

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΕΦΑΡΜΟΣΜΕΝΑ ΗΛΕΚΤΡΟΝΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Σύστημα καταγραφής ημερομηνίας, ώρας,  
θερμοκρασίας και υγρασίας σε εξωτερική μνήμη με  
δυνατότητα εμφάνισης των δεδομένων σε υπολογιστή.»



Του φοιτητή:  
Κούγια Παράσχου  
Αρ. Μητρώου: 51809Μ

Επιβλέπων  
Καζακόπουλος Αριστοτέλης  
Καθηγητής Δ.Π.Ε.

Ημερομηνία 10/10/2021

Τίτλος Δ.Ε.: Σύστημα καταγραφής ημερομηνίας, ώρας, θερμοκρασίας και υγρασίας σε εξωτερική μνήμη με δυνατότητα εμφάνισης των δεδομένων σε υπολογιστή

Κωδικός Δ.Ε.: 2009

Όνοματεπώνυμο φοιτητή: Κούγιας Παράσχος

Όνοματεπώνυμο εισηγητή: Καζακόπουλος Αριστοτέλης

Ημερομηνία ανάληψης Δ.Ε.: 3/4/2020

Ημερομηνία περάτωσης Δ.Ε.: 15/10/2021

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Μεταπτυχιακό Πρόγραμμα Σπουδών «Εφαρμοσμένα Ηλεκτρονικά Συστήματα» στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Κούγια Παράσχου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

*«Στην αγαθή Τύχη»*



## Πρόλογος

Η διπλωματική εργασία εκπονήθηκε στα πλαίσια του Μεταπτυχιακού προγράμματος στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος. Η επιλογή του θέματος έγινε με σκοπό την έρευνα, εξοικείωση και την πρακτική εξάσκηση του φοιτητή με τις πιο σύγχρονες τεχνολογίες και εργαλεία στον τομέα του “διαδικτύου των πραγμάτων” (IoT). Το θέμα που επιλέχθηκε έχει τίτλο «Σύστημα καταγραφής ημερομηνίας, ώρας, θερμοκρασίας και υγρασίας σε εξωτερική μνήμη με δυνατότητα εμφάνισης των δεδομένων σε υπολογιστή» και το τελικό αποτέλεσμα αποσκοπεί στην δημιουργία ενός πρωτότυπου μιας συσκευής του “διαδικτύου των πραγμάτων “. Το πρωτότυπο της συσκευής αποσκοπεί να παρουσιάσει ένα βασικό πυρήνα συσκευής IoT που θα μπορεί να επεκταθεί και να χρησιμοποιηθεί στο μέλλον σε σύγχρονες εφαρμογές του “διαδικτύου των πραγμάτων”.

## Περίληψη

Ο τίτλος της διπλωματικής εργασίας είναι «Σύστημα καταγραφής ημερομηνίας, ώρας, θερμοκρασίας και υγρασίας σε εξωτερική μνήμη με δυνατότητα εμφάνισης των δεδομένων σε υπολογιστή». Οι στόχοι της εργασίας αυτής είναι η δημιουργία μιας συσκευής του “διαδικτύου των πραγμάτων” που καταγράφει μετρήσεις θερμοκρασίας, υγρασίας και της ώρας που συλλέχθηκαν. Τα δεδομένα αυτά αποθηκεύονται σε τρία στάδια εξωτερική μνήμης, αρχικά σε κάρτα μνήμης στην συσκευή IoT, σε τοπικό επίπεδο σε μία βάση χρονοσειρών και τέλος αποθηκεύονται σε μια υποδομή “υπολογιστικού νέφους” (cloud). Η απεικόνιση των δεδομένων προς το χρήστη γίνεται σε τοπικό επίπεδο μέσω του λογισμικού Grafana και με απομακρυσμένη πρόσβαση είτε στο διαδικτυακό περιβάλλον (web browser) είτε στο κινητό μέσω της cloud πλατφόρμας ThingSpeak.

Η δομή της εργασίας ξεκίνησε με την οριοθέτηση των παραπάνω στόχων και έπειτα έγινε έρευνα για την εξέλιξη στους τομείς του “διαδικτύου των πραγμάτων” και συναφείς τομείς όπως η βιομηχανική εξέλιξη 4ης γενιάς (Industry 4.0), το υπολογιστικό “νέφος” (cloud) και τα “έξυπνα” σπίτια (smart home). Τα αποτελέσματα της έρευνας αυτής οδήγησαν την σχεδίαση της αρχιτεκτονικής και των επιμέρους υποσυστημάτων του πρωτότυπου της συσκευής IoT. Στην συνέχεια ακολούθησε η υλοποίηση του πρωτότυπου με την χρήση του μικροεπεξεργαστή ESP32 για την συλλογή των μετρήσεων θερμοκρασίας, υγρασίας, εντοπισμού κίνησης και μετρικών για την κατάσταση του υπολογιστικού συστήματος IoT όπως το μέγεθος διαθέσιμης μνήμης-εκτέλεσης, μέγεθος διαθέσιμης μνήμης στην κάρτα, κ.α.). Το πρωτότυπο της συσκευής IoT συνδέεται στο προσωπικό δίκτυο (οικιακό ή επαγγελματικό) όπου αποστέλλει τις μετρήσεις σε έναν εξυπηρετητή συλλογής μετρήσεων ο οποίος είναι υπεύθυνος για την συλλογή, αποθήκευση, προβολή σε τοπικό επίπεδο και προώθηση των δεδομένων προς το υπολογιστικό σύννεφο.

Οι στόχοι της εργασίας αυτής επιτεύχθηκαν διότι η πρωτότυπη συσκευή IoT, ο εξυπηρετητής συλλογής δεδομένων, η απεικόνιση των δεδομένων σε διαφορετικά επίπεδα χρήστη και η χρήση της εφαρμογής υπολογιστικού νέφους (cloud) αποτελεί την βάση μιας εφαρμογής από άκρη-σε-άκρη (end-to-end) που μπορεί να χρησιμοποιηθεί τόσο σε σύγχρονες εφαρμογές του διαδικτύου των πραγμάτων όσο και σε βιομηχανικές εφαρμογές της βιομηχανικής εξέλιξης 4ης γενιάς καθώς και σε εφαρμογές “έξυπνου” σπιτιού.

«IoT device prototype for recording date, time, temperature and humidity in external memory with the ability to display data on a computer. »

«Paraschos Kougias»

## **Abstract**

The title of the dissertation is "IoT device prototype for recording date, time, temperature and humidity in external memory with the ability to display data on a computer." The objectives of this work are to create a device of the "internet of things" that records measurements of temperature, humidity and current timestamp. This data is stored in three stages of external memory, initially on a memory card on the IoT device, locally on a time series database and finally stored in a "cloud computing" infrastructure. The visualization of the data to the user is done locally through the Grafana software and with remote access either through the web application or through the mobile app connected to the cloud platform ThingSpeak. The structure of the work started with the definition of the objectives above and continued with a research on the leading edge areas of the "internet of things" and closely related areas such as the Industry 4.0, the cloud computing and the Smart Houses. The results of this research led to the design principles of the architecture and the individual subsystems of the IoT device prototype. Next step was the implementation of the prototype using the ESP32 microprocessor for the collection of temperature, humidity, motion detection and metrics for the status of the IoT device such as the size of available execution memory or the size of available memory on the card. The prototype of the IoT device connects to the personal network where it sends the measurements to a measurement collection server which is responsible for collecting, storing, viewing locally and forwarding the data to the cloud. The objectives of this work was fulfilled because the prototype of the IoT device, the data collection server, the display of data at different user levels and the use of the cloud application is an end-to-end application which can be used both either in modern applications of the Internet of Things or in industrial applications in Industry 4.0 or as well in Smart Home applications.

## **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω τους καθηγητές, φίλους, συναδέλφους και την οικογένεια μου που με την καθοδήγηση, την υπομονή και την βοήθεια τους συνέβαλαν καθοριστικά στην ολοκλήρωση της διπλωματικής μου εργασίας.

# Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract.....	vii
Ευχαριστίες.....	viii
Περιεχόμενα.....	ix
Κατάλογος Σχημάτων.....	xi
Κατάλογος Πινάκων.....	xiii
Συνομογραφίες.....	xv
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Σκοπός της διπλωματικής.....	1
1.2 Σχετική έρευνα και πρότυπα σύγχρονων τεχνολογιών.....	1
1.2.1 Διαδίκτυο των πραγμάτων - Internet of Things (IoT).....	2
1.2.2 Υπολογιστικό “σύννεφο” - Cloud computing.....	3
1.2.3 Βιομηχανία 4ης γενιάς - Industry 4.0.....	5
1.2.4 “Εξυπνα” σπίτια - Smart homes.....	6
1.3 Αποτελέσματα Έρευνας.....	11
Κεφάλαιο 2ο: Σχεδίαση αρχιτεκτονικής - Σύστημα IoT.....	13
2.1 Εισαγωγή.....	13
2.2 Συνολικό διάγραμμα αρχιτεκτονικής.....	14
2.3 Ανάλυση επιμέρους υποσυστημάτων.....	14
2.3.1 Υλικό.....	15
2.3.2 Λογισμικό.....	18
Κεφάλαιο 3ο: Υλοποίηση πρωτότυπου - Σύστημα IoT.....	22
3.1 Εισαγωγή.....	22
3.2 Συνολικό διάγραμμα αρχιτεκτονικής υλοποίησης.....	22
3.3 Περιγραφή υλοποίησης των επιμέρους υποσυστημάτων.....	23
3.3.1 Υλικό.....	23
3.3.2 Λογισμικό.....	34
Κεφάλαιο 4ο: Αποτελέσματα.....	41
4.1 Μεταφορά μετρήσεων από άκρη-σε-άκρη.....	41
4.2 Σημεία αποθήκευσης σε εξωτερική μνήμη.....	44
4.3 Διεπαφή χρήστη.....	44

4.3.1 Τοπικός χρήστης.....	45
4.3.2 Απομακρυσμένος χρήστης.....	45
Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης.....	47
5.1 Συμπεράσματα.....	47
5.1.1 Σε γενικό επίπεδο.....	47
5.1.2 Σε τεχνικό επίπεδο.....	48
5.2 Προτάσεις βελτίωσης.....	48
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	51
ΠΑΡΑΡΤΗΜΑ Α : Κώδικας συσκευής IoT.....	53
ΠΑΡΑΡΤΗΜΑ Β - Κώδικας εξυπηρετητή.....	58
ΠΑΡΑΡΤΗΜΑ C - Κώδικας προώθησης και αποθήκευσης των μετρήσεων.....	60
ΠΑΡΑΡΤΗΜΑ D - Αρχείο ρυθμίσεων.....	63
ΠΑΡΑΡΤΗΜΑ Ε - Πρότυπη φόρμα μετρήσεων JSON.....	64
ΠΑΡΑΡΤΗΜΑ F -Απαιτήσεις πακέτων λογισμικού για τον εξυπηρετητή Python-Flask.....	65

## Κατάλογος Σχημάτων

Εικόνα 1-1 - Διαδίκτυο των πραγμάτων.....	2
Εικόνα 1-2 - Τα βασικά χαρακτηριστικά του Υπολογιστικού “νέφους” (cloud ) [9].....	3
Εικόνα 1-3 - Πηγαίνοντας προς την Βιομηχανία της 4ης γενιάς [15][20].....	5
Εικόνα 1-4 - Εκτίμηση της αύξησης των συνδεδεμένων συσκευών IoT στο περιβάλλον του "έξυπνου" σπιτιού για το έτος 2022.....	11
Εικόνα 2-1 - Σχεδίαση - Μπλοκ διάγραμμα της αρχιτεκτονικής μιας πρωτότυπης εφαρμογής IoT με πυρήνα την συσκευή IoT .....	13
Εικόνα 2-2 - Αριστερά: Απεικονίζεται η αρχιτεκτονική κατά την σχεδίαση. Δεξιά: Απεικονίζεται η στοίβα του λογισμικού (Software Stack) με τις διαφορετικές λειτουργίες αντιστοιχισμένες στα 5 επίπεδα της εφαρμογής IoT. Με γκρι και πορτοκαλί χρώμα απεικονίζεται το λογισμικό χρήσης όπως βιβλιοθήκες λογισμικού ενώ με μπλε, πράσινο και πορτοκαλί χρώμα περιγράφεται το λογισμικό που αναπτύχθηκε.....	15
Εικόνα 3-1 - Η αρχιτεκτονική της τελικής υλοποίησης του συστήματος IoT σε σχέση με την αρχιτεκτονική σχεδίασης. Αριστερά: Απεικονίζονται τα εικονίδια των τελικών επιλογών του υλικού. Δεξιά: Απεικονίζεται η στοίβα του λογισμικού με τα εικονίδια των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη των επιμέρους υποσυστημάτων της εφαρμογής IoT.....	22
Εικόνα 3-2 - Αισθητήρας DHT11.....	24
Εικόνα 3-3 - Αισθητήρας DHT11 ενσωματωμένος στην πλακέτα της συσκευής IoT.....	24
Εικόνα 3-4 - Αισθητήρας Εντοπισμού Κίνησης (PIR).....	25
Εικόνα 3-5 - Προφίλ αισθητήρα Εντοπισμού Κίνησης (PIR) ενσωματωμένο στην συσκευή IoT.....	25
Εικόνα 3-6 - Εσωτερική πίσω όψη του αισθητήρα Εντοπισμού Κίνησης (PIR) στην συσκευή IoT...	25
Εικόνα 3-7 - Μητρική πλακέτα Wemos D1 R32 που χρησιμοποιήθηκε για την ανάπτυξη της συσκευής IoT.....	26
Εικόνα 3-8 -Μητρική πλακέτα Espressif ESP32-WROOM-D32 που χρησιμοποιήθηκε τόσο για την ανάπτυξη αλλά κυρίως για την τελική της ενσωμάτωση στο πρωτότυπο της συσκευής IoT.....	27
Εικόνα 3-9 - Συνδεσμολογία του αισθητήρα DHT και PIR κατά την ανάπτυξη του κώδικα για την συλλογή των μετρήσεων με την μητρική πλακέτα Wemos D1 R32.....	27
Εικόνα 3-10 - Συνδεσμολογία της μητρικής πλακέτας Espressif ESP32-WROOM-D32 με την κάρτα μνήμης κατά την ανάπτυξη του κώδικα αποθήκευσης των δεδομένων των μετρήσεων στην εξωτερική κάρτα μνήμης.....	28
Εικόνα 3-11 - Συναρμολόγηση και κόλληση της τελικής πλακέτας του πρωτότυπου συστήματος IoT. Σε αυτή την εικόνα φαίνεται η πάνω πλευρά της πλακέτας με την μητρική του επεξεργαστή ESP32, τους αισθητήρες και την κάρτα μνήμης.....	28
Εικόνα 3-12 - Κόλληση της κάτω πλευράς των συνδέσεων της τελικής μητρικής πλακέτας του συστήματος IoT.....	29
Εικόνα 3-13 - 3D σχέδιο του κουτιού της τελικής όψης του συστήματος IoT σε σχεδιαστικό πρόγραμμα CAD.....	29
Εικόνα 3-14 - 3D εκτύπωση του τελικού κουτιού του πρωτότυπου υλοποίησης του συστήματος IoT.30	
Εικόνα 3-15 - Τοποθέτηση της τελικής πλακέτας στη βάση του κουτιού της συσκευής IoT και σύνδεση των εσωτερικών συνδέσεων των αισθητήρων και της τροφοδοσίας.....	30
Εικόνα 3-16 - Εγκατάσταση και συνδεσμολογία του αισθητήρα εντοπισμού κίνησης (PIR) με το πάνω μέρος της συσκευής.....	31
Εικόνα 3-17 - Τροφοδότηση και δοκιμές λειτουργίας της συσκευής IoT. ....	31

Εικόνα 3-18 - Πλάγια όψη της συσκευή IoT. Οι κυψελωτές τρύπες στο πλάι του κουτιού της συσκευή έχουν σκοπό την άμεση επαφή του αισθητήρα υγρασίας και θερμοκρασίας με τον αέρα του περιβάλλοντα χώρου. ....	32
Εικόνα 3-19 - Μητρική πλακέτα Raspberry Pi. Χρησιμοποιήθηκε ως υπολογιστής για να φιλοξενήσει τον εξυπηρετητή συλλογής, αποθήκευσης και αποστολής των μετρήσεων.....	32
Εικόνα 3-20 - Οικιακός δρομολογητής (router). Αποτελεί την πύλη για την αποστολή δεδομένων από την συσκευή IoT προς το υπολογιστικό “σύννεφο”.....	33
Εικόνα 3-21 - Φορητή συσκευή (κινητό Android) με εγκατεστημένη την εφαρμογή ThingShow για την απομακρυσμένη πρόσβαση τα δεδομένα των μετρήσεων μέσω της εφαρμογής υπολογιστικού “νέφους” ThingSpeak.....	33
Εικόνα 3-22 - Απεικόνιση της βάσης δεδομένων χρονοσειρών InfluxDB μέσα από το λογισμικό Grafana ως διαθέσιμη πηγή δεδομένων.....	37
Εικόνα 3-23 - Απεικόνιση δεδομένων με το λογισμικό Grafana σε επίπεδο τοπικού χρήστη. Στιγμιότυπο των μετρήσεων των δύο τελευταίων ημερών φιλτραρισμένων με την τεχνική του “κινητού μέσου όρου” με ένα εύρος μετρήσεων 30 δείγματα.....	38
Εικόνα 3-24 - Ρυθμίσεις κλειδιών αυθεντικοποίησης στην εφαρμογή ThingShow για την ανάγνωση των δεδομένων των μετρήσεων από την προγραμματιστική διεπαφή της εφαρμογής του “νέφους” ThinkSpeak.....	39
Εικόνα 3-25 - Επιτυχής σύνδεση, ανάγνωση και απεικόνιση των δεδομένων των μετρήσεων μέσω της εφαρμογής ThingShow.....	39
Εικόνα 3-26 - Γενική εικόνα από τον φυλλομετρητή του καναλιού συλλογής δεδομένων στην διαδικτυακή πύλη της εφαρμογής υπολογιστικού νέφους ThingSpeak.....	40
Εικόνα 4-1 - 3/4 όψη του πρωτότυπου της συσκευής IoT.....	41
Εικόνα 4-2 - Πλάγια όψη του πρωτότυπου της συσκευής IoT.....	41
Εικόνα 4-3 - Το πρωτότυπο της συσκευής IoT και το RaspberryPi που φιλοξενεί των εξυπηρετητή συλλογής δεδομένων στο ακρογωνιαίο δίκτυο.....	41
Εικόνα 4-4 - Απεικόνιση του δείγματος. Η τιμή και ο χρόνος είναι ελάχιστα διαφορετικοί γιατί η απεικόνιση των δεδομένων είχε προ-επεξεργαστεί με βάση τον αλγόριθμο του μέσου όρου ανα πέντε δείγματα που διατίθεται από το λογισμικό Grafana για την εξομάλυνση των άκρων της κυματομορφής. ....	42
Εικόνα 4-5 - Απεικόνιση του δείγματος υγρασίας στο φυλλομετρητή μέσα από την εφαρμογή ThingSpeak. Η τιμή και ο χρόνος είναι ελάχιστα διαφορετική γιατί η απεικόνιση των δεδομένων είχε προ-επεξεργαστεί με βάση τον αλγόριθμο του μέσου όρου ανα 15 δείγματα που διατίθεται από το λογισμικό ThingSpeak για την εξομάλυνση των άκρων της κυματομορφής.....	43
Εικόνα 4-6 - Απεικόνιση του δείγματος υγρασίας στο κινητό μέσα από την εφαρμογή ThingShow...43	43
Εικόνα 4-7 - Απεικόνιση των μετρήσεων της υγρασίας και της θερμοκρασίας στον τοπικό χρήστη μέσω του λογισμικού Grafana.....	45
Εικόνα 4-8 - Διεπαφή απομακρυσμένου χρήστη μέσω του φυλλομετρητή.....	46
Εικόνα 4-9 - Διεπαφή απομακρυσμένου χρήστη μέσω του κινητού.....	46

## Κατάλογος Πινάκων

Πίνακας 1 - Συντομογραφίες.....	xv
Πίνακας 1-1 - Περιγραφή των συσκευών "Ενεργειακές πύλες πληροφοριών" για τα "έξυπνα" σπίτια.	6
Πίνακας 1-2 - Περιγραφή των μετρητών φορτίων για τα "έξυπνα" σπίτια.....	7
Πίνακας 1-3 - Περιγραφή των έξυπνων οικιακών συσκευών για τα "έξυπνα" σπίτια.....	7
Πίνακας 1-4 - Περιγραφή των έξυπνων θερμοστατών για τα "έξυπνα" σπίτια.....	8
Πίνακας 1-5 - Περιγραφή του έξυπνου θερμοσίφωνα για τα "έξυπνα" σπίτια.....	9
Πίνακας 1-6 - Περιγραφή των έξυπνων κόμβων μετρήσεων για τα "έξυπνα" σπίτια.....	9
Πίνακας 1-7 - Περιγραφή των έξυπνων πριζών για τα "έξυπνα" σπίτια.....	10
Πίνακας 1-8 - Περιγραφή έξυπνων φώτων για τα "έξυπνα" σπίτια.....	10
Πίνακας 2-1 - Περιγραφή της λειτουργικότητας των επιμέρους υποσυστημάτων κατά την σχεδίαση σε επίπεδο "Υλικού"(Hardware).....	15
Πίνακας 2-2 - Περιγραφή της λειτουργικότητας των επιμέρους υποσυστημάτων κατά την σχεδίαση σε επίπεδο λογισμικού (software).....	18
Πίνακας 3-1 - Περιγραφή των αισθητήρων που χρησιμοποιήθηκαν στο πρωτότυπο της συσκευής IoT.23	
Πίνακας 3-2 - Περιγραφή και εικόνες των σταδίων ανάπτυξης της μητρικής πλακέτας της συσκευής IoT.....	27
Πίνακας 3-3 - Περιγραφή και απεικόνιση του υλικού που χρησιμοποιήθηκε για την υλοποίηση του επιπέδου του "ακρογωνιαίου" δικτύου(Edge network).....	32
Πίνακας 3-4 - Φορητή συσκευή - "Έξυπνο" κινητό τηλέφωνο με λογισμικό Android.....	33
Πίνακας 3-5 - Μορφή της δομής δεδομένων JSON για την οργανωμένη συλλογή, αποθήκευση και αποστολή των δεδομένων των μετρήσεων.....	35
Πίνακας 3-6 - Σημεία πρόσβασης του εξυπηρετητή συλλογής δεδομένων μέσω του πρωτοκόλλου HTTP.....	35
Πίνακας 3-7 - Συνάρτηση της Python κλάσης DataCollectionServer για την αποθήκευση των μετρήσεων στην βάση χρονοσειρών InfluxDB.....	36
Πίνακας 3-8 - Συνάρτηση της Python κλάσης DataCollectionServer για την αποστολή των μετρήσεων στην εφαρμογή υπολογιστικού νέφους ThingSpeak.....	38
Πίνακας 3-9 - Ρυθμίσεις και προβολή των δεδομένων των μετρήσεων σε κινητό τηλέφωνο μέσω της εφαρμογής ThingShow που επικοινωνεί με την εφαρμογή υπολογιστικού "νέφους" ThingSpeak.....	39
Πίνακας 4-1 - Το πρωτότυπο της συσκευής IoT ως αποτέλεσμα αυτής της διπλωματικής εργασίας..	41
Πίνακας 4-2 - Τα στάδια ακολουθεί μια μέτρηση υγρασίας από την συλλογή της στην συσκευή IoT μέχρι την αποθήκευση της στο υπολογιστικό σύννεφο και την απεικόνιση της στο χρήστη.....	42
Πίνακας 4-3 - Απεικόνιση ενός τμήματος των μετρήσεων που καταγράφονται στην κάρτα μνήμης της συσκευής IoT.....	44
Πίνακας 4-4 - Διεπαφή του απομακρυσμένου χρήστη για την προβολή των μετρήσεων.Στα αριστερά η εικόνα στο φυλλομετρητή και στα δεξιά η εικόνα στο κινητό.....	45
Πίνακας 5 - Πηγαίος κώδικας σε C/C++ για την υλοποίηση του λογισμικού του μικροεπεξεργαστή ESP32. Ο κώδικας αναπτύχθηκε σε περιβάλλον Arduino IDE.....	53
Πίνακας 6 - Πηγαίος κώδικας σε Python για την υλοποίηση του λογισμικού του εξυπηρετητή με το πακέτο Flask. Ο εξυπηρετητής είναι υπεύθυνος για την συλλογή, διαχείριση, αποθήκευση και αποστολή των δεδομένων των μετρήσεων.....	58
Πίνακας 7 - Πηγαίος κώδικας σε Python για την υλοποίηση της κλάσης του εξυπηρετητή που του δίνει δυνατότητες για την διαχείρισης, αποθήκευσης και αποστολή των δεδομένων των μετρήσεων..	60

Πίνακας 8 - Αρχείο διαχείρισης των ρυθμίσεων του λογισμικού του εξυπηρετητή για την εύκολη αρχικοποίηση, διαχείριση και συντήρηση των διαφορετικών ρυθμίσεων κάθε πακέτου λογισμικού στην τελική εφαρμογή IoT.....	63
Πίνακας 9 - Παράδειγμα της μορφής του αρχείου JSON. Αυτή η δομή δεδομένων χρησιμοποιείται για την αποστολή και αποθήκευση των δεδομένων των μετρήσεων.....	64
Πίνακας 10 - Οι εκδόσεις των πακέτων της Python 3 που χρειάζονται για την εκτέλεση του εξυπηρετητή συλλογής δεδομένων της εφαρμογής IoT.....	65

## Συντομογραφίες

Πίνακας 1 - Συντομογραφίες

#	Συντομογραφία	Ολογράφως
1	Δ.Ε.	Διπλωματική Εργασία
2	Δ.Ι.Π.Α.Ε.	Διεθνές Πανεπιστήμιο Ελλάδος
3	Π.Ε.	Πτυχιακή Εργασία
4	ΔτΠ - IoT	Διαδίκτυο των Πραγμάτων - Internet of Things - IoT
5	NIST	Αμερικανικό Ινστιτούτο Προτύπων και Τεχνολογίας - US National Institute of Standards and Technology (NIST)
6	SaaS	Λογισμικό ως υπηρεσία - Software as a Service
7	PaaS	Πλατφόρμα ως υπηρεσία - Platform as a Service
8	IaaS	Υποδομή ως υπηρεσία - Infrastructure as a Service
9	ΤΠΕ	Τεχνολογίας πληροφοριών και επικοινωνιών
10	JRC	Ευρωπαϊκή Ένωση Επιστημονικής Έρευνας (JRC)
11	SD	Κάρτα μνήμης - SD card
12	Router	Δρομολογητής
13	Υλικό	Με τον όρο “Υλικό” αναφερόμαστε στο hardware της συσκευής IoT.
14	Λογισμικό	Με τον όρο “λογισμικό” αναφερόμαστε σε όλα τα κομμάτια που αποτελούνται από πηγαίο κώδικα τόσο για την συσκευή IoT όσο και για άλλες βιβλιοθήκες ή εφαρμογές.
15	Arduino Uno	Ο Arduino Uno είναι μια μητρική πλακέτα ανοικτού κώδικα πάνω στην οποία σχεδιάστηκε μια από τις μητρικές πλακέτες του μικροεπεξεργαστή EPS32 που χρησιμοποιήθηκε κατά την ανάπτυξη της συνδεσμολογίας και του κώδικα της συσκευής IoT.
16	Arduino IDE	Το Arduino IDE (Integrated Development Environment) είναι ένα περιβάλλον ανάπτυξης πηγαίου κώδικα για μητρικές πλακέτες συμβατές με αυτό. Μια από αυτές τις πλακέτες είναι και οι μητρικές πλακέτες του

		μικροεπεξεργαστή EPS32.
17	Ακρογωνιαίο Δίκτυο	Ο όρος “ακρογωνιαίο” δίκτυο (Edge network) χρησιμοποιείται για να περιγράψει υπολογιστικά συστήματα που μεσολαβούν κατά την σύνδεση συσκευών IoT στο υπολογιστικό νέφος.
18	Προσωπικό δίκτυο	Προσωπικό δίκτυο σημαίνει ένα δίκτυο το οποίο είναι ιδιωτικό και έχει ελεγχόμενη πρόσβαση. Ένα οικιακό δίκτυο θεωρείται προσωπικό(private network).
19	Δημόσιο δίκτυο	Δημόσιο δίκτυο σημαίνει ένα δίκτυο το οποίο είναι δημόσια προσβάσιμο συνήθως χωρίς ελεγχόμενη πρόσβαση. Το διαδίκτυο και δημόσια προσβάσιμοι εξυπηρετητές ανήκουν στο δημόσιο δίκτυο.Η επικοινωνία προσωπικών δικτύων είναι συνήθως προσβάσιμη μέσα από το δημόσιο δίκτυο (public network).

## Κεφάλαιο 1ο: Εισαγωγή

### 1.1 Σκοπός της διπλωματικής

Τα τελευταία χρόνια, οι εξελίξεις στην τεχνολογία σε συνδυασμό με την ανάπτυξη των επεξεργαστών, του διαδικτύου, και με την αλληλεπίδραση διαφόρων συσκευών και συστημάτων, επηρεάζουν διάφορους τομείς της οικονομίας και της καθημερινής ζωής της σύγχρονης κοινωνίας. Χαρακτηριστικοί τομείς εξέλιξης αποτελούν το υπολογιστικό “σύννεφο” (cloud), το διαδίκτυο των πραγμάτων (IoT), η βιομηχανία 4ης γενιάς και τα “έξυπνα” σπίτια.

Σκοπός της διπλωματικής αυτής είναι,

1. Η ανάπτυξη ενός συστήματος του διαδικτύου των πραγμάτων που καταγράφει μετρήσεις από αισθητήρες θερμοκρασίας και υγρασίας καθώς και την ημερομηνία και ώρα που συλλέχθηκαν.
2. Οι μετρήσεις αυτές αποθηκεύονται σε εξωτερική μνήμη σε τρία διαφορετικά επίπεδα. Πρώτον, σε κάρτα μνήμης στο σύστημα του διαδικτύου των πραγμάτων (IoT). Δεύτερον, σε μια βάση δεδομένων του ακρογωνιαίου συστήματος (Edge network) που αποτελεί πύλη για την αποστολή στο υπολογιστικό “σύννεφο” (Cloud). Τρίτον, τα δεδομένα αποθηκεύονται και στην μνήμη της εφαρμογής του υπολογιστικού “σύννεφου”.
3. Η εμφάνιση των δεδομένων των μετρήσεων στο χρήστη μέσω φιλικών διεπαφών με την βοήθεια εφαρμογών διαδικτύου ή κινητού τηλεφώνου.
4. Τελευταίος αλλά εξίσου σημαντικό στόχος της διπλωματικής είναι η εξοικείωση και πρακτική ενασχόληση του φοιτητή με σύγχρονες τεχνολογίες και τεχνικές τόσο σε ερευνητικό επίπεδο όσο και στο επίπεδο της σύγχρονης αγοράς-εργασίας.

Αυτή η διπλωματική εργασία περιγράφει την σχεδίαση και υλοποίηση σε πρωτότυπο ενός συστήματος συλλογής δεδομένων στο “διαδίκτυο των πραγμάτων”. Ένα τέτοιο σύστημα μπορεί να αποτελέσει τον πυρήνα πληθώρας εφαρμογών που συνδυάζουν τις παρακάτω σύγχρονες τεχνολογίες:

1. Διαδίκτυο των πραγμάτων - Internet of Things
2. Υπολογιστικό σύννεφο - Cloud
3. Βιομηχανία 4ης γενιάς - Industry 4.0
4. “Έξυπνα” σπίτια - Smart homes

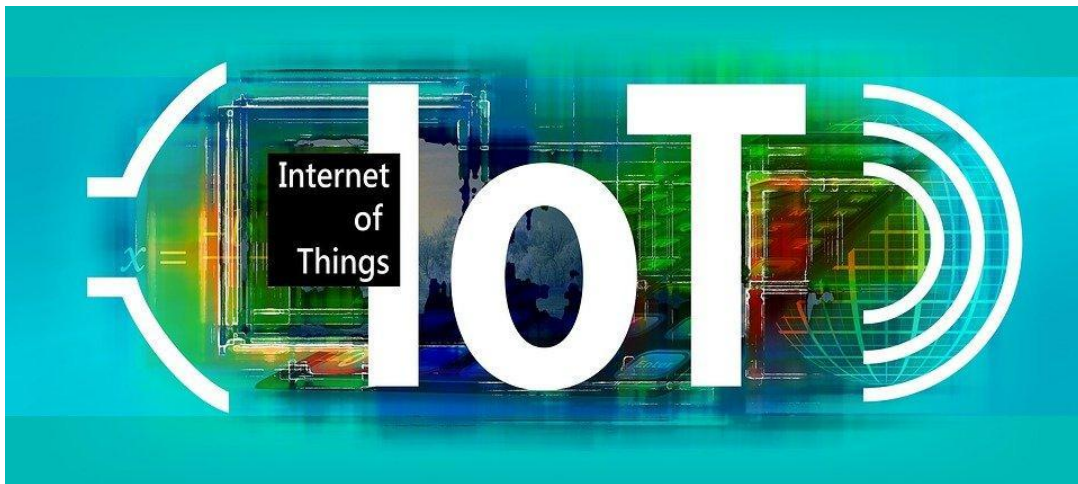
### 1.2 Σχετική έρευνα και πρότυπα σύγχρονων τεχνολογιών

Στην ενότητα αυτή παρουσιάζεται η έρευνα, που έγινε στα πλαίσια αυτής της διπλωματικής εργασίας, για τις πιο σύγχρονες τεχνολογίες και την μελλοντική τους κατεύθυνση. Η έρευνα είχε σκοπό τον εντοπισμό των θεμελιωδών χαρακτηριστικών και απαιτήσεων των σύγχρονων πεδίων ερευνών και εφαρμογών με σκοπό την επιλογή των σχεδιαστικών χαρακτηριστικών ενός σύγχρονου πυρήνα ενός συστήματος του “διαδικτύου των πραγμάτων”.

Στην έρευνα αυτή έμφαση έχει δοθεί στις τεχνολογίες του υπολογιστικού “σύννεφου” (cloud), του διαδικτύου των πραγμάτων (IoT), της βιομηχανίας 4ης γενιάς και τα “έξυπνα” σπίτια. Στην συνέχεια δίνονται οι ορισμοί των τεχνολογιών αυτών και μερικών εφαρμογών τους μέσα από ερευνητικές εργασίες και σύγχρονα πρότυπα τεχνολογιών όπως ορίζονται από οργανισμούς προτυποποίησης όπως είναι η Ευρωπαϊκή Ένωση Επιστημονικής Έρευνας (JRC) [21] και ο Αμερικανικός οργανισμός προτυποποίησης τεχνολογίας (NIST) [9], [10].

### 1.2.1 Διαδίκτυο των πραγμάτων - Internet of Things (IoT)

Το “Διαδίκτυο των Πραγμάτων” (“Internet of Things” – IoT) είναι μια συλλογή από ενσωματωμένα συστήματα με ηλεκτρονικά, λογισμικά, αισθητήρες, ενεργοποιητές τα οποία είναι συνδεδεμένα με μικροελεγκτές [26]. Οι μικροελεγκτές παρέχουν της βασική λειτουργικότητα για την επεξεργασία, προσωρινή αποθήκευση και αποστολή των δεδομένων με τη χρήση του διαδικτύου. Το διαδίκτυο και διάφορα άλλα πρωτόκολλα επικοινωνίας χρησιμοποιούνται με σκοπό τη συλλογή και ανταλλαγή δεδομένων μεταξύ των διαφόρων συσκευών αλλά και από/προς τους χρήστες . Οι συσκευές IoT διαθέτουν πληθώρα αισθητήρων, υπολογιστικής ισχύος και διάφορα πρωτόκολλα επικοινωνίας που τις επιτρέπουν να αναπτυχθούν σε πολλά περιβάλλοντα ανάλογα της απαιτήσεως της εκάστοτε εφαρμογής.



Εικόνα 1-1 - Διαδίκτυο των πραγμάτων

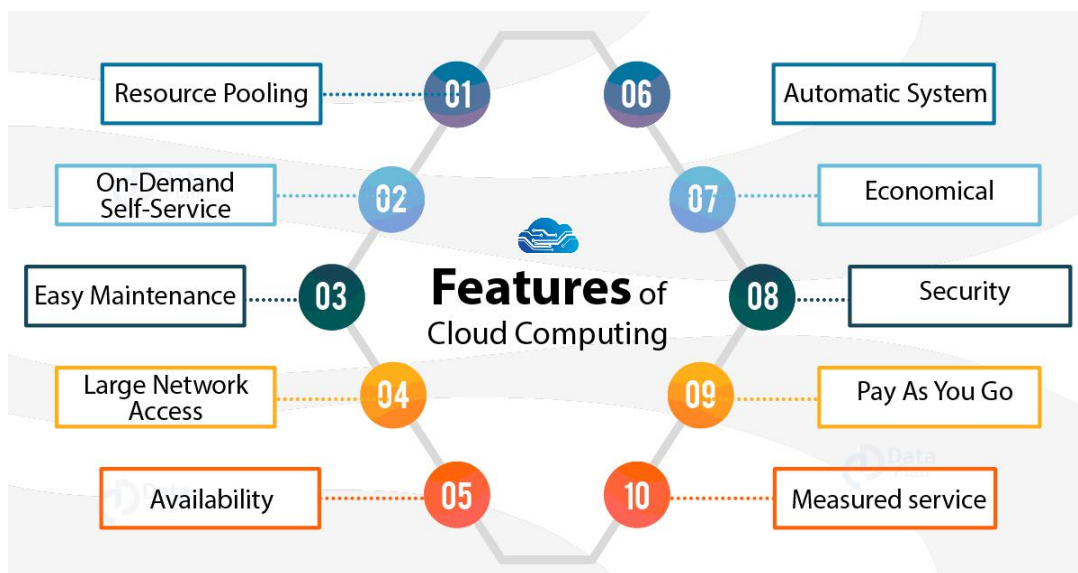
Ο ορισμός των «πραγμάτων» από την οπτική του IoT είναι πολύ ευρύς και περιλαμβάνει μια ποικιλία φυσικών στοιχείων. Αυτά περιλαμβάνουν προσωπικά αντικείμενα που μεταφέρει ο καθένας μας στην καθημερινότητα του, όπως έξυπνα τηλέφωνα, tablet και ψηφιακές φωτογραφικές μηχανές. Περιλαμβάνει επίσης στοιχεία στο περιβάλλον του ανθρώπου (π.χ. σπίτι, όχημα ή εργασία), στις βιομηχανίες (π.χ. μηχανές, μοτέρ, ρομπότ) καθώς και πράγματα εφοδιασμένα με ετικέτες (π.χ. RFID) που συνδέονται μέσω συσκευής πύλης (π.χ. ένα έξυπνο τηλέφωνο). Στα επόμενα χρόνια, η εξέλιξη αυτή των “πραγμάτων” του διαδικτύου θα οδηγήσει σε ένα τεράστιο αριθμό συσκευών που θα συνδεθεί με το διαδίκτυο. Το καθένα από τα αυτά θα παρέχει δεδομένα και πληροφορίες και μερικά από αυτά, ακόμη και υπηρεσίες [25]. Ουσιαστικά, θα δημιουργηθεί ένα οικοσύστημα IoT που θα αποτελείται από “έξυπνες” συσκευές που χρησιμοποιούν ενσωματωμένους επεξεργαστές, αισθητήρες και υλικό επικοινωνίας για τη συλλογή, αποστολή και επεξεργασία δεδομένων που αποκτούν από το περιβάλλον τους. Τα δεδομένα που συλλέγουν προωθούνται μέσω μιας πύλης IoT ή ακρογωνιαίας συσκευής υπολογιστικού “νέφους”, όπου τα δεδομένα αποστέλλονται είτε στο υπολογιστικό “σύννεφο” για την περαιτέρω ανάλυση ή αναλύονται τοπικά. Μερικές φορές, αυτές οι συσκευές επικοινωνούν με άλλες IoT συσκευές και παίρνουν αποφάσεις με βάση τις πληροφορίες που λαμβάνουν μεταξύ τους. Σε τέτοια συστήματα, οι συσκευές πραγματοποιούν το μεγαλύτερο μέρος της εργασίας χωρίς ανθρώπινη παρέμβαση, παρόλο που οι άνθρωποι μπορούν να αλληλεπιδρούν με τις συσκευές.

Ο όρος «Διαδίκτυο των Πραγμάτων» και στα Αγγλικά (“Internet of Things”–IoT), πρωτοχρησιμοποιήθηκε το 1999 από τον Βρετανό Kevin Ashton στην προσπάθειά του να παρουσιάσει ένα σύστημα στο οποίο τα αντικείμενα του φυσικού κόσμου θα είχαν τη δυνατότητα να

συνδεθούν στο Διαδίκτυο, χρησιμοποιώντας 9 αισθητήρες. Αυτός ο προσδιορισμός ήθελε να δώσει έμφαση στις δυνατότητες που μπορεί να έχει ένα σύστημα ταυτοποίησης με τη χρήση ραδιοσυχνότητων (“Radio Frequency Identification” – RFID) [23]. Αργότερα, το IoT συνδέθηκε από τους ερευνητές και με άλλες τεχνολογίες, όπως οι αισθητήρες, οι ενεργοποιητές, οι συσκευές εντοπισμού θέσεις (“Global Positioning System” – GPS) και οι κινητές συσκευές.

## 1.2.2 Υπολογιστικό “σύννεφο” - Cloud computing

Ένας από τους πιο διαδεδομένους ορισμούς του υπολογιστικού “Σύννεφου” (Cloud computing) προέρχεται από το Αμερικανικό Ινστιτούτο Προτύπων και Τεχνολογίας (NIST). Σύμφωνα με τον επίσημο ορισμό του NIST [9], το υπολογιστικό “σύννεφο” είναι ένα υπολογιστικό μοντέλο που επιτρέπει ευρεία, φιλική προς το χρήστη και κατ'απαίτηση (on-demand) πρόσβαση σε ένα δίκτυο που αποτελείται από ένα σύνολο κοινόχρηστων και διαμορφώσιμων υπολογιστικών πόρων (π.χ. δίκτυα, διακομιστές, χώρος αποθήκευσης, εφαρμογές και υπηρεσίες). Οι πόροι αυτοί διατίθενται στους χρήστες εύκολα και άμεσα με ελάχιστη διαχείριση ή αλληλεπίδραση με τους παρόχους υπηρεσιών.



Εικόνα 1-2 - Τα βασικά χαρακτηριστικά του Υπολογιστικού “νέφους” (cloud) [9]

Ο ορισμός NIST απαριθμεί πέντε βασικά χαρακτηριστικά του υπολογιστικού “σύννεφου”:

1. **Αυτοεξυπηρέτηση κατά παραγγελία.** Ένας καταναλωτής μπορεί να καταναλώνει υπολογιστικούς πόρους όπως π.χ. χρόνο χρήσης διακομιστή και αποθήκευση δεδομένων στο δίκτυο, χωρίς να απαιτείται ανθρώπινη αλληλεπίδραση με τον πάροχο υπηρεσιών.
2. **Ευρεία πρόσβαση στο δίκτυο.** Οι υπηρεσίες παροχής είναι διαθέσιμες μέσω του δικτύου και είναι προσβάσιμες μέσω συμβατών μηχανισμών μέσα από μια πληθώρα εφαρμογών και τεχνολογιών που υποστηρίζονται από τις πλατφόρμες των πελατών (π.χ. κινητά τηλέφωνα, tablet, φορητούς υπολογιστές και σταθμούς εργασίας).
3. **Ομαδοποίηση πόρων.** Οι υπολογιστικοί πόροι του παρόχου ομαδοποιούνται για να εξυπηρετήσουν πολλούς και διαφορετικής κλίμακας καταναλωτές. Η ομαδοποίηση αυτή βασίζεται στην δυναμική διαχείριση διαφορετικών φυσικών και εικονικών πόρων ανάλογα με τη ζήτηση των καταναλωτών. Σε αυτό το μοντέλο διαχείρισης υπολογιστικών πόρων, ο καταναλωτής δεν καταλαβαίνει που ακριβώς είναι η φυσική τοποθεσία του εξυπηρετητή που χρησιμοποιεί παρά μόνο σε επίπεδο υπολογιστικού κέντρου ή χώρας. Παραδείγματα

μίσθωσης υπολογιστικών πόρων από τους καταναλωτές περιλαμβάνουν η αποθήκευση, η υπολογιστική ισχύς, η μνήμη και το εύρος ζώνης δικτύου.

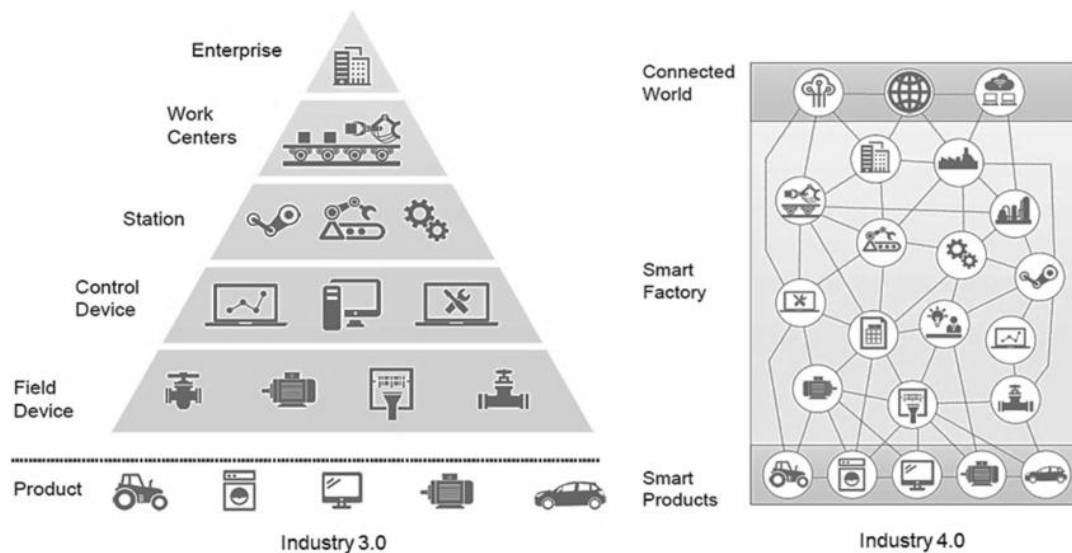
4. **Ταχεία ελαστικότητα.** Οι υπηρεσίες παροχής μπορούν να ενεργοποιηθούν και να αποδεσμευτούν με ελαστικό τρόπο και σε ορισμένες περιπτώσεις αυτόματα. Η ελαστικότητα στην ενεργοποίηση των παρεχόμενων υπηρεσιών επιτρέπει την γρήγορη κλιμάκωση του υπολογιστικού συστήματος τόσο από την μεριά του καταναλωτή όσο και από την μεριά του παρόχου. Με αυτό τον τρόπο ο καταναλωτής μπορεί να έχει πρόσβαση σε υπολογιστικούς πόρους διαφόρων μεγεθών και ανά πάσα στιγμή ανάλογα με τις ανάγκες του την εκάστοτε στιγμή.
5. **Μέτρηση χρήσης παρεχόμενων υπηρεσιών.** Τα συστήματα υπολογιστικού “νέφους” ελέγχουν και βελτιστοποιούν αυτόματα, τη χρήση υπολογιστικών πόρων. Αυτό είναι δυνατό επειδή διαθέτουν μηχανισμούς μέτρησης κατάλληλους για τον κάθε τύπο υπηρεσίας (π.χ. μέγεθος διαθέσιμου αποθηκευτικού χώρου, παρεχόμενη υπολογιστική ισχύς, εύρος ζώνης δικτύου και λογαριασμοί ενεργών χρηστών). Η χρήση πόρων γίνεται με διαφάνεια τόσο για τον πάροχο όσο και για τον χρήστη καθώς οι μηχανισμοί μέτρησης επιτρέπουν της παρακολούθηση, τον έλεγχο και την οπτικοποίηση της χρησιμοποιούμενης υπηρεσίας.

Επιπλέον, είναι σημαντικό σε αυτό το σημείο να αναφερθεί ότι ο ορισμός NIST απαριθμεί και τα μοντέλα υπηρεσιών του υπολογιστικού “σύννεφου”:

1. **Λογισμικό ως υπηρεσία (SaaS).** Οι υπηρεσίες παροχής προς τον καταναλωτή είναι η χρήση εφαρμογών που λειτουργούν πάνω στην υποδομή του παρόχου υπολογιστικού “νέφους”. Οι εφαρμογές είναι προσβάσιμες από διάφορες συσκευές μέσω μιας απλής διεπαφής, όπως ένα πρόγραμμα περιήγησης ιστού (π.χ. email μέσω διαδικτύου) ή μια διεπαφή προγράμματος. Ο καταναλωτής είναι απαλλαγμένος από την διαχείριση της υποδομής υπολογιστικού “νέφους” που περιλαμβάνει δίκτυο, διακομιστές, λειτουργικά συστήματα, χώρο αποθήκευσης ή ακόμη και μεμονωμένες δυνατότητες εφαρμογής, με εξαίρεση ενδεχομένως τον έλεγχο του αριθμού των χρηστών και συγκεκριμένες ρυθμίσεις διαμόρφωσης της εφαρμογής στις ανάγκες του πελάτη.
2. **Πλατφόρμα ως υπηρεσία (PaaS).** Οι υπηρεσίες παροχής σε αυτή την περίπτωση απευθύνεται σε καταναλωτές που θέλουν να χρησιμοποιήσουν τις υποδομές του παρόχου υπολογιστικού νέφους με σκοπό την εγκατάσταση και ανάπτυξη εφαρμογών με στόχο άλλους καταναλωτές-χρήστες. Σε αυτές τις υπηρεσίες παροχής ο καταναλωτής δεν έχει τον έλεγχο διαχείρισης της υποδομής των πόρων του υπολογιστικού “νέφους” (π.χ. δίκτυο, εξυπηρετητές, λειτουργικά συστήματα ή αποθήκευση), παρά μόνο ελέγχει τις εγκατεστημένες υπηρεσίες-εφαρμογές και μπορεί να ρυθμίζει τα λειτουργικά συστήματα και περιβάλλοντα λογισμικού που φιλοξενούν αυτές τις εφαρμογές.
3. **Υποδομή ως υπηρεσία (IaaS).** Οι υπηρεσίες παροχής στον καταναλωτή είναι η παροχή επεξεργασίας, αποθήκευσης, δικτύων και άλλων θεμελιωδών υπολογιστικών πόρων όπου το ο καταναλωτής είναι σε θέση να αναπτύξει και να εκτελέσει το δικό του λογισμικό, το οποίο μπορεί να περιλαμβάνει από λειτουργικά συστήματα μέχρι και εφαρμογές. Σε αυτή την περίπτωση, ο καταναλωτής δεν διαχειρίζεται ούτε ελέγχει το υπολογιστικό κέντρο του “σύννεφου” ωστόσο έχει δικαιώματα ελέγχου στα λειτουργικά συστήματα, την αποθήκευση και τις εφαρμογές που αναπτύσσονται και ενδεχομένως περιορισμένο έλεγχο στις ρυθμίσεις δικτύου (π.χ. τείχη προστασίας κεντρικού υπολογιστή).

### 1.2.3 Βιομηχανία 4ης γενιάς - Industry 4.0

Το Industry 4.0 είναι ένας όρος που εφαρμόζεται σε μια ομάδα ραγδαίων τεχνολογικών εξελίξεων στα βιομηχανικά επίπεδα του σχεδιασμού, της κατασκευής, λειτουργίας και συντήρησης συστημάτων και προϊόντων παραγωγής. Ο χαρακτηρισμός Industry 4.0 προέρχεται από την Ευρωπαϊκή Ένωση και αποτελεί ορόσημο και οδηγό στο μέλλον της βιομηχανίας [15], [20].



Εικόνα 1-3 - Πηγαίνοντας προς την Βιομηχανία της 4ης γενιάς [15][20]

Η βιομηχανία 4ης γενιάς εξαρτάται από μια σειρά νέων και καινοτόμων τεχνολογικών εξελίξεων όπως:

- Η εφαρμογή **τεχνολογίας πληροφοριών και επικοινωνιών (ΤΠΕ)** για την ψηφιοποίηση της πληροφορίας και την ενσωμάτωση συστημάτων σε όλα τα στάδια της παραγωγής, δημιουργίας και χρήσης του προϊόντος (συμπεριλαμβανομένων της εφοδιαστικής και προμηθευτικής αλυσίδας), τόσο εντός των εταιρειών όσο και πέραν των ορίων της εταιρείας
- **Ενσωματωμένα συστήματα** που χρησιμοποιούν ΤΠΕ για την παρακολούθηση και τον έλεγχο φυσικών διαδικασιών και συστημάτων. Αυτά μπορεί να περιλαμβάνουν ενσωματωμένους αισθητήρες, έξυπνα ρομπότ που μπορούν να προσαρμόζονται στο άμεσο προϊόν που πρόκειται να παραχθεί ή σε **προσθετικές κατασκευαστικές τεχνικές** όπως είναι η τρισδιάστατη εκτύπωση.
- **Επικοινωνίες δικτύου**, συμπεριλαμβανομένων ασύρματων και επίγειων συνδέσεων διαδικτύου που εξυπηρετούν την σύνδεση των μηχανών, των προϊόντων εργασίας, των συστημάτων αλλά και των ανθρώπων, τόσο εντός του κατασκευαστικού χώρου του εργοστασίου, όσο και με τους προμηθευτές και διανομείς.
- **Προσομοίωση, μοντελοποίηση και απεικόνιση** τόσο στο σχεδιασμό των προϊόντων όσο και στην εγκατάσταση των διαδικασιών παραγωγής.
- **Συλλογή τεράστιων ποσοτήτων δεδομένων** με σκοπό την ανάλυση και εκμετάλλευσής άμεσα στον κατασκευαστικό χώρο του εργοστασίου ή μέσω της αποστολής και ανάλυσης μεγάλου όγκου δεδομένων από το υπολογιστικό “νέφος”.
- **Επαυξημένη πραγματικότητα και “έξυπνα” εργαλεία** με σκοπό την μεγαλύτερη υποστήριξη και ασφάλεια των εργαζομένων και των συνεργαζόμενων ρομπότ στο χώρο του εργοστασίου.

### 1.2.4 “Έξυπνα” σπίτια - Smart homes

Το έξυπνο σπίτι θα μπορούσε να χαρακτηριστεί ως το σύνολο των διατάξεων, με τις οποίες ομαδοποιούνται, οργανώνονται και αυτοματοποιούνται οι λειτουργίες μιας κατοικίας, σύμφωνα με τις ανάγκες του εκάστοτε ιδιοκτήτη. Υπάρχουν διάφοροι τύποι “έξυπνων” συσκευών και τεχνολογιών “έξυπνου” σπιτιού. Ωστόσο, καθένα από αυτά μπορεί να έχει διαφορετική τελική χρήση, πρωτόκολλο σύνδεσης και διεπαφή. Στην έκθεση του 2015 [21], ο Karlin, B. προτείνει ξεχωριστά προϊόντα που συγκεντρώνονται σε τρεις ομάδες, υπό μια κοινή ονοματολογία των οικιακών/σπιτικών συστημάτων διαχείρισης ενέργειας και ταιριάζουν στο υπό μελέτη οικοσύστημα του “έξυπνου” σπιτιού και των “έξυπνων” διασυνδεδεμένων συσκευών.

Παρακάτω παρουσιάζονται οι πιο σημαντικές γενικές κατηγορίες συσκευών IoT στα πλαίσια του “έξυπνου” σπιτιού και των “έξυπνων” διασυνδεδεμένων συσκευών και διαφοροποιούνται με βάση τα χαρακτηριστικά τους όπως είναι η λειτουργία τους, η διεπαφή με το χρήστη, το υλικό και το λογισμικό.

#### 1.2.4.1 Ενεργειακές Πύλες πληροφοριών

Οι ενεργειακές πύλες είναι μια εφαρμογή που συλλέγει και παρουσιάζει με φιλικό τρόπο στο χρήστη πληροφορίες για την κατανάλωση ενέργειας του “έξυπνου” σπιτιού. Αυτός ο τύπος εφαρμογών παρέχει μια πιο λεπτομερή και άμεση ενημέρωση από τους παραδοσιακούς λογαριασμούς και συνήθως παρέχονται ως υπηρεσία από επιχειρήσεις κοινής ωφέλειας.

Πίνακας 1-1 - Περιγραφή των συσκευών "Ενεργειακές πύλες πληροφοριών" για τα "έξυπνα" σπίτια.

<b>Ενεργειακές Πύλες πληροφοριών</b>	
<b>Λειτουργία</b>	Συλλογή και Μεταφορά Δεδομένων Ενέργειας προς τον τελικό καταναλωτή
<b>Συγκεκριμένες λειτουργίες</b>	<ul style="list-style-type: none"> <li>• Λαμβάνει πληροφορίες κατανάλωσης ενέργειας από το “έξυπνους” μετρητές, έξυπνες συσκευές και άλλα έξυπνα προϊόντα εντός του νοικοκυριού.</li> <li>• Επιτρέπει πιο λεπτομερή και σχεδόν σε πραγματικό χρόνο πληροφόρηση του τελικού καταναλωτή για την κατανάλωση-ενέργειας σε σχέση με τους παραδοσιακούς λογαριασμούς</li> <li>• Με βάση τις πληροφορίες που παρέχει η πύλη, επιτρέπει στους χρήστες να ενεργούν είτε με τηλεχειρισμό είτε μέσω του πάνελ ελέγχου.</li> <li>• Παρέχει άμεση ενημέρωση για προτεινόμενες ενέργειες ή ρυθμίσεις με σκοπό την εξοικονόμηση ενέργειας αλλά και την βελτίωση του ενεργειακού προφίλ του χρήστη για χαμηλότερο κόστος.</li> </ul>
<b>Διεπαφή</b>	Έξυπνα κινητά, Διαδικτυακές εφαρμογές, Προσωπικός υπολογιστής, λογισμικό
<b>Επικοινωνία</b>	Wi-Fi, LAN
<b>Αλληλεπίδραση</b>	Αμφίδρομη επικοινωνία από και προς άλλες έξυπνες συσκευές και έξυπνα προϊόντα.

### 1.2.4.2 Μετρητές Φορτίων

Οι μετρητές φορτίου παρέχουν απλές πληροφορίες κατανάλωσης ενέργειας για την λειτουργία μιας έξυπνης συσκευής. Αυτά συνδέονται μεταξύ της πρίζας και της πραγματικής συσκευής και συλλέγουν μετρήσεις για την κατανάλωση ενέργειας.

Πίνακας 1-2 - Περιγραφή των μετρητών φορτίων για τα "έξυπνα" σπίτια

<b>Μετρητές Φορτίων</b>	
<b>Λειτουργία</b>	Άμεση συλλογή των δεδομένων κατανάλωσης μιας οικιακής συσκευής.
<b>Συγκεκριμένες λειτουργίες</b>	Εγκατεστημένο μεταξύ των πριζών παροχής και των συσκευών λαμβάνει πληροφορίες κατανάλωσης ενέργειας σε πραγματικό χρόνο απευθείας από μεμονωμένες συσκευές. Ο υπολογισμός του τελικού κόστους κατανάλωσης είναι έμμεσος από την τιμή της μονάδας κατανάλωσης και της τελικής ποσότητας του φορτίου.
<b>Διεπαφή</b>	Η οθόνη της συσκευής ή έξυπνη εφαρμογή κινητού τηλεφώνου ή υπολογιστή.
<b>Επικοινωνία</b>	Wi-Fi, LAN, Bluetooth
<b>Αλληλεπίδραση</b>	Μονόδρομη επικοινωνία για την μέτρηση και αποστολή/ένδειξη του φορτίου κατανάλωσης.

### 1.2.4.3 Έξυπνες οικιακές συσκευές

Οι έξυπνες οικιακές συσκευές ορίζονται ως συσκευές με δυνατότητα επικοινωνίας. Αυτό το κανάλι επικοινωνίας χρησιμοποιείται για την πρόσβαση και ρύθμιση των λειτουργιών της συσκευής από τον τελικό χρήστη είτε από άλλες πλατφόρμες/συσκευές του έξυπνου σπιτιού.

Πίνακας 1-3 - Περιγραφή των έξυπνων οικιακών συσκευών για τα "έξυπνα" σπίτια

<b>Έξυπνες οικιακές συσκευές</b>	
<b>Λειτουργία</b>	Οικιακές συσκευές με δυνατότητα επικοινωνίας με τον χρήστη και με άλλες πλατφόρμες και υπηρεσίες.
<b>Συγκεκριμένες λειτουργίες</b>	<ul style="list-style-type: none"> <li>• Επικοινωνία μεταξύ του έξυπνου μετρητή, παρέχοντας πληροφορίες για την κατανάλωση ενέργειας.</li> <li>• Δυνατότητα αλλαγής του προφίλ κατανάλωσης της συσκευής.</li> <li>• Δυνατότητα υποστήριξης μεταβλητής τιμολόγησης με βάση την ζήτηση στην αγορά ενέργειας την επόμενη μέρα.</li> </ul>
<b>Διεπαφή</b>	Η οθόνη της συσκευής ή έξυπνη εφαρμογή κινητού τηλεφώνου ή υπολογιστή, Περιφερειακές οθόνες μέσα στο έξυπνο σπίτι, διαδικτυακές εφαρμογές και η ενεργειακή πύλη πληροφοριών

<b>Επικοινωνία</b>	Ασύρματη και ενσύρματα πρωτόκολλα επικοινωνίας.
<b>Αλληλεπίδραση</b>	Αμφίδρομη επικοινωνία ανάμεσα στο χρήστη, τις εφαρμογές μετρήσεων και ακόμα και τον πάροχο ηλεκτρικής ενέργειας.

#### 1.2.4.4 Έξυπνοι θερμοστάτες

Οι έξυπνοι θερμοστάτες έχουν τελικά την ίδια κύρια λειτουργικότητα με τους παραδοσιακούς θερμοστάτες όπως τον έλεγχο ενός συστήματος κλιματισμού. Τα πρόσθετα χαρακτηριστικά του έξυπνου θερμοστάτη είναι ο ευέλικτος προγραμματισμός με αλγόριθμους μηχανικής μάθησης και η φιλική απεικόνιση της κατάστασης του προγράμματος του συστήματος κλιματισμού προς το χρήστη σε πραγματικό χρόνο.

Πίνακας 1-4 - Περιγραφή των έξυπνων θερμοστατών για τα "έξυπνα" σπίτια

<b>Έξυπνες θερμοστάτες</b>	
<b>Λειτουργία</b>	Έλεγχος της θερμοκρασίας μέσα από τον έλεγχο ενός συνόλου παραμέτρων κατανάλωσης.
<b>Συγκεκριμένες λειτουργίες</b>	<ul style="list-style-type: none"> <li>• Αυτόνομη μηχανική μάθηση του προφίλ θέρμανσης του καταναλωτή</li> <li>• Ενεργοποίηση/απενεργοποίηση κατά την εγγύτητα του καταναλωτή προς/από το σπίτι.</li> <li>• Ανίχνευση παρουσίας για την ρύθμιση της θερμοκρασίας κλιματισμού.</li> <li>• Επικοινωνία με τον χρήστη και δυνατότητα για τηλεχειρισμό και έλεγχο μέσω άλλων συσκευών</li> <li>• Αλληλεπίδραση με άλλες έξυπνες οικιακές συσκευές</li> </ul>
<b>Διεπαφή</b>	Η οθόνη της συσκευής ή έξυπνη εφαρμογή κινητού τηλεφώνου ή υπολογιστή, Περιφερειακές οθόνες μέσα στο έξυπνο σπίτι, διαδικτυακές εφαρμογές
<b>Επικοινωνία</b>	Ασύρματη πρωτόκολλα επικοινωνίας.
<b>Αλληλεπίδραση</b>	Αμφίδρομη επικοινωνία ανάμεσα στο χρήστη και τις εφαρμογές μετρήσεων.

#### 1.2.4.5 Έξυπνοι Θερμοσίφωνες

Οι έξυπνοι θερμοσίφωνες είναι οι κλασσικοί ενσωματωμένοι ελεγκτές θερμοσίφωνα που μετατρέπουν το πετρέλαιο, φυσικό αέριο ή ηλεκτρισμό σε θερμότητα ζεσταίνοντας το νερό. Ωστόσο το έξυπνο χαρακτηριστικό τους είναι ότι επιτρέπουν στον χρήστη να ελέγχει την θερμοκρασία του νερού, να προγραμματίζει την θέρμανση του νερού ενεργοποιώντας/απενεργοποιώντας τον θερμοσίφωνα εντός/εκτός σπιτιού.

Πίνακας 1-5 - Περιγραφή του έξυπνου θερμοσίφωνα για τα "έξυπνα" σπίτια

<b>Έξυπνες θερμοσίφωνες</b>	
<b>Λειτουργία</b>	Βελτιστοποίηση του προγραμματισμού και της κατανάλωσης ενέργειας κατά την λειτουργία θέρμανσης του νερού.
<b>Συγκεκριμένες λειτουργίες</b>	<ul style="list-style-type: none"> <li>• Απομακρυσμένος έλεγχος θερμοσίφωνα</li> <li>• Σύνδεση του θερμοσίφωνα με τα έξυπνα συστήματα κλιματισμού για την μείωση της κατανάλωσης ενέργειας.</li> </ul>
<b>Διεπαφή</b>	Η οθόνη της συσκευής ή έξυπνη εφαρμογή κινητού τηλεφώνου ή υπολογιστή, Περιφερειακές οθόνες μέσα στο έξυπνο σπίτι, διαδικτυακές εφαρμογές
<b>Επικοινωνία</b>	Ασύρματη πρωτόκολλα επικοινωνίας.
<b>Αλληλεπίδραση</b>	Αμφίδρομη επικοινωνία ανάμεσα στο χρήστη και τις εφαρμογές μετρήσεων.

#### 1.2.4.6 Έξυπνοι Κόμβοι Μετρήσεων

Οι έξυπνοι κόμβοι μετρήσεων είναι συσκευές που συγκεντρώνουν πολλές μετρήσεις από τις έξυπνες συνδεδεμένες συσκευές εντός του οικιακού περιβάλλοντος. Ο κύριος στόχος των έξυπνων κόμβων είναι η ενσωμάτωση των λειτουργιών από όλες τις αυτές τις συσκευές και επικοινωνούν με σκοπό τον συντονισμό και την λήψη αποφάσεων για διάφορα συστήματα του έξυπνου οικιακού περιβάλλοντος.

Πίνακας 1-6 - Περιγραφή των έξυπνων κόμβων μετρήσεων για τα "έξυπνα" σπίτια

<b>Έξυπνοι Κόμβοι Μετρήσεων</b>	
<b>Λειτουργία</b>	Σύνδεση και ομαδοποίηση όλων των συνδεδεμένων συσκευών του έξυπνου σπιτιού.
<b>Συγκεκριμένες λειτουργίες</b>	<ul style="list-style-type: none"> <li>• Απομακρυσμένος έλεγχος των συνδεδεμένων συσκευών</li> <li>• Σύνδεση συνδεδεμένων συσκευών που τις καθιστά ικανές να επικοινωνούν μεταξύ τους</li> <li>• Πρόσβαση στο διαδίκτυο</li> <li>• Χρήση των πληροφοριών για την διασύνδεση με έξυπνα συστήματα ψυχαγωγίας ή φωτισμού.</li> </ul>
<b>Διεπαφή</b>	Η οθόνη της συσκευής ή έξυπνη εφαρμογή κινητού τηλεφώνου, διαδικτυακές εφαρμογές
<b>Επικοινωνία</b>	Ασύρματη πρωτόκολλα επικοινωνίας.
<b>Αλληλεπίδραση</b>	Αμφίδρομη επικοινωνία ανάμεσα στο χρήστη, τις εφαρμογές μετρήσεων και φυσικά τις άλλες συνδεδεμένες συσκευές.

### 1.2.4.7 Έξυπνες Πρίζες

Οι έξυπνες πρίζες είναι συσκευές που βρίσκονται μεταξύ μιας συμβατής πρίζας ενέργειας και μιας ηλεκτρικής συσκευής. Αυτές οι συσκευές μετρούν την κατανάλωση ενέργειας από την συσκευή και μπορούν να ενεργοποιούν/απενεργοποιούν την παροχή ηλεκτρικής ενέργειας προς την συσκευή.

Πίνακας 1-7 - Περιγραφή των έξυπνων πριζών για τα "έξυπνα" σπίτια

<b>Έξυπνες Πρίζες</b>	
<b>Λειτουργία</b>	Έλεγχος και συλλογή μετρήσεων κατανάλωσης των οικιακών συσκευών.
<b>Συγκεκριμένες λειτουργίες</b>	<ul style="list-style-type: none"> <li>● Απομακρυσμένος έλεγχος συσκευών</li> <li>● Μπορεί να μετατρέψει μια μη "έξυπνη" συσκευή ή πρίζα σε "έξυπνη"</li> <li>● Επικοινωνία με τον χρήστη</li> <li>● Αλληλεπίδραση με έξυπνους οικιακούς κόμβους</li> </ul>
<b>Διεπαφή</b>	έξυπνη εφαρμογή κινητού τηλεφώνου, διαδικτυακές εφαρμογές
<b>Επικοινωνία</b>	Ασύρματη πρωτόκολλα επικοινωνίας.
<b>Αλληλεπίδραση</b>	Αμφίδρομη επικοινωνία ανάμεσα στο χρήστη, τις εφαρμογές μετρήσεων.

### 1.2.4.8 Έξυπνα Φώτα

Τα έξυπνα φώτα είναι συσκευές φωτισμού που ενσωματώνουν κανονικό φωτισμό με ενσωματωμένο έλεγχο που επιτρέπει την ενεργοποίηση/απενεργοποίηση, την αλλαγή της φωτεινότητας ή του χρώματος του φωτός, προγραμματισμό στο χρόνο φωτισμού με σκοπό την μείωση της κατανάλωσης ενέργειας.

Πίνακας 1-8 - Περιγραφή έξυπνων φώτων για τα "έξυπνα" σπίτια

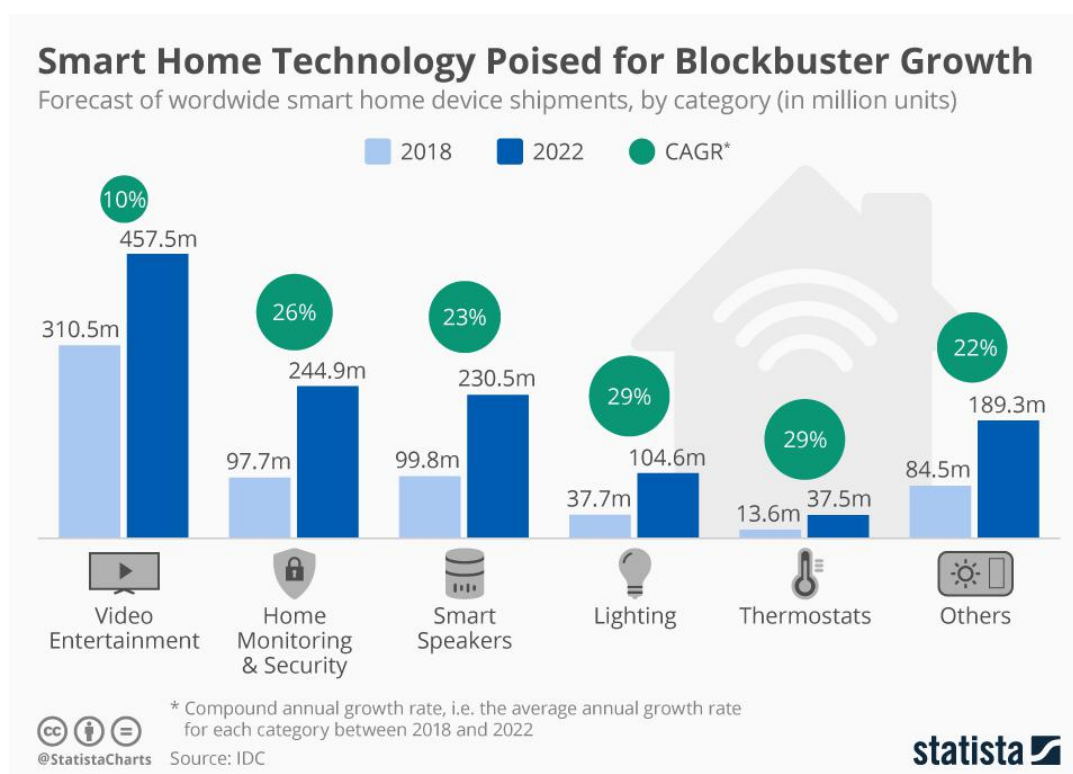
<b>Έξυπνα Φώτα</b>	
<b>Λειτουργία</b>	Εφαρμογές φωτισμού με ποικίλα χαρακτηριστικά ελέγχου
<b>Συγκεκριμένες λειτουργίες</b>	<ul style="list-style-type: none"> <li>● Αισθητήρας φωτισμού</li> <li>● Δυνατότητα εξασθένισης</li> <li>● Ανίχνευση παρουσίας</li> <li>● Προγραμματισμός φωτισμού</li> <li>● Επικοινωνία με τον χρήστη</li> <li>● Απομακρυσμένη πρόσβαση</li> <li>● Αλληλεπίδραση με έξυπνους οικιακούς κόμβους</li> <li>● Αλλαγή χρώματος</li> </ul>
<b>Διεπαφή</b>	Έξυπνη εφαρμογή κινητού τηλεφώνου, διαδικτυακές εφαρμογές
<b>Επικοινωνία</b>	Ασύρματη πρωτόκολλα επικοινωνίας.

<b>Αλληλεπίδραση</b>	Αμφίδρομη επικοινωνία ανάμεσα στο χρήστη, τις εφαρμογές μετρήσεων και φυσικά τις άλλες συνδεδεμένες συσκευές.
----------------------	---

### 1.3 Αποτελέσματα Έρευνας

Με βάση την παραπάνω έρευνα και το προγνωστικό διάγραμμα που φαίνεται στην Εικόνα 1-4 παρατηρείται ότι το 2022 αναμένεται μια αύξηση της τάξης του 30% στις εφαρμογές έξυπνων θερμοστατών και 26% για τις εφαρμογές παρακολούθησης μετρήσεων από τα έξυπνα σπιτία.

Τα προγνωστικά αυτά κάνουν το πρωτότυπο της εργασίας αυτής να αποτελεί μια σύγχρονη ολοκληρωμένη εφαρμογή δεδομένου ότι συλλέγει, επεξεργάζεται, αποθηκεύει, παρουσιάζει και αποστέλλει στο υπολογιστικό “νέφος” αλλά και τοπικά μετρήσεις θερμοκρασίας, υγρασίας και ανίχνευσης κίνησης χρονο-προσδιορισμένες με βάση το διεθνές πρότυπο ώρας ISO8601 [8].



Εικόνα 1-4 - Εκτίμηση της αύξησης των συνδεδεμένων συσκευών IoT στο περιβάλλον του "έξυπνου" σπιτιού για το έτος 2022

Μια ακόμα σύγχρονη εφαρμογή χρήσης του πρωτότυπου αυτής της διπλωματικής εργασίας είναι η χρήση του στα πλαίσια της βιομηχανικής εξέλιξης 4ης γενιάς. Το σύστημα του διαδικτύου των πραγμάτων που αναπτύχθηκε μπορεί να χρησιμοποιηθεί σαν πυρήνας σε εφαρμογές μετρήσεων από αισθητήρες θερμοκρασίας, υγρασίας και ανίχνευσης κίνησης. Οι μετρήσεις αυτές είναι σημαντικές γιατί θα μπορούσαν να χρησιμοποιηθούν σε διάφορα εργοστασιακά περιβάλλοντα τόσο για την μέτρηση εργοστασιακών διεργασιών όσο και για την μέτρηση των κλιματικών συνθηκών που επικρατούν σε αποθήκες και χώρους εργασίας.

Ο πυρήνας του συστήματος IoT που αναπτύχθηκε μπορεί να αποτελέσει την βάση για την προσθήκη και άλλων εργοστασιακών αισθητήρων με σκοπό την συλλογή περισσότερων μετρήσεων και αποστολή τους στα κέντρα ελέγχου της βιομηχανίας. Ο εξυπηρετητής συλλογής δεδομένων θα

## Κεφάλαιο 1

μπορούσε να εξελιχθεί σε έναν κόμβο διανομής των μετρήσεων προς τα εργοστασιακά ρομπότ και ταυτόχρονα να προγραμματιστεί με λογική λήψης αποφάσεων προσανατολισμένη στις εργοστασιακές απαιτήσεις της παραγωγής.

Οι παραπάνω εφαρμογές τοποθετούν το πρωτότυπο σύστημα IoT που θα παραχθεί από αυτή την διπλωματική εργασία ως ένα επεκτάσιμο πυρήνα ενός σύγχρονου συστήματος εφαρμογών έξυπνου σπιτιού ή/και βιομηχανικής εξέλιξης 4ης γενιάς.

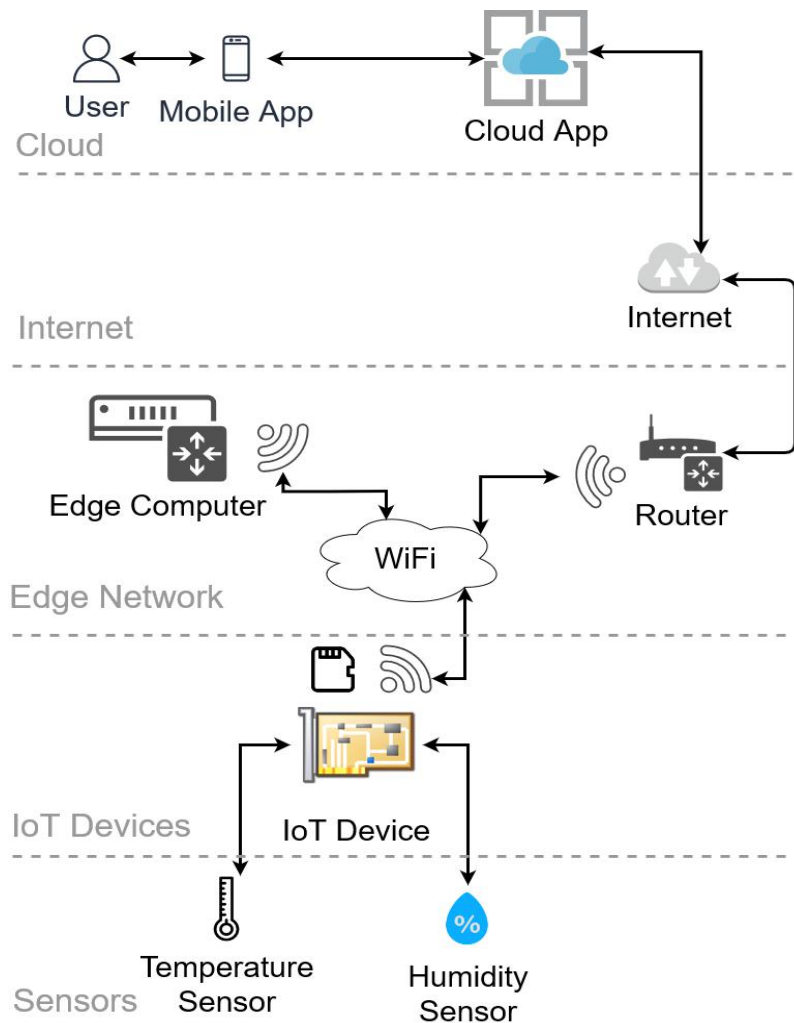
## Κεφάλαιο 2ο: Σχεδίαση αρχιτεκτονικής - Σύστημα IoT

### 2.1 Εισαγωγή

Με βάση τα αποτελέσματα της έρευνας και των πιο σύγχρονων τεχνολογιών στην αγορά, αυτή η ενότητα περιγράφει την μεθοδολογία σχεδίασης του συστήματος IoT. Στην αρχή περιγράφεται η αρχιτεκτονική του συστήματος σε μορφή μπλοκ διαγράμματος εμφανίζοντας τα διάφορα “επίπεδα” που υπάρχουν σε ένα σύστημα του διαδικτύου των πραγμάτων.

Στην συνέχεια περιγράφονται τα επιμέρους υποσυστήματα που συνθέτουν το ολικό σύστημα του διαδικτύου των πραγμάτων IoT. Για το κάθε υποσύστημα, αναφέρεται το “επίπεδο” που ανήκει σχετικά με το συνολικό σύστημα IoT, η τεχνολογία που χρησιμοποιήθηκε και τέλος η λειτουργικότητα του τόσο στα γενικά πλαίσια μιας IoT εφαρμογής όσο και φυσικά, στα πλαίσια χρήσης τους στο πρωτότυπο της διπλωματικής εργασίας.

Η περιγραφή των υποσυστημάτων είναι χωρισμένη σε δύο μεγάλες κατηγορίες. Την κατηγορία του “Υλικού” (hardware) όπου περιγράφονται τα υποσυστήματα που σχετίζονται μόνο με το “υλικό” του συστήματος IoT. Στην δεύτερη κατηγορία, περιγράφονται τα υποσυστήματα του “λογισμικού” του συστήματος IoT. Η διάκριση αυτή φαίνεται και στην Εικόνα 2-1.



Εικόνα 2-1 - Σχεδίαση - Μπλοκ διάγραμμα της αρχιτεκτονικής μιας πρωτότυπης εφαρμογής IoT με πυρήνα την συσκευή IoT

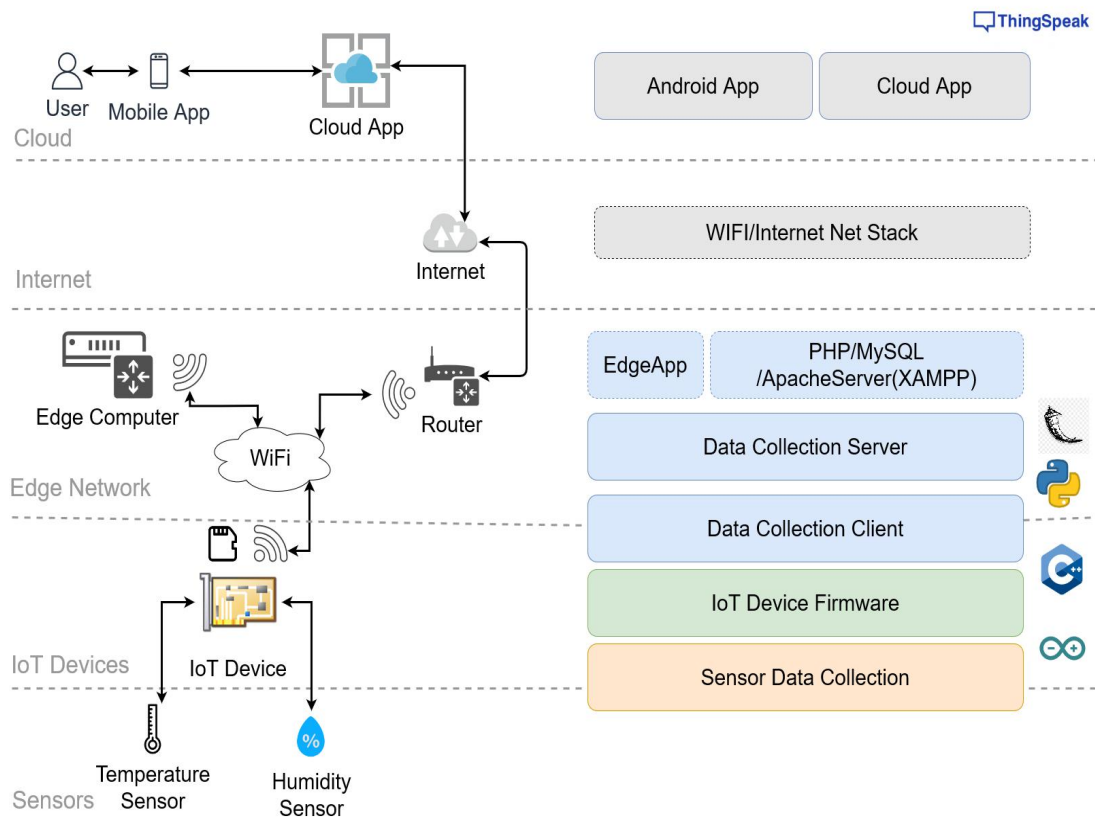
## 2.2 Συνολικό διάγραμμα αρχιτεκτονικής

Η Εικόνα 2-1 παρουσιάζει την σχεδίαση του συστήματος σε μορφή μπλοκ διαγράμματος. Τα μπλοκ διαγράμματα παρουσιάζουν τα διάφορα στοιχεία της αρχιτεκτονικής του συστήματος, τον τρόπο διασύνδεσης μεταξύ τους καθώς και τα διάφορα επίπεδα λειτουργικότητας του κάθε υποσυστήματος. Η λειτουργικότητα και οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση κάθε μπλοκ διαγράμματος περιγράφονται στον Πίνακα 2-1.

1. **Αισθητήρες (Sensors):** Σε αυτό το επίπεδο περιγράφονται οι αισθητήρες. Οι αισθητήρες είναι αναλογικές, ψηφιακές ή και “έξυπνες” μετρητικές διατάξεις για την συλλογή και αποστολή των δεδομένων των μετρήσεων προς τον μικροελεγκτή.
2. **Μικροελεγκτές/Συσκευές IoT (MCU/IoT Devices):** Σε αυτό το επίπεδο περιγράφεται ο μικροελεγκτής του συστήματος ή γενικότερα η μητρική της IoT συσκευής. Η χρήση των “μητρικών-ανάπτυξης” συνηθίζεται κατά την ανάπτυξη ενός IoT συστήματος καθώς οι εταιρείες παρέχουν τους μικροελεγκτές τοποθετημένους σε μητρικές πλακέτες με ενσωματωμένες διεπαφές τόσο για τον προγραμματισμό μέσω υπολογιστή όσο και για την διασύνδεση τους με μια πληθώρα αισθητήρων.
3. **Ακρογωνιαίο Δίκτυο (Edge network):** Σε αυτό το επίπεδο βρίσκεται εξοπλισμός με υψηλότερη υπολογιστική ισχύ από τους μικροελεγκτές και δικτυακός εξοπλισμός συνδεδεμένος με το δημόσιο διαδίκτυο με σκοπό την επεξεργασία και προώθηση των δεδομένων που προέρχονται από τα περιφερειακά συστήματα IoT. Στην περίπτωση της Εικόνα 2-1 η υπολογιστική ισχύς παρέχεται από έναν προσωπικό υπολογιστή και ο διαδικτυακός εξοπλισμός σύνδεσης με το διαδίκτυο είναι ένας οικιακός δρομολογητής (router).
4. **Internet:** Σε αυτό το επίπεδο είναι το γνωστό σε όλους διαδίκτυο. Η πρόσβαση στο διαδίκτυο είναι δυνατή μέσω ακαδημαϊκών συνδέσεων ή συνδέσεων με πάγιες χρεώσεις από τους παρόχους διαδικτύου. Σε εφαρμογές IoT, οι συνδέσεις αυτές μπορεί να διαφέρουν από τις κλασικές συνδέσεις καταναλωτών ανάλογα με τις ανάγκες της “άνω/κάτω ζεύξης” (upload/download link) ανάλογα με τον όγκο των δεδομένων.
5. **Σύννεφο (Cloud):** Αυτό το επίπεδο είναι το υπολογιστικό “σύννεφο” που είναι διαθέσιμο από παρόχους υπηρεσιών υπολογιστικής ισχύος, αποθηκευτικού χώρου στο “σύννεφο” και διεπαφής με το χρήστη μέσω εφαρμογών διαδικτύου και κινητών.

## 2.3 Ανάλυση επιμέρους υποσυστημάτων

Σε αυτή την ενότητα περιγράφονται τα επιμέρους υποσυστήματα και η λειτουργικότητα τους στο σύνολο της αρχιτεκτονικής που προκύπτει από την σχεδίαση του IoT συστήματος όπως φαίνεται στην Εικόνα 2-2 .



Εικόνα 2-2 - Αριστερά: Απεικονίζεται η αρχιτεκτονική κατά την σχεδίαση. Δεξιά: Απεικονίζεται η στοίβα του λογισμικού (Software Stack) με τις διαφορετικές λειτουργίες αντιστοιχισμένες στα 5 επίπεδα της εφαρμογής IoT. Με γκρι και πορτοκαλί χρώμα απεικονίζεται το λογισμικό χρήσης όπως βιβλιοθήκες λογισμικού ενώ με μπλε, πράσινο και πορτοκαλί χρώμα περιγράφεται το λογισμικό που αναπτύχθηκε

### 2.3.1 Υλικό

Στον Πίνακα 2-1 περιγράφεται κάθε στοιχείο που ανήκει στο “υλικό” του συστήματος IoT. Τα χαρακτηριστικά των επιμέρους υποσυστημάτων περιλαμβάνουν το επίπεδο του, το όνομα του, την τεχνολογία του και τέλος την λειτουργικότητα του τόσο σε γενικό επίπεδο μιας εφαρμογής IoT όσο και την συγκεκριμένη χρήση του στο πρωτότυπο της διπλωματικής αυτής.

Πίνακας 2-1 - Περιγραφή της λειτουργικότητας των επιμέρους υποσυστημάτων κατά την σχεδίαση σε επίπεδο "Υλικού"(Hardware)

Υλικό			
A/A	Επίπεδο	Όνομα	Τεχνολογία
1	Αισθητήρες (Sensors)	Αισθητήρας θερμοκρασίας	Θερμίστορ
<b>Γενική Λειτουργικότητα</b>			

<p>Οι αισθητήρες θερμοκρασίας είναι πλέον απαραίτητοι σε κάθε υπολογιστικό σύστημα. Χρησιμοποιούνται</p> <ol style="list-style-type: none"> <li>για την μέτρηση της θερμοκρασίας του συστήματος λειτουργίας με σκοπό την πυροπροστασία και την αποφυγή καταστροφών του συστήματος λόγω υψηλών ή χαμηλών θερμοκρασιών.</li> <li>για την μέτρηση της θερμοκρασίας περιβάλλοντος.</li> <li>για την διόρθωση ευαισθησίας άλλων αισθητήρων που επηρεάζονται από την θερμοκρασία</li> <li>για την συσχέτιση με άλλες μετρήσεις αισθητήρων ως διακριτικά χαρακτηριστικά μηχανικής μάθησης.</li> </ol>			
<p><b>Λειτουργικότητα στο σύστημα IoT</b></p>			
<p>Σε αυτό το σύστημα ο αισθητήρας θερμοκρασίας θα χρησιμοποιηθεί για την μέτρηση της θερμοκρασίας του περιβάλλοντα χώρου.</p>			
2	Αισθητήρες (Sensors)	Αισθητήρας υγρασίας	Μέτρηση σχετικής υγρασίας
<p><b>Γενική Λειτουργικότητα</b></p>			
<p>Οι αισθητήρες υγρασίας παίζουν σημαντικό ρόλο σε εφαρμογές στην βιομηχανία, σε εφαρμογές πληροφορικής σε θαλάσσια περιβάλλοντα και σε περιβαλλοντικές μετρήσεις σε κλειστούς και ανοιχτούς χώρους. Χρησιμοποιούνται</p> <ol style="list-style-type: none"> <li>για την μέτρηση της σχετικής υγρασίας στον περιβάλλοντα χώρο είτε είναι εργοστάσιο είτε το δωμάτιο ενός σπιτιού.</li> <li>για την μέτρηση της σχετικής υγρασίας με σκοπό τον έλεγχο απομακρυσμένων κλιματιστικών συσκευών σε αποθήκες τροφίμων και άλλων ευαίσθητων εμπορευμάτων.</li> <li>για την συσχέτιση με άλλες μετρήσεις αισθητήρων ως διακριτικά χαρακτηριστικά μηχανικής μάθησης.</li> </ol>			
<p><b>Λειτουργικότητα στο σύστημα IoT</b></p>			
<p>Σε αυτό το σύστημα ο αισθητήρας υγρασίας θα χρησιμοποιηθεί για την μέτρηση της υγρασίας του περιβάλλοντα χώρου.</p>			
3	Συσκευές IoT (MCU/IoT Devices)	Μητρική μικροεπεξεργαστή ESP32	Ολοκληρωμένο Σύστημα σε πυρίτιο (System on Chip)
<p><b>Γενική Λειτουργικότητα</b></p>			
<p>Οι μικροελεγκτές έχουν γνωρίσει μεγάλη εξέλιξη τα τελευταία χρόνια και έχουν περάσει από απλές αρχιτεκτονικές επεξεργαστών σε ολοκληρωμένα συστήματα πάνω σε τσιπς. Χαρακτηριστικό παράδειγμα είναι η εξέλιξη του AVR που εξελίχθηκε στην μητρική πλακέτα που είναι γνωστή σήμερα ως Arduino. Ταυτόχρονα, οι επεξεργαστές ARM έχουν κερδίσει αξιόλογο έδαφος στις φορητές εφαρμογές και στα ενσωματωμένα συστήματα με χαρακτηριστικό παράδειγμα αυτό του Raspberry Pi. Πλέον οι σημερινές μητρικές πλακέτες αναπτυξιακού περιβάλλοντος και τα ολοκληρωμένα τσιπάκια επεξεργαστών</p>			

<p>χρησιμοποιούνται,</p> <ol style="list-style-type: none"> <li>1. Σε εφαρμογές διαδικτύου των πραγμάτων για την συλλογή δεδομένων από αισθητήρες και την αποστολή τους στο διαδίκτυο.</li> <li>2. Σε εφαρμογές απομακρυσμένες για την συλλογή δεδομένων από αισθητήρες και την αποστολή τους μέσω του δικτύου κινητής τηλεφωνίας και του διαδικτύου.</li> <li>3. Σε συστήματα πραγματικού χρόνου για τον έλεγχο βιομηχανικών εφαρμογών ή ρομποτ.</li> <li>4. Σε οικιακές συσκευές για την διευκόλυνση του χρήστη και της διεπαφής μέσω κινητού τηλεφώνου από το δίκτυο Bluetooth.</li> </ol>			
<b>Λειτουργικότητα στο σύστημα IoT</b>			
<p>Στο σύστημα αυτό θα χρησιμοποιηθεί για την συλλογή δεδομένων από τους αισθητήρες θερμοκρασίας και υγρασίας. Πάνω στην μητρική θα συνδεθούν οι αισθητήρες θερμοκρασίας και υγρασίας παρέχοντας την απαραίτητη τάση και ρεύμα μέσα από το σύστημα τροφοδοσίας της μητρικής πλακέτας. Η μητρική πλακέτα που θα επιλεγεί φέρει ένα ολοκληρωμένο σύστημα IoT με συνδεσιμότητα στα ασύρματα δίκτυα WiFi.</p>			
4	Ακρογωνιαίο Δίκτυο (Edge network)	PC/laptop/RasberyPi	Ισχυροί επεξεργαστές x86_64, amd64, arm
<b>Γενική Λειτουργικότητα</b>			
<p>Το ακρογωνιαίο δίκτυο (Edge network) είναι κυρίως μια έννοια που προέρχεται από το υπολογιστικό “σύννεφο” και αναφέρεται σε υπολογιστικά συστήματα που συγκεντρώνουν πληροφορία από/πρός τα δίκτυα IoT με κατεύθυνση από και προς το υπολογιστικό “σύννεφο”. Σκοπός των ακρογωνιαίων υπολογιστικών συστημάτων είναι η αποσυμφόρηση του φόρτου εργασίας από τα χαμηλής επεξεργαστικής ισχύος κατώτερα επίπεδα όπως είναι οι IoT συσκευές αλλά και ταυτόχρονα η προώθηση μεγαλύτερων πακέτων δεδομένων προς το υπολογιστικό “σύννεφο” με σκοπό την καλύτερη εκμετάλλευση διαδικτυακών και υπολογιστικών πόρων προς το “σύννεφο”. Τα ακρογωνιαία υπολογιστικά συστήματα χρησιμοποιούνται σε,</p> <ol style="list-style-type: none"> <li>1. Εφαρμογές συλλογής και επεξεργασίας δεδομένων από συσκευές “διαδικτύου των πραγμάτων”.</li> <li>2. Εφαρμογές με υψηλές υπολογιστικές απαιτήσεις στα χαμηλότερα επίπεδα των συσκευών του διαδικτύου των πραγμάτων όπως είναι η επεξεργασία χιλιάδων χρονοσειρών.</li> <li>3. Εφαρμογές μηχανικής μάθησης και τεχνητής νοημοσύνης που απαιτούν την εκπαίδευση των μοντέλων εκτός του υπολογιστικού “σύννεφου”.</li> <li>4. Εφαρμογές πραγματικού χρόνου με σκοπό την επίτευξη της εκτέλεσης γρήγορων υπολογιστικών εργασιών σε μικρούς χρόνους.</li> </ol>			
<b>Λειτουργικότητα στο σύστημα IoT</b>			
<p>Στο σύστημα αυτό θα χρησιμοποιηθεί ως ακρογωνιαίο υπολογιστικό σύστημα ένας προσωπικός υπολογιστής (laptop) για την συλλογή δεδομένων των μετρήσεων και την αποσυμφόρηση της περιορισμένης μνήμης του μικροεπεξεργαστή καθώς και για την φιλοξενία του εξυπηρετητή αναμετάδοσης των μετρήσεων προς το υπολογιστικό νέφος.</p>			
5	Συσκευές IoT	SD card	Flash memory

	(MCU/IoT Devices)		
<b>Γενική Λειτουργικότητα</b>			
Οι κάρτες μνήμης SD card έχουν αρχίσει να εκλείπουν από τις προσωπικές συσκευές δεδομένου ότι οι σκληροί δίσκοι και οι μνήμες τυχαίας προσπέλασης έχουν εξελιχθεί και παρέχουν τεράστιες χωρητικότητες σε μικρά μεγέθη. Αυτό έχει επηρεάσει ακόμα και τα ενσωματωμένα συστήματα τα οποία και με την εμφάνιση του υπολογιστικού “σύννεφου” έχουν αρχίσει να προωθούν τα δεδομένα προς τα εκεί.			
<b>Λειτουργικότητα στο σύστημα IoT</b>			
Σε αυτό το υπολογιστικό σύστημα η SD card θα συνδεθεί στον μικροελεγκτή του συστήματος IoT με σκοπό την αποθήκευση των δεδομένων σε εξωτερική τοπική μνήμη για καταστάσεις έκτακτης ανάγκης και μη διαθεσιμότητας σύνδεσης με την εφαρμογή του υπολογιστικού “σύννεφου”..			
6	Ακρογωνιαίο Δίκτυο (Edge network)/ Internet	Δρομολογητής (Router)	OSI model – Επίπεδο 3
<b>Γενική Λειτουργικότητα</b>			
Οι δρομολογητές του διαδικτύου, σε όλους μας τα γνωστά ρούτερ αποτελούν την πύλη από το προσωπικό δίκτυο προς το δημόσιο διαδίκτυο μέσα από παρόχους διαδικτυακών συνδέσεων			
<b>Λειτουργικότητα στο σύστημα IoT</b>			
Σε αυτό το υπολογιστικό σύστημα χρησιμοποιείται ένας απλός οικιακός δρομολογητής παρέχοντας το προσωπικό δίκτυο στις συσκευές του διαδικτύου των πραγμάτων και την πρόσβαση στο διαδίκτυο για τον ακρογωνιαίο υπολογιστικό κόμβο με σκοπό την αποστολή των μετρήσεων στο υπολογιστικό “νέφος”.			

### 2.3.2 Λογισμικό

Στον Πίνακα 2-2 περιγράφεται κάθε στοιχείο που ανήκει στο λογισμικό του συστήματος IoT. Τα χαρακτηριστικά του περιέχουν το επίπεδο του, το όνομα του, την τεχνολογία του και τέλος την λειτουργικότητα του τόσο σε γενικό επίπεδο όσο και την συγκεκριμένη χρήση του στο πρωτότυπο της διπλωματικής αυτής.

Πίνακας 2-2 - Περιγραφή της λειτουργικότητας των επιμέρους υποσυστημάτων κατά την σχεδίαση σε επίπεδο λογισμικού (software)

<b>Λογισμικό</b>			
<b>A/A</b>	<b>Επίπεδο</b>	<b>Όνομα</b>	<b>Τεχνολογία</b>
1	Μικροελεγκτές/ Συσκευές IoT	Firmware	C/C++ κώδικας σε Arduino IDE

	(MCU/IoT Devices)		
<b>Γενική Λειτουργικότητα</b>			
<p>Το λογισμικό των μικροεπεξεργαστών και των ολοκληρωμένων συστημάτων σε τσιπ (SoC) ονομάζεται firmware. Το firmware είναι συνήθως γραμμένο σε γλώσσες προγραμματισμού χαμηλού επιπέδου όπως C/C++. Η διαδικασία αποσφαλμάτωσης και μεταγλώττισης του κώδικα είναι άρρηκτα συνδεδεμένη με την μητρική πλακέτα και την αρχιτεκτονική του εκάστοτε μικροεπεξεργαστή. Το περιβάλλον ανάπτυξης είναι ένας συνδυασμός ανάπτυξης κώδικα, ηλεκτρονικών συνδέσεων και δοκιμών πάνω στην μητρική πλακέτα.</p>			
<b>Λειτουργικότητα στο σύστημα IoT</b>			
<p>Σε αυτό το υπολογιστικό σύστημα το firmware έχει σκοπό</p> <ol style="list-style-type: none"> <li>1. Την αρχικοποίηση του συστήματος διαδικτύου των πραγμάτων.</li> <li>2. Την συνεχή εκτέλεση του κώδικα με συνθήκες λογικής και αποφάσεων του μικροελεγκτή σε ένα συνεχές loop.</li> <li>3. την συλλογή των τιμών των μετρήσεων από του αισθητήρες σε δυαδική μορφή λέξης.</li> <li>4. την επεξεργασία και αποθήκευση των τιμών στην κάρτα μνήμης.</li> <li>5. τον χρονο-προσδιορισμό των μετρήσεων</li> <li>6. την σύνδεση με τον εξυπηρετητή στον ακρογωνιαίο υπολογιστή</li> </ol> <p>την αποστολή των μετρήσεων στον ακρογωνιαίο εξυπηρετητή.</p>			
2	Ακρογωνιαίο Δίκτυο (Edge network)	Flask server	REST API
<b>Γενική Λειτουργικότητα</b>			
<p>Το πακέτο λογισμικού Flask είναι ένα πακέτο λογισμικού γραμμένο στην γλώσσα προγραμματισμού Python. Το πακέτο Flask είναι ένα λιτό αλλά ιδιαίτερα χρήσιμο πακέτο για την ανάπτυξη διαδικτυακών εφαρμογών μικρής κλίμακας όπως στις εφαρμογές του “διαδικτύου των πραγμάτων”. Το πακέτο flask είναι γραμμένο με βάση την λογική των templates και δεν έχει κάποιες συγκεκριμένες εξαρτήσεις από άλλα εργαλεία ή βιβλιοθήκες λογισμικού. Ωστόσο, παρέχει την δυνατότητα στον προγραμματιστή να υλοποιήσει και να χρησιμοποιήσει οποιοδήποτε πακέτο της Python με σκοπό την προσαρμογή της εφαρμογής του στις εκάστοτε απαιτήσεις. Έτσι το πακέτο Flask αποτελεί ένα πανίσχυρο εργαλείο καθώς η Python διαθέτει πληθώρα πακέτων για την υιοθέτησή τεχνικών ελέγχου της φόρμας των δεδομένων, χαρτογράφηση αντικειμένων, διαχείριση ανταλλαγής αρχείων, αυθεντικοποίηση και πολλά άλλα.</p>			
<b>Λειτουργικότητα στο σύστημα IoT</b>			
<p>Σε αυτό το υπολογιστικό σύστημα το πακέτο Flask χρησιμοποιήθηκε ως εξυπηρετητής για την συλλογή και διαχείριση των μετρήσεων. Ο εξυπηρετητής Flask από την μεριά του συστήματος του διαδικτύου των πραγμάτων IoT χρησιμοποιήθηκε ως συλλέκτης δεδομένων. Στην συνέχεια, ο εξυπηρετητής χρησιμοποιείται για την αποθήκευση των δεδομένων σε μια τοπική βάση για την απεικόνιση των μετρήσεων στον χρήστη τοπικά μέσω του φυλλομετρητή. Ταυτόχρονα, ο εξυπηρετητής Flask είναι υπεύθυνος για την</p>			

προώθηση των δεδομένων προς το υπολογιστικό σύννεφο.			
3	Ακρογωνιαίο Δίκτυο (Edge network)/ Internet	InfluxDB, Grafana	Βάση δεδομένων χρονοσειρών, επεξεργασία και απεικόνιση χρονοσειρών αντίστοιχα.
<b>Γενική Λειτουργικότητα</b>			
<p>Η InfluxDB είναι μια ανοικτού κώδικα βάση δεδομένων συγκεκριμένα για χρονοσειρές. Χρησιμοποιείται ιδιαίτερα για την αποθήκευση και συλλογή δεδομένων σε εφαρμογές παρακολούθησης λειτουργικών παραμέτρων, συλλογή δεδομένων από συστήματα αισθητήρων IoT και εφαρμογές μετρικών για την ανάλυση σε πραγματικό χρόνο.</p> <p>Το πακέτο Grafana είναι ένα ανοικτού κώδικα λογισμικό για την ανάλυση και διαδραστική απεικόνιση δεδομένων στο χρήστη. Διαθέτει μια μεγάλη ποικιλία γραφημάτων, τεχνικών απεικόνισης δεδομένων καθώς και ένα σύστημα ειδοποιήσεων μέσω διαδικτύου. Το πακέτο λογισμικού Grafana παρέχει ιδιαίτερη ευελιξία στο χρήστη καθώς μπορεί να συνδεθεί σε διάφορες βάσεις δεδομένων ή άλλες πηγές δεδομένων και ταυτόχρονα να δημιουργήσει μια πληθώρα πολύπλοκων διαδραστικών γραφημάτων για την απεικόνιση και κατανόηση των δεδομένων.</p>			
<b>Λειτουργικότητα στο σύστημα IoT</b>			
<p>Σε αυτό το υπολογιστικό σύστημα χρησιμοποιήθηκε η InfluxDB ως βάση δεδομένων για την αποθήκευση των χρονοσειρών από τη συλλογή των μετρήσεων από τους αισθητήρες. Στην συνέχεια, η InfluxDB συνδέεται με το πακέτο λογισμικού Grafana για την απεικόνιση και παρουσίαση των μετρήσεων στον χρήστη σε τοπικό επίπεδο.</p>			
4	Cloud App	ThingSpeak	Εφαρμογή Υπολογιστικού σύννεφου για την φιλοξενία μετρήσεων από IoT συστήματα.
<b>Γενική Λειτουργικότητα</b>			
<p>Η εφαρμογή υπολογιστικού “σύννεφου” ThingSpeak είναι ανοικτού κώδικα και επιτρέπει την αλληλεπίδραση των χρηστών με συσκευές του διαδικτύου των πραγμάτων. Η εφαρμογή αυτή παρέχει στους χρήστες της μια διεπαφή και τα κατάλληλα εργαλεία για την αυθεντικοποίηση της συσκευής και του χρήστη με σκοπό την αποστολή και συλλογή δεδομένων και προς τις δύο κατευθύνσεις με δικαιώματα τόσο εγγραφής όσο και ανάγνωσης των δεδομένων</p>			
<b>Λειτουργικότητα στο σύστημα IoT</b>			
<p>Σε αυτό το υπολογιστικό σύστημα χρησιμοποιείται κυρίως η αποστολή δεδομένων από τον εξυπηρετητή Flask στην εφαρμογή ThingSpeak. Η συλλογή των δεδομένων σε μια</p>			

εφαρμογή υπολογιστικού νέφους αποθηκεύει τα δεδομένα εξωτερικά από το σύστημα του διαδικτύου των πραγμάτων αυξάνοντας έτσι την προσβασιμότητα, την αποθήκευση και την ακεραιότητα των δεδομένων των μετρήσεων.			
5	Mobile App	ThingSpeak viewer	Anrdoid εφαρμογή
<b>Γενική Λειτουργικότητα</b>			
Το πακέτο λογισμικού ThingSpeak Viewer είναι μια εφαρμογή για κινητά με λογισμικό Android.			
<b>Λειτουργικότητα στο σύστημα IoT</b>			
Σε αυτό το σύστημα χρησιμοποιείται για την απεικόνιση των δεδομένων σε φορητή κινητή συσκευή και την προσβασιμότητα από το χρήστη στην πληροφορία των μετρήσεων που συλλέγει το σύστημα του διαδικτύου των πραγμάτων από οποιοδήποτε σημείο με κάλυψη σήματος κινητής τηλεφωνίας.			

## Κεφάλαιο 3ο: Υλοποίηση πρωτότυπου - Σύστημα IoT

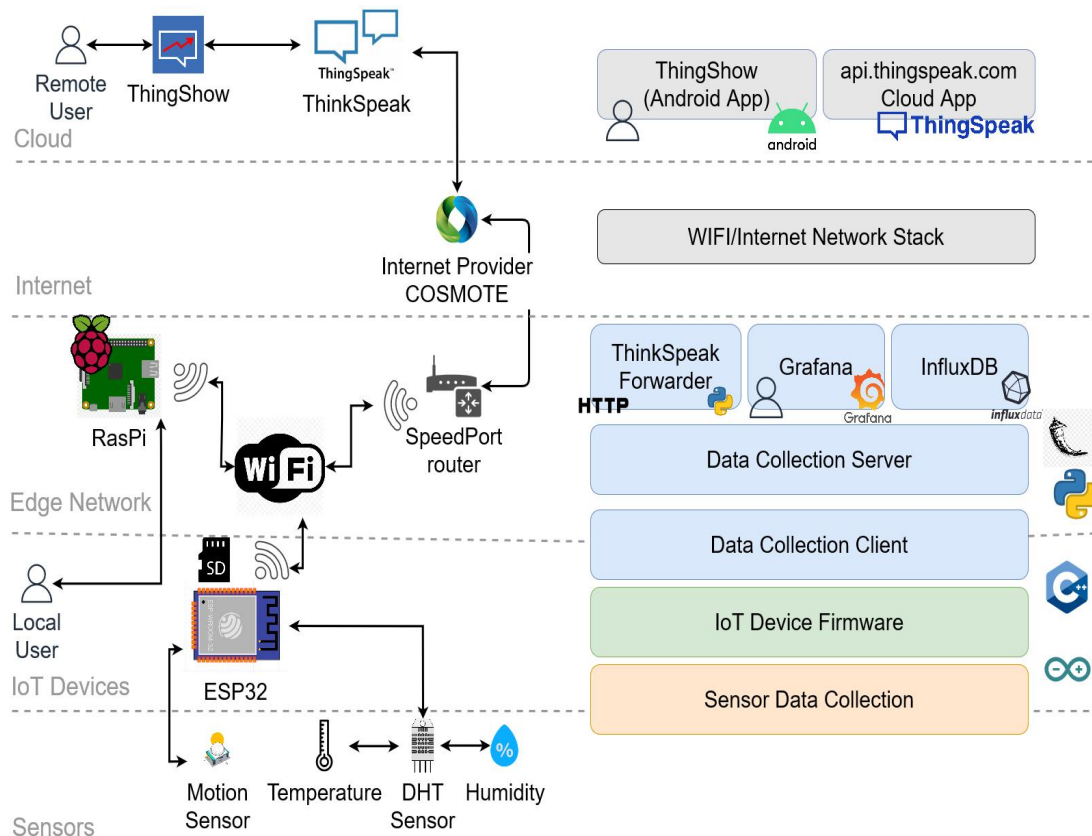
### 3.1 Εισαγωγή

Σε αυτή την ενότητα περιγράφεται η υλοποίηση του πρωτότυπου του συστήματος IoT δηλαδή η τελική αρχιτεκτονική υλοποίησης, ο εξοπλισμός που χρησιμοποιήθηκε και το λογισμικό που αναπτύχθηκε σε κάθε στάδιο του συστήματος του “διαδικτύου των πραγμάτων”-IoT.

Η περιγραφή της υλοποίησης του συστήματος IoT ξεκινάει με την παρουσίαση του τελικού μπλοκ διαγράμματος της υλοποιημένης αρχιτεκτονικής του συστήματος IoT όπως φαίνεται στην Εικόνα 3-1 και στην συνέχεια βασίζεται στην περιγραφή της υλοποίησης των επιμέρους υποσυστημάτων του τελικού πρωτότυπου του συστήματος IoT τόσο σε επίπεδο λογισμικού όσο και στο επίπεδο του υλικού στην ενότητα 3.3

### 3.2 Συνολικό διάγραμμα αρχιτεκτονικής υλοποίησης

Η συνολική αρχιτεκτονική του υλοποιημένου συστήματος του διαδικτύου των πραγμάτων φαίνεται στην Εικόνα 3-1.



Εικόνα 3-1 - Η αρχιτεκτονική της τελικής υλοποίησης του συστήματος IoT σε σχέση με την αρχιτεκτονική σχεδίαση. Αριστερά: Απεικονίζονται τα εικονίδια των τελικών επιλογών του υλικού. Δεξιά: Απεικονίζεται η στοίβα του λογισμικού με τα εικονίδια των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη των επιμέρους υποσυστημάτων της εφαρμογής IoT

Η Εικόνα 3-1 είναι άρρηκτα συνδεδεμένη με το μπλοκ διάγραμμα της αρχιτεκτονικής κατά την σχεδίαση (Εικόνα 2-1) δεδομένου ότι έχουν εφαρμοστεί οι σχεδιαστικές επιλογές που περιγράφονται στο κεφάλαιο της σχεδίασης. Ωστόσο, το διάγραμμα της Εικόνα 3-1 διαφέρει γιατί αποτελεί το τελικό

υλοποιημένο μπλοκ διάγραμμα του πρωτότυπου με βάση την τελική υλοποίηση. Το διάγραμμα της Εικόνας 3-1 διαθέτει λεπτομέρειες για όλα τα εργαλεία, τα πακέτα λογισμικού καθώς και τον εξοπλισμό που είτε αναπτύχθηκε είτε χρησιμοποιήθηκε κατά την υλοποίηση του συστήματος του διαδικτύου των πραγμάτων.

### 3.3 Περιγραφή υλοποίησης των επιμέρους υποσυστημάτων

Σε αυτή την ενότητα γίνεται η περιγραφή και ανάλυση της υλοποίησης των επιμέρους υποσυστημάτων του τελικού συστήματος του διαδικτύου των πραγμάτων. Για την καλύτερη οργάνωση, αυτή η ανάλυση και περιγραφή της υλοποίησης θα βασιστεί στο διάγραμμα της υλοποιημένης αρχιτεκτονικής στην Εικόνα 3-1.

#### 3.3.1 Υλικό

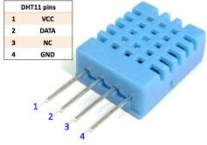
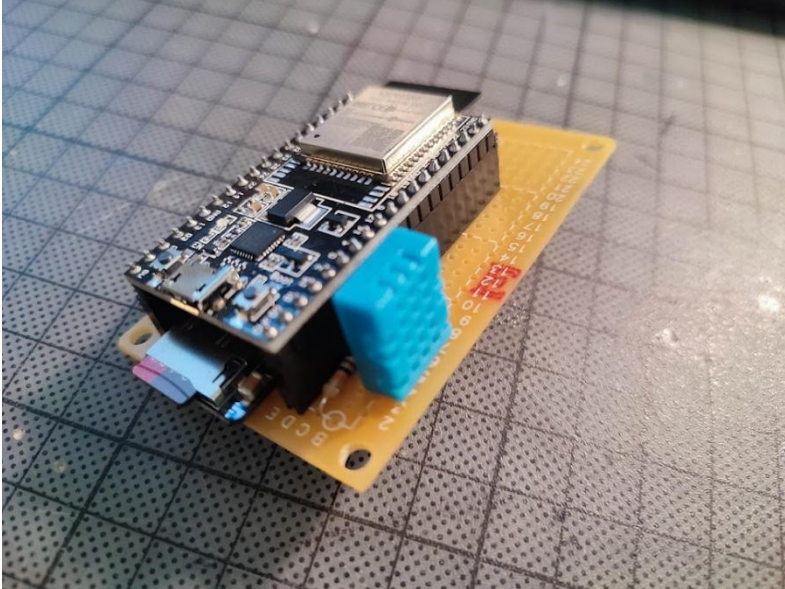
Σε αυτή την υποενότητα περιγράφονται τα επιμέρους συστήματα του πρωτότυπου που σχετίζονται μόνο με το “υλικό” (hardware).

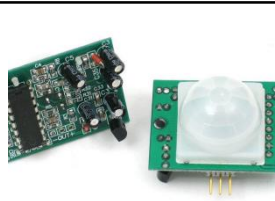
##### 3.3.1.1 Αισθητήρες

Στον Πίνακα 3-1 παρουσιάζονται οι αισθητήρες που χρησιμοποιήθηκαν στο πρωτότυπο της συσκευής IoT.

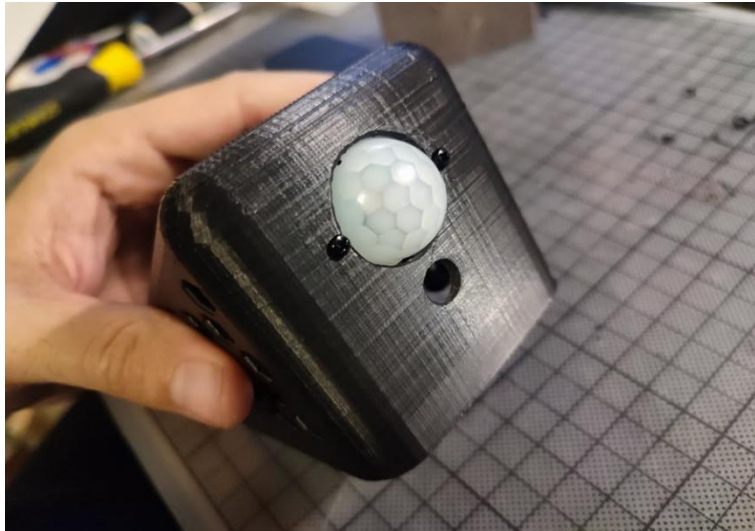
Πίνακας 3-1 - Περιγραφή των αισθητήρων που χρησιμοποιήθηκαν στο πρωτότυπο της συσκευής IoT.

#	Όνομα αισθητήρα	Περιγραφή
1	DHTXX	Στην Αισθητήρας DHT11 φαίνεται ο αισθητήρας DHTXX. Ο αισθητήρας DHT είναι ουσιαστικά ένας συνδυασμός δύο αισθητήρων. Ο ένας αισθητήρας χωρητικότητας που μετράει σχετική υγρασία και ο άλλος είναι ένα θερμίστορ δηλαδή μια μη γραμμική αντίσταση που μεταβάλλεται με την αλλαγή της θερμοκρασίας. Ο αισθητήρας DHT διαθέτει επίσης και ένα τσιπάκι το οποίο χρησιμοποιείται για την μετατροπή των αναλογικών τιμών σε ψηφιακές και στην συνέχεια την μετάδοση των μετρήσεων σε μορφή ψηφιακής λέξης προς τον μικροεπεξεργαστή ESP32. Η θέση του αισθητήρα προς την μητρική του μικροεπεξεργαστή ESP32 φαίνεται στην Εικόνα 3-3
Εικόνα		Εικόνα στο σύστημα IoT

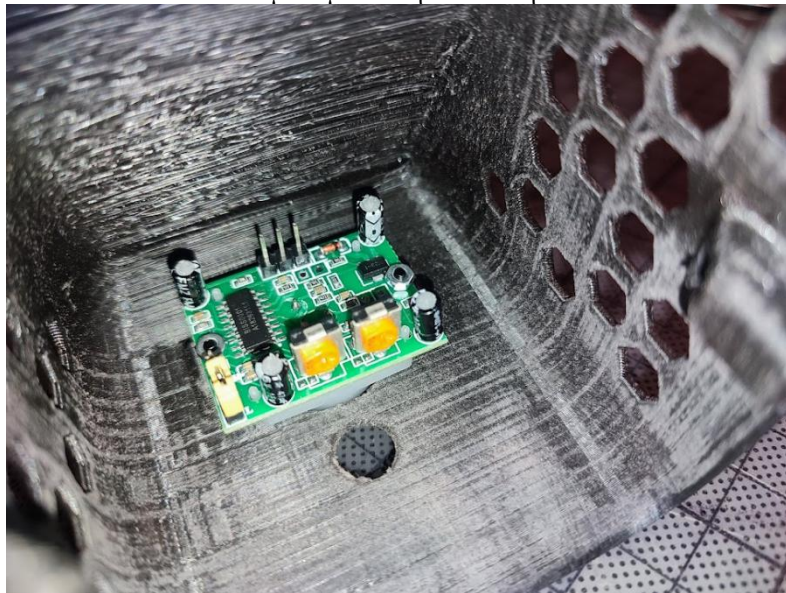
 <p>Εικόνα 3-2 - Αισθητήρας DHT11</p>	 <p>Εικόνα 3-3- Αισθητήρας DHT11 ενσωματωμένος στην πλακέτα της συσκευής IoT.</p>
<p>2 PIR</p>	<p>Στις Εικόνα 3-4, Εικόνα 3-5, Εικόνα 3-6 φαίνεται ο αισθητήρας PIR είναι ένα μικρό-σύστημα ανίχνευσης κίνησης. Ο αισθητήρας αυτός διαθέτει μια πυροηλεκτρική επιφάνεια ευαίσθητη στην υπέρυθη ακτινοβολία. Όλα τα θερμά σώματα εκπέμπουν υπέρυθη ακτινοβολία και αυτό διεγείρει την πυροηλεκτρική επιφάνεια. Επειδή αυτός ο αισθητήρας προσπαθεί να εντοπίσει κίνηση και όχι να μετρήσει την ακτινοβολία χωρίζεται σε δύο μέρη. Τα δύο αυτά μέρη είναι ηλεκτρικά συνδεδεμένα έτσι ώστε σε κατάσταση ηρεμίας τα αλληλοαναιρούνται. Έτσι εάν ένα από τα δύο μισά εκτεθεί σε μικρότερη ή μεγαλύτερη υπέρυθη ακτινοβολία τότε η έξοδος της πυροηλεκτρικής διάταξης θα μεταβεί σε μια ψηφιακή κατάσταση “υψηλού” ή “χαμηλού” δυναμικού, εντοπίζοντας έτσι την κίνηση του ανθρώπου μπροστά από τον αισθητήρα.</p>
<p>Εικόνα</p>	<p>Εικόνα στο σύστημα IoT</p>



Εικόνα 3-4 -  
Αισθητήρας Εντοπισμού  
Κίνησης (PIR)



Εικόνα 3-5 - Προφίλ αισθητήρα Εντοπισμού Κίνησης (PIR)  
ενσωματωμένο στην συσκευή IoT



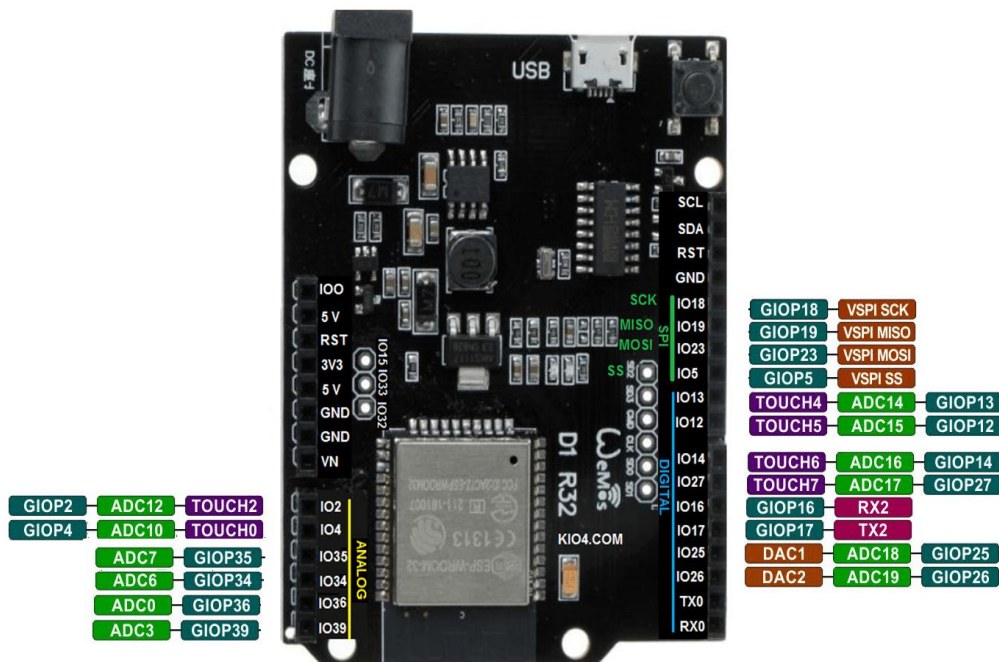
Εικόνα 3-6 - Εσωτερική πίσω όψη του αισθητήρα Εντοπισμού Κίνησης  
(PIR) στην συσκευή IoT

### 3.3.1.2 Συσκευή “διαδικτύου των πραγμάτων”(IoT) - ESP32

Στον Εικόνα 3-7 και Εικόνα 3-8 φαίνεται η υλοποίηση του πυρήνα του συστήματος IoT δηλαδή η μητρική πλακέτα που χρησιμοποιήθηκε για την ανάπτυξη και υλοποίηση του συστήματος IoT. Η μητρική πλακέτα έχει πάνω της έναν μικροεπεξεργαστή Tensilica Xtensa LX6 που είναι ενσωματωμένος στο μικροελεγκτή ESP32.

Για την εκπόνηση της διπλωματικής αυτής χρησιμοποιήθηκαν δύο διαφορετικές μητρικές πλακέτες που φέρουν τον ίδιο επεξεργαστή (ESP32). Η μία πλακέτα ονομάζεται Wemos D1 R32 και φαίνεται στην Εικόνα 3-7. Η άλλη μητρική πλακέτα ονομάζεται Espressif ESP32-WROOM-D3, χρησιμοποιήθηκε κυρίως για την εγκατάσταση της στο πρωτότυπο της συσκευής IoT και φαίνεται στην Εικόνα 3-8.

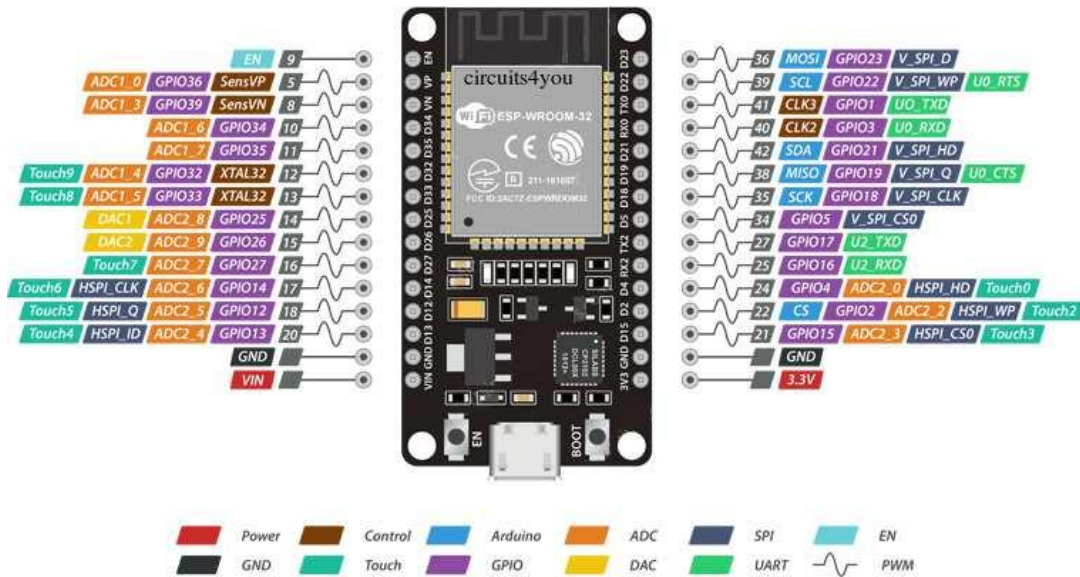
Η πλακέτα Wemos D1 R32 βασίζεται στον μικροεπεξεργαστή ESP32 ο οποίος ενσωματώνει και τις λειτουργίες δικτύου WiFi και Bluetooth. Η συσκευή διαθέτει ένα σύνολο εισόδων/εξόδων και το σχηματικό της διάγραμμα μοιάζει με αυτό του Arduino Uno. Ο Arduino Uno αποτελεί μία από τις πιο ευρέως διαδεδομένες συσκευές στην κοινότητα των ηλεκτρονικών τόσο για επαγγελματίες όσο και για ερασιτέχνες.



Εικόνα 3-7 - Μητρική πλακέτα Wemos D1 R32 που χρησιμοποιήθηκε για την ανάπτυξη της συσκευής IoT.

Στην Εικόνα 3-7, φαίνεται ο αριθμός των ψηφιακών και αναλογικών εισόδων και εξόδων της συσκευής Wemos D1 R32. Η συσκευή αυτή μπορεί να προγραμματιστεί μέσω της θύρας micro usb που διαθέτει καθώς και το ενσωματωμένο FTDI chipset. Τέλος, η συσκευή μπορεί να λάβει ρεύμα είτε μέσω του power barrel είτε μέσω της θύρας micro usb είτε με τη χρήση μπαταρίας από τα αντίστοιχα pin.

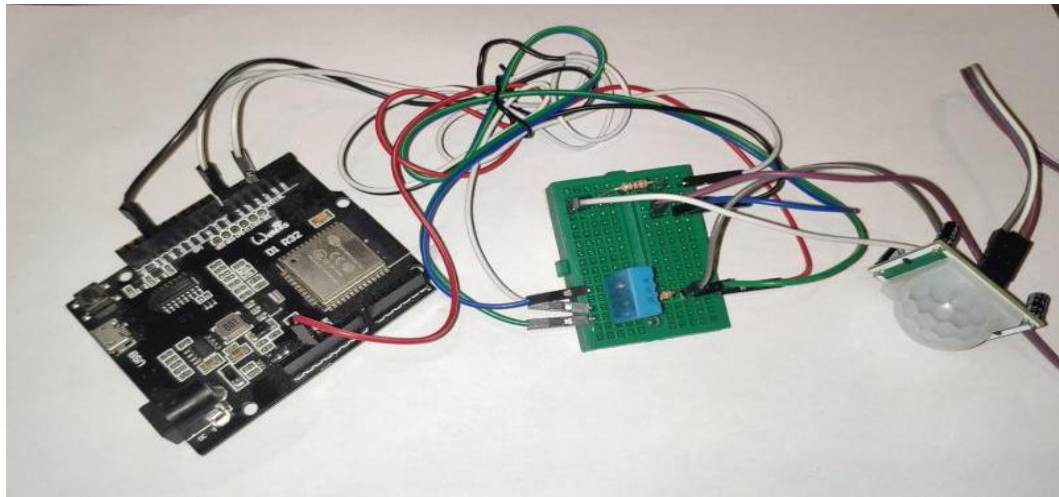
Στην Εικόνα 3-8 παρουσιάζεται η άλλη μητρική πλακέτα ανάπτυξης Espressif ESP32-WROOM-D32 που εγκαταστήθηκε στην συσκευή IoT. Η πλακέτα αυτή διαθέτει έναν μικροεπεξεργαστή ESP32 και αποτελεί την επιλογή πολλών ηλεκτρονικών μηχανικών εξαιτίας του μειωμένου μεγέθους της και της χαμηλής τιμής της. Η συσκευή μπορεί να λάβει ρεύμα μέσω της θύρας micro usb ή μέσω των αντίστοιχων pin. Στην Εικόνα 3-8 φαίνονται οι ψηφιακές και αναλογικές εισοδοι και εξοδοι της συσκευής.



Εικόνα 3-8 -Μητρική πλακέτα Espressif ESP32-WROOM-D32 που χρησιμοποιήθηκε τόσο για την ανάπτυξη αλλά κυρίως για την τελική της ενσωμάτωση στο πρωτότυπο της συσκευής IoT

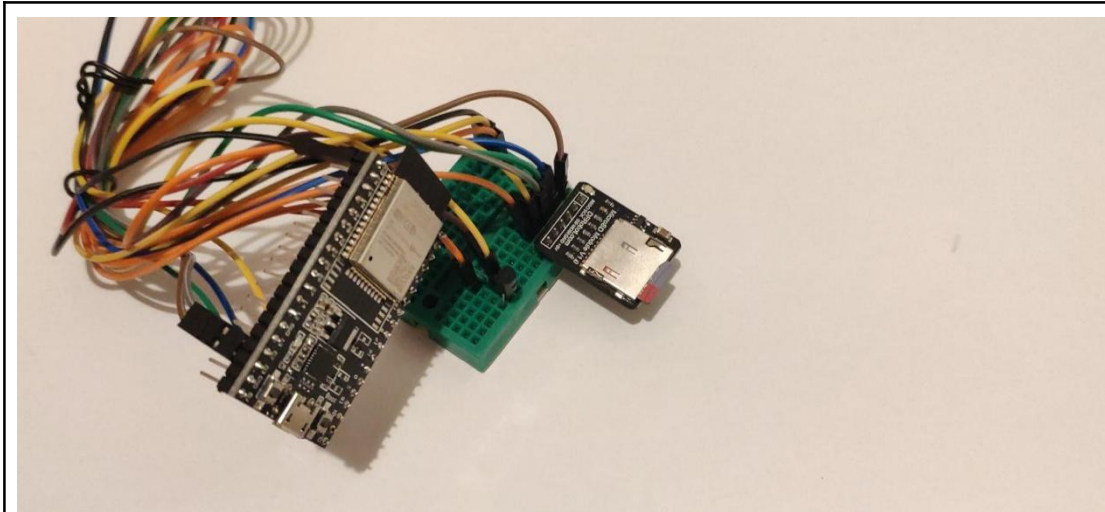
Στο Πίνακα 3-2 παρουσιάζονται διάφορα στάδια κατά την ανάπτυξη του πρωτότυπου συστήματος της συσκευής IoT.Τα στάδια αυτά περιλαμβάνουν τα αρχικά στάδια της βασικής σύνδεσης των αισθητήρων, την κάρτα μνήμης, την κόλληση της πλακέτας, την εκτύπωση του κουτιού της συσκευής IoT με 3d printer<sup>1</sup>, την συναρμολόγηση και το τελικό αποτέλεσμα.

Πίνακας 3-2 - Περιγραφή και εικόνες των σταδίων ανάπτυξης της μητρικής πλακέτας της συσκευής IoT.

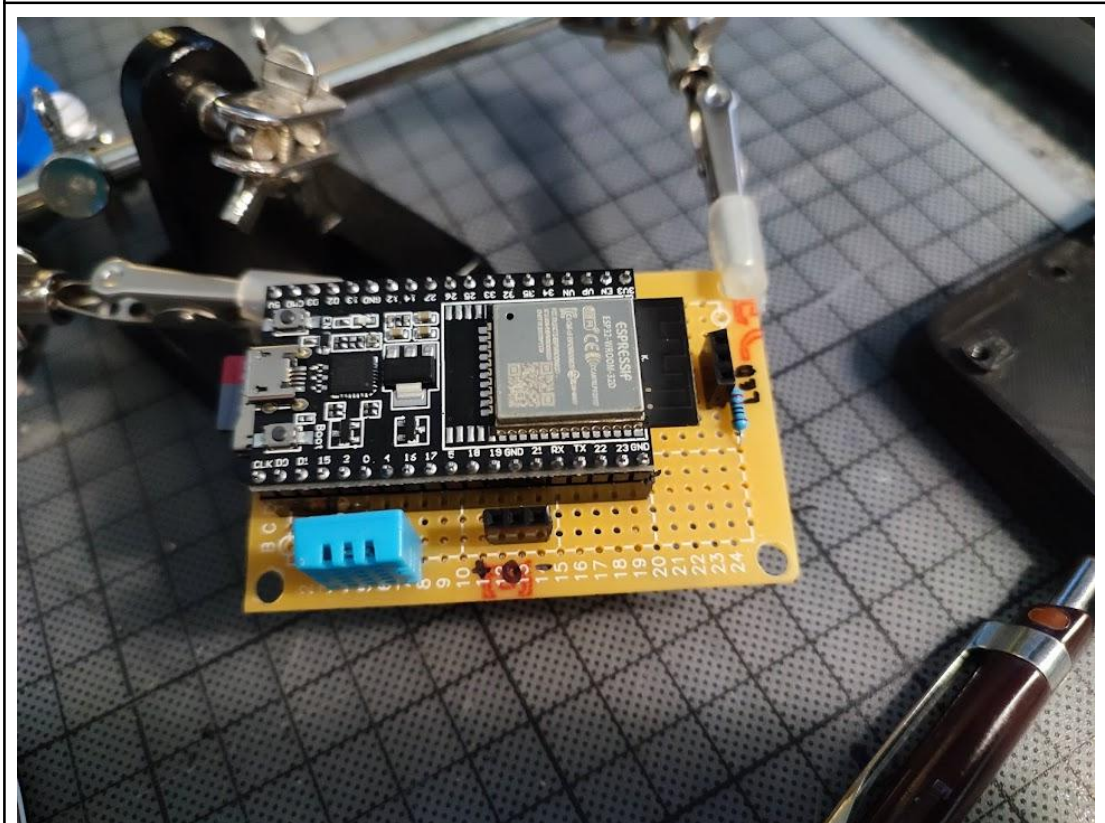


Εικόνα 3-9 - Συνδεσμολογία του αισθητήρα DHT και PIR κατά την ανάπτυξη του κώδικα για την συλλογή των μετρήσεων με την μητρική πλακέτα Wemos D1 R32

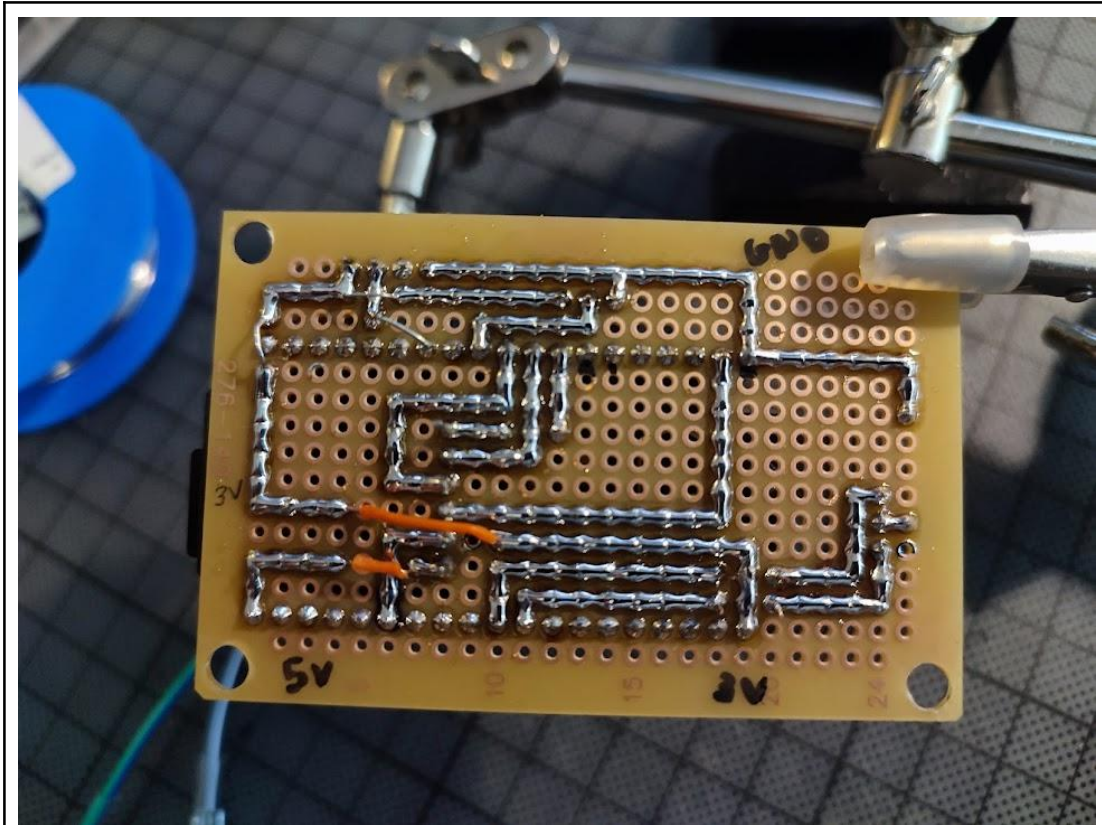
<sup>1</sup> Για το σχέδιο της συσκευής σε CAD και ο 3d εκτυπωτής είναι φιλική συνεισφορά στην εργασία αυτή.



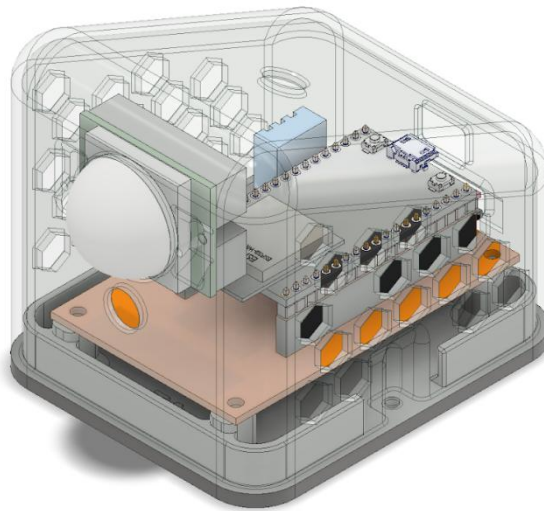
Εικόνα 3-10 - Συνδεσμολογία της μητρικής πλακέτας Espressif ESP32-WROOM-D32 με την κάρτα μνήμης κατά την ανάπτυξη του κώδικα αποθήκευσης των δεδομένων των μετρήσεων στην εξωτερική κάρτα μνήμης



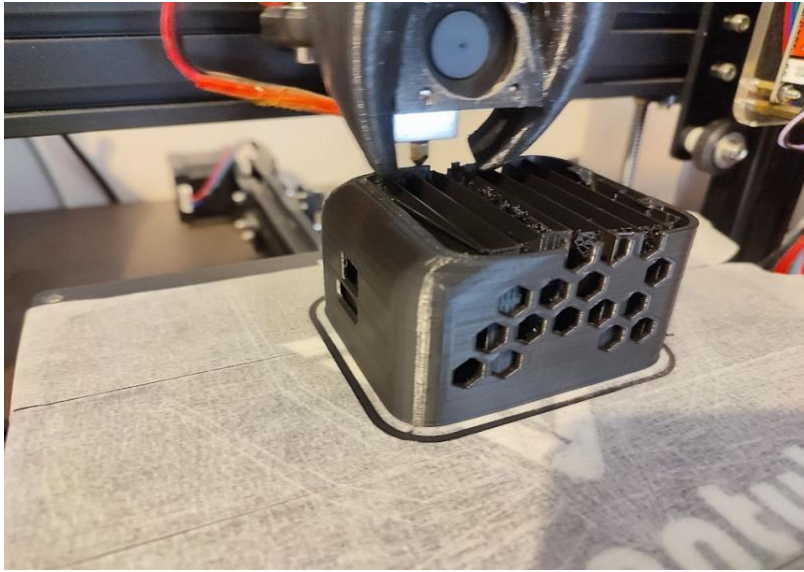
Εικόνα 3-11 - Συναρμολόγηση και κόλληση της τελικής πλακέτας του πρωτότυπου συστήματος IoT. Σε αυτή την εικόνα φαίνεται η πάνω πλευρά της πλακέτας με τον μικροελεγκτή ESP32, τους αισθητήρες και την κάρτα μνήμης



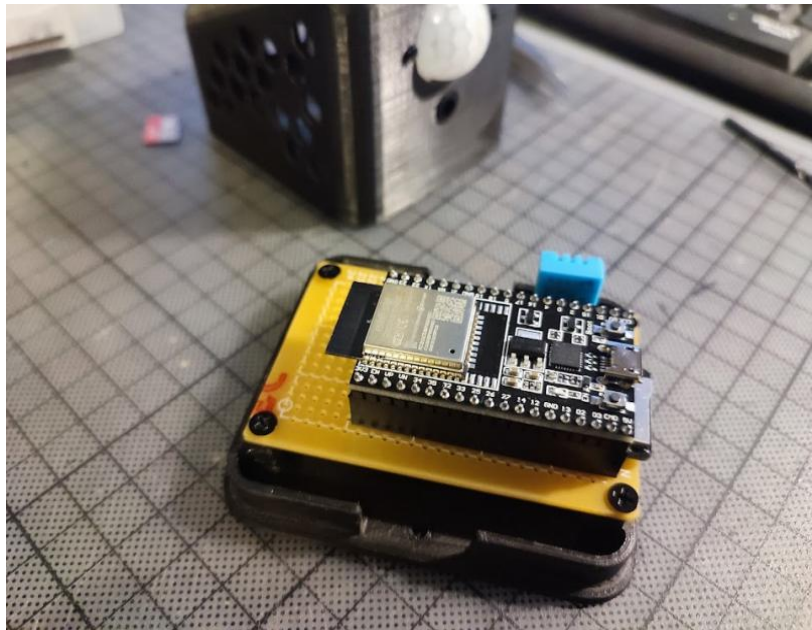
Εικόνα 3-12 - Κόλληση της κάτω πλευράς των συνδέσεων της τελικής μητρικής πλακέτας του συστήματος IoT



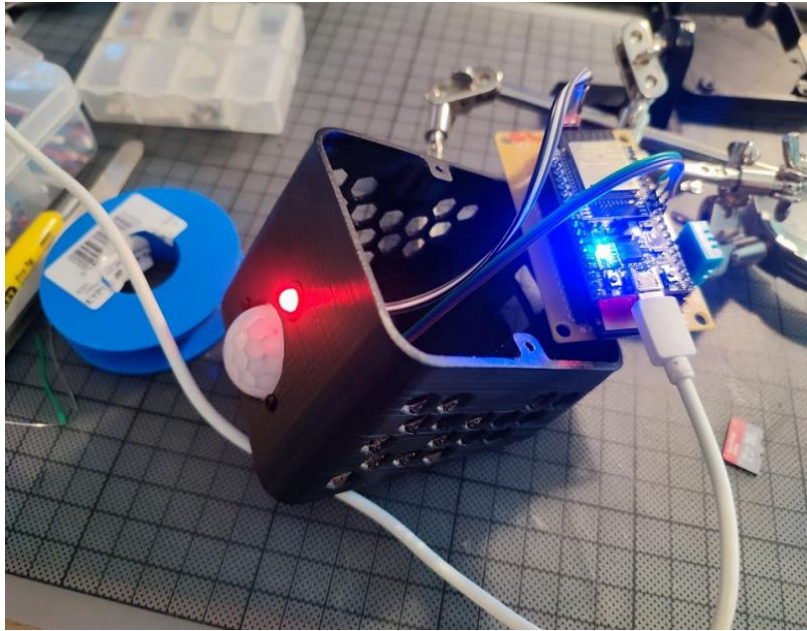
Εικόνα 3-13 - 3D σχέδιο του κουτιού της τελικής όψης του συστήματος IoT σε σχεδιαστικό πρόγραμμα CAD.



Εικόνα 3-14 - 3D εκτύπωση του τελικού κουτιού του πρωτότυπου υλοποίησης του συστήματος IoT.



Εικόνα 3-15 - Τοποθέτηση της τελικής πλακέτας στη βάση του κουτιού της συσκευής IoT και σύνδεση των εσωτερικών συνδέσεων των αισθητήρων και της τροφοδοσίας.



Εικόνα 3-16 - Εγκατάσταση και συνδεσμολογία του αισθητήρα εντοπισμού κίνησης (PIR) με το πάνω μέρος της συσκευής.



Εικόνα 3-17 - Τροφοδότηση και δοκιμές λειτουργίας της συσκευής IoT.



Εικόνα 3-18 - Πλάγια όψη της συσκευή IoT. Οι κυψελωτές τρύπες στο πλάι του κουτιού της συσκευή έχουν σκοπό την άμεση επαφή του αισθητήρα υγρασίας και θερμοκρασίας με τον αέρα του περιβάλλοντα χώρου.

### 3.3.1.3 Ακρογωνιαίο Δίκτυο - Edge Network

Στο Πίνακα 3-3 φαίνεται ο εξοπλισμός που συνθέτει το ακρογωνιαίο επίπεδο στην υλοποίηση του πρωτότυπου συστήματος IoT.

Πίνακας 3-3 - Περιγραφή και απεικόνιση του υλικού που χρησιμοποιήθηκε για την υλοποίηση του επιπέδου του “ακρογωνιαίου” δικτύου(Edge network)



Εικόνα 3-19 - Μητρική πλακέτα Raspberry Pi. Χρησιμοποιήθηκε ως υπολογιστής για να φιλοξενήσει τον εξυπηρετητή συλλογής, αποθήκευσης και αποστολής των μετρήσεων.

Στην Εικόνα 3-19 φαίνεται η μητρική πλακέτα Raspberry Pi που χρησιμοποιήθηκε ως ακρογωνιαίος “υπολογιστής” για την επεξεργασία, συλλογή, αποθήκευση και αποστολή των μετρήσεων σε τοπικό επίπεδο. Στο Raspberry Pi φιλοξενείται,

1. το λογισμικό του εξυπηρετητή για την συλλογή των μετρήσεων από το ESP32.
2. η βάση δεδομένων για χρονοσειρές InfluxDB για την αποθήκευση των μετρήσεων.
3. το πακέτο λογισμικού διεπαφής χρήστη Grafana για την απεικόνιση των μετρήσεων.
4. το πακέτο λογισμικού για την αποστολή των μετρήσεων στο ThnigSpeak.



Εικόνα 3-20 - Οικιακός δρομολογητής (router). Αποτελεί την πύλη για την αποστολή δεδομένων από την συσκευή IoT προς το υπολογιστικό “σύννεφο”.

Στην Εικόνα 3-20 φαίνεται ο δρομολογητής (router) που ανήκει στον εξοπλισμό του παρόχου πρόσβασης στο διαδίκτυο σε οικιακά περιβάλλοντα. Ο δρομολογητής δεν αποτελεί κομμάτι του πρωτότυπου συστήματος IoT ωστόσο έγιναν κάποιες ρυθμίσεις με σκοπό την εύρυθμη λειτουργία του συστήματος IoT στο οικιακό-προσωπικό δίκτυο. Οι ρυθμίσεις είναι,

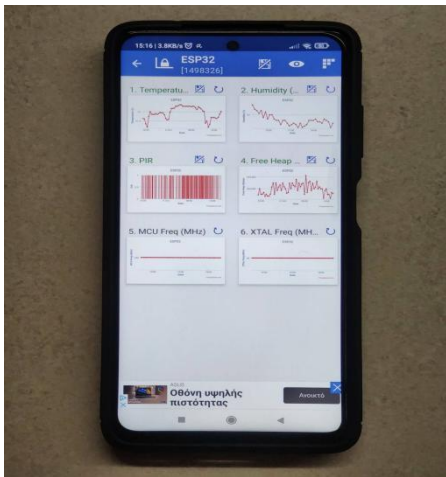
1. Ανάθεση στατικής διεύθυνσης IP στην μητρική Raspberry Pi .
2. Ανάθεση στατικής διεύθυνσης IP στην μητρική του ESP32.

Η χρήση της στατικής διεύθυνσης IP διευκόλυνε την ανάπτυξη και αδιάκοπη λειτουργία του συστήματος IoT μέσα στο προσωπικό οικιακό δίκτυο.

### 3.3.1.4 Πρόσβαση στο υπολογιστικό “σύννεφο” (Cloud) - Φορητή συσκευή Android

Στην Εικόνα 3-21 φαίνεται το “έξυπνο” κινητό τηλέφωνο λογισμικού Android ωστόσο θα μπορούσε να χρησιμοποιηθεί οποιαδήποτε φορητή συσκευή με πρόσβαση στο διαδίκτυο όπως τάμπλετ ή φορητός υπολογιστή (laptop).

Πίνακας 3-4 - Φορητή συσκευή - "Έξυπνο" κινητό τηλέφωνο με λογισμικό Android



Εικόνα 3-21 - Φορητή συσκευή (κινητό Android) με εγκατεστημένη την εφαρμογή ThingShow για την απομακρυσμένη πρόσβαση τα δεδομένα των μετρήσεων μέσω της εφαρμογής υπολογιστικού “νέφους” ThingSpeak.

Το “έξυπνο” κινητό με λογισμικό Android και εγκατεστημένη την εφαρμογή ThingShow για την απεικόνιση των δεδομένων των μετρήσεων από οποιοδήποτε εξουσιοδοτημένο απομακρυσμένο χρήστη. Μπορεί να χρησιμοποιηθεί σε οποιαδήποτε συσκευή μπορεί να εγκατασταθεί η εφαρμογή ThingShow

### 3.3.2 Λογισμικό

Σε αυτή την υποενότητα περιγράφονται τα επιμέρους συστήματα του πρωτότυπου που σχετίζονται μόνο με το λογισμικό (software).

#### 3.3.2.1 Λογισμικό συσκευής IoT - Firmware ESP32

Ο κώδικας για το λογισμικό του επεξεργαστή ESP32 βρίσκεται στο ΠΑΡΑΡΤΗΜΑ Α και είναι υπεύθυνο για τις εξής λειτουργίες,

1. Αρχικοποίηση του μικροεπεξεργαστή, των περιφερειακών μονάδων του και η σύνδεση στο τοπικό δίκτυο WiFi. Αυτό πραγματοποιείται μέσω της συνάρτησης `setup()` και των αρχικών `global` μεταβλητών.
2. Εκτέλεση ενός συνεχούς βρόχου ο οποίος υλοποιείται με την εκτέλεση της συνάρτησης `loop()`,
  - a. Η συλλογή των μετρήσεων από τους αισθητήρες.
  - b. Η συλλογή των μετρήσεων από εσωτερικούς καταχωρητές.
  - c. Ο χρόνο- προσδιορισμός των μετρήσεων με το πρωτόκολλο NTP.
  - d. Η αποθήκευση των δεδομένων στην κάρτα μνήμης.
  - e. Ο επανέλεγχος της κατάστασης σύνδεσης στο τοπικό δίκτυο, όπου εάν είναι επιτυχής συνεχίζει με την αποστολή των δεδομένων προς τον εξυπηρετητή συλλογής δεδομένων Flask.

Για τον προγραμματισμό των μητρικών ESP32 χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον προγραμματισμού Arduino IDE [10]. Το Arduino IDE χρησιμοποιείται για τον προγραμματισμό μιας πληθώρας συσκευών και επιτρέπει στον προγραμματιστή να επαναχρησιμοποιήσει εν μέρει τον κώδικα του σε διαφορετικών χαρακτηριστικών συσκευές (π.χ. συστήματα αρχιτεκτονικής IA-32, x86-64, ARM). Ο προγραμματιστής δύναται να γράψει κώδικα στις γλώσσες προγραμματισμού C/C++ κάνοντας το ιδιαίτερα δημοφιλές στην κοινωνία των προγραμματιστών. Τόσο η άδεια χρήσης του (GNU GPL) όσο και η δημοτικότητα του οδήγησαν εταιρείες κατασκευής συσκευών να δημιουργούν βιβλιοθήκες για την υποστήριξη των συσκευών τους από το Arduino IDE.

Οι μητρικές ESP32 που χρησιμοποιήθηκαν φέρουν τον ίδιο μικροεπεξεργαστή και για τον προγραμματισμό τους μοιράζονται τα ίδια χαρακτηριστικά σύνδεσης/προγραμματισμού. Οι βιβλιοθήκες που χρησιμοποιήθηκαν για τη λειτουργία τόσο των αισθητήρων όσο και των πρωτοκόλλων μεταβίβασης πληροφορίας είναι,

- **ESP32 Dev Module:** αφορά τα βασικά χαρακτηριστικά του επεξεργαστή ESP32 όπως η διαχείριση του WiFi, αναλογικά και ψηφιακά κανάλια επικοινωνίας, μνήμη προγράμματος, κτλ.
- **WiFi:** Η βιβλιοθήκη αυτή παρέχει όλες τις συναρτήσεις για την σύνδεση του μικροεπεξεργαστή ESP32, ESP8266 σε ασύρματα δίκτυα WiFi.
- **ArduinoJSON** επιτρέπει την δημιουργία και ανάγνωση δομών δεδομένων τύπου JSON. Η χρήση κωδικοποίησης JSON είναι ευρέως διαδεδομένη καθώς επιτρέπει τη δομημένη οργάνωση της πληροφορίας με τρόπο ώστε να γίνεται κατανοητό τόσο από τους ανθρώπους όσο και από τις μηχανές.
- **HTTIClient** Η βιβλιοθήκη αυτή επιτρέπει την δημιουργία και διαχείριση αιτημάτων του πρωτοκόλλου HTTP. Μέσα από την βιβλιοθήκη αυτή ήταν δυνατή η επικοινωνία της συσκευής IoT με τον εξυπηρετητή συλλογής δεδομένων. Η συσκευή IoT αποστέλλει τα δεδομένα με ένα αίτημα POST προς τον εξυπηρετητή.

- **DHT** Η βιβλιοθήκη αυτή περιέχει συναρτήσεις και εντολές δυαδικού επιπέδου (εγγραφή στη μνήμη του αισθητήρα, διάβασμα από τη μνήμη του αισθητήρα, αποκωδικοποίηση του μηνύματος, κ.α.). Η βιβλιοθήκη αυτή χρησιμοποιήθηκε για τη λήψη των μετρήσεων θερμοκρασίας και υγρασίας μέσω του αισθητήρα DHT 11 .
- **Wire** η βιβλιοθήκη αυτή περιέχει συναρτήσεις απαραίτητες για την επικοινωνία συσκευών μέσω του πρωτοκόλλου I2C. Διάλογοι επικοινωνίας I2C χρησιμοποιήθηκαν από την συσκευή IoT για την καταγραφή των μετρήσεων στην κάρτα μνήμης.

Στο Πίνακα 3-5 βλέπουμε τη δομή του JSON μηνύματος που αποστέλλεται στον εξυπηρετητή συλλογής δεδομένων. Επιλέχθηκε αυτή η μορφή γιατί όπως φαίνεται είναι επεκτάσιμη και μπορεί εύκολα να προστεθούν νέες μετρήσεις στην περίπτωση σύνδεσης και άλλου αισθητήρα στον ESP32.

Πίνακας 3-5 - Μορφή της δομής δεδομένων JSON για την οργανωμένη συλλογή, αποθήκευση και αποστολή των δεδομένων των μετρήσεων.

```

{
1. "device": "DEVICE1",
2. "api_key": "paras2021",
3. "temperature": 23.5,
4. "humidity": 50,
5. "PIR": 1,
6. "esp_time": 2021-09-30T03:21:44,
7. "sd_total": 15185,
8. "sd_used": 251,
9. }
    
```

### 3.3.2.2 Ακρογωνιαίο Δίκτυο - Edge Network

Ο κώδικας για το λογισμικό της μητρικής πλακέτας Raspberry Pi είναι χωρισμένος στις εξής λειτουργίες,

1. **Εξυπηρετητής Flask:** Αποτελεί την υλοποίηση του εξυπηρετητή συλλογής των δεδομένων των μετρήσεων από το ESP32. Ο εξυπηρετητής flask είναι ένας HTTP εξυπηρετητής που διαθέτει 2 σημεία πρόσβασης στα δεδομένα των μετρήσεων.

Πίνακας 3-6 - Σημεία πρόσβασης του εξυπηρετητή συλλογής δεδομένων μέσω του πρωτοκόλλου HTTP.

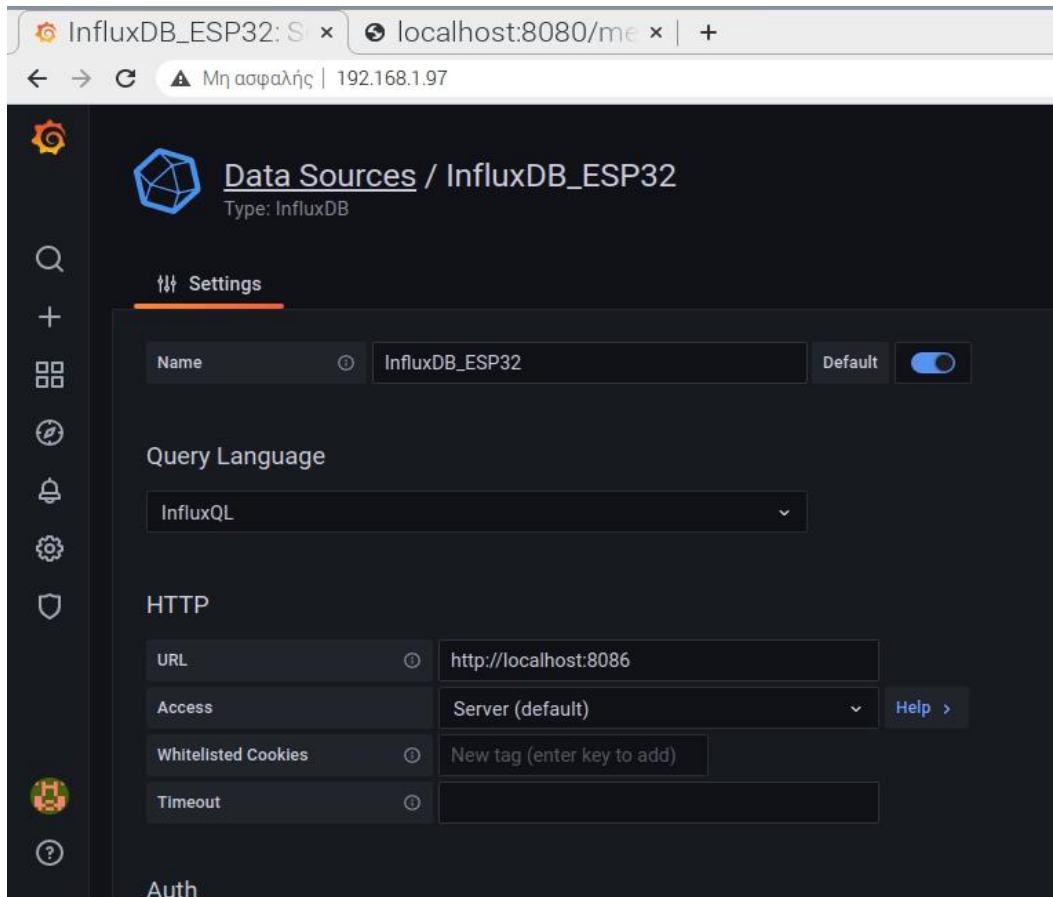
Μέθοδος	Διεύθυνση	Ορίσματα	Περιγραφή
GET	http://<διεύθυνση IP:8080>/measurements	Κανένα όρισμα.	Το σημείο αυτό επιστρέφει τις 100 τελευταίες μετρήσεις που έχουν αποθηκευτεί από τον εξυπηρετητή.
POST	http://<διεύθυνση IP:8080>/addvalue	Body = ”{ "device": "DEVICE1", "api_key": "paras2021", "temperature": 23.5, "humidity": 50, "PIR": 1, "esp_time":2021-09-30T03:21:44,	Αποστολή δεδομένων προς τον εξυπηρετητή με την μεταφορά των μετρήσεων μέσα στο σώμα του αιτήματος POST με την μορφή JSON.

		<pre> “sd_total”: 15185, “sd_used”: 251, }” </pre>	
--	--	--	--

2. **Κλάση Python τύπου DataCollectionServer - Εξυπηρετητής διαχείρισης δεδομένων.**  
Αυτό το κομμάτι λογισμικού είναι μια κλάση σε Python η οποία χρησιμοποιείται για την επέκταση της λειτουργικότητας του εξυπηρετητή Flask. Ο εξυπηρετητής Flask δημιουργεί ένα αντικείμενο της κλάσης αυτής και μπορεί να αποκτήσει την δυνατότητα αποθήκευσης των δεδομένων στην βάση χρονοσειρών InfluxDB καθώς και την αποστολή των δεδομένων των μετρήσεων προς την υπολογιστική εφαρμογή του ThingSpeak. Ταυτόχρονα αυτή η κλάση δίνει την δυνατότητα χρήσης ενός αρχείου ρυθμίσεων όπως φαίνεται στο ΠΑΡΑΡΤΗΜΑ D. Αυτό το αρχείο αυξάνει την φορητότητα και διευκολύνει την ρύθμιση του εξυπηρετητή συλλογής δεδομένων κατά την μεταφορά του σε άλλα υπολογιστικά συστήματα ή την επανεκκίνηση του με νέες ρυθμίσεις.
3. **InfluxDB** Η βάση χρονοσειρών InfluxDB αποτελεί ένα από τα επίπεδα αποθήκευσης των δεδομένων με σκοπό την ακεραιότητα, επεξεργασία, διαχείριση και έλεγχο των δεδομένων των μετρήσεων σε τοπικό επίπεδο. Η InfluxDB είναι ορατή ως πηγή δεδομένων για το πακέτο απεικόνισης λογισμικού Grafana. Στην Εικόνα 3-22 φαίνεται η InfluxDB εγκατεστημένη στο Raspberry Pi ως διαθέσιμη πηγή δεδομένων για το λογισμικό Grafana. Η αποθήκευση των δεδομένων στην InfluxDB γίνεται με την χρήση της συνάρτησης,

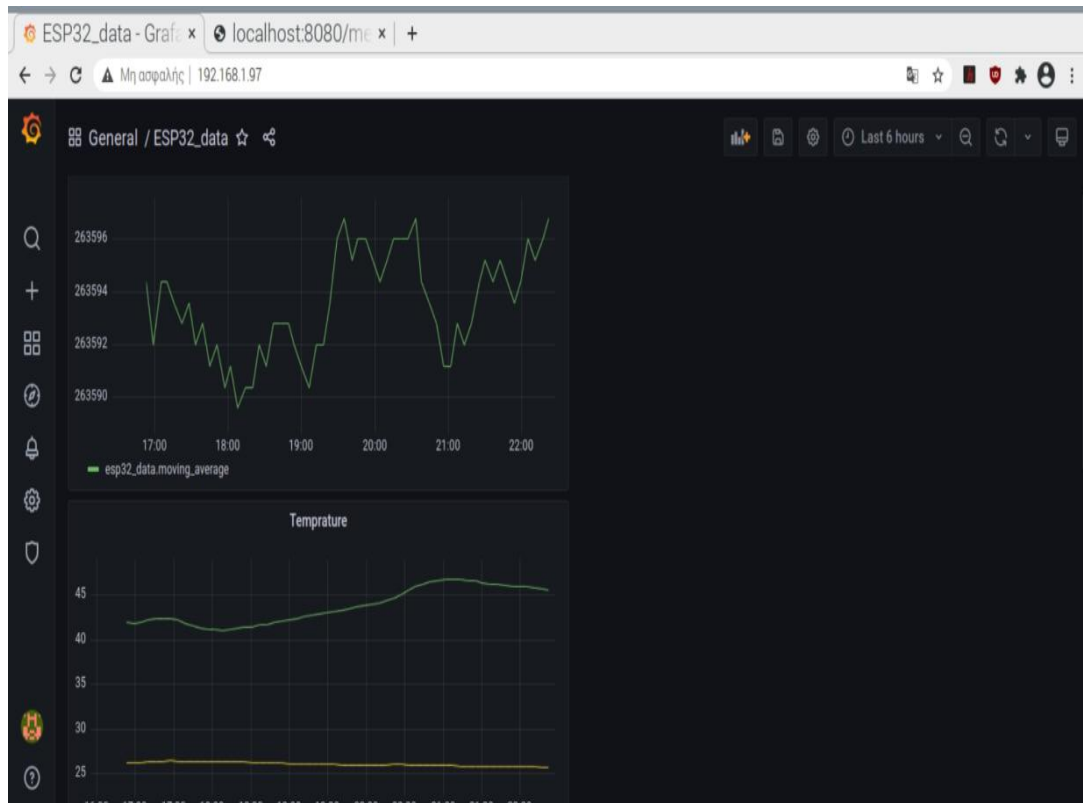
Πίνακας 3-7 - Συνάρτηση της Python κλάσης DataCollectionServer για την αποθήκευση των μετρήσεων στην βάση χρονοσειρών InfluxDB.

Όνομα Συνάρτηση	Ορίσματα	Περιγραφή
DataCollectionServer.influx_update(sample)	<pre> sample = { “measurement” : “&lt;name&gt;”, “time” :“&lt;timestamp&gt;”, “body” : { “field0” :“value0”, “field1” :“value1”, ... “fieldn” :“valuen”, } } </pre>	<p>Η κλήση της συνάρτησης αυτής κάνει τον εξυπηρετητή να πακετάρει τις μετρήσεις από τον ESP32 σε μια Python δομή τύπου dict() όπως φαίνεται στην στήλη του ορίσματος και να την αποθηκεύσει στην βάση δεδομένων InfluxDB.</p>



Εικόνα 3-22 - Απεικόνιση της βάσης δεδομένων χρονοσειρών InfluxDB μέσα από το λογισμικό Grafana ως διαθέσιμη πηγή δεδομένων.

4. **Grafana** - Τέλος, πολύ σημαντικό είναι και το πακέτο λογισμικού Grafana που είναι εγκατεστημένο στην μητρική πλακέτα του Raspberry Pi και είναι υπεύθυνο για την προεπεξεργασία και απεικόνιση των μετρήσεων στο χρήστη σε τοπικό επίπεδο. Τα όρια του τοπικού χρήστη οριοθετούνται από τα όρια του προσωπικού-οικιακού δικτύου. Σε ένα εταιρικό περιβάλλον με διαμοιραζόμενα υπο-δίκτυα θα μπορούσε αυτή η απεικόνιση των δεδομένων να είναι προσβάσιμη και από την άλλη άκρη του κόσμου. Όπως φαίνεται στην Εικόνα 3-23, το Grafana μπορεί να δεχθεί διάφορες πηγές δεδομένων. Στην περίπτωση αυτή έχει οριστεί ως διαθέσιμη πηγή δεδομένων η InfluxDB και το Grafana εμφανίζει διαθέσιμες επιλογές για την απεικόνιση των μετρήσεων της βάση InfluxDB\_ESP32. Το Grafana δίνει την δυνατότητα προσαρμογής των απεικονίσεων στις εκάστοτε ανάγκες του χρήστη. Έτσι στην Εικόνα 3-23 παρουσιάζεται ένα στιγμιότυπο των μετρήσεων των δύο τελευταίων ημερών φιλτραρισμένων με την τεχνική του “κινητού μέσου όρου” με ένα εύρος μετρήσεων 30 δείγματα.



Εικόνα 3-23 - Απεικόνιση δεδομένων με το λογισμικό Grafana σε επίπεδο τοπικού χρήστη. Στιγμιότυπο των μετρήσεων των δύο τελευταίων ημερών φιλτραρισμένων με την τεχνική του “κινητού μέσου όρου” με ένα εύρος μετρήσεων 30 δείγματα

### 3.3.2.3 Πρόσβαση στο υπολογιστικό “σύννεφο” (Cloud) - Χρήση φυλλομετρητή ή Android εφαρμογής

Η εφαρμογή του υπολογιστικού “σύννεφου” ThingSpeak και η εφαρμογή προβολής ThingShow σε κινητό με λογισμικό Android έπρεπε να ρυθμιστούν κατάλληλα όπως αντίστοιχα και ο εξυπηρετητής Python για την αποστολή των δεδομένων των μετρήσεων στο υπολογιστικό “σύννεφο”.

Όλα ξεκινούν με την εγγραφή του χρήστη στην εφαρμογή υπολογιστικού “σύννεφου” ThingSpeak. Με την σύνδεση στο “σύννεφο” ο χρήστης έχει την δυνατότητα να δημιουργήσει κανάλια επικοινωνίας με συσκευές IoT. Κάθε κανάλι έχει έναν ξεχωριστό κωδικό που αποτελεί την ταυτότητα του (channel ID) και διαθέτει και δύο διαφορετικά κλειδιά επικοινωνίας με την προγραμματιστική διεπαφή του ThingSpeak API. Τα κλειδιά επικοινωνίας το ένα είναι για να γράφεις και το άλλο είναι για να διαβάζεις τις μετρήσεις από το κανάλι.

Για την αποστολή των δεδομένων προς το κανάλι από τον εξυπηρετητή Python χρησιμοποιείται η συνάρτηση,

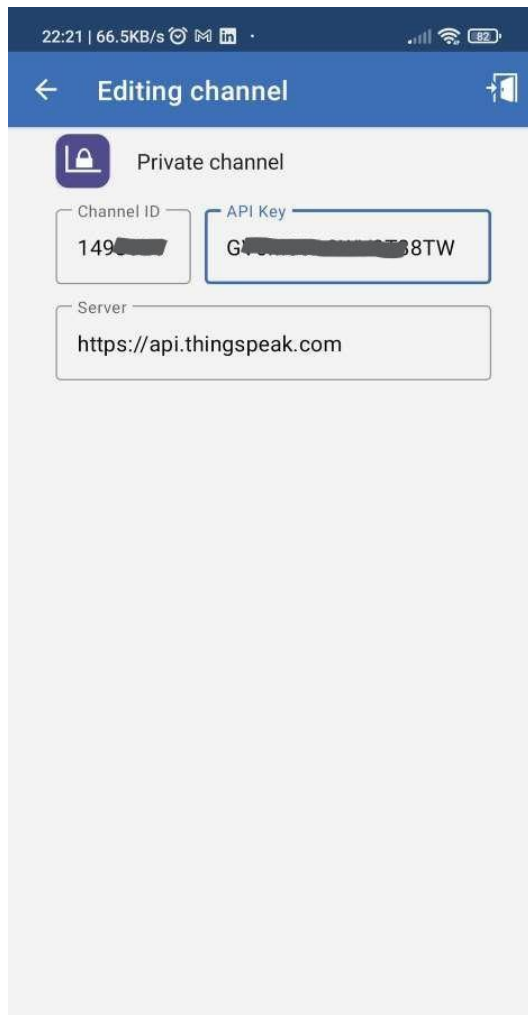
Πίνακας 3-8 - Συνάρτηση της Python κλάσης DataCollectionServer για την αποστολή των μετρήσεων στην εφαρμογή υπολογιστικού νέφους ThingSpeak.

Όνομα Συνάρτηση	Ορίσματα	Περιγραφή
DataCollectionServer.thingspeak_update(sa	sample = { “field0” :“value0”, “field1” :“value1”,	Η κλήση της συνάρτησης αυτής κάνει τον εξυπηρετητή να πακετάρει τις μετρήσεις από τον ESP32 σε μια Python δομή τύπου

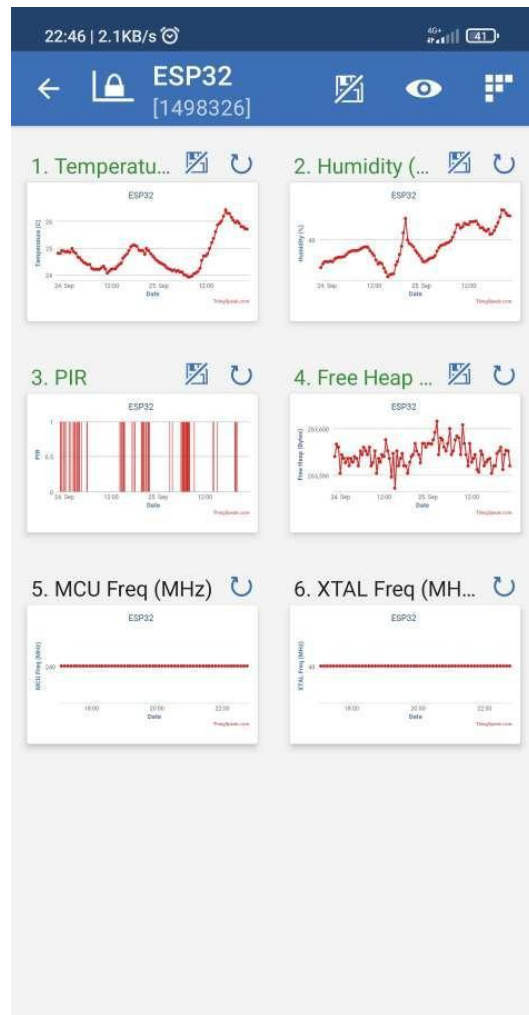
<p>mple)</p>	<pre>... "field" : "valuen" }</pre>	<p>String και να αποστέλλεται προς την προγραμματιστική διεπαφή του ThingSpeak μέσω του URL ενός POST αιτήματος της μορφής  <a href="https://api.thingspeak.com/update?api_key=****&amp;field1=25.7&amp;field2=45.3&amp;field3=0&amp;field4=263592&amp;field5=240&amp;field6=40&amp;created_at=2021-09-25T19:40:15.949358+00:00">[https://api.thingspeak.com/update?api_key=****&amp;field1=25.7&amp;field2=45.3&amp;field3=0&amp;field4=263592&amp;field5=240&amp;field6=40&amp;created_at=2021-09-25T19:40:15.949358+00:00]</a>.</p>
--------------	-------------------------------------	--

Για την προβολή των δεδομένων στην κινητή συσκευή απαιτείται η εγκατάσταση και ρύθμιση της εφαρμογής ThingShow σε κινητό τύπου Android και χρήση του κλειδιού ανάγνωσης για το συγκεκριμένο κανάλι επικοινωνίας. Η Εικόνα 3-24 παρουσιάζει την σελίδα των ρυθμίσεων της εφαρμογής Android και η Εικόνα 3-25 παρουσιάζει την επιτυχή σύνδεση και απεικόνιση στο κανάλι των μετρήσεων που φιλοξενείται στην εφαρμογή υπολογιστικού “νέφους” ThingSpeak.

Πίνακας 3-9 - Ρυθμίσεις και προβολή των δεδομένων των μετρήσεων σε κινητό τηλέφωνο μέσω της εφαρμογής ThingShow που επικοινωνεί με την εφαρμογή υπολογιστικού “νέφους” ThingSpeak.



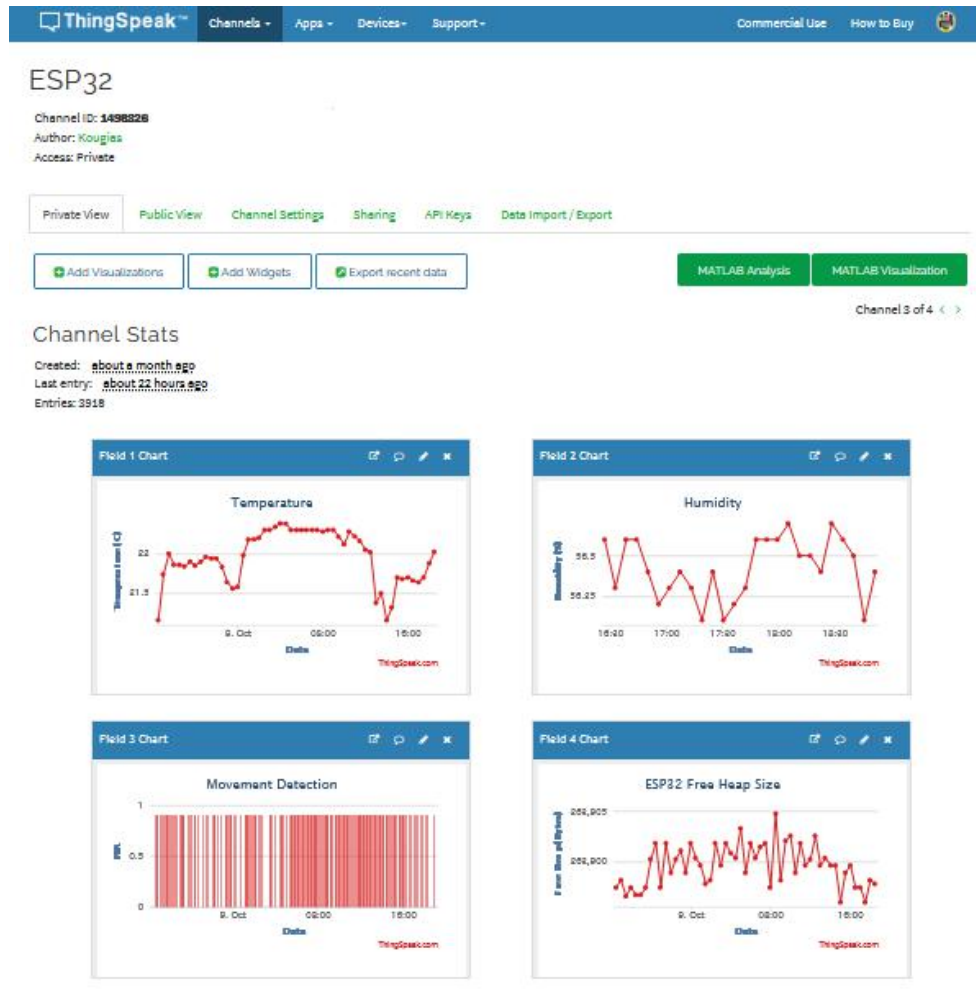
Εικόνα 3-24 - Ρυθμίσεις κλειδιών αυθεντικοποίησης στην εφαρμογή ThingShow για την ανάγνωση των δεδομένων των μετρήσεων από την προγραμματιστική διεπαφή της εφαρμογής του “νέφους” ThinkSpeak.



Εικόνα 3-25 - Επιτυχής σύνδεση, ανάγνωση και απεικόνιση των δεδομένων των μετρήσεων μέσω της εφαρμογής ThinkShow.

## Κεφάλαιο 3

Για την προβολή των δεδομένων στο φυλλομετρητή (web-browser) χρειάζεται απλά η σύνδεση του χρήστη στο λογαριασμό του στο ThingSpeak και η επιλογή του καναλιού που συλλέγει τις μετρήσεις της συσκευής IoT όπως φαίνεται στην Εικόνα 3-26.

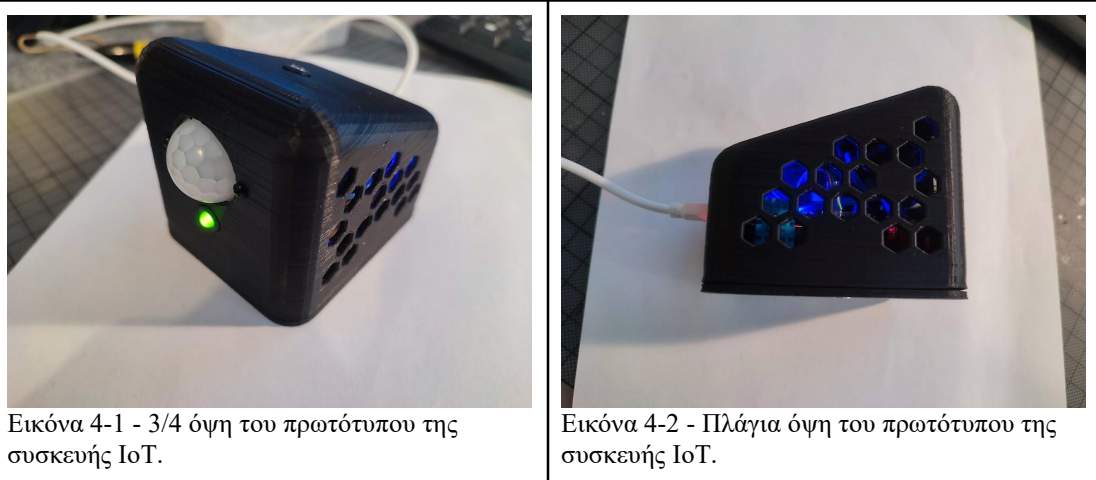


Εικόνα 3-26 - Γενική εικόνα από τον φυλλομετρητή του καναλιού συλλογής δεδομένων στην διαδικτυακή πύλη της εφαρμογής υπολογιστικού νέφους ThinkSpeak.

## Κεφάλαιο 4ο: Αποτελέσματα

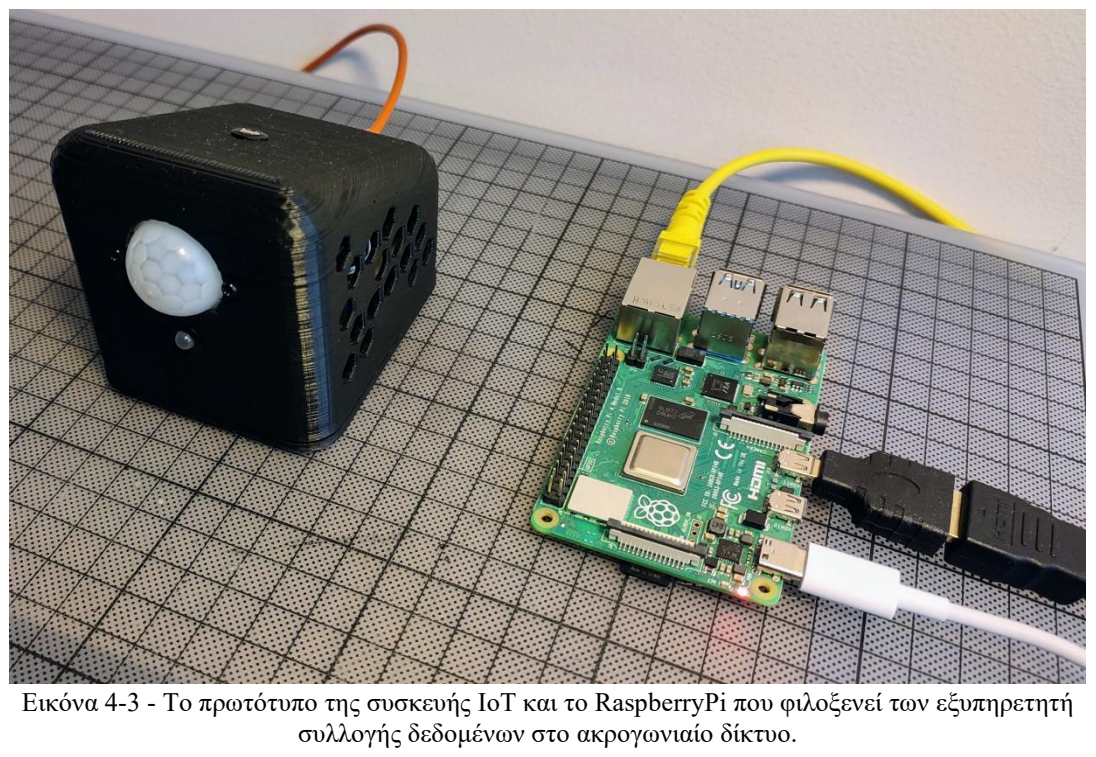
Σε αυτή την ενότητα παρουσιάζονται τα αποτελέσματα της διπλωματική εργασίας σε αντιστοίχιση με τους αρχικούς στόχους της. Όπως φαίνεται στο Πίνακας 4-1 το πρωτότυπο της συσκευής IoT και η ενσωμάτωσή του σε μια σύγχρονη ολοκληρωμένη εφαρμογή του “διαδικτύου των πραγμάτων” ολοκληρώνει τους στόχους της εργασίας αυτής. Παρακάτω περιγράφονται πιο αναλυτικά τα αποτελέσματα.

Πίνακας 4-1 - Το πρωτότυπο της συσκευής IoT ως αποτέλεσμα αυτής της διπλωματικής εργασίας.



Εικόνα 4-1 - 3/4 όψη του πρωτότυπου της συσκευής IoT.

Εικόνα 4-2 - Πλάγια όψη του πρωτότυπου της συσκευής IoT.



Εικόνα 4-3 - Το πρωτότυπο της συσκευής IoT και το RaspberryPi που φιλοξενεί των εξυπηρετητή συλλογής δεδομένων στο ακρογωνιαίο δίκτυο.

### 4.1 Μεταφορά μετρήσεων από άκρη-σε-άκρη

Αρχικός στόχος της εργασίας αυτής είναι η ανάπτυξη ενός συστήματος του “διαδικτύου των πραγμάτων” που καταγράφει μετρήσεις από αισθητήρες θερμοκρασίας και υγρασίας καθώς και την ημερομηνία και ώρα που συλλέχθηκαν.

Το σύστημα IoT που σχεδιάστηκε και υλοποιήθηκε πετυχαίνει αυτό τον στόχο διότι,

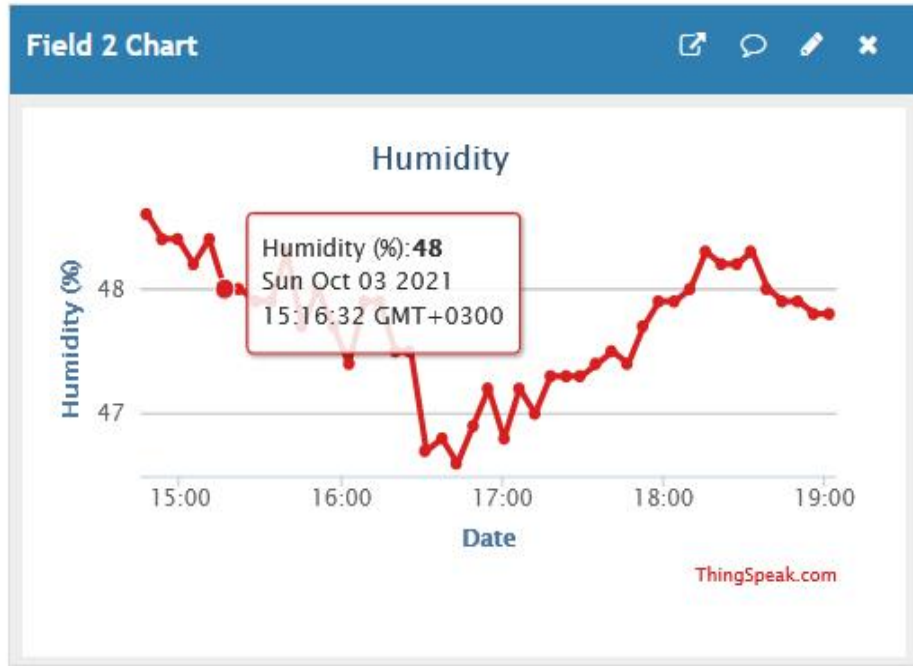
- Το υλοποιημένο IoT σύστημα συλλέγει όχι μόνο μετρήσεις θερμοκρασίας και υγρασίας αλλά και ανίχνευσης κίνησης και κατάστασης της μνήμης του μικροελεγκτή ESP32 και ταυτόχρονα τα χαρακτηρίζει με την ώρα και ημερομηνία συλλογής με βάση διεθνείς μορφές ώρας [11] που είναι συμβατές με όλα τα υπολογιστικά συστήματα. Πέρα από αυτό όμως το υλοποιημένο σύστημα IoT δεν είναι απλά μια συσκευή καταγραφής μετρήσεων, είναι μια πολυεπίπεδη εφαρμογή (Fullstack) τεχνολογιών από άκρη-σε-άκρη (end-to-end). Δηλαδή, στην μία άκρη είναι ο αισθητήρας ενός συστήματος IoT και στην άλλη άκρη είναι ο χρήστης που μπορεί να λαμβάνει και να επεξεργάζεται τα δεδομένα σε γραφήματα τόσο σε τοπικό όσο και σε απομακρυσμένο επίπεδο πρόσβασης, ακόμα και στο κινητό του.

Ο Πίνακας 4-2 παρουσιάζει τα στάδια που περνάει μια μέτρηση από το ένα άκρο στο άλλο (end-to-end), δηλαδή από το σύστημα IoT για να φτάσει στο υπολογιστικό νέφος και στους χρήστες.

Πίνακας 4-2 - Τα στάδια ακολουθεί μια μέτρηση υγρασίας από την συλλογή της στην συσκευή IoT μέχρι την αποθήκευση της στο υπολογιστικό σύννεφο και την απεικόνιση της στο χρήστη.

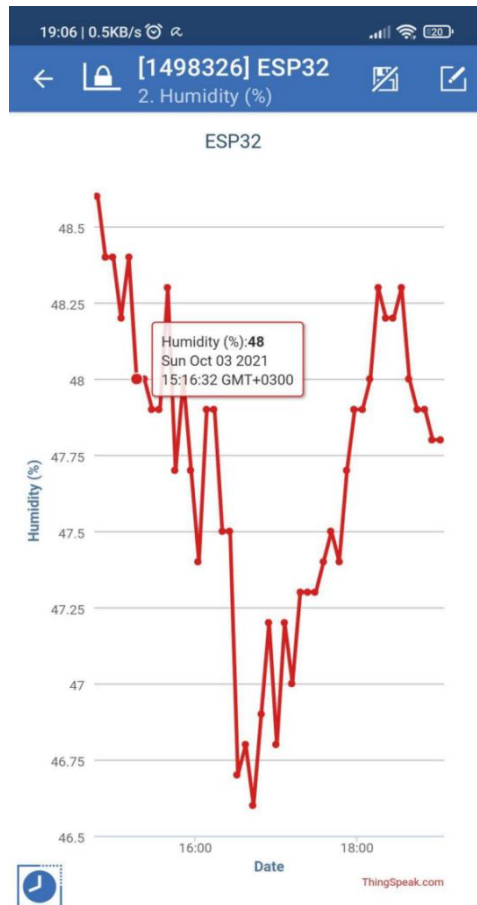
Επίπεδο	Τιμή/Απεικόνιση
Μέτρηση IoT Συσκευή	<pre>{ "device": "DEVICE2",   "api_key": "paras2021",   "temperature": 22.700,   "<b>humidity": 48.20,</b>   "PIR": 1,   "mcu_freq": 240,   "xtal_freq": 40,   "free_heap": 263896,   "esp_time": "2021-10-03T15:15:47" }</pre>
Ακρογωνιαίο Δίκτυο - Αποθήκευση στην Influx DB	<pre>{ 'measurement': 'esp32_data', 'time': '2021-10-03T15:15:47.148620+00:00', 'fields':   { 'temperature': 22.7, '<b>humidity': 48.2,</b> 'PIR': 0, 'mcu_freq': 240, 'xtal_freq': 40, 'free_heap': 263900 } }</pre>
Ακρογωνιαίο δίκτυο - Απεικόνιση μετρήσεων σε τοπικό επίπεδο με το λογισμικό Grafana.	 <p>Εικόνα 4-4 - Απεικόνιση του δείγματος. Η τιμή και ο χρόνος είναι ελάχιστα διαφορετικοί γιατί η απεικόνιση των δεδομένων είχε προ-επεξεργαστεί με βάση τον αλγόριθμο του μέσου όρου ανα πέντε δείγματα που διατίθεται από το λογισμικό Grafana για την εξομάλυνση των άκρων της κυματομορφής.</p>

Εφαρμογή φυλλομετρητή - Απεικόνιση μετρήσεων σε απομακρυσμένο επίπεδο με το λογισμικό ThingSpeak στον υπολογιστή.



Εικόνα 4-5 - Απεικόνιση του δείγματος υγρασίας στο φυλλομετρητή μέσα από την εφαρμογή ThingSpeak. Η τιμή και ο χρόνος είναι ελάχιστα διαφορετική γιατί η απεικόνιση των δεδομένων είχε προ-επεξεργαστεί με βάση τον αλγόριθμο του μέσου όρου ανα 15 δείγματα που διατίθεται από το λογισμικό ThingSpeak για την εξομάλυνση των άκρων της κυματομορφής.

Εφαρμογή κινητού - Απεικόνιση μετρήσεων σε απομακρυσμένο επίπεδο με το λογισμικό ThingSpeak σε φορητή συσκευή



Εικόνα 4-6 - Απεικόνιση του δείγματος υγρασίας στο κινητό μέσα από την εφαρμογή ThingShow.

## 4.2 Σημεία αποθήκευσης σε εξωτερική μνήμη

Δεύτερος στόχος της εργασίας αυτής είναι οι μετρήσεις από τους αισθητήρες να αποθηκεύονται σε εξωτερική μνήμη. Για το λόγο αυτό επιλέχθηκε τα δεδομένα να αποθηκεύονται σε τρία σημεία εξωτερικής μνήμης. Αρχικά το πρώτο επίπεδο είναι η κάρτα μνήμης στην συσκευή IoT, σε δεύτερο επίπεδο τα δεδομένα αποθηκεύονται στο ακρογωνιαίο σύστημα (Edge network) και στο τελευταίο και τρίτο επίπεδο τα δεδομένα αποθηκεύονται στο υπολογιστικό “σύννεφο” (Cloud).

Το σύστημα IoT που σχεδιάστηκε και υλοποιήθηκε πετυχαίνει αυτό τον στόχο όχι με έναν αλλά με τρεις διαφορετικούς τρόπους,

- Πρώτο και βασικότερο, σε περίπτωση που χαθεί η σύνδεση με το ακρογωνιαίο δίκτυο είτε με το διαδίκτυο το σύστημα IoT αποθηκεύει τα δεδομένα από τις μετρήσεις στην κάρτα μνήμης. Στο Πίνακα 4-3 φαίνεται ένα μέρος των μετρήσεων που καταγράφονται στην κάρτα μνήμης της συσκευής IoT.

Πίνακας 4-3 - Απεικόνιση ενός τμήματος των μετρήσεων που καταγράφονται στην κάρτα μνήμης της συσκευής IoT

```
{
  "PIR":1,"api_key":"paras2021","created_at":"2021-10-03T15:21:33.974968+00:00","device":"DEVICE2","free_heap":263896,"humidity":48.2,"mcu_freq":240,"temperature":22.7,"xtal_freq":40},
  {"PIR":0,"api_key":"paras2021","created_at":"2021-10-03T15:27:20.811743+00:00","device":"DEVICE2","free_heap":263900,"humidity":48.2,"mcu_freq":240,"temperature":22.7,"xtal_freq":40},
  {"PIR":1,"api_key":"paras2021","created_at":"2021-10-03T15:33:07.541137+00:00","device":"DEVICE2","free_heap":263900,"humidity":48.3,"mcu_freq":240,"temperature":22.7,"xtal_freq":40},
  {"PIR":1,"api_key":"paras2021","created_at":"2021-10-03T15:38:54.366138+00:00","device":"DEVICE2","free_heap":263892,"humidity":48.0,"mcu_freq":240,"temperature":22.7,"xtal_freq":40},
  {"PIR":1,"api_key":"paras2021","created_at":"2021-10-03T15:44:41.181074+00:00","device":"DEVICE2","free_heap":263896,"humidity":47.9,"mcu_freq":240,"temperature":22.7,"xtal_freq":40},
  {"PIR":1,"api_key":"paras2021","created_at":"2021-10-03T15:50:28.027859+00:00","device":"DEVICE2","free_heap":263896,"humidity":47.9,"mcu_freq":240,"temperature":22.7,"xtal_freq":40}
}
```

- Σε δεύτερο επίπεδο η αποθήκευση των δεδομένων των μετρήσεων γίνεται στο Raspberry Pi που ανήκει στο ακρογωνιαίο δίκτυο στην ειδική βάση χρονοσειρών InfluxDB όπως φαίνεται στην Εικόνα 3-22 και Εικόνα 3-23.
- Τρίτο και τελευταίο επίπεδο, με την αποστολή των δεδομένων των μετρήσεων στο υπολογιστικό “σύννεφο”, οι μετρήσεις αποθηκεύονται σε απομακρυσμένα κέντρα εξυπηρετητών (farm servers) με αποτέλεσμα τα δεδομένα να είναι προσβάσιμα από απομακρυσμένους χρήστες.

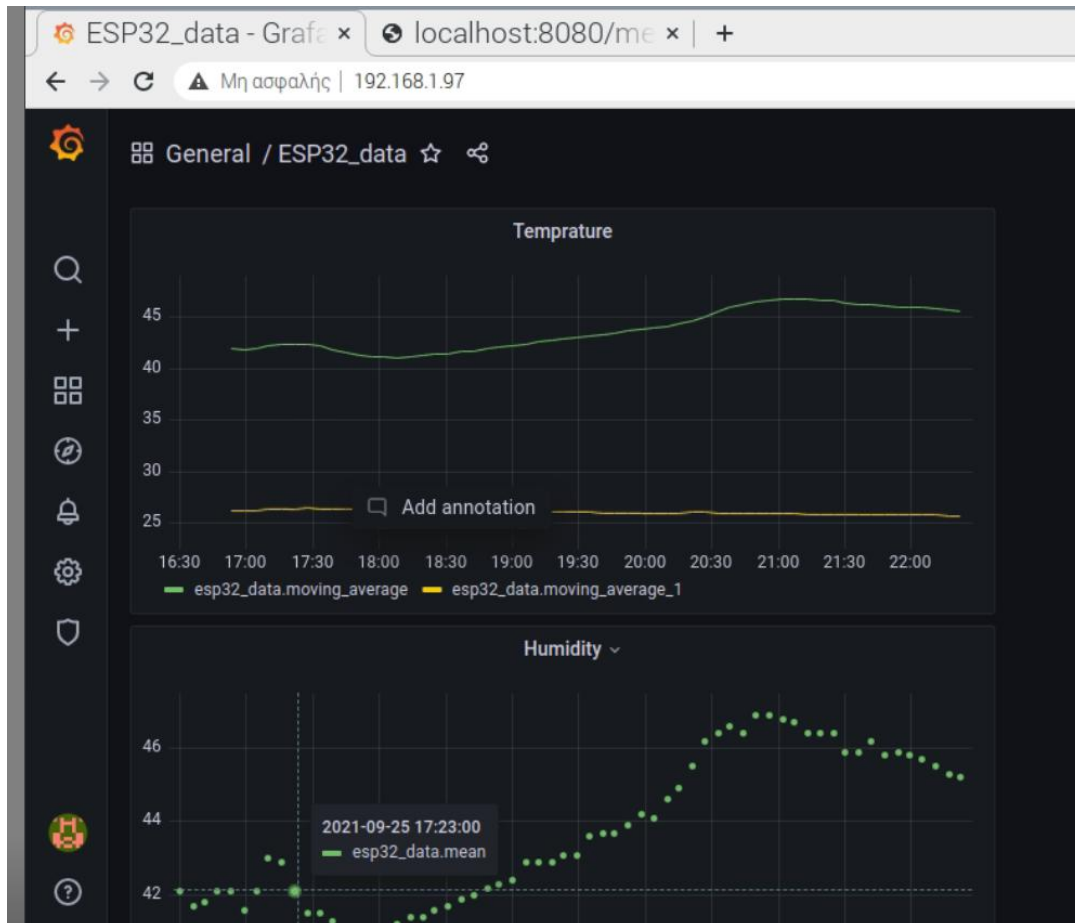
## 4.3 Διεπαφή χρήστη

Τρίτος στόχος της εργασίας αυτής είναι η εμφάνιση αυτών των αποτελεσμάτων στο χρήστη μέσω της χρήσης μιας συμβατής εφαρμογής διαδικτύου ή κινητού τηλεφώνου με σκοπό την διευκόλυνση της προσβασιμότητας του χρήστη στα δεδομένα των αισθητήρων.

Το σύστημα IoT που σχεδιάστηκε και υλοποιήθηκε πετυχαίνει αυτό τον στόχο σε δύο διαφορετικά επίπεδα, τοπικά και απομακρυσμένα.

### 4.3.1 Τοπικός χρήστης

Τοπικός χρήστης σημαίνει ο χρήστης που έχει πρόσβαση στο προσωπικό δίκτυο (private network) που είναι εγκατεστημένη η βάση χρονοσειρών InfluxDB και το λογισμικό Grafana. Με αυτό το τρόπο ο τοπικός χρήστης θα μπορεί να βλέπει τα δεδομένα των μετρήσεων μέσα από το προσωπικό δίκτυο στην μορφή που φαίνονται στην Εικόνα 4-7.

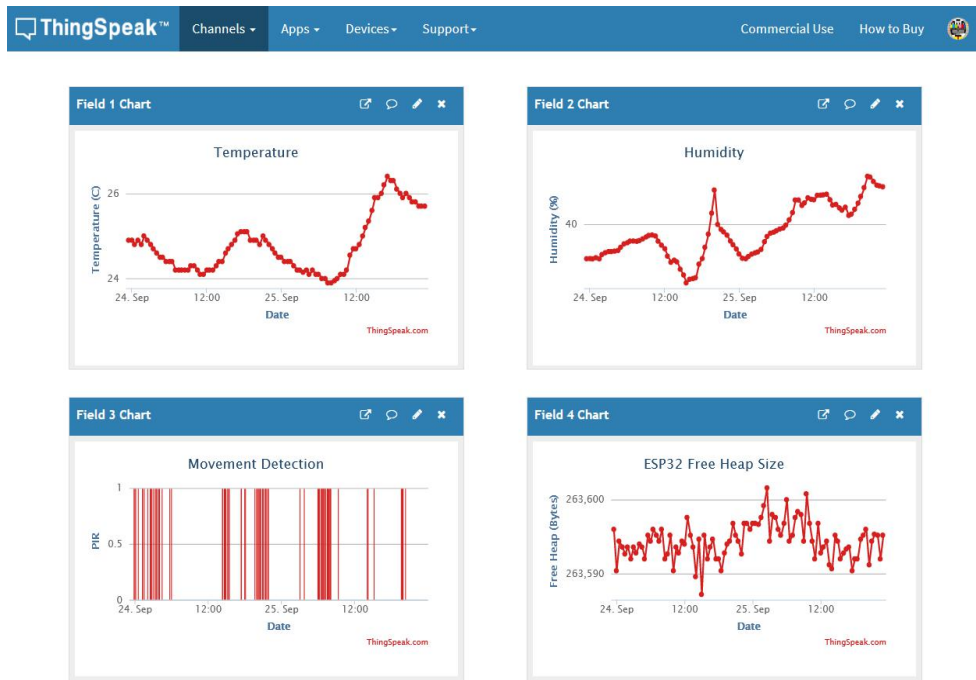


Εικόνα 4-7 - Απεικόνιση των μετρήσεων της υγρασίας και της θερμοκρασίας στον τοπικό χρήστη μέσω του λογισμικού Grafana

### 4.3.2 Απομακρυσμένος χρήστης

Απομακρυσμένος χρήστης σημαίνει ο χρήστης που δεν έχει πρόσβαση στο προσωπικό δίκτυο (private network) αλλά έχει τα κλειδιά ανάγνωσης από την διαδικτυακή εφαρμογή του υπολογιστικού “νέφους” Thingspeak και είτε μπορεί να συνδεθεί μέσω του φυλλομετρητή είτε μέσω εφαρμογής κινητού τηλεφώνου όπως η εφαρμογή ThingShow. Στο Πίνακα 4-4 φαίνονται οι προβολές των μετρήσεων για τον απομακρυσμένο χρήστη, στην Εικόνα 4-8 φαίνεται η διεπαφή για το φυλλομετρητή και στην Εικόνα 4-9 φαίνεται η διεπαφή στο κινητό.

Πίνακας 4-4 - Διεπαφή του απομακρυσμένου χρήστη για την προβολή των μετρήσεων. Στα αριστερά η εικόνα στο φυλλομετρητή και στα δεξιά η εικόνα στο κινητό.



Εικόνα 4-8 - Διεπαφή απομακρυσμένου χρήστη μέσω του φυλλομετρητή.



Εικόνα 4-9 - Διεπαφή απομακρυσμένου χρήστη μέσω του κινητού.

## Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης

Σε αυτή την ενότητα αναφέρονται τα συμπεράσματα και οι προτάσεις βελτίωσης που προέκυψαν κατά την ανάπτυξη του πρωτότυπου του πυρήνα ενός σύγχρονου συστήματος του διαδικτύου των πραγμάτων (IoT).

### 5.1 Συμπεράσματα

Η υλοποίηση ενός πρωτότυπου συστήματος IoT οδήγησε στα παρακάτω συμπεράσματα τόσο σε γενικό επίπεδο όσο και σε τεχνικό επίπεδο.

#### 5.1.1 Σε γενικό επίπεδο

- Η σωστή σχεδίαση και ο προσδιορισμός των αναγκών του πρωτότυπου συστήματος IoT έπαιξε σημαντικό ρόλο στην διευκόλυνση της τελικής υλοποίησης της συσκευής IoT. Η έρευνα που προηγήθηκε βοήθησε στην κατανόηση των βασικών κατευθύνσεων των σύγχρονων τεχνολογιών του διαδικτύου των πραγμάτων και την διασύνδεση του με εφαρμογές του υπολογιστικού νέφους αλλά και την ανάγκη για πρόσβαση των χρηστών τόσο σε τοπικό όσο και σε απομακρυσμένο επίπεδο. Τέλος, η επιλογή αποθήκευσης των δεδομένων σε κάρτα μνήμης έγινε με σκοπό την κάλυψη καταστάσεων έκτακτης ανάγκης σε περίπτωση διακοπής της σύνδεσης στο δίκτυο και την ακεραιότητα των δεδομένων.
- Η υλοποίηση ενός σύγχρονου συστήματος IoT είναι μια πολύ-επίπεδη εφαρμογή τεχνολογιών. Ξεκινά από το χαμηλότερο επίπεδο με την κατανόηση και επιλογή των αισθητήρων που χρειάζονται και συνεχίζει με την επιλογή και τον προγραμματισμό του μικροεπεξεργαστή. Σε επόμενο βήμα ακολουθεί η επιλογή της γλώσσας προγραμματισμού και του πρωτοκόλλου επικοινωνίας όπως είναι η Python και το πρωτόκολλο HTTP. Σε αυτό το σημείο, η επιλογή διασύνδεσης με το υπολογιστικό νέφος δημιούργησε την ανάγκη της εγκατάστασης ενός ακρογωνιαίου κόμβου όπως είναι το Raspberry Pi με σκοπό την διευκόλυνση της συλλογής, διαχείρισης, αποθήκευσης και αποστολής των δεδομένων προς την εφαρμογή υπολογιστικού νέφους. Η ολοκλήρωση της στοίβας των τεχνολογιών δεν θα ήταν αρκετή εάν δεν μπορεί ο τελικός χρήστης να έχει πρόσβαση στα δεδομένα αυτά. Με αυτό το σκεπτικό, εγκαταστήθηκε το λογισμικό Grafana και η εφαρμογή κινητού ThingShow με σκοπό την απεικόνιση των μετρήσεων τόσο σε τοπικό όσο και σε απομακρυσμένο επίπεδο.
- Η συλλογή, διαχείριση, αποθήκευση και αποστολή των δεδομένων των μετρήσεων μαζί με το χρονικό προσδιορισμό τους είναι μια απαιτητική εφαρμογή για την διατήρηση της ακεραιότητας των μετρήσεων. Η συσκευή IoT καταγράφει τα δεδομένα των μετρήσεων και χρησιμοποιεί το διαδίκτυο για τον συγχρονισμό του ρολογιού του (NTP) αποτελώντας την πηγή των δεδομένων. Ωστόσο, η αποστολή των δεδομένων στα ενδιάμεσα στάδια του ακρογωνιαίου κόμβου και στην εφαρμογή ThingSpeak στο υπολογιστικό νέφος πρέπει να διατηρήσουν την ακρίβεια του χρονικού προσδιορισμού και των τιμών των μετρήσεων. Αυτό γίνεται εφικτό με την διαχείριση του χρόνου σε μορφή διεθνούς προτύπου (ISO 8601) που υποστηρίζεται τόσο από την γλώσσα προγραμματισμού Python όσο και από την εφαρμογή ThingSpeak. Η διαχείριση των τιμών δεν αποτελεί πρόβλημα καθώς σε όλα τα επίπεδα υποστηρίζεται η διατήρηση των δεκαδικών ψηφίων. Ωστόσο, κατά την απεικόνιση των δεδομένων τόσο στο λογισμικό Grafana και στην εφαρμογή ThingSpeak υπάρχουν αλγόριθμοι φιλτραρίσματος και απεικόνισης των δεδομένων που μπορεί

να παρουσιάσουν ελαφρώς “αλλαγμένες” τις τιμές σύμφωνα με το αποτέλεσμα του αλγορίθμου (μέσες τιμές ανά επιλεγμένο αριθμό δειγμάτων, ενδιάμεσος μέσος, συσσωρευτικό άθροισμα κ.α).

### 5.1.2 Σε τεχνικό επίπεδο

- Η δημιουργία μιας εφαρμογής του διαδικτύου των πραγμάτων απαιτεί από τον ηλεκτρονικό μηχανικό ένα σύνολο κατανόησης και γνώσεων. Το υλικό απαιτεί γνώσεις προγραμματισμού χαμηλού επιπέδου με την γλώσσα προγραμματισμού C/C++ και ταυτόχρονα την κατανόηση της αρχιτεκτονικής του μικροελεγκτή ESP32 για τον προσδιορισμό των δυνατοτήτων και περιορισμών που εισάγει στην εφαρμογή. Οι τεχνολογίες των προγραμματιστικών διεπαφών και η ενασχόληση με τα προσωπικά και δημόσια δίκτυα χρειάζεται τόσο γνώσεις αντικειμενοστραφούς προγραμματισμού όσο και βασικές γνώσεις δικτύων για την εύρυθμη λειτουργία του πρωτότυπου της συσκευής IoT.
- Η επιλογή του μικροελεγκτή σε τέτοιες εφαρμογές είναι κομβικής σημασίας καθώς αποτελεί τον πυρήνα της συσκευής IoT. Στην εργασία αυτή επιλέχθηκε ο μικροελεγκτής ESP32 γιατί διαθέτει ενσωματωμένη στοιβία για σύνδεση σε ασύρματα δίκτυα WiFi αλλά και ισχυρό επεξεργαστή στα 240MHz. Οι περιφερειακές του διεπαφές ήταν αρκετές για τις σχεδιαστικές ανάγκες της διπλωματικής εργασίας. Επιπλέον ο μικροελεγκτής ESP32 διατίθεται σε χαμηλό κόστος και γ'αυτό υποστηρίζεται από μια μεγάλη κοινότητα μηχανικών ανοικτού υλικού και λογισμικού.
- Η δημιουργία λογισμικού για πολυεπίπεδες εφαρμογές αυξάνει την πολυπλοκότητα των ρυθμίσεων και μειώνει τη φορητότητα του κώδικα σε άλλα υπολογιστικά συστήματα. Για το λόγο αυτό, ο εξυπηρετητής συλλογής δεδομένων χρησιμοποιεί ένα αρχείο ρυθμίσεων το οποίο λειτουργεί σαν ένα μοναδικό σημείο διαχείρισης των παραμέτρων όλων των επιπέδων της εφαρμογής και αυξάνει την επεκτασιμότητα του κώδικα.

Με την ενασχόληση με όλα αυτά τα υπολογιστικά επίπεδα και με την χρήση τόσο σύγχρονων εργαλείων, ο φοιτητής εξοικειώθηκε και απέκτησε γνώση και εμπειρία σε τεχνολογίες που αποτελούν την αιχμή τόσο σε ερευνητικό όσο και εμπορικό επίπεδο.

## 5.2 Προτάσεις βελτίωσης

Η διπλωματική αυτή παρουσιάζει την υλοποίηση και χρήση ενός πρωτότυπου πυρήνα συστήματος IoT τόσο για εφαρμογές “έξυπνου” σπιτιού όσο και σε εφαρμογές για την εξέλιξη της βιομηχανίας 4ης γενιάς. Το πρωτότυπο θα μπορούσε να επεκταθεί και να βελτιωθεί σε διάφορα σημεία είτε να αποτελέσει την βάση για μελλοντική ανάπτυξη εφαρμογών IoT. Τα σημεία αυτά είναι:

- Η επιλογή και προσθήκη διαφορετικών αισθητήρων είτε ψηφιακών είτε αναλογικών. Οι αναλογικοί αισθητήρες πρέπει να συνδεθούν με τους διαθέσιμους αναλογικούς-ψηφιακούς μετατροπείς (ADC) του μικροελεγκτή ESP32. Από την άλλη μεριά, η σύνδεση ψηφιακών αισθητήρων είναι πιο διαδεδομένη γιατί επιτρέπει την χρήση των ψηφιακών πρωτοκόλλων επικοινωνίας (π.χ. I2C, UART, κ.α). με αποτέλεσμα την χρήση μεγαλύτερου αριθμού αισθητήρων και ταυτόχρονα την μεγαλύτερη αξιοπιστία στην μέτρηση. Η αύξηση της ποικιλίας των αισθητήρων μπορεί να εισάγει στο σύστημα IoT περισσότερα δεδομένα και άρα καλύτερη πληροφόρηση ανάλογα την εφαρμογή. Δηλαδή, στην περίπτωση του “έξυπνου” σπιτιού, η ποικιλία μετρήσεων θα μπορούσε να οδηγήσει στην καλύτερη απεικόνιση της συνολικής εικόνας του σπιτιού αλλά και στην λήψη καλύτερων αποφάσεων εξοικονόμησης ενέργειας, ψυχαγωγίας ή κλιματισμού. Στην περίπτωση της εξέλιξης της βιομηχανίας 4ης γενιάς, περισσότεροι βιομηχανικοί αισθητήρες θα μπορούσαν να ενισχύσουν την ποιότητα και τον έλεγχο της γραμμής

παραγωγής με την καλύτερη λήψη αποφάσεων από το κέντρο ελέγχου ή την καλύτερη συνεργασία με τα εργοστασιακά ρομπότ.

- Η ενίσχυση της λογικής του εξυπηρετητή συλλογής δεδομένων. Ο εξυπηρετητής δεδομένων έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι επεκτάσιμος. Η χρήση της αντικειμενοστραφούς γλώσσας προγραμματισμού Python σε συνδυασμό με τον εξυπηρετητή Flask δίνει την δυνατότητα στον ηλεκτρονικό μηχανικό να αναπτύξει προγραμματιστικές κλάσεις στοχευμένες στην επέκταση της εκάστοτε εφαρμογής. Για παράδειγμα, στην περίπτωση του έξυπνου σπιτιού θα μπορούσαν να αναπτυχθούν αλγόριθμοι μηχανικής μάθησης που θα προσαρμόζονταν στις συνθήκες των ενοίκων με σκοπό την δημιουργία ενός οικολογικού προφίλ θέρμανσης και κατανάλωσης ενέργειας. Στο παράδειγμα, της βιομηχανίας 4ης γενιάς, ο ηλεκτρονικός μηχανικός θα μπορούσε να αναπτύξει προγραμματιστικές κλάσεις Python με σκοπό την επεξεργασία και ομαδοποίηση των δεδομένων για την επίβλεψη και βελτιστοποίηση της παραγωγικής γραμμής ή ακόμα και για την λήψη αποφάσεων τόσο σε τοπικό όσο και σε απομακρυσμένο επίπεδο.
- Η σύνδεση της συσκευής IoT με ενεργοποιητές (actuators). Οι ενεργοποιητές είναι συσκευές που δέχονται εντολές και ενεργούν στο περιβάλλον τους. Χαρακτηριστικά παραδείγματα ενεργοποιητών είναι τα ρελέ/διακόπτες και οι κινητήρες. Η χρήση των ενεργοποιητών επιβάλλει την ενεργοποίηση αμφίδρομης λειτουργίας ελέγχου, δηλαδή απομακρυσμένοι χρήστες ή εξυπηρετητές που μπορούν να στέλνουν εντολές στους ενεργοποιητές για την εκτέλεση ενεργειών. Στην περίπτωση του έξυπνου σπιτιού, η εφαρμογή θα μπορούσε να είναι ο εντοπισμός κίνησης στο δωμάτιο και η αυτόματη ενεργοποίηση του φωτισμού ή ακόμα και η ενεργοποίηση της θέρμανσης με την παρουσία ατόμων στην κατοικία. Η σύνδεση ενεργοποιητών σε εφαρμογές της βιομηχανίας 4ης γενιάς είναι απαραίτητη. Οι γραμμές παραγωγής στην βιομηχανία είναι γεμάτοι διακόπτες και κινητήρες και οι υψηλές ταχύτητες παραγωγής επιβάλλουν την λήψη αποφάσεων και εκτέλεση εντολών από τις μηχανές. Μια πολλά υποσχόμενη μελλοντική χρήση στην βιομηχανία 4ης γενιάς, είναι τα αυτόνομα ρομπότ τα οποία θα λειτουργούν σαν ενεργοποιητές και θα μπορούν να συνεργάζονται με τους εργάτες αλλά και να παίρνουν αποφάσεις με βάση μετρήσεις που προέρχονται από τον περιβάλλοντα χώρο από συσκευές IoT.
- Η αμφίδρομη σύνδεση συσκευών IoT με κέντρα ελέγχου και εφαρμογές του διαδικτύου των πραγμάτων και του υπολογιστικού νέφους τόσο για εφαρμογές του έξυπνου σπιτιού όσο και για εφαρμογές της βιομηχανίας 4ης γενιάς θα οδηγήσει στην ανάγκη για διατήρηση της ιδιωτικότητας των δεδομένων αλλά και της ασφάλειας των εφαρμογών αυτών. Το πρωτότυπο της συσκευής IoT θα μπορούσε να ενισχυθεί με μηχανισμούς κυβερνοασφάλειας τόσο σε επίπεδο υλικού όσο και σε επίπεδο λογισμικού. Σε επίπεδο υλικού θα μπορούσαν να κρυπτογραφηθούν τα δεδομένα των μετρήσεων όταν αποθηκεύονται στην κάρτα μνήμης. Σε επίπεδο λογισμικού θα μπορούσε να χρησιμοποιηθεί το πρωτόκολλο επικοινωνίας HTTPS κατά την αποστολή των μετρήσεων από τη συσκευή IoT στον εξυπηρετητή συλλογής δεδομένων. Στα επόμενα στάδια, το πρωτότυπο IoT υιοθετεί τα επίπεδα ασφαλείας των εφαρμογών Grafana και ThingSpeak. Για την σύνδεση στο περιβάλλον του Grafana και της διαδικτυακής πύλης του ThingSpeak απαιτείται η χρήση κωδικού και ονόματος χρήστη. Ενώ για την πρόσβαση στην προγραμματιστική διεπαφή του ThingSpeak χρειάζεται η χρήση του μοναδικού κλειδιού με δικαιώματα ανάγνωσης ή εγγραφής αντίστοιχα.
- Τελευταίο σημείο βελτίωσης είναι η ενίσχυση της διεπαφής του χρήστη με την συσκευή IoT. Για την αλλαγή των ρυθμίσεων και τον έλεγχο της συσκευής θα μπορούσε να εγκατασταθεί στον

## Κεφάλαιο 4

ESP32 ένας μικροσκοπικός εξυπηρετητής HTTP παρόμοιος με αυτών που έχουν τα οικιακά ρουτερ. Μέσα από το περιβάλλον του εξυπηρετητή ο χρήστης θα μπορούσε να ρυθμίζει τις παραμέτρους σύνδεσής στο προσωπικό δίκτυο και να ρυθμίζει τις βασικές λειτουργίες του μικροεπεξεργαστή EPS32.

Οι μελλοντικές επεκτάσεις που περιγράφονται παραπάνω σκοπό έχουν να προτείνουν τα επόμενα βήματα για την εξέλιξη του πρωτότυπου της συσκευής IoT σε ένα ολοκληρωμένο σύστημα IoT που εξυπηρετεί τις πιο σύγχρονες απαιτήσεις τόσο σε ακαδημαϊκό επίπεδο όσο και στο επίπεδο της αγοράς.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

Ο τρόπος γραφής των βιβλιογραφικών αναφορών γίνεται σύμφωνα με τα παρακάτω παραδείγματα (IEEE style):

### Βιβλία

[1] Shubham Sarda, *Object-oriented Programming with Python - A Practical Guide.*, 2019, ISBN: 9781839216855, O'Reilly

[2] Syed Omar Faruk Towaha, *.Learning C for Arduino*, Released March 2017, ISBN: 9781787120099, O'Reilly

### Application Note

[3] ThingSpeak Documentation - <https://www.mathworks.com/help/thingspeak>

[4] ThingSpeak Tutorial - <https://www.mathworks.com/help/thingspeak/getting-started-with-thingspeak.html>

[5] Grafana with InfluxDB tutorial - <https://grafana.com/docs/grafana/latest/getting-started/getting-started-influxdb/>

[6] InfluxDB Documentation & Tutorial - <https://docs.influxdata.com/influxdb/v1.8/introduction/getting-started/>

[7] Arduino IDE 1.8.16 - Λογισμικό Ανάπτυξης του πηγαίου κώδικα για τον ESP32 - <https://www.arduino.cc/en/software>

### Πατέντες

[8] ISO 8601 - Date time format - <https://www.iso.org/iso-8601-date-and-time-format.html>

[9] NIST - Final Version of the NIST Definition of Cloud Computing - SP 800-145, 2011 - <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-146.pdf>

[10] NIST - Draft NIST Special Publication 800-213 - Establishing IoT Device Cybersecurity Requirement - <https://doi.org/10.6028/NIST.SP.800-213-draft>

### Data Sheet

[11] ESP32 - Espressif - [https://cdn.sparkfun.com/datasheets/IoT/esp32\\_datasheet\\_en.pdf](https://cdn.sparkfun.com/datasheets/IoT/esp32_datasheet_en.pdf) .

[12] Raspberry Pi M4 Datasheet - <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>

[13] DHT 11 - Αισθητήρας Υγρασίας και Θερμοκρασίας - Datasheet - <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>

[14] PIR sensor - Datasheet - <https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>

### Internet Site

[15] Industry 4.0 - Digitization for productivity and grow - Sep 2015 - [https://www.europarl.europa.eu/thinktank/en/document.html?reference=EPRS\\_BRI%282015%29568337](https://www.europarl.europa.eu/thinktank/en/document.html?reference=EPRS_BRI%282015%29568337)

[16] Smart Home Technology Poised for Blockbuster Growth - url (<https://www.statista.com/chart/15736/smart-home-market-forecast/>)

### **Paper in Conference Proceedings**

[17] C. G. C. Carducci, A. Monti, M. H. Schraven, M. Schumacher and D. Mueller, "Enabling ESP32-based IoT Applications in Building Automation Systems," 2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT), 2019, pp. 306-311, doi: 10.1109/METROI4.2019.8792852.[8] H. A. Nimr, "Defuzzification of the outputs of fuzzy controllers," presented at 5th International Conference on Fuzzy Systems, Cairo, Egypt, 2006. - <https://ieeexplore.ieee.org/document/8792852>

[18] A. Zare and M. T. Iqbal, "Low-Cost ESP32, Raspberry Pi, Node-Red, and MQTT Protocol Based SCADA System," 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2020, pp. 1-5, doi: 10.1109/IEMTRONICS51293.2020.9216412.- <https://ieeexplore.ieee.org/document/9216412>

[19] M. S. Mahamud, M. S. R. Zishan, S. I. Ahmad, A. R. Rahman, M. Hasan and M. L. Rahman, "Domicile - An IoT Based Smart Home Automation System," 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST), 2019, pp. 493-497, doi: 10.1109/ICREST.2019.8644349. - <https://ieeexplore.ieee.org/document/8644349>

### **Journal Articles**

[20] Industry 4.0 in Germany - The Obstacles Regarding Smart Production in the Manufacturing Industry - Aug 2018

[21] JRC Technical Reports - Smart home and appliances: State of the art - Serrenho T., Bertoldi P. 2019

[22] A. Maier, A. Sharp and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," 2017 Internet Technologies and Applications (ITA), 2017, pp. 143-148, doi: 10.1109/ITECHA.2017.8101926. - <https://ieeexplore.ieee.org/document/8101926>

[23] Ashton, K (2009) That "Internet of Things" Thing: In the Real World Things Matter More than Ideas. RFID Journal.

[24] <http://www.rfidjournal.com/articles/view?4986>

[25] H. U. Rehman, M. Asif and M. Ahmad, "Future applications and research challenges of IOT," 2017 International Conference on Information and Communication Technologies (ICICT), 2017, pp. 68-74, doi: 10.1109/ICICT.2017.8320166.

[26] I. Zhou et al., "Internet of Things 2.0: Concepts, Applications, and Future Directions," in IEEE Access, vol. 9, pp. 70961-71012, 2021, doi: 10.1109/ACCESS.2021.3078549.

## ΠΑΡΑΡΤΗΜΑ Α : Κώδικας συσκευής IoT

Πίνακας 5 - Πηγαίος κώδικας σε C/C++ για την υλοποίηση του λογισμικού του μικροεπεξεργαστή ESP32. Ο κώδικας αναπτύχθηκε σε περιβάλλον Arduino IDE

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <DHT.h>
#include "FS.h"
#include "SD.h"
#include "SPI.h"
#include "time.h"

#define DHTPIN 4 //DHT temperature & humidity sensor
#define RLEDPIN 25 //RED LED
#define GLEDPIN 26 //GREEN LED
#define PIRPIN 17 //PIR SENSOR
#define LIGHTPIN 34 //PHOTORESISTOR
#define SDCSPIN 5 //SD CARD MODULE
#define DHTTYPE DHT11 //DHT Sensor type (DHT11 or DHT22)
#define LIGHTZONES 5 //NUMBER OF LUMINOCITY LEVELS
#define SDATTEMPTS 5 //NUMBER OF ATTEMPTS TO MOUNT SD CARD BEFORE ABORTING
//Temperature & Humidity Sensor type
DHT dht(DHTPIN, DHTTYPE);
//WIFI COnfiguration
const char* ssid = "*****";
const char* password = "*****";
String serverName = "http://192.168.1.97:8080/addvalue";
String api_key = "*****";
HTTPClient http;
WiFiClient client;
//Data variables
int p;//PIR
int l;//light
float h;//humidity
```

```

float t;//temperature
uint32_t cpu_f;//cpu frequency
uint32_t xtal_f;//xtal frequency
uint32_t free_heap;//free heap memory on ESP32
String sd_total_size;//total SD size in MB
String sd_used_size;//used SD size in MB
String mac_add;//MAC Address of the ESP32
//Time details for timestamps
const char* ntpServer = "pool.ntp.org";
const long  gmtoffset_sec = 7200; //+2hours Athens
const int   daylightOffset_sec = 3600;
String timestamp;
struct tm timeinfo;
//data packet message to server and SD Logging
String post_body;
//flag to check if SD is working or not(failed mount or no card inserted)
int sd_working = 0;
void setup() {
    Serial.begin(115200);
    Serial.println("Device Initialization");
    //Pins Init
    pinMode(PIRPIN, INPUT);
    pinMode(LIGHTPIN, INPUT);
    pinMode(RLEDPIN, OUTPUT);
    pinMode(GLEDPIN, OUTPUT);
    digitalWrite(RLEDPIN, HIGH);//RED LED ON indicating setup running
    digitalWrite(GLEDPIN, LOW);
    //DHT Sensor Init
    dht.begin();
    //Wifi Init
    WiFi.begin(ssid, password);
    Serial.print("Connecting ");
    while (WiFi.status() != WL_CONNECTED) {

```

```

        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("Connected to WiFi network with IP Address: ");
    Serial.println(WiFi.localIP());
    //Time Init
    configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
    //SD Card Init
    Serial.println("SD Card Mounting");
    int sd_mount_attempts = 0;
    while ((!SD.begin(SDCSPIN)) && (sd_mount_attempts < SDATTEMPTS)) {
        delay(500);
        sd_mount_attempts++;
        Serial.print(".");
    }
    Serial.println("");
    if (sd_mount_attempts == SDATTEMPTS) {
        Serial.println("Card Mount Failed");
    }
    else {
        sd_working = 1;
        Serial.println("Card Mount Success");
    }
    uint8_t cardType = SD.cardType();
    if (cardType == CARD_NONE) {
        Serial.println("No SD card attached");
    }
    digitalWrite(RLEDPIN, LOW); //RED LED OFF indicating setup completed
}
void loop() {
    digitalWrite(GLEDPIN, HIGH); //GREEN LED ON indicating the start of each measuring
loop
    //Photoresistor

```

```

l = analogRead(LIGHTPIN);
l = map(l, 0, 4095, 5, 1); //map analog values from 0 (bright) to 4095 (dark) to 5
zones (1:dark 5:bright)
//PIR
int p = digitalRead(PIRPIN);
//DHT11
h = dht.readHumidity();
t = dht.readTemperature();
//ESP32 Parameters
mac_add = WiFi.macAddress();
cpu_f = getCpuFrequencyMhz();
xtal_f = getXtalFrequencyMhz();
free_heap = ESP.getFreeHeap();
//Timestamp
if (!getLocalTime(&timeinfo, 20000)) {
    Serial.println("[TIME] NO NTP RESPONSE");
    timestamp = "NA";
}
else {
    char timeStringBuff[50]; //50 chars should be enough
    strftime(timeStringBuff, sizeof(timeStringBuff), "%Y-%m-%dT%H:%M:%S",
&timeinfo);
    String asString1(timeStringBuff);
    timestamp = asString1;
}
//Data packet preparation
post_body = "{\"device\": \"" + mac_add + "\", \"api_key\": \"" + api_key + "\", \"temperature\": "
+ String(t, 1) + ", \"humidity\": " + String(h, 1)
+ ", \"light\": " + String(l) + ", \"PIR\": " + String(p, 10) + ", \"mcu_freq\": " +
String(cpu_f, 10)
+ ", \"xtal_freq\": " + String(xtal_f, 10) + ", \"free_heap\": " +
String(free_heap, 10) + ", \"esp_time\": \"" + timestamp + "\"";
//SD Data preparation
if (sd_working) {
    //add card related data to packet

```

```

char buffer[20];
sprintf(buffer, "%llu", SD.totalBytes() / (1024 * 1024));
sd_total_size = (String)buffer;
sprintf(buffer, "%llu", SD.usedBytes() / (1024 * 1024));
sd_used_size = (String)buffer;
post_body = post_body + ", \"sd_total\":" + sd_total_size + ", \"sd_used\":" +
sd_used_size + "}";
//write new data packet to card
appendFile(SD, "/data.txt", post_body.c_str());
}
else {
//complete data packet without card related data
Serial.println("");
post_body = post_body + ", \"sd_total\":NA, \"sd_used\":NA}";
}
Serial.println(post_body); //Serial print the complete data packet
//Data packet sending to server
if (WiFi.status() == WL_CONNECTED) {
if (http.begin(client, "http://192.168.1.97:8080/addvalue")) {
http.addHeader("Content-Type", "application/json");
int httpResponse = http.POST(post_body);
if (httpResponse) {
Serial.println("http.POST SUCCESS\n\n");
}
else {
Serial.println("http.POST FAILED\n\n");
}
}
else {
Serial.println("http.begin FAILED\n\n");
}
}
digitalWrite(GLEDPIN, LOW); //GREEN off until next loop starts

```

```

    delay(172800); //172800
}
//append data to file in SD Card
void appendFile(fs::FS &fs, const char * path, const char * message) {
    Serial.printf("Appending to file: %s\n", path);
    File file = fs.open(path, FILE_APPEND);
    if (!file) {
        Serial.println("Failed to open file for appending");
        return;
    }
    if (file.print(message)) {
        Serial.println("Message appended");
    }
    else {
        Serial.println("Append failed");
    }
    file.close();
}

```

## ΠΑΡΑΡΤΗΜΑ Β - Κώδικας εξυπηρετητή

Πίνακας 6 - Πηγαίος κώδικας σε Python για την υλοποίηση του λογισμικού του εξυπηρετητή με το πακέτο Flask. Ο εξυπηρετητής είναι υπεύθυνος για την συλλογή, διαχείριση, αποθήκευση και αποστολή των δεδομένων των μετρήσεων.

```

from flask import Flask, request, Response, jsonify
import json
from datetime import datetime, timezone
import logging
from tinydb import TinyDB, Query
from dcserverutils import DataCollectionServer

logger = logging.getLogger()
logger.setLevel(logging.DEBUG)
handler = logging.StreamHandler()
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s -

```

```

%(message)s')
handler.setFormatter(formatter)
logger.addHandler(handler)

# server for data collection
server = DataCollectionServer()
# Save values in json file
db = TinyDB('db/tinydb.json' )
datahandler = Query()
measurements = db.table("measurements")

app = Flask(__name__)
@app.route("/")
def hello_world():
    return "<p>Hello from IoT Sensor Project!</p>"

@app.route('/addvalue', methods=['POST'])
def addvalue():
    logger.debug("received request. post data: {0}".format(request.get_data()))
    esp32_request = request.get_json()
    api_key = esp32_request["api_key"]
    if api_key == 'paras2021':
        # Add timestamp
        dt = datetime.now(timezone.utc)
        esp32_request["created_at"] = dt.isoformat()
        # Add the db handling
        print(esp32_request)
        measurements.insert(esp32_request)
        server.thingspeak_update(esp32_request)
        server.influxdb_update(esp32_request)
        return json.dumps({"success": True}), 200
    else:
        return json.dumps({"success": False}), 401

@app.route('/measurements', methods=['GET'])
def getallmeasurements():
    json_data = measurements.all()
    return jsonify(json_data)

if __name__ == "__main__":
    app.run(host=server.ip, port=server.port, debug=False)

```

## ΠΑΡΑΡΤΗΜΑ C - Κώδικας προώθησης και αποθήκευσης των μετρήσεων

Πίνακας 7 - Πηγαίος κώδικας σε Python για την υλοποίηση της κλάσης του εξυπηρετητή που του δίνει δυνατότητες για την διαχείρισης, αποθήκευσης και αποστολή των δεδομένων των μετρήσεων

```
import traceback
import configparser
import urllib
from datetime import datetime, timezone
import influxdb

class DataCollectionServer():
    def __init__(self):
        self.configObj = configparser.ConfigParser()
        self.loadDataCollectionServerConfig()

    def applysettings(self):
        self.flaskSettings()
        self.thinkSpeakSettings()
        self.influxDBSettings()

    def flaskSettings(self):
        self.host = self.DataCollectionSettings['dc_server_hostname']
        self.ip = self.DataCollectionSettings['dc_server_ip']
        self.port = int(self.DataCollectionSettings['dc_server_port'])
        self.topics = self.DataCollectionSettings['dc_server_api_key']

    def thinkSpeakSettings(self):
        self.thinkspeak_endpoint = self.PushDataCollectionSettings['url']
        self.channelid = self.PushDataCollectionSettings['channel_id']
        self.write_api_key = self.PushDataCollectionSettings['write_api_key']
        self.read_api_key = self.PushDataCollectionSettings['read_api_key']
        fields = [field.strip() for field in
self.PushDataCollectionSettings['fields'].split(',')]
        labels = [label.strip() for label in
self.PushDataCollectionSettings['labels'].split(',')]
```

```

self.fields = dict(zip(labels,fields))
print("Thinkspeak fields: " + str(self.fields))

def influxDBSettings(self):
    self.influx_user = self.InfluxDBDataCollectionSettings['influx_user']
    self.influx_pass = self.InfluxDBDataCollectionSettings['influx_pass']
    self.influx_db = self.InfluxDBDataCollectionSettings['influx_db']
    self.influx_host = self.InfluxDBDataCollectionSettings['influx_host']
    self.influx_port = self.InfluxDBDataCollectionSettings['influx_port']
    self.influx_measurement_name =
self.InfluxDBDataCollectionSettings['influx_measurement_name']
    self.influx_body_fields = [ field.strip() for field in
self.InfluxDBDataCollectionSettings['influx_body_fields'].split(',') ]
    print("InfluxDB fields: " + str(self.influx_body_fields))

def loadDataCollectionServerConfig(self):
    try:
        self.configObj.read("config/iot.ini")
        self.DataCollectionSettings =
dict(self.configObj.items('flask'))
        self.PushDataCollectionSettings =
dict(self.configObj.items('thinkspeak'))
        self.InfluxDBDataCollectionSettings =
dict(self.configObj.items('influxdb'))
        self.applysettings()
    except:
        traceback.print_exc()
        print("Check the configuration file or the home directory of
the executable\n")
        # sys.exit(1)
        # Debug prints - To be removed.
        print("Config Sections: " + str(self.configObj.sections()))
        print("The DataCollectionSettings: " +
str(self.DataCollectionSettings))

def thingspeak_update(self, sample):
    if sample is None:
        print("ERROR - Thinkspeak Update failed - No Sample")
    url = self.thingspeak_endpoint + "update?api_key="

```

```

url += self.write_api_key
arguments = ""
for (value, field) in self.fields.items():
    if value in sample.keys():
        #print(value, field, sample[value])
        arguments += ("&" + field + "={}".format(sample[value]))
update_url = url + arguments
try:
    urllib.request.urlopen(update_url)
except urllib.error.HTTPError as e:
    print("ERROR - HTTP Error - Update Thinkspeak failed\n")
    print(e.reason)
except urllib.error.URLError as e:
    print("ERROR - BAD_URL - Update Thinkspeak failed\n")
    print(e.reason)
else:
    print("SUCCESS - Update Thinkspeak channel " + self.channelid)
    print("Thinkspeak POST: " + update_url)

def influxdb_update(self, sample):
    if sample is None:
        print("ERROR - InfluxDB Update failed - No Sample")
    body = {}
    body["measurement"] = self.influx_measurement_name
    if "created_at" not in sample.keys():
        dt = datetime.now(timezone.utc)
        body["time"] = dt.isoformat()
    body["time"] = sample["created_at"]
    body_fields = {}
    for (field, value) in sample.items():
        if field in self.influx_body_fields:
            body_fields[field] = value
    body["fields"] = body_fields
    try:
        self.influxclient = influxdb.InfluxDBClient(
            self.influx_host, self.influx_port,
            self.influx_user, self.influx_pass, self.influx_db)
        self.influxclient.write_points([body])
    except influxdb.exceptions.InfluxDBClientError as e:

```

```

        print("ERROR - InfluxDBClientError - Update InfluxDB failed\n")
        print(e.reason)
    else:
        print("SUCCESS - Update InfluxDB: " + self.influx_db)
        print("InfluxDB Body: " + str(body))

```

## ΠΑΡΑΡΤΗΜΑ D - Αρχείο ρυθμίσεων

Πίνακας 8 - Αρχείο διαχείρισης των ρυθμίσεων του λογισμικού του εξυπηρετητή για την εύκολη αρχικοποίηση, διαχείριση και συντήρηση των διαφορετικών ρυθμίσεων κάθε πακέτου λογισμικού στην τελική εφαρμογή IoT.

```

[flask]
dc_server_hostname = localhost
dc_server_ip = 0.0.0.0
dc_server_port = 8080
dc_server_api_key = *****

[thingspeak]
url = https://api.thingspeak.com/
channel_id = *****
write_api_key = *****
read_api_key = *****
fields = field1, field2, field3, field4, field5, field6, created_at
labels = temperature, humidity, PIR, free_heap, mcu_freq, xtal_freq,
created_at

[influxdb]
influx_user = grafana
influx_pass = *****
influx_db = esp32
influx_host = 127.0.0.1
influx_port = 8086
influx_measurement_name = esp32_data
influx_body_fields = temperature, humidity, PIR, free_heap, mcu_freq,
xtal_freq

[esp32]
esp32_ip = 192.168.1.103

```

## ΠΑΡΑΡΤΗΜΑ Ε - Πρότυπη φόρμα μετρήσεων JSON

Πίνακας 9 - Παράδειγμα της μορφής του αρχείου JSON. Αυτή η δομή δεδομένων χρησιμοποιείται για την αποστολή και αποθήκευση των δεδομένων των μετρήσεων

```
{
  "measurements": {
    "1": {
      "device": "esp22",
      "api_key": "****",
      "temperature": 24.9,
      "humidity": 28.9,
      "PIR": 0,
      "mcu_freq": 240,
      "xtal_freq": 40,
      "free_heap": 267700,
      "time": "23-09-2021 16:35:30.672439 UTC +0000"
    },
    "2": {
      "device": "esp22",
      "api_key": "****",
      "temperature": 24.8,
      "humidity": 30,
      "PIR": 0,
      "mcu_freq": 240,
      "xtal_freq": 40,
      "free_heap": 264484,
      "time": "23-09-2021 16:41:16.413772 UTC +0000"
    },
    "3": {
      "device": "esp22",
      "api_key": "****",
      "temperature": 24.8,
      "humidity": 30.1,
      "PIR": 0,
      "mcu_freq": 240,
      "xtal_freq": 40,
      "free_heap": 264380,
      "time": "23-09-2021 16:47:02.147135 UTC +0000"
    },
    "4": {
      "device": "esp22",
```

```

"api_key": "****",
"temperature": 24.8,
"humidity": 31,
"PIR": 0,
"mcu_freq": 240,
"xtal_freq": 40,
"free_heap": 264276,
"time": "23-09-2021 16:52:47.878119 UTC +0000"
},
"5": {
"device": "esp22",
"api_key": "****",
"temperature": 24.8,
"humidity": 31.1,
"PIR": 0,
"mcu_freq": 240,
"xtal_freq": 40,
"free_heap": 264184,
"time": "23-09-2021 16:58:33.608773 UTC +0000"
}
}

```

## ΠΑΡΑΡΤΗΜΑ F -Απαιτήσεις πακέτων λογισμικού για τον εξυπηρετητή Python-Flask

Πίνακας 10 - Οι εκδόσεις των πακέτων της Python 3 που χρειάζονται για την εκτέλεση του εξυπηρετητή συλλογής δεδομένων της εφαρμογής IoT.

```

certifi==2021.5.30
charset-normalizer==2.0.6
click==8.0.1
dataclasses==0.6
Flask==2.0.1
idna==3.2
importlib-metadata==4.8.1
influxdb==5.3.1
itsdangerous==2.0.1
Jinja2==3.0.1
MarkupSafe==2.0.1

```

```
msgpack==1.0.2
pkg-resources==0.0.0
python-dateutil==2.8.2
pytz==2021.1
requests==2.26.0
six==1.16.0
tinydb==4.5.1
typing-extensions==3.10.0.2
urllib3==1.26.7
Werkzeug==2.0.1
zipp==3.5.0
```