβάλλον που καταπονείται από ποικίλες μεταβολές. Ειδικότερα, η ανίχνευση φωτιάς είναι ένα πεδίο κρίσιμης σημασίας τόσο για τη βιοποικιλότητα αλλά και για τα αστικά κέντρα διαβίωσης [16], [17], [18]. Τα κλασσικά συστήματα, ενέχουν ένα σύνολο αισθητήρων συνδεδεμένα σε μια πλακέτα υλικού, όπου η πλακέτα στέλνει τα δεδομένα σε ένα κεντρικό εξυπηρετητή για περαιτέρω επεξεργασία.

Το προτεινόμενο σύστημα-ιδέα της διπλωματικής εργασίας, αφενός περιλαμβάνει ποιοτικούς αισθητήρες με μεγάλη ακρίβεια sensing, αφετέρου, ακολουθεί μια προσέγγιση data-driven αξιοποιώντας την υπηρεσία The Things Network, κάνοντας τα συλλεγόμενα δεδομένα διαθέσιμα για οποιοδήποτε τρίτο σύστημα, εφαρμογή ή χρήστη επιθυμεί να τα διαβάσει. Επίσης, οι τεχνολογίες και τα frameworks είναι σύγχρονα εξασφαλίζοντας τη μέγιστη απόδοση αλλά και την ασφάλεια του συστήματος. Επιπρόσθετα, έχει υλοποιηθεί ο μηχανισμός του threshold, δηλαδή γίνεται συνεχώς παρακολούθηση σε πραγματικό χρόνο για τα όρια των τιμών έτσι ώστε η εμφάνιση φωτιάς να γίνει άμεσα αντιληπτή.

5.2 Υπάρχουσα Προβλήματα-Προκλήσεις

Στο χώρο της ανίχνευσης φωτιάς με βάση IoT συστήματα, υπάρχουν ακόμη ανοιχτά ζητήματα που χρίζουν λύσεων ή και διαφορετικών, πιο έξυπνων μεθόδων. Υπάρχουν περιπτώσεις στις οποίες οι έξυπνοι αισθητήρες παρουσιάζουν μια μικρή απόκλιση ή χάνουν τα δεδομένα. Συνεπώς, η παρακολούθηση της φωτιάς μπορεί να είναι άκυρη ή ψευδής. Σε αυτό το σημείο, απαιτείται ο σχεδιασμός ενός μηχανισμού ικανού να αντιλαμβάνεται αυτό το κενό και να το διορθώνει.

Επιπλέον, σε μια περιοχή παρακολούθησης, υπάρχουν πολλαπλά σημεία τα οποία αποτελούν δυνητικά σημεία εστίας φωτιάς. Γίνεται αντιληπτό ότι, απαιτείται και ένα μαζικό πλήθος έξυπνων αισθητήρων κατανεμημένων στο χώρο για τη πλήρη κάλυψη της παρακολούθησης. Μια προσέγγιση μαζικής ύπαρξης συσκευών, τείνει να αυξήσει το κόστος, τη πολυπλοκότητα στη διαχείριση αλλά και στη συνδεσιμότητα. Υπό αυτό το πρίσμα, η ερευνητική κοινότητα οφείλει να αναζητήσει, σχεδιάσει νέα πρωτόκολλα, τεχνολογίες για να αντιμετωπίσει τα παραπάνω θέματα με γνώμονα πάντα τη βιωσιμότητα των λύσεων αλλά και του πεδίου εφαρμογής.

5.3 Μελλοντική Εργασία

Τα τελευταία χρόνια, η τεχνητή νοημοσύνη ενσωματώνεται ολοένα σε περισσότερα IoT συστήματα, είτε βιομηχανικά, είτε περιβάλλοντα γεωργίας, ή περιβάλλοντος. Αποτελεί μια μελλοντική κατεύθυνση η εξερεύνηση μοντέλων ή μεθόδων τεχνητής νοημοσύνης (AI), και ειδικά ο εντοπισμός φωτιάς από συλλεγόμενες εικόνες έτσι ώστε να μην υπάρχει πλήρη εξάρτηση από τους αισθητήρες αλλά από ένα αυτόνομο ευφυή σύστημα που θα μπορεί να αποφασίζει γρήγορα και αξιόπιστα μέσα από εικόνες για την ανίχνευση φωτιάς [8], [19], [20].

Επίσης, ένα άλλο εξίσου σημαντικό σημείο που αποτελεί μελλοντική εργασία είναι, η αξιοποίηση και μελέτη της χρήσης μη-επανδρωμένων οχημάτων (UAVs) για εναέρια επιτήρηση με στόχο τη δυναμική παρακολούθηση σε πραγματικό χρόνο, την εύκολη μετακίνηση του σε διαφορετικά σημεία ενδιαφέροντος αλλά και τη λήψη εικόνων με στόχο την αναγνώριση της φωτιάς έτσι ώστε να μειωθεί ο αριθμός των ανεπτυγμένων αισθητήρων στο πεδίο αλλά και να γίνεται πιο γρήγορα η διαδικασία της ανίχνευσης [21].

Βιβλιογραφία

- [1] C. C. Sobin, “A Survey on Architecture, Protocols and Challenges in IoT,” *Wirel. Pers. Commun.*, vol. 112, no. 3, pp. 1383–1429, Jun. 2020, doi: 10.1007/s11277-020-07108-5.
- [2] D. Lymperis, “Σύλλογή IoT δεδομένων και σημασιολογικός εμπλουτισμός τους με εφαρμογή στην παρακολούθηση ποιότητας αέρα σε αστικό περιβάλλον”.
- [3] A. A. Alli and M. M. Alam, “The fog cloud of things: A survey on concepts, architecture, standards, tools, and applications,” *Internet Things*, vol. 9, p. 100177, Mar. 2020, doi: 10.1016/j.iot.2020.100177.
- [4] M. Mansour *et al.*, “Internet of Things: A Comprehensive Overview on Protocols, Architectures, Technologies, Simulation Tools, and Future Directions,” *Energies*, vol. 16, no. 8, p. 3465, Apr. 2023, doi: 10.3390/en16083465.
- [5] J. Vergara, J. Botero, and L. Fletscher, “A Comprehensive Survey on Resource Allocation Strategies in Fog/Cloud Environments,” *Sensors*, vol. 23, no. 9, p. 4413, Apr. 2023, doi: 10.3390/s23094413.
- [6] J. Jiang, Z. Li, Y. Tian, and N. Al-Nabhan, “A Review of Techniques and Methods for IoT Applications in Collaborative Cloud-Fog Environment,” *Secur. Commun. Netw.*, vol. 2020, pp. 1–15, Sep. 2020, doi: 10.1155/2020/8849181.
- [7] M. R. Rezaee, N. A. W. Abdul Hamid, M. Hussin, and Z. A. Zukarnain, “Fog Offloading and Task Management in IoT-Fog-Cloud Environment: Review of Algorithms, Networks, and SDN Application,” *IEEE Access*, vol. 12, pp. 39058–39080, 2024, doi: 10.1109/ACCESS.2024.3375368.
- [8] S. Ahmad, I. Shakeel, S. Mehruz, and J. Ahmad, “Deep learning models for cloud, edge, fog, and IoT computing paradigms: Survey, recent advances, and future directions,” *Comput. Sci. Rev.*, vol. 49, p. 100568, Aug. 2023, doi: 10.1016/j.cosrev.2023.100568.
- [9] “Computational Intelligence and Neuroscience - 2023 - Santhosh Kumar - A Comprehensive Survey on Machine Learning-Based.pdf.”
- [10] S. Balaji, K. Nathani, and R. Santhakumar, “IoT Technology, Applications and Challenges: A Contemporary Survey,” *Wirel. Pers. Commun.*, vol. 108, no. 1, pp. 363–388, Sep. 2019, doi: 10.1007/s11277-019-06407-w.
- [11] M. Jouhari, N. Saeed, M.-S. Alouini, and E. M. Amhoud, “A Survey on Scalable LoRaWAN for Massive IoT: Recent Advances, Potentials, and Challenges,” *IEEE Commun. Surv. Tutor.*, vol. 25, no. 3, pp. 1841–1876, 2023, doi: 10.1109/COMST.2023.3274934.
- [12] B. Shilpa, H. P. Gupta, R. K. Jha, and S. S. Hashmi, “LoRa interference issues and solution approaches in dense IoT networks: a review,” *Telecommun. Syst.*, Jul. 2024, doi: 10.1007/s11235-024-01192-9.
- [13] S. Iyer *et al.*, “A Survey on Semantic Communications for Intelligent Wireless Networks,” *Wirel. Pers. Commun.*, vol. 129, no. 1, pp. 569–611, Mar. 2023, doi: 10.1007/s11277-022-10111-7.

- [14]R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar, and B. Qureshi, “An Overview of IoT Sensor Data Processing, Fusion, and Analysis Techniques,” *Sensors*, vol. 20, no. 21, p. 6076, Oct. 2020, doi: 10.3390/s20216076.
- [15]Md. O. Qays, I. Ahmad, A. Abu-Siada, Md. L. Hossain, and F. Yasmin, “Key communication technologies, applications, protocols and future guides for IoT-assisted smart grid systems: A review,” *Energy Rep.*, vol. 9, pp. 2440–2452, Dec. 2023, doi: 10.1016/j.egyr.2023.01.085.
- [16]“thesis_Anastasiou_Nikolaos.pdf.”
- [17]“icsd16107,icsd16110 .pdf.”
- [18]S. F. Sulthana, C. T. A. Wise, C. V. Ravikumar, R. Anbazhagan, G. Idayachandran, and G. Pau, “Review Study on Recent Developments in Fire Sensing Methods,” *IEEE Access*, vol. 11, pp. 90269–90282, 2023, doi: 10.1109/ACCESS.2023.3306812.
- [19]A. Sairi, S. Labed, B. Miles, and A. Kout, “A review on early forest fire detection using IoT-enabled WSN,” in *2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAEECS)*, BLIDA, Algeria: IEEE, Mar. 2023, pp. 1–6. doi: 10.1109/ICAEECS56710.2023.10104887.
- [20]A. Secilmis, N. Aksu, F. A. Dael, I. Shayea, and A. A. El-Saleh, “Machine Learning-Based Fire Detection: A Comprehensive Review and Evaluation of Classification Models,” *JOIV Int. J. Inform. Vis.*, vol. 7, no. 3–2, p. 1982, Nov. 2023, doi: 10.30630/joiv.7.3-2.2332.
- [21]O. M. Bushnaq, A. Chaaban, and T. Y. Al-Naffouri, “The Role of UAV-IoT Networks in Future Wildfire Detection,” *IEEE Internet Things J.*, vol. 8, no. 23, pp. 16984–16999, Dec. 2021, doi: 10.1109/JIOT.2021.3077593.