ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Συστημάτων Επικοινωνιών, Ηλεκτρονικής
& Συστημάτων Πληροφορικής

Ανάπτυξη NB-IoT συστήματος για την παρακολούθηση κλίσης κεραιών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΠΑΠΑΚΥΡΙΤΣΗ ΓΙΩΡΓΟΥ

Επιβλέπων: Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.

ΕΡΓΑΣΤΗΡΙΟ ΔΙΚΤΥΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Αθήνα, Ιούνιος 2020

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Συστημάτων Επικοινωνιών, Ηλεκτρονικής
& Συστημάτων Πληροφορικής

Ανάπτυξη NB-IoT συστήματος για την παρακολούθηση κλίσης κεραιών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΠΑΠΑΚΥΡΙΤΣΗ ΓΙΩΡΓΟΥ

Επιβλέπων: Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Ιουνίου 2020:

.....	.....	.....
Ευστάθιος Συκάς Καθηγητής Ε.Μ.Π.	Συμεών Παπαβασιλείου Καθηγητής Ε.Μ.Π.	Ιωάννα Ρουσσάκη Επίκουρη Καθηγήτρια

ΕΡΓΑΣΤΗΡΙΟ ΔΙΚΤΥΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Αθήνα, Ιούνιος 2020

.....

Παπακυρίτσης Γεώργιος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Τα τελευταία χρόνια, η 4η βιομηχανική επανάσταση είναι προ των πυλών. Εντός αυτού του πλαισίου, κομβικές τεχνολογίες όπως το διαδίκτυο των πραγμάτων (IoT) αναπτύσσονται με ραγδαίους ρυθμούς . Το διαδίκτυο των πραγμάτων είναι το πλαίσιο εκείνο που ενσωματώνει πρωτόκολλα επικοινωνίας, λογισμικό και αισθητήρες, επιδιώκοντας την συνδεσιμότητα με απώτερο στόχο την γρήγορη και ευφυή ανταλλαγή δεδομένων. Η συγκεκριμένη τεχνολογία θα συνεχίσει να αναπτύσσεται προωθώντας την περαιτέρω αυτοματοποίηση και βελτιστοποίηση καθημερινών διαδικασιών.

Σημαντικό ζήτημα της τεχνολογίας IoT είναι ότι οι τελικές συσκευές συχνά δεν διαθέτουν πρόσβαση σε σταθερή παροχή ρεύματος ή βρίσκονται σε απομακρυσμένα σημεία, σημεία στα οποία τα συμβατικά πρωτόκολλα επικοινωνίας δεν έχουν κάλυψη.

Τα δύο παραπάνω ζητήματα είναι κομβικά για έναν αισθητήρα που πρόκειται να τοποθετηθεί σε κεραιοσυστήματα ύψους δεκάδων μέτρων.

Μια λύση σε αυτά τα προβλήματα έρχεται να δώσει το πρωτόκολλο επικοινωνίας Narrow Band IoT, το οποίο εξασφαλίζει χαμηλές καταναλώσεις ενέργειας, καθώς και αυξημένη εμβέλεια.

Η διπλωματική πραγματοποιήθηκε σε συνεργασία με την COSMOTE .

Λέξεις Κλειδιά :

Αρχιτεκτονική IoT, Πρωτόκολλα επικοινωνίας, Narrow Band IoT, Arduino, Pycom, MQTT, Grafana, InfluxDB , Αδρανοποίηση, Εξοικονόμηση Ενέργειας, Φωτοβολταϊκό Πάνελ, C++, MicroPython.

Abstract

We live in the era of the 4th industrial revolution. Within this framework, key technologies such as the Internet of Things (IoT) are developing rapidly. The Internet of Things is the framework that integrates networks, software and sensors,in order to enable connectivity for the purpose of fast and intelligent data exchange. This technology will continue to be developed further promoting the automation and optimization of daily processes.
An important issue with IoT technology is that end-devices often do not have access to a fixed power supply or are located at remote locations where conventional communication protocols have no coverage.
These issues are critical to a sensor that is to be mounted on telecommunication antennas spanning tens of meters high.
A solution to these problems comes with the Narrow Band IoT communication protocol, which ensures low power consumption as well as increased range.
This thesis was conducted in collaboration with COSMOTE .

Keywords :
IoT Architecture, Communication Protocols, Narrow Band IoT, Arduino, Pycom, MQTT, Grafana, InfluxDB , Deepsleep, Energy Efficiency, Solar Panel, C++, MicroPython.

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω θερμά τον επιβλέπωντα καθηγητή κ.Συκά καθώς και τους συναδέλφους στην Cosmote, Γ.Λυμπερόπουλο, Θ.Πάζιο και Α.Πλασκασοβίτη. Χωρίς αυτούς η διπλωματική εργασία που κρατάτε στα χέρια σας δεν θα μπορούσε να υπάρξει.
Πιότερο από όλους θα ήθελα να ευχαριστήσω τους ανθρώπους με τους οποίους συλλογικοποιηθήκαμε και αγωνιστήκαμε εντος και εκτός του Αυτοδιαχειριζόμενου Στεκιού Πολυτεχνείου. Όλους αυτούς, που σε χρόνια δύσκολα, χτίσαμε μαζί συνείδηση ταξική και κοινωνική. Αυτή η εμπειρία, εντός και έναντια, σε μια ακαδημαϊκή σφαίρα διαχωρισμένη από το υπόλοιπο κοινωνικό, θα συνεχίσει να ζεί στις μελλοντικές επιλογές της ζωής και του αγώνα.

Περιεχόμενα

Πίνακας εικόνων

Κεφάλαιο 1
Εισαγωγή

1.1 Περιγραφή Προβλήματος

Η ασύρματη επικοινωνία ραδιοκυμάτων αποτέλεσε ένα σημείο τομής για την ανθρώπινη δραστηριότητα. Για πρώτη φορά -κατά την σύλληψη της στα τέλη του 19ου αιώνα - άνθρωποι σε απόσταση χιλιάδων χιλιομέτρων, μπορούσαν να επικοινωνήσουν άμεσα χωρίς την χρήση εκτενούς υλικής υποδομής (καλώδια – τηλέγραφος). Οι εφαρμογές των ασύρματων τηλεπικοινωνιών υπήρξαν επαναστατικές σε τομείς όπως το εμπόριο, τη βιομηχανία, την πολιτισμική επικοινωνία, τη διάχυση των ειδήσεων, την πλατιά επικοινωνία όλων των κοινωνικών ζητημάτων. Όμως οι ασύρματες επικοινωνίες δεν είναι επ' ουδενί απλώς μια τεχνολογική κατάκτηση του παρελθόντος αλλά αποτελούν σε μεγάλο βαθμό τις τεχνολογίες πάνω στις οποίες οικοδομείται η 4η βιομηχανική επανάσταση.

Τεχνολογίες αιχμής όπως τα αυτόνομα οχήματα, οι έξυπνες οικιακές συσκευές, οι βιομηχανικοί αυτοματισμοί κ.α. βασίζουν την λειτουργία τους πάνω στην ύπαρξη ποιοτικών και κυρίως αξιόπιστων τηλεπικοινωνιακών δικτύων νέας γενιάς (4/5G, NB-IoT κ.α.).

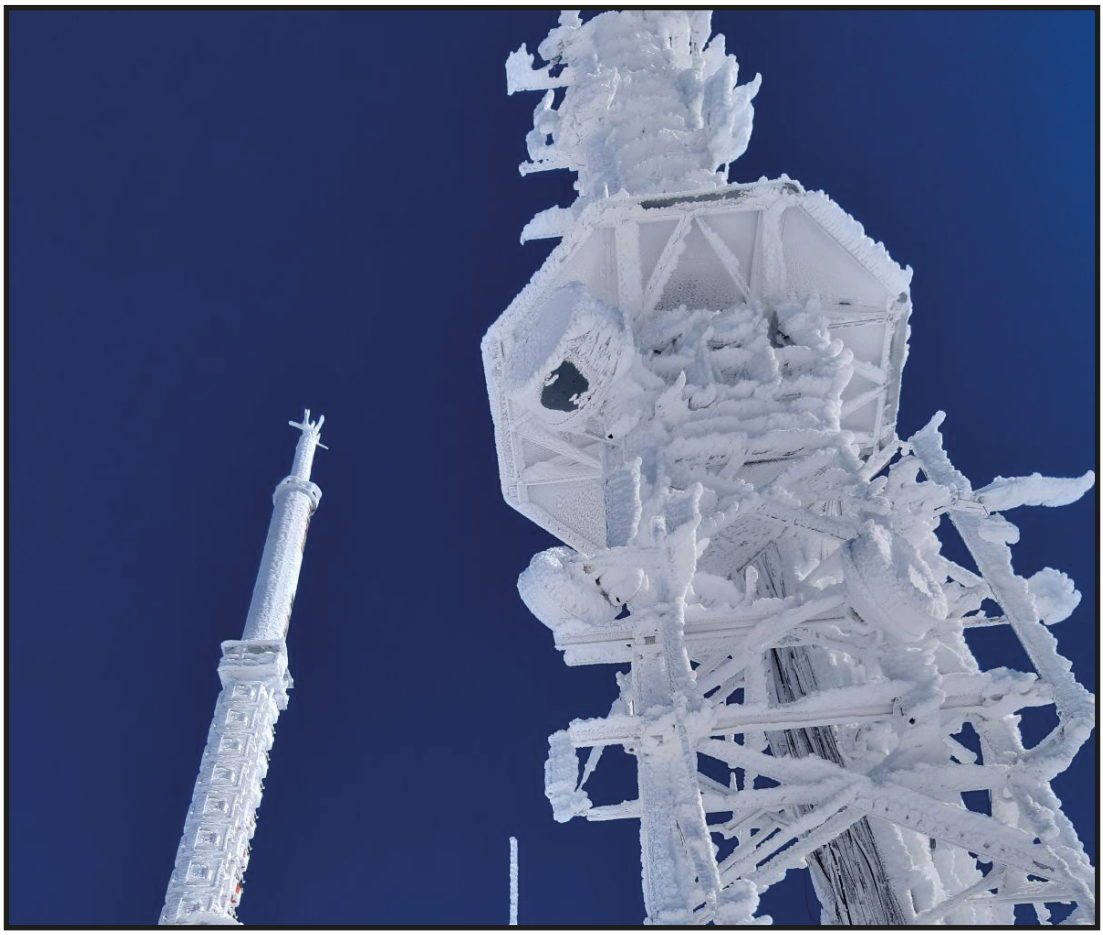
Η υποδομή που φέρει εις πέρας το παραπάνω τηλεπικοινωνιακό έργο, είναι ένα πολύπλοκο δίκτυο που περιλαμβάνει από γιγαντιαία κεραιοσυστήματα ύψους δεκάδων μέτρων και σταθμούς βάσης χιλιάδων συνδρομητών, μέχρι την μικροκλίμακα των μικρο-κεραιών (small cells) εγκατεστημένων σε σημεία με πληθυσμιακές πυκνώσεις και εξειδικευμένες ανάγκες.

Ας εστιάσουμε στο πρώτο σκέλος, αυτό των τηλεπικοινωνιακών πύργων που αποτελούν την ραχοκοκκαλιά (backbone) μιας υποδομής

ασύρματων επικοινωνιών. Έχουμε να κάνουμε με κατασκευές μεγάλου ύψους οι ίδιες, τοποθετημένες σε στρατηγικά σημεία μεγάλου υψομέτρου και φέροντας εξοπλισμό πολλών εκατομμυρίων ευρώ. Εκεί υπάρχουν εγκατεστημένες δεκάδες μεμονωμένες κεραίες, εκ των οποίων κάποιες καλύπτουν το τηλεπικοινωνιακό έργο μεγάλων περιοχών, ενώ άλλες στοχεύουν στην διασύνδεση μεταξύ τηλεπικοινωνιακών πύργων με κατευθυντικές ασύρματες ζεύξεις ακριβείας. Η τοποθέτηση των κεραιών ως προς την κλίση είναι ένα ζήτημα κομβικό που υπαγορεύεται από προσομοιώσεις μοντέλων κάλυψης. Απόκλιση έστω και λίγων μοιρών νομοτελειακά οδηγεί σε απώλεια τηλεπικοινωνιακού έργου.

Λόγω της φύσης τους (τοποθέτηση σε μεγάλο υψόμετρο) , τα κεραιοσυστήματα δέχονται πολύ υψηλές καταπονήσεις από τα καιρικά φαινόμενα. Αυτές οι καταπονήσεις αναπόφευκτα οδηγούν σε αλλαγές στην κλίση των κεραιών με τα ανάλογα προβλήματα που προκύπτουν από αυτό. Λόγω της αντικειμενικής δυσκολίας πρόσβασης στους πύργους, η απώλεια της ορθής κλίσης γίνεται αντιληπτή πολύ αργά, όταν αυτή έχει αρχίσει

εικόνα 1.1: Κεραιοσύστημα τοποθετημένο σε ύψος που επιβλέπει αστικό ιστό



εικόνα 1.2: Το μεγάλο ύψος και τα καιρικά φαινόμενα καταπονούν τα κεραιοσυστήματα

να προκαλεί προβλήματα στις επικοινωνίες. Παράλληλα η διαδικασία ελέγχου των κεραιών για απώλεια κλίσης και η επιδιόρθωση τους, είναι μια διαδικασία δύσκολη και επικίνδυνη. Η λύση του προβλήματος της απομακρυσμένης επίβλεψης της κλίσης των κεραιοσυστημάτων αποτελεί μια πρόκληση και θα πρέπει να εξασφαλίζει την ενεργειακή αυτονομία αλλά και την αποτελεσματική και αδιάλειπτη λειτουργία υπό συνθήκες έντονων καταπονήσεων. [1]

1.2 Σκοπός Διπλωματικής

Σκοπός της παρούσας διπλωματικής είναι η υλοποίηση μιας λύσης Διαδικτύου των Πραγμάτων (Internet of Things – IoT) στο πρόβλημα

που περιγράφηκε παραπάνω, δηλαδή η κατασκευή ενός ολοκληρωμένου συστήματος το οποίο θα εποπτεύει την κλίση τηλεπικοινωνιακών κεραιών. Πιο συγκεκριμένα στην παρούσα διπλωματική αναπτύσσεται ένα IoT σύστημα το οποίο θα έχει τις παρακάτω λειτουργίες :

- Καταγραφή της κλίσης με την χρήση αισθητήρων επιτάχυνσης.
- Σύνδεση στο δίκτυο Narrow Band IoT
- Αποστολή των καταγραφών κλίσης στην υποδομή του εργαστηρίου.
- Αποθήκευση των καταγραφών σε κατάλληλη βάση δεδομένων.
- Αποτύπωση των δεδομένων κλίσης σε κατάλληλο γραφικό περιβάλλον.
- Δυνατότητα ειδοποιήσεων προς τον τελικό χρήστη σε περίπτωση που καταγράψουμε μη επιτρεπτές τιμές κλίσης.

Οι κύριες προκλήσεις του συστήματος είναι η αδιάλειπτη λειτουργία, η ενεργειακή αυτονομία και η ανθεκτικότητα για μεγάλα διαστήματα λειτουργίας και υπό αντίξοες συνθήκες, καθώς η λύση σχεδιάστηκε για εφαρμογή σε κεραίες μεγάλου ύψους και τοποθετημένες σε μεγάλο υψόμετρο, οπού η ανθρώπινη παρέμβαση είναι δυσχερής, τα καιρικά φαινόμενα ισχυρά και η ηλεκτροδότηση μη εξασφαλισμένη.

1.3 Απαιτήσεις

Στο πλαίσιο του προβλήματος που παρουσιάστηκε στα παραπάνω κεφάλαια, προκύπτουν οι παρακάτω απαιτήσεις για την λύση που σχεδιάστηκε στα πλαίσια αυτής της διπλωματικής.

Σταθερή περιοδική λειτουργία για μεγάλο χρονικό διάστημα.

Η φύση του προβλήματος απαιτεί τον επανέλεγχο της κλίσης ανά

τακτά χρονικά διαστήματα. Η κλίση δεν είναι ένα μέγεθος που απαιτεί την συχνή μέτρηση του (πάνω από 3 φορές την ημέρα) όσο την μέτρηση του με μια σταθερότητα σε μεγάλο χρονικό βάθος.

Υψηλή δυνατότητα εξοικονόμησης ενέργειας.

Καθότι η λύση θα τοποθετηθεί σε κεραιοσυστήματα χωρίς εγγυημένη παροχή ενέργειας καθ' όλο το μήκος, η απαίτηση για ενεργειακή εξοικονόμηση είναι κομβική. Θέλουμε ελαχιστοποίηση της κατανάλωσης κατά τα διαστήματα που δεν παίρνουμε μετρήσεις, καθώς και φόρτιση του συσσωρευτή με την χρήση ηλιακού πάνελ.

Ακρίβεια των μετρήσεων.

Έχοντας να κάνουμε με ένα πρόβλημα όπου η διαφοροποίηση μεταξύ ορθής και λανθασμένης λειτουργίας είναι στην κλίμακα των ~5^{o} μοιρών, γίνεται αντιληπτό ότι η πιστή αποτύπωση της γωνίας κλίσης είναι κεντρικής σημασίας.

Αποτελεσματική και ελαφριά (lightweight) μεταφορά δεδομένων μέσω δικτύου.

Η απαίτηση για σίγουρη μεταφορά δεδομένων είναι αυτονόητη. Άλλο τόσο αυτονόητη – αφού μιλάμε για ένα αυτόνομο IoT σύστημα - είναι η απαίτηση για μεταφορά δεδομένων με χαμηλή χρήση πόρων, ενεργειακών και δικτυακών.

Ανθεκτικότητα στις καιρικές συνθήκες.

Οι αντίξοες καιρικές συνθήκες που επικρατούν στην κορυφή των τηλεπικοινωνιακών πύργων, είναι ο καθοριστικός παράγοντας που οδηγεί στο αρχικό πρόβλημα στο οποίο ερχόμαστε να δώσουμε λύση. Αυτονόητα η ανθεκτικότητα του συστήματός μας σε αυτές τις συνθήκες, είναι ιδιαίτερα σημαντική.

1.4 Περιγραφή Λύσης

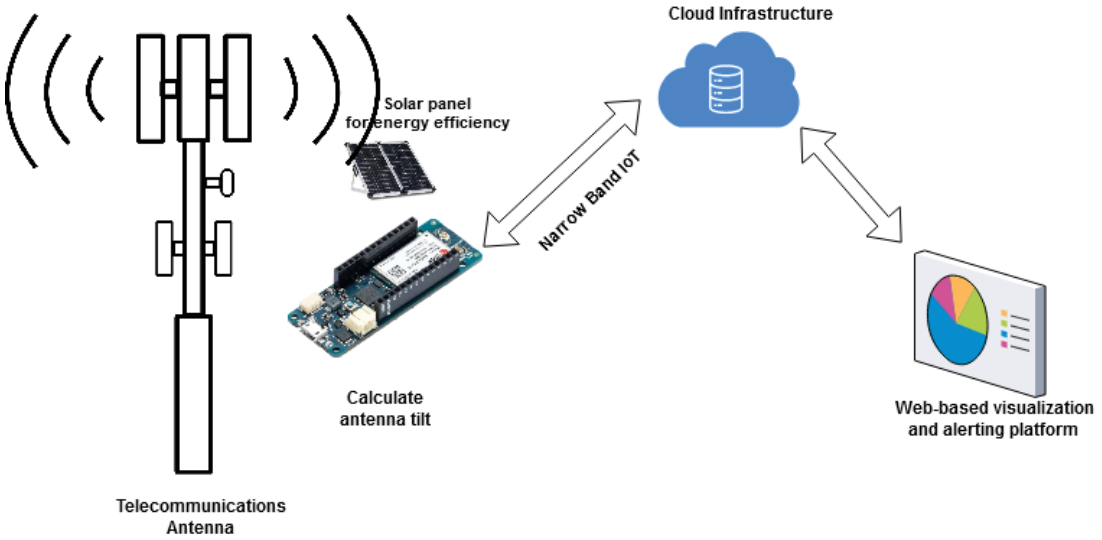
Ο ανωτέρω σκοπός ενός συστήματος καταγραφής κλίσης υλοποιείται με την ενοποίηση και ολοκλήρωση των παρακάτω υποσυστημάτων:

- ***Ακραία Στοιχεία***

Τελικές συσκευές Internet of Things κατάλληλα προσαρμοσμένες με αισθητήρες καταγραφής κλίσης, καθώς και σύστημα συσσωρευτών και φωτοβολταϊκού πάνελ. Οι εν λόγω συσκευές τοποθετημένες κατάλληλα στις κεραίες, θα καταγράφουν την κλίση και με την χρήση του NB-IoT δικτύου θα την αποστέλλουν στην υποδομή του εργαστηρίου.

- ***Υποδομή λήψης και αποθήκευσης των καταγραφών***

Backend υποδομή που χρησιμοποιείτε για την λήψη των μηνυμάτων με την χρήση πρωτοκόλλου επικοινωνίας χαμηλής χρήσης πόρων, την κατάλληλη επεξεργασία και αποθήκευση τους σε βάση δεδομένων.



εικόνα 1.3: Υψηλού επιπέδου περιγραφή του συστήματος

• ***Γραφικό περιβάλλον αποτύπωσης των δεδομένων και αποστολής ειδοποιήσεων***

Διαδικτυακό περιβάλλον μέσω του οποίου ο τελικός χρήστης έχει την δυνατότητα να διαχειρίζεται τα δεδομένα του και να εξάγει οργανωμένα συμπεράσματα για την ορθή λειτουργία του υπό παρακολούθηση συστήματος. Παράλληλα θα διαθέτει την δυνατότητα δημιουργίας ειδοποιήσεων ώστε να ενημερώνεται σε πραγματικό χρόνο για αλλαγές κλίσης εκτός φυσιολογικού πλαισίου.

Κατά την διάρκεια της διπλωματικής αναπτύχθηκαν δυο παρεμφερείς IoT λύσεις για το παραπάνω πρόβλημα. Πίο συγκεκριμένα, ύστερα από σύγκριση πολλαπλών IoT πλατφορμών ανάπτυξης (prototyping boards) επιλέχθηκαν οι δύο με τις μεγαλύτερες δυνατότητες και προοπτικές να επιλύσουν το πρόβλημα. Στην συνέχεια αναπτύχθηκαν τα ανάλογα λογισμικά ανά συσκευή και δοκιμάστηκε η αποτελεσματική τους λειτουργία με παράλληλη σύγκριση μεταξύ τους. Οι συσκευές συνδυάστηκαν με αισθητήρες επιτάχυνσης κατάλληλους για τον υπολογισμό της κλίσης ενώ φέραν ενσωματωμένα τα αναγκαία κυκλώματα για την σύνδεση στο δίκτυο και την αποστολή των πειραματικών δεδομένων. Λόγω της ιδιαιτερότητας της τοποθέτησης σε κεραίες, μελετήθηκαν οι ενεργειακές απαιτήσεις των συστημάτων καθώς και οι απαιτήσεις ανθεκτικότητας, ακρίβειας και αξιοπιστίας των λύσεων. Συγκεκριμένα μελετήθηκαν οι ενεργειακές καταναλώσεις των αισθητήρων σε διαφορετικά σενάρια χρήσης, η επίδραση των καιρικών φαινομένων στην ακρίβεια των μετρήσεων κλίσης , η πιστότητα μέτρησης των αισθητήρων κλίσης υπό διαφορετικές γωνίες καθώς και ορισμένες προσομοιώσεις λειτουργίας των λύσεων σε εργαστηριακό περιβάλλον.

Για την επικοινωνία των αισθητήρων με το εργαστήριο εγκαταστάθηκε και παραμετροποιήθηκε κατάλληλα, ένας εξυπηρετητής επικοινωνίας, με σκοπό την υλοποίηση ενός πρωτοκόλλου επικοινωνίας χαμηλής χρήσης πόρων δικτύου και άρα φιλικού ως προς τα IoT δίκτυα.

Για την αναζήτηση και παρουσίαση των δεδομένων στο γραφικό

περιβάλλον χρησιμοποιήθηκε βάση δεδομένων χρονοσειρών. Για την παρουσίαση των δεδομένων κλίσης και την παροχής της δυνατότητα επίβλεψης και ενημέρωσης για κρίσιμες αλλαγές, υλοποιήθηκε ένα διαδικτυακό γραφικό περιβάλλον που επικοινωνώντας άμεσα με την βάση δεδομένων, μας δίνει την δυνατότητα οπτικοποίησης των μετρήσεων αλλά και αποστολής ειδοποιήσεων με βάση τις επιλογές του τελικού χρήστη.

1.5 Δομή Διπλωματικής

Η διπλωματική εργασία αποτελείται από συνολικά 7 Κεφάλαια. Στο παρόν κεφάλαιο γίνεται μία περιγραφή του προβλήματος που αντιμετωπίζεται και μία συνοπτική περιγραφή της IoT λύσης που αναπτύχθηκε. Στο 2ο Κεφάλαιο γίνεται εισαγωγή στην τεχνολογία Διαδικτύου των Πραγμάτων. Στο 3ο Κεφάλαιο περιγράφεται η αρχιτεκτονική που χρησιμοποιήθηκε, τόσο στο κομμάτι της επιλογής των ακραίων στοιχείων όσο και στο κομμάτι της σταθερής υποδομής, όπως οι εξυπηρετητές δικτυακής κίνησης, βάσης δεδομένων και διαδικτυακής απεικόνισης. Στο 4ο Κεφάλαιο παρουσιάζονται τα πειράματα που πραγματοποιήθηκαν Στο 5ο Κεφάλαιο συνοψίζονται οι παρατηρήσεις και τα συμπεράσματα που προέκυψαν. Στο 6ο Κεφάλαιο παρουσιάζονται τα προβλήματα που αντιμετωπίστηκαν, καταγράφονται ορισμένες διορθώσεις, καθώς και προβληματισμοί και ιδέες για την μελλοντική εξέλιξη της λύσης. Στο Παράρτημα παρουσιάζονται οι πηγαίοι κώδικες. Τέλος παρατίθεται η βιβλιογραφία.

Κεφάλαιο 2
Τεχνολογίες Διαδικτύου των Πραγμάτων

2.1 Ορισμός

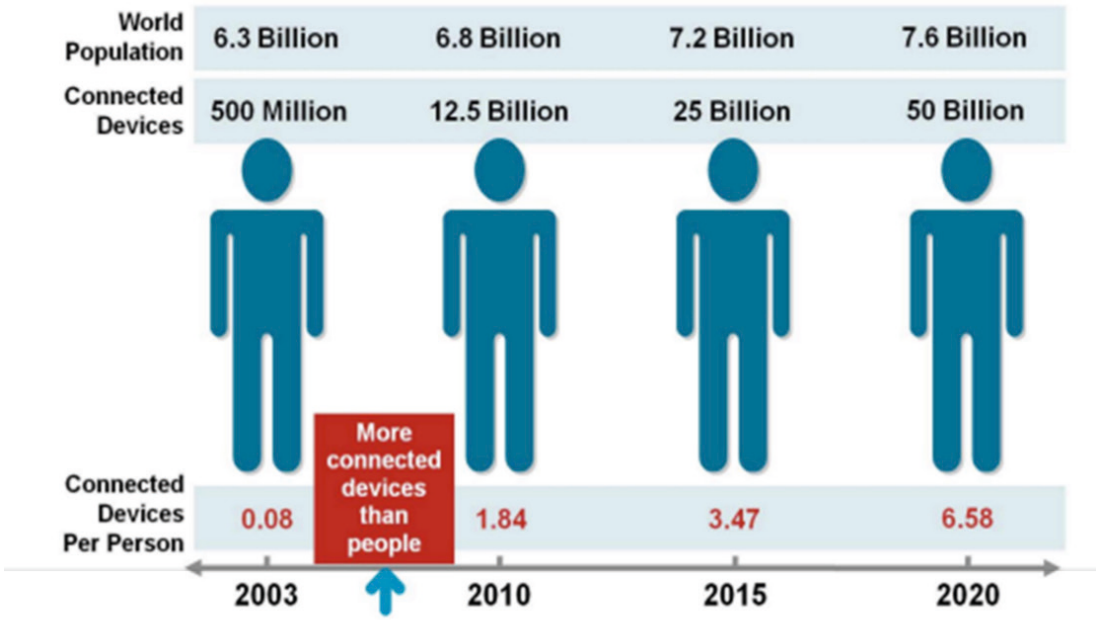
Ως Διαδίκτυο των πραγμάτων (αγγλικά : Internet of Things) μπορούμε να ορίσουμε το νέο εκείνο τεχνολογικό ρεύμα στο οποίο συσκευές προσαρμοσμένες με τα κατάλληλα κυκλώματα και πρωτόκολλα δικτυακής λειτουργίας, με αισθητήρες και ενεργοποιητές καθώς και το ανάλογο λογισμικό συνδέονται μεταξύ τους ή με το ευρύτερο διαδίκτυο ώστε να παρέχουν χρήσιμες υπηρεσίες. Όταν αναφερόμαστε σε "πράγματα", δεν αναφερόμαστε σε κάτι περιορισμένο, αλλά σε συσκευές οι οποίες υπάρχουν σε κάθε πλευρά της ανθρώπινης δραστηριότητας, και που πλέον αποκτούν δυνατότητα διασύνδεσης, βελτιστοποιώντας την λειτουργικότητά τους. Ενδεικτικά κινητά τηλέφωνα, έξυπνα πλυντήρια, έξυπνα συστήματα άρδευσης, οχήματα, αισθητήρες θερμοκρασίας, κίνησης, υγρασίας κ.α. συνδεδεμένα στο δικτυακό νέφος, συνθέτουν κάποια από τα "πράγματα" του Διαδικτύου των πραγμάτων. [2]

2.2. Ιστορία του Διαδικτύου των Πραγμάτων

Η ιδέα για τη δημιουργία ενός δικτύου έξυπνων συσκευών άρχισε να συζητείται από το 1982 στο Carnegie Mellon University, όπου δημιουργήθηκε η πρώτη συνδεδεμένη με το διαδίκτυο συσκευή, ένας αυτόματος πωλητής αναψυκτικών, που ανέφερε αν τα αναψυκτικά που είχαν προστεθεί ήταν κρύα. Περί το 1991 (MarkWeiser) έγινε μια σειρά από δημοσιεύσεις σχετικά με την εξέλιξη των υπολογιστών τον 21ο

αιώνα, η οποία έθεσε τη βάση για την ανάπτυξη του IoT. Το 1994 (Reza Raji) έγινε η πρώτη περιγραφή της ιδέας του IoT η οποία ήταν η εξής: «η μεταφορά μικρών πακέτων δεδομένων σε ένα μεγάλο σύνολο κόμβων, προκειμένου να ενοποιηθούν και να αυτοματοποιηθούν τα πάντα, από μικρές οικιακές συσκευές μέχρι ολόκληρα εργοστάσια». Μεταξύ του 1993 και του 1996 αρκετές εταιρείες πρότειναν κάποιες λύσεις , όμως τελικά το 1999 άρχισε η πραγματική εξέλιξη του κλάδου, όταν προτάθηκε η επικοινωνία συσκευής με συσκευή στο Παγκόσμιο Οικονομικό Φόρουμ. Η ιδέα του Internet of Things έγινε πρώτη φορά γνωστή το 1999 μέσω του Auto-ID Center του Πανεπιστημίου της Μασαχουσέτης (MIT), μέσα από δημοσιεύσεις σχετικές με την ανάλυση της αγοράς. Η ταυτοποίηση μέσω ραδιοσυχνοτήτων (Radio Frequency Identification, RFID) θεωρήθηκε από τους ιδρυτές του Auto-ID ως προαπαιτούμενο για την ανάπτυξη του IoT. Η λογική ήταν ότι εάν όλα τα αντικείμενα είχαν κάποια διακριτικά (identifiers), τότε οι υπολογιστές θα μπορούσαν να τα διαχειριστούν και να τα καταγράψουν ευκολότερα. Εκτός από τη χρήση του RFID για τον χαρακτηρισμό των αντικειμένων θα μπορούσαν να χρησιμοποιηθούν και άλλες τεχνολογίες, όπως η επικοινωνία κοντινού πεδίου (Near Field Communication, NFC), τα barcodes, τα QR codes καθώς και το ψηφιακό υδατογράφημα. Ως αρχική σκέψη, στόχος ήταν η υλοποίηση του IoT μέσω του εξοπλισμού όλων των αντικειμένων παγκοσμίως με μικρές συσκευές αναγνώρισης ή διακριτικά που μπορούν να αναγνωστούν από μηχανές, κάτι που θα είχε ως αποτέλεσμα μια πλήρη αλλαγή στην καθημερινότητα. Ο όρος Internet of Things εισήχθη από τον Kevin Ashton έναν εκ των ιδρυτών του Auto-ID Center στο οποίο έγινε αναφορά παραπάνω. Τον χρησιμοποίησε για μία παρουσίαση το 1999 και έκτοτε καθιερώθηκε σαν όρος. Την τελευταία δεκαετία ο ορισμός αυτός κατέληξε να προσδιορίζει ένα ολοένα και μεγαλύτερο φάσμα εφαρμογών στους τομείς της υγείας, των δημοσίων υπηρεσιών, των μεταφορών κλπ. Παρόλο που η έννοια του όρου 'Thing' έχει μεταβληθεί με την εξέλιξη της τεχνολογίας, ο στόχος της χρήσης πληροφοριών από υπολογιστικά συστήματα χωρίς την ανθρώπινη παρέμβαση παραμένει ο ίδιος. Σύμφωνα με την Cisco IBSG το Internet of Things «γεννήθηκε» πραγματικά τα έτη 2008-2009 όταν

και είχαμε εκρηκτική αύξηση στον αριθμό των συσκευών με δυνατότητα σύνδεσης στο διαδίκτυο. Πιο συγκεκριμένα, το έτος 2003 με πληθυσμό 6.3 δισεκατομμύρια είχαμε 500 εκατομμύρια συνδεδεμένες συσκευές. Το έτος 2010 ο πληθυσμός της γης αυξήθηκε σε 6.8 δισεκατομμύρια και οι συνδεδεμένες συσκευές σε 12.5 δισεκατομμύρια που σημαίνει ότι αναλογούσαν 1.84 συσκευές στον κάθε άνθρωπο. Σύμφωνα με εκτιμήσεις της ίδιας εταιρείας το 2020 θα έχουμε 50 δισεκατομμύρια συνδεδεμένες στο διαδίκτυο συσκευές. Αυτή η εκτίμηση φαίνεται να βρίσκεται πολύ κοντά στα πραγματικά σημερινά δεδομένα. [3][4][5][6]



εικόνα 2.1: Παρελθόν, παρόν και μέλλον του IoT

2.3 Η χρησιμότητα της τεχνολογίας IoT

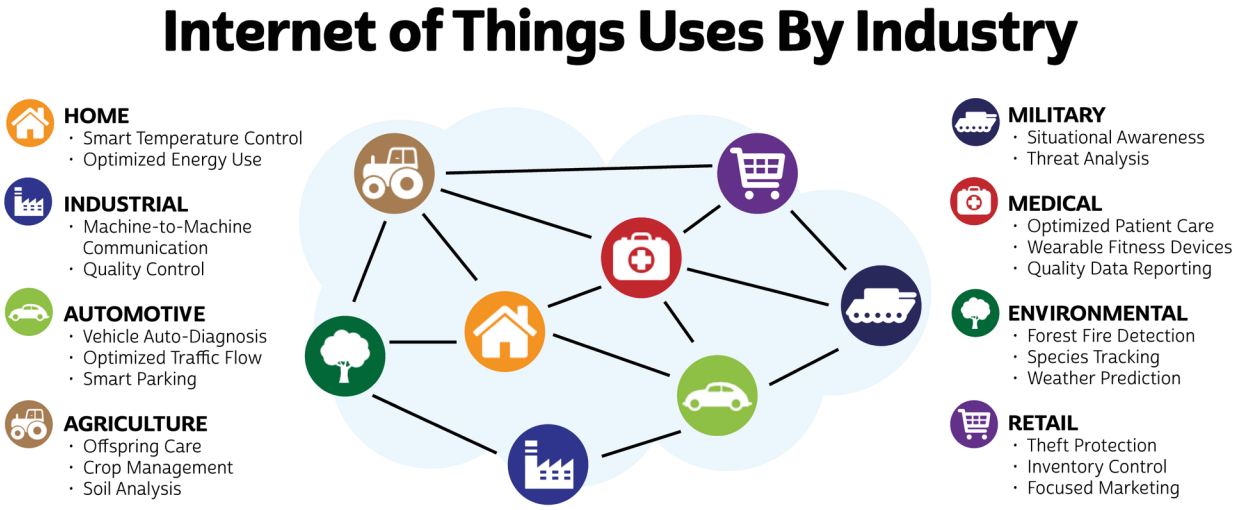
Η ενσωμάτωση της τεχνολογίας IoT στη ζωή του ανθρώπου μπορεί να προσφέρει πολλαπλά πλεονεκτήματα καθώς θα βοηθήσει τα άτομα, και το κοινωνικό σύνολο σε καθημερινή βάση. Τομείς οπού αυτό το νέο τεχνολογικό πεδίο μπορεί να προσφέρει, είναι η υγεία, η απλοποίηση

και ο αυτοματισμός στην καθημερινή ζωή καθώς και οι αυτοματισμοί και οι βελτιστοποιήσεις στο εργασιακό πεδίο. Η εφαρμογή τους στην υγεία θα μπορούσε να αποδειχτεί εξαιρετικά επωφελής τόσο για ένα άτομο όσο και για μία κοινωνία. Σένσορες κατάλληλα ενσωματωμένοι σε ένα IoT νέφος θα μπορούσαν να βοηθήσουν στην καταγραφή ζωτικών λειτουργιών ασθενών επιτρέποντας στα νοσοκομεία να παρακολουθούν την κατάσταση τους σε πραγματικό χρόνο, προσφέροντας ένα σημαντικό εργαλείο για την μείωση της θνησιμότητας. Παράλληλα, καθώς το μέγεθος των συσκευών μειώνεται, και σε συνδυασμό με την αύξηση της ενεργειακής αυτονομίας τους, δεν θα ήταν απίθανο να δούμε στο μέλλον μικρο-συσκευές που χρησιμοποιώντας τεχνολογίες IoT και αλγόριθμους σμήνους θα πλοηγούνται στο εσωτερικό του ανθρώπινου σώματος, δίνοντας λύσεις σε μέχρι στιγμής ανίατα ιατρικά ζητήματα. Προφανώς όλα τα παραπάνω υπό μια αυστηρή και κοινωνική εποπτεία πάνω σε ζητήματα βιοηθικής και προστασίας της ιδιωτικότητας.

Επίσης η τεχνολογία IoT θα μπορούσε να προσφέρει πολλά σε πολιτικές/οικιακές χρήσεις. Ένα τυπικό παράδειγμα είναι οι – βασισμένοι σε IoT – οικιακοί αυτοματισμοί. Μέσω αυτών μπορούμε να βελτιστοποιήσουμε την θέρμανση και την ψύξη των κατοικιών, εξοικονομώντας σε οικονομικό επίπεδο αλλά και βοηθώντας στην μείωση του περιβαλλοντικού αποτυπώματος. Εφαρμογές IoT χρησιμοποιούνται για την αυτόματη ρύθμιση του φωτισμού, για την αυτόματη και εξειδικευμένη ύδρευση ενός οικιακού κήπου καθώς και για την καταγραφή των αποθεμάτων τροφίμων και την αυτόματη παραγγελία όταν αυτά εξαντληθούν. Άλλες εφαρμογές ευφυών βρίσκονται στο πεδίο των IX και περιλαμβάνουν από εφαρμογές αυτόματης οδήγησης, και εύρεσης της βέλτιστης διαδρομής, μέχρι συστήματα ασφαλείας που ειδοποιούν τις υπηρεσίες υγείας σε περίπτωση ατυχήματος.

Σε επίπεδο δημόσιας σφαίρας, οι εφαρμογές είναι πολλαπλές. Συστήματα IoT μπορούν να εγκατασταθούν από δήμους με πολλαπλά πλεονεκτήματα. Μια εφαρμογή βρίσκεται στο πεδίο αποκομιδής των σκουπιδιών (το κύριο έργο -βάση προϋπολογισμού- των δήμων), όπου

με την χρήση έξυπνων κάδων που επικοινωνούν την πληρότητα τους μπορούμε να βελτιστοποιήσουμε τα δρομολόγια. Μια άλλη βρίσκεται με την καταγραφή σε πραγματικό χρόνο των ελεύθερων θέσεων δημοτικής στάθμευσης και η προβολή της πληροφορίας στους δημότες, με εξειδικευμένη εφαρμογή στο κινητό.



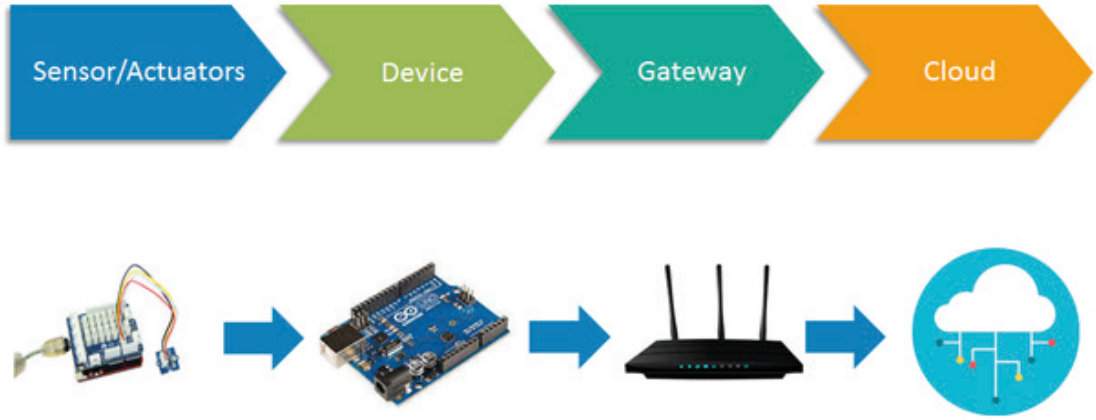
εικόνα 2.2: Εφαρμογές του IoT ανά κλάδο

Στο κομμάτι της εργασίας οι αυτοματισμοί και οι βελτιστοποιήσεις που μπορούν να γίνουν πραγματικότητα με την χρήση των εν λόγω τεχνολογιών, έχουν την προοπτική να απελευθερώσουν ένα τεράστιο κομμάτι ανθρώπινου μόχθου ανοίγοντας τον δρόμο για την αύξηση της παραγωγικότητας της εργασίας και γιατί όχι την ανάλογη μείωση του ωραρίου εργασίας. Στην βιομηχανία θα συνεισφέρουν στην παρακολούθηση των μηχανημάτων σε πραγματικό χρόνο και έτσι θα συντελέσουν στην κατάλληλη συντήρηση τους και στον σωστό συγχρονισμό τους. Επίσης με την παρακολούθηση της χημικής σύστασης των χώρων θα εξασφαλίζεται ασφαλές και υγιές περιβάλλον για τους εργαζόμενους. Τα έξυπνα ηλεκτρικά δίκτυα θα βοηθήσουν στην αποτελεσματική σύνδεση των ανανεώσιμων πηγών ενέργειας, στην βελτίωση της αξιοπιστίας του συστήματος και

στην εξοικονόμηση ενέργειας διότι οι συσκευές θα μπορούν να λαμβάνουν αποφάσεις και να προσαρμόζονται χωρίς ανθρώπινη καθοδήγηση. Στη γεωργία η παραγωγή θα μπορούσε να αυξηθεί, τα έξοδα από ζημιές να μειωθούν και οι ρύποι να ελαχιστοποιηθούν. Για παράδειγμα ο έλεγχος τα συστατικών του εδάφους θα μπορούσε να προσφέρει σε σημαντικό βαθμό για την υλοποίηση των συγκεκριμένων στόχων. [7][8]

2.4 Αρχιτεκτονική συστήματος IoT

Στην ουσία, η αρχιτεκτονική IoT είναι το σύστημα πολλών στοιχείων: αισθητήρες, πρωτόκολλα, ενεργοποιητές, υπηρεσίες cloud και στρώματα. Λόγω της πολυπλοκότητάς της, την αναλύουμε σε 4 στάδια . Αυτός ο διαχωρισμός σε 4 στάδια επιλέγεται για να συμπεριλαμβάνει όλους αυτούς τους διάφορους τύπους στοιχείων σε ένα ενοποιημένο δίκτυο με έναν τρόπο κατανοητό, βοηθητικό προς την ανάπτυξη εφαρμογών και την ορθή κλιμάκωσή τους.



εικόνα 2.3: Αρχιτεκτονική συστήματος IoT

Επίπεδο 1: Αισθητήρες και ενεργοποιητές:

Οι αισθητήρες συλλέγουν τα δεδομένα από το περιβάλλον ή το αντικείμενο υπό μέτρηση και τα μετατρέπουν σε χρήσιμα δεδομένα. Τα

παραδείγματα βρίσκονται ανάμεσα μας. Ένα καθημερινό όσο και περίπλοκο παράδειγμα της καθημερινής ζωής είναι οι εξειδικευμένοι αισθητήρες του κινητού τηλεφώνου, που ανιχνεύουν το διάνυσμα της βαρυτικής έλξης - και τη σχετική θέση του τηλεφώνου πάνω στον πλανήτη - και τη μετατρέπουν σε δεδομένα που το τηλέφωνό μπορεί να χρησιμοποιήσει για τον προσανατολισμό της συσκευής.

Οι ενεργοποιητές (actuators) είναι συσκευές εκείνες που παρεμβαίνουν για να αλλάξουν τις φυσικές συνθήκες που παράγουν τα δεδομένα. Ένας ενεργοποιητής μπορεί, για παράδειγμα, να κλείσει μια τροφοδοσία ρεύματος, να ρυθμίσει μια βαλβίδα ροής αέρα ή να μετακινήσει μια ρομποτική λαβή σε μια διαδικασία συναρμολόγησης.

Το επίπεδο ανίχνευσης / ενεργοποίησης καλύπτει τα πάντα από παλαιές βιομηχανικές συσκευές έως συστήματα ρομποτικής κάμερας, ανιχνευτές στάθμης νερού, αισθητήρες ποιότητας αέρα, επιταχυνσιόμετρα και μετρητές καρδιακού ρυθμού.

Και το πεδίο εφαρμογής των IoT αισθητήρων ή ενεργοποιητών επεκτείνεται ραγδαία, χάρη εν μέρει στις τεχνολογίες δικτύου ασύρματων αισθητήρων χαμηλής κατανάλωσης (NB-IoT, Lora κλπ) που επιτρέπουν αυξημένη ενεργειακή αυτονομία καθώς και της τεχνολογίας Power over Ethernet, η οποία επιτρέπει τη λειτουργία συσκευών σε ένα ενσύρματο LAN χωρίς την ανάγκη για σύνδεση με μία πηγή εναλλασσόμενου ρεύματος.

Στην αρχιτεκτονική IoT, κάποια επεξεργασία δεδομένων μπορεί να συμβεί σε κάθε ένα από τα τέσσερα στάδια. Ωστόσο, ενώ γίνεται να επεξεργαστούμε τα δεδομένα στον αισθητήρα, αυτό που μπορούμε να κάνουμε είναι περιορισμένο από την ισχύ επεξεργασίας που είναι διαθέσιμη σε κάθε συσκευή IoT. Τα δεδομένα όμως βρίσκονται στο επίκεντρο μιας αρχιτεκτονικής IoT και πρέπει να επιλέγουμε ανάμεσα στην άμεση παροχή τους και στην σε βάθος ανάλυση και επεξεργασία τους. Όσο πιο άμεσα και ανεπεξέργαστα χρειαζόμαστε τα δεδομένα, τόσο πιο κοντά στις τελικές συσκευές (αισθητήρες) πρέπει να γίνεται η επεξεργασία σας.

Αντίθετα για πιο αναλυτικές πληροφορίες που απαιτούν την

εκτεταμένη επεξεργασία και σύνθεση των αρχικών δεδομένων, θα χρειαστεί να μεταφερθούν αυτά σε συστήματα βασισμένα ή κέντρο δεδομένων και μπορεί να φέρει μαζί διάφορες πηγές δεδομένων.

Επίπεδο 2: Επίπεδο δικτύου

Τα δεδομένα από τους αισθητήρες ξεκινούν σε αναλογική μορφή. Τα δεδομένα αυτά πρέπει να συγκεντρωθούν και να μετατραπούν σε ψηφιακές ροές για περαιτέρω επεξεργασία καθώς προχωράνε στα παρακάτω στάδια. Τα συστήματα απόκτησης δεδομένων (Data acquisition systems - DAS) εκτελούν αυτές τις λειτουργίες απόκτησης και μετασχηματισμού δεδομένων. Το DAS συνδέεται με το δίκτυο αισθητήρων, συγκεντρώνει τα δεδομένα και πραγματοποιεί την μετατροπή από αναλογικά δεδομένα σε ψηφιακή πληροφορία. Η πύλη Δικτύου λαμβάνει τα συγκεντρωτικά και ψηφιοποιημένα δεδομένα και τα δρομολογεί μέσω ασύρματων και/ή ενσύρματων δικτύων χρησιμοποιώντας συνήθως αρχιτεκτονική του TCP/IP stack για την προώθηση τους στο Επίπεδο 3 για περαιτέρω επεξεργασία.

Τα συστήματα βαθμίδας 2 συχνά βρίσκοντας σε στενή απόσταση από τους αισθητήρες και τους ενεργοποιητές. Για παράδειγμα, μια αντλία μπορεί να περιέχει δεκάδες αισθητήρες και ενεργοποιητές που τροφοδοτούν τα δεδομένα σε μια συσκευή συσσωμάτωσης δεδομένων που ψηφιοποιεί τα δεδομένα. Αυτή η συσκευή μπορεί να είναι φυσικά συνδεδεμένη με την αντλία. Μια παρακείμενη συσκευή θα επεξεργάζεται τότε τα δεδομένα και θα τα διαβιβάζει στα συστήματα του Επιπέδου 3 ή 4. Προφανώς όσο τα συστήματα βελτιώνονται τόσο γίνεται όλα και πιο σφιχτή ενσωμάτωση των στοιχείων του Επιπέδου 2 σε ενιαίες συσκευές. Χαρακτηριστικό παράδειγμα είναι συσκευές όπως το NodeMCU ή η σειρά Arduino MKR οπού συνδυάζουν τους μηχανισμούς ψηφιοποίησης των αναλογικών δεδομένων και μηχανισμούς σύνδεσης σε IP based δίκτυα.

Γιατί να επεξεργαστούμε τα δεδομένα; Οι αναλογικές ροές δεδομένων που προέρχονται από τους αισθητήρες δημιουργούν γρήγορα μεγάλους όγκους δεδομένων. Οι μετρήσιμες ιδιότητες του φυσικού κόσμου - κίνηση, τάση, δόνηση και ούτω καθεξής - μπορούν να δημιουργήσουν τεράστιες

ποσότητες συνεχώς μεταβαλλόμενων δεδομένων. Αρκεί να σκεφτούμε πόσο δεδομένα μπορεί να δημιουργήσει μια σύνθετη μηχανή όπως μια μηχανή αεροσκαφών σε μια ημέρα και δεν υπάρχει θεωρητικό όριο στον αριθμό των αισθητήρων που θα μπορούσαν να τροφοδοτούν τα δεδομένα σε ένα σύστημα IoT. Επιπλέον, ένα σύστημα IoT είναι πάντα σε λειτουργία, παρέχοντας συνεχή συνδεσιμότητα και τροφοδοσίες δεδομένων. Οι ροές δεδομένων του Διαδικτύου μπορεί να είναι τεράστιες κάνοντας ωφέλιμη το προεπεξεργασία τους.

Ένας άλλος λόγος που δεν θέλουμε την μεταβίβαση των δεδομένων σε αυτή τη μορφή είναι ότι τα αναλογικά δεδομένα έχουν συγκεκριμένα χρονικά και δομικά χαρακτηριστικά που απαιτούν επεξεργασία με την χρήση εξειδικευμένου λογισμικού. Είναι καλύτερο να μετατρέψουμε τα δεδομένα σε ψηφιακή μορφή και αυτό συμβαίνει στο Επίπεδο 2.

Οι έξυπνες πύλες, ουσιαστικά πύλες με επαυξημένες λειτουργίες, προσθέτουν δυνατότητες όπως στατιστικές/χρονικές αναλύσεις, προστασία από κακόβουλο λογισμικό και υπηρεσίες διαχείρισης δεδομένων. Τα συστήματα αυτά επιτρέπουν την ανάλυση των ροών δεδομένων σε πραγματικό χρόνο. Παρόλο που η παροχή πληροφοριών για τα δεδομένα είναι λίγο λιγότερο άμεση στην πύλη δικτύου απ' ότι θα ήταν όταν αποστέλλονται απευθείας από το επίπεδο αισθητήρα / ενεργοποιητή, η πύλη διαθέτει την υπολογιστική ισχύ για να καταστήσει τις πληροφορίες σε μορφή που είναι πιο κατανοητή από τους ενδιαφερόμενους.

Οι πύλες εξακολουθούν να είναι συσκευές ακροδεκτών - είναι εξωτερικές ως προς το κέντρο δεδομένων – οπότε η τοποθεσία έχει σημασία. Οι συσκευές DAS και οι πύλες δικτύου μπορεί να καταλήξουν σε ένα ευρύ φάσμα περιβαλλόντων από το πάτωμα ενός εργοστασίου έως φορητούς σταθμούς πεδίου, έτσι ώστε τα συστήματα αυτά να είναι συνήθως φορητά, εύχρηστα και ανθεκτικά ώστε να αντέχουν σε διακυμάνσεις της θερμοκρασίας, της υγρασίας, και δόνηση.

Στάδιο 3. "Μεθοριακά" συστήματα πληροφορικής (Edge IT)

Μόλις τα δεδομένα IoT έχουν ψηφιοποιηθεί και συγκεντρωθεί,

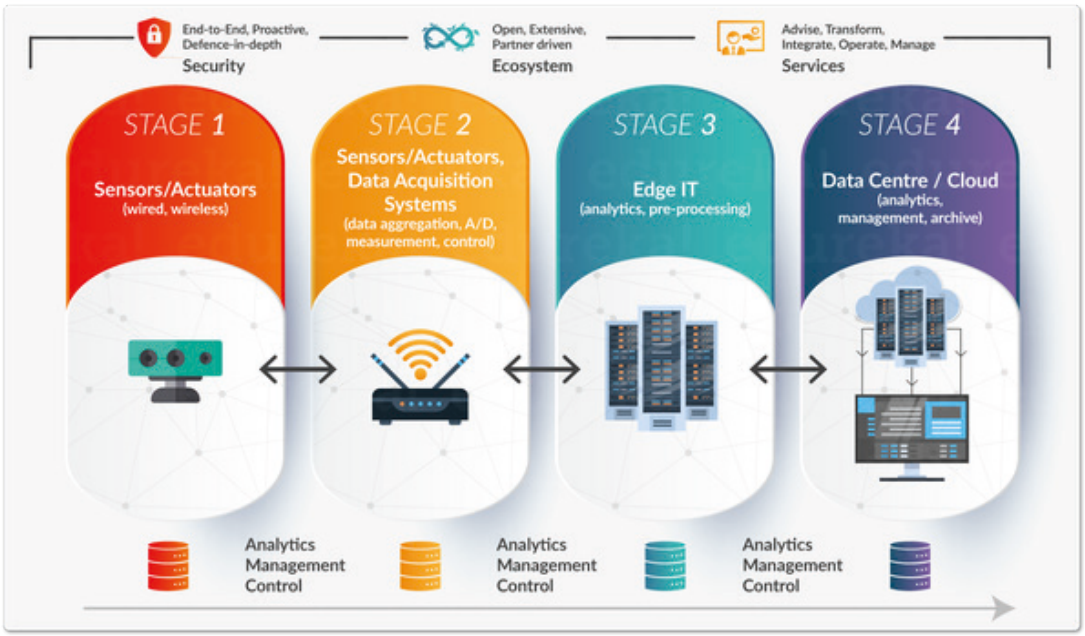
είναι έτοιμα να περάσουν στη σφαίρα της πληροφορικής. Ωστόσο, τα δεδομένα ενδέχεται να απαιτούν περαιτέρω επεξεργασία προτού εισέλθουν στο κέντρο δεδομένων. Κάπου εδώ "μπαίνουν στο παιχνίδι" τα edge IT συστήματα τα οποία εκτελούν αυτή την περαιτέρω ανάλυση. Αυτά τα συστήματα επεξεργασίας πληροφοριών μπορεί να βρίσκονται σε απομακρυσμένα γραφεία ή σε άλλες θέσεις στην περιφέρεια των δικτύων μας, αλλά γενικά αυτά βρίσκονται στην εγκατάσταση ή τη θέση όπου οι αισθητήρες βρίσκονται πιο κοντά στους αισθητήρες, όπως σε μια ντουλάπα καλωδίωσης.

Επειδή τα δεδομένα IoT μπορούν εύκολα να καταναλώσουν το εύρος ζώνης μιας ζεύξης δικτύου και να απορροφήσουν τους πόρους του κέντρου δεδομένων μας, είναι καλύτερο να διαθέτουμε συστήματα στα άκρα - στις εισόδους – του δικτύου μας, ικανά να εκτελέσουν αναλυτικές εργασίες ως έναν τρόπο να μειωθεί η επιβάρυνση της κεντρικής υποδομής. Εάν απλά είχαμε ένα μεγάλο "αγωγό δεδομένων" που θα οδηγούσε στο κέντρο δεδομένων και την cloud υποδομή, θα χρειαζόμασταν δυσανάλογα τεράστια χωρητικότητα. Θα αντιμετωπίζαμε επίσης προβλήματα ασφάλειας, προβλήματα αποθήκευσης και καθυστερήσεις στην επεξεργασία των δεδομένων. Με μια πιο σταδιακή προσέγγιση, μπορούμε να προεπεξεργαστούμε τα δεδομένα, να βγάλουμε συμπεράσματα από αυτά και να προωθήσουμε μόνο αυτά (τα συμπεράσματα) . Για παράδειγμα, αντί να μεταβιβάσουμε ανεπεξέργαστα δεδομένα δονήσεων από μία σειρά αντλίες νερού, θα μπορούσατε να συγκεντρώσουμε τα δεδομένα, να τα αναλύσουμε και να στείλετε μόνο εκτιμήσεις σχετικά με το πότε κάθε συσκευή λειτουργεί ή θα χρειαστεί service.

Επίπεδο 4: Το κέντρο δεδομένων και η υποδομή νέφους

Δεδομένα που απαιτούν περισσότερη επεξεργασία σε βάθος και όπου η ανατροφοδότηση δεν χρειάζεται να είναι άμεση, προωθούνται στο κέντρο δεδομένων είτε αυτό είναι φυσικό μηχάνημα είναι είναι μια σειρά αλληλοσυνδεόμενων συστημάτων βασισμένα σε υποδομές νέφους (cloud infrastracture) , όπου ισχυρά συστήματα πληροφορικής μπορούν

να αναλύουν, να διαχειρίζονται και να αποθηκεύουν τα δεδομένα με ασφάλεια. Προφανώς χρειαζόμαστε περισσότερο χρόνο για την εξαγωγή αποτελεσμάτων από το Επίπεδο 4, αλλά μπορούμε έτσι να έχουμε μια πιο εμπεριστατωμένη ανάλυση, καθώς και να συνδυάσουμε τα δεδομένα αισθητήρων μας με δεδομένα από άλλες πηγές για βελτιστοποίηση και κανονικοποίηση των αποτελεσμάτων. Η επεξεργασία του επιπέδου 4 μπορεί να γίνει σε φυσικές υποδομές, στο νέφος ή σε ένα υβριδικό σύστημα, αλλά ο τύπος επεξεργασίας που εκτελείται σε αυτό το στάδιο παραμένει ο ίδιος, ανεξάρτητα από την πλατφόρμα. [9]



εικόνα 2.4: Αρχιτεκτονική συστήματος IoT (2)

2.5 Τεχνολογίες IoT

2.5.1 Πρωτόκολλα Ασύρματης Επικοινωνίας

Αξίζει αναφερθεί ότι υπάρχουν δεκάδες πρωτόκολλα ασύρματης και ενσύρματης επικοινωνίας που αξιοποιούνται από συσκευές του Διαδικτύου των πραγμάτων. Για λόγους οικονομίας χώρου αλλά και λόγω του πλαισίου

της διπλωματικής που εστιάζει σε συσκευές ασύρματης επικοινωνίας, θα περιορίσουμε την αναφορά μας στα πιο σημαντικά από αυτά επιλέγοντας να περιλάβουμε ενδεικτικά τεχνολογίες επικοινωνίας μικρών, μεσαίων και μεγάλων αποστάσεων. Για το πρωτόκολλο ***Narrow Band IoT*** το οποίο και χρησιμοποιείται στην παρούσα διπλωματική εργασία θα γίνει ξεχωριστή αναλυτική αναφορά.

2.5.1.1 NFC(Near field communication)

Η επικοινωνία κοντινού πεδίου (NFC) είναι ένα σύνολο πρωτοκόλλων επικοινωνίας που επιτρέπουν σε δύο ηλεκτρονικές συσκευές, μία εκ των οποίων είναι συνήθως φορητή συσκευή για να καθιερώσουν επικοινωνία σε απόσταση 4 εκατοστών μεταξύ τους. Οι συσκευές NFC χρησιμοποιούνται σε συστήματα επαφών χωρίς πληρωμή, πχ σε χρεωστικές/πιστωτικές κάρτες. Μια πρόσφατη χρήση για το NFC είναι η ενσωμάτωση με το smartphone, ή ακόμα και με το smartwatch ώστε να ενσωματώσουν κυρίως εφαρμογές ανέπαφης πληρωμής.

2.5.1.2 RFID

Το RFID είναι τα αρχικά του όρου Radio Frequency Identification, η απόδοση του στα ελληνικά ορίζεται ως «ταυτοποίηση μέσω ραδιοσυχνοτήτων». Τα συστήματα RFID αποτελούν ένα υποσύνολο των Συστημάτων Αυτόματου Προσδιορισμού (Automatic Identification Systems). Ειδικότερα λειτουργεί ως γενικός όρος των τεχνολογιών που χρησιμοποιούν ραδιοκύματα για να προσδιορίσουν αυτόματα ανθρώπους ή αντικείμενα και αποτελεί την τεχνολογική εξέλιξη των ραβδωτών κωδίκων (barcode]). Η τεχνολογία RFID είναι γνωστή εδώ και 50 χρόνια. Χρησιμοποιήθηκε για πρώτη φορά από την πολεμική αεροπορία της Αγγλίας κατά τη διάρκεια του Β' Παγκοσμίου, για την αναγνώριση και τη διάκριση των εχθρικών από τα φιλικά αεροπλάνα. Κατά τη διάρκεια των επόμενων δεκαετιών, άρχισε να εδραιώνεται η χρήση και εκμετάλλευσή της. Αρχικά, σε πειραματικό στάδιο και σε εργαστηριακό επίπεδο, για να φτάσουμε στο σήμερα, όπου γίνεται λόγος για εφαρμογή της τεχνολογίας

RFID στην καθημερινή ζωή των ανθρώπων, κυρίως μέσω του εμπορίου. Δεδομένου ότι οι ετικέτες RFID μπορούν να συνδεθούν σε μετρητά, ρούχα, ζώα και ανθρώπους, η δυνατότητα ανάγνωσης προσωπικών πληροφοριών χωρίς συγκατάθεση δημιούργησε σοβαρές ανησυχίες για την προστασία της ιδιωτικής ζωής. Αυτές οι ανησυχίες οδήγησαν στην ανάπτυξη τυποποιημένων προδιαγραφών για θέματα ιδιωτικότητας και ασφάλειας.

2.5.1.3 Bluetooth L.E

Το Bluetooth Low Energy (Bluetooth LE) είναι μια ασύρματη τεχνολογία ασύρματου δικτύου που σχεδιάστηκε και διατέθηκε στο εμπόριο από την Bluetooth Special Interest Group (Bluetooth SIG) με στόχο τις νέες εφαρμογές στον τομέα της υγειονομικής περίθαλψης, των φαναριών, και τις βιομηχανίες οικιακής ψυχαγωγίας. Σε σύγκριση με το κλασικό Bluetooth, το Bluetooth Low Energy προορίζεται να προσφέρει σημαντικά μειωμένη κατανάλωση ενέργειας και κόστος ενώ διατηρεί ένα παρόμοιο εύρος επικοινωνίας. Τα λειτουργικά συστήματα κινητής τηλεφωνίας, συμπεριλαμβανομένων των iOS, Android, Windows Phone και BlackBerry, καθώς και τα λειτουργικά προσωπικών υπολογιστών macOS, Linux, Windows 8 και Windows 10, υποστηρίζουν εγγενώς τη χαμηλή ενέργεια Bluetooth. Το Bluetooth SIG προβλέπει ότι μέχρι το 2018 περισσότερο από το 90 τοις εκατό των smartphones με δυνατότητα Bluetooth θα υποστηρίζει Bluetooth Low Energy.

2.5.1.4 Z-wave

Το Z-Wave είναι ένα πρωτόκολλο ασύρματων επικοινωνιών που χρησιμοποιείται κυρίως για οικιακό αυτοματισμό. Χρησιμοποιεί ραδιοκύματα χαμηλής ενέργειας για να επικοινωνεί από συσκευή σε συσκευή, επιτρέποντας τον ασύρματο έλεγχο οικιακών συσκευών και άλλων συσκευών, όπως συστήματα ελέγχου ασφαλείας, θερμοστάτες, παράθυρα, κλειδαριές, πισίνες και πόρτες γκαράζ. Όπως και άλλα πρωτόκολλα και συστήματα που στοχεύουν στην αγορά αυτοματισμού

κατοικιών και γραφείων, ένα σύστημα αυτοματισμού Z-Wave μπορεί να ελεγχθεί από ασύρματο πληκτρολόγιο, πληκτρολόγιο τοίχου ή smart-phones, tablet ή υπολογιστές με πύλη Z-Wave ή συσκευή κεντρικού ελέγχου.

2.5.1.5 802.11 (WiFi)

Το IEEE 802.11 είναι μια οικογένεια προτύπων της IEEE για ασύρματα τοπικά δίκτυα (WLAN) που είχαν ως σκοπό να επεκτείνουν το 802.3 (Ethernet, το συνηθέστερο πρωτόκολλο ενσύρματης δικτύωσης υπολογιστών) στην ασύρματη περιοχή. Τα πρότυπα 802.11 είναι ευρύτερα γνωστά ως «WiFi» επειδή η WiFi Alliance, ένας οργανισμός ανεξάρτητος της IEEE, παρέχει την πιστοποίηση για τα προϊόντα που εμπίπτουν στις προδιαγραφές του 802.11. Αυτή η οικογένεια πρωτοκόλλων αποτελεί το καθιερωμένο πρότυπο της βιομηχανίας στο χώρο των ασύρματων τοπικών δικτύων.

Η ονομασία WiFi χρησιμοποιείται για να προσδιορίσει τις συσκευές WLAN που βασίζονται στην προδιαγραφή IEEE 802.11 b/g/n και εκπέμπουν σε συχνότητες 2.4GHz. Ωστόσο το WiFi έχει επικρατήσει και ως όρος αναφερόμενος συνολικά στα ασύρματα τοπικά δίκτυα. Συνήθεις εφαρμογές του είναι η παροχή ασύρματων δυνατοτήτων πρόσβασης στο Internet, τηλεφωνίας μέσω διαδικτύου (VoIP) και διασύνδεσης μεταξύ ηλεκτρονικών συσκευών όπως τηλεοράσεις, ψηφιακές κάμερες, DVD Player και ηλεκτρονικοί υπολογιστές. Σε φορητές ηλεκτρονικές συσκευές το 802.11 βρίσκει εφαρμογές ασύρματης μετάδοσης, όπως π.χ. στη μεταφορά φωτογραφιών από ψηφιακές κάμερες σε υπολογιστές για περαιτέρω επεξεργασία και εκτύπωση, αν και σε αυτόν τον τομέα έχει υποσκελιστεί από το πρωτόκολλο Bluetooth για τα πολύ μικρότερης εμβέλειας ασύρματα προσωπικά δίκτυα.

2.5.1.6 2G/3G/4G/5G

Το G αντιπροσωπεύει το “Generation” του δικτύου κινητής τηλεφωνίας και όσο πιο μεγάλος είναι ο αριθμός τόσο υψηλότερη

απόδοση υπάρχει στο ασύρματο δίκτυο. Στα δίκτυα δεύτερης γενιάς(2G) τα σήματα μεταδόθηκαν σε ψηφιακή μορφή και αυτό βελτίωσε δραματικά την ποιότητα των κλήσεων και μείωσε επίσης την πολυπλοκότητα της μετάδοσης δεδομένων. Η τρίτη γενιά κινητών δικτύων έχει γίνει δημοφιλής σε μεγάλο βαθμό χάρη στην ικανότητα των χρηστών να έχουν πρόσβαση στο Internet μέσω συσκευών όπως τα κινητά και τα tablet. Σήμερα, οι εταιρείες κινητής τηλεφωνίας προσφέρουν υπηρεσίες 4G με την θεωρητικά μέγιστη ταχύτητα να φτάνει τα 100Mbps. Το επόμενο μεγάλο βήμα στις τεχνολογίες κινητής τηλεφωνίας είναι η τεχνολογία 5G που προς το παρόν δεν έχει διατεθεί μαζικά στην αγορά. Το 5G θα προσφέρει ταχύτητες εώς 10Gbps με πολύ χαμηλό latency ενώ θα αντιμετωπίζει δραστικά το ζήτημα της μείωσης της παρεχόμενης υπηρεσίας με βάση των αριθμό των χρηστών

2.5.1.7 LORA

Το LoRa επιτρέπει την μετάδοση σε πολύ μεγάλες αποστάσεις

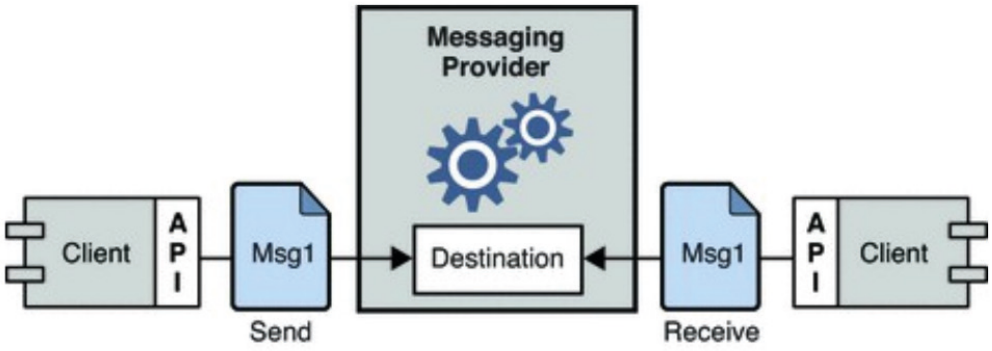
Τεχνολογία	Συχνότητα	Ρυθμός Δεδομένων	Απόσταση	Ισχύς	Κόστος
Z-wave	908.42 MHz	40Kb/s	30m	Χαμηλή	Μέτριο
WirelessHart	2.4 GHz	250Kb/s	100m	Μέτρια	Μέτριο
Zigbee	2.4 GHz	250Kb/s	100m	Χαμηλή	Μέτριο
802.15.4	2.4 GHz	250Kb/s	100m	Χαμηλή	Χαμηλό
Bluetooth	2.4 GHz	1.2-1.3 Mb/s	100m	Χαμηλή	Χαμηλό
LoRa	<1 GHz	<50Kb/s	2-5km	Χαμηλή	Μέτριο
LTE Cat 0/1	Cellular bands	1-10Mb/s	Αρκετά km	Μέτρια	Υψηλό
NB-IoT	Cellular bands	0.1-1Mb/s	Αρκετά km	Μέτρια	Υψηλό
SIGFOX	<1GHz	Πολύ χαμηλό	Αρκετά km	Χαμηλή	Μέτριο
Weightless	<1GHz	0.1-24Mb/s	Αρκετά km	Χαμηλή	Χαμηλό
Wi-Fi(11f/h)	2,4,5<1GHz	0.1-1Mb/s	Αρκετά km	Μέτρια	Χαμηλό
2G/3G	Cellular bands	10Mb/s	Αρκετά km	Υψηλή	Υψηλό

εικόνα 2.5: Σύγκριση πρωτοκόλλων ασύρματης επικοινωνίας

(πάνω από 10 χιλιόμετρα σε αγροτικές περιοχές) με χαμηλή κατανάλωση ενέργειας. Η τεχνολογία παρουσιάζεται σε δύο μέρη - το LoRa στο φυσικό στρώμα και το LoRaWAN στα ανώτερα στρώματα. Το LoRaWAN αντιπροσωπεύει το δίκτυο ευρείας περιοχής μεγάλης εμβέλειας (Long Rang Wide Area Network). Είναι ένα πρότυπο για την ασύρματη επικοινωνία που επιτρέπει στις συσκευές IoT να επικοινωνούν σε μεγάλη απόσταση με ελάχιστη κατανάλωση για τον συσσωρευτή.

2.5.2 Πρωτόκολλα επικοινωνίας MOM

Τα MOM (Message-oriented Middleware) πρωτόκολλα χρησιμοποιούνται για την αποστολή και λήψη των μηνυμάτων μεταξύ ετερογενών πλατφορμών. Δημιουργείται ένα κατανεμημένο επίπεδο επικοινωνιών που απομονώνει τον προγραμματιστή της εφαρμογής από τις λεπτομέρειες του εκάστοτε λειτουργικού συστήματος και διεπαφής. Ελαττώνεται η ανάμειξη του προγραμματιστή με τους μηχανισμούς master/slave (ή server/client), παρέχοντάς του το API για την απαιτούμενη επικοινωνία. Οι συσκευές επικοινωνούν μεταξύ τους (D2D –Device to Device- communication), τα δεδομένα συλλέγονται και αποστέλλονται στον server (D2S –Device to Server- communication) ο οποίος μοιράζεται την πληροφορία με άλλους server (S2S -Server to Server- communication) και με άλλες συσκευές και προγράμματα (clients). Εκτενέστερη αναφορά θα γίνει στο πρωτόκολλο MQTT το οποίο και χρησιμοποιούμε στην λύση



εικόνα 2.6: Πρωτόκολλα επικοινωνίας MOM

μας

2.5.2.1 MQTT

Το MQTT (Message Queuing Telemetry Transport) είναι ένα πρωτόκολλο του Data επιπέδου και λειτουργεί πάνω από TCP. Είναι ελαφρύ με μικρό footprint (χώρο που δεσμεύει στην RAM), γι' αυτό και είναι κατάλληλο για χρήση σε συσκευές περιορισμένων δυνατοτήτων και δίκτυα χαμηλού bandwidth ή/και υψηλού latency, συνεπώς σε εφαρμογές και δίκτυα IoT.

2.5.2.1.1 Μοντέλο Δημοσίευσης-Συνδρομής(Publish-Subscribe)

Το MQTT βασίζεται στο μοντέλο Δημοσίευσης-Συνδρομής (Publish-Subscribe - pub/sub). Το συγκεκριμένο μοντέλο στηρίζεται στην λογική ότι οι αποστολείς μηνυμάτων (publishers) δεν γνωρίζουν απευθείας τον παραλήπτη του κάθε μηνύματος. Εν αντιθέσει κατηγοριοποιούν τα μηνύματά τους με κάποιον συγκεκριμένο τρόπο και τα κάνουν publish (δημοσιεύουν) με βάση την κατηγορία τους. Αντίστοιχα οι αποδέκτες των μηνυμάτων (subscribers) δηλώνουν το ενδιαφέρον τους για κάποια συγκεκριμένη κατηγόρια και λαμβάνουν τα αντίστοιχα μηνύματα. Η κατηγοριοποίηση των μηνυμάτων γίνεται βασικά με δύο τρόπους:

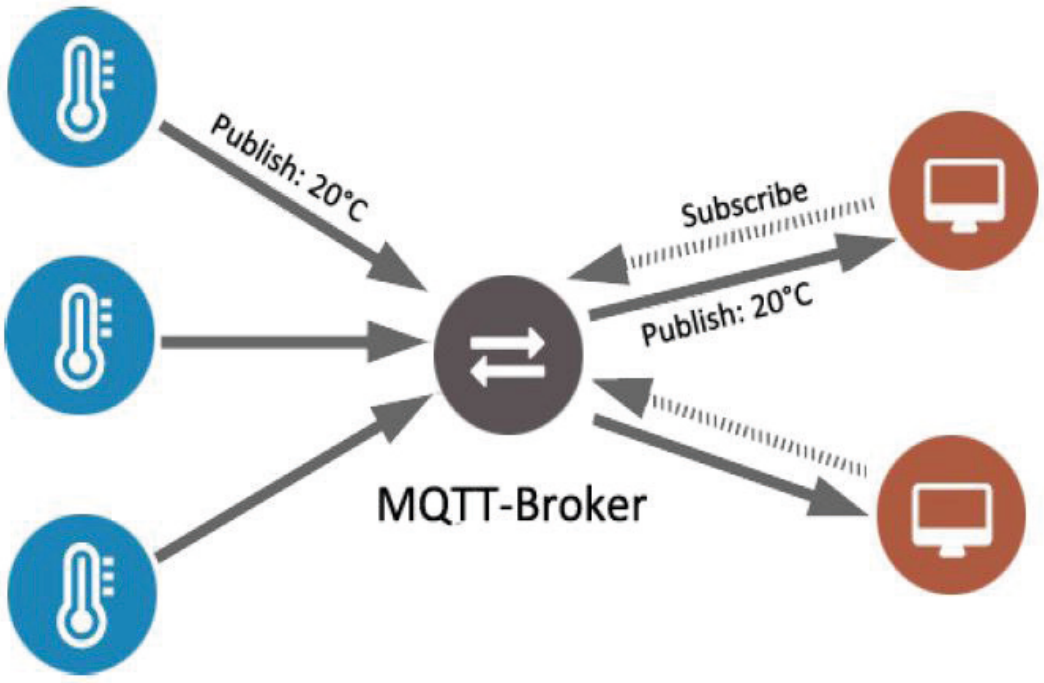
Βάσει του θέματος (subject-based filtering).

Σε αυτό τον τρόπο, κάθε μήνυμα γίνεται published σε κάποιο συγκεκριμένο topic(θέμα)και λαμβάνεται από όλους τους subscribers(συνδρομητές)που έχουν δηλώσει ενδιαφέρον στο topic αυτό. Με τον όρο topic περιγράφετε μια αυθαίρετη κατηγοριοποίηση με βάση την οποία κατευθύνεται η ροή των μηνυμάτων. Στην περίπτωση αυτή, οι publishers είναι αυτοί που είναι υπεύθυνοι για τον ορισμό των κατηγοριοποιήσεων/topics.

Βάσει του περιεχομένου (content-based filtering).

Σε αυτό τον τρόπο, κάθε μήνυμα φέρει κάποια συγκεκριμένα

χαρακτηριστικά και οι subscribers δηλώνουν τις προτιμήσεις τους σε χαρακτηριστικά. Στην περίπτωση αυτή, οι subscribers είναι ουσιαστικά υπεύθυνοι για το φιλτράρισμα των μηνυμάτων. Φυσικά μπορεί να χρησιμοποιηθεί και ο συνδυασμός των δύο παραπάνω μεθόδων. Δηλαδή μηνύματα που φέρουν χαρακτηριστικά αλλά γίνονται publish και σε topics και αντίστοιχα subscribers που δηλώνουν ενδιαφέρον σε συγκεκριμένα χαρακτηριστικά, εντός συγκεκριμένων topics. Συνηθισμένη πρακτική όταν χρησιμοποιούνται αντικειμενοστραφείς γλώσσες είναι και το φιλτράρισμα βάσει τύπου. Για παράδειγμα μπορεί ένας subscriber να επιθυμεί να λαμβάνει όλα τα μηνύματα που αφορούν Exceptions. Κάθε pub/sub σύστημα απαιτεί έναν τρόπο διαχείρισης της κίνησης των διαφόρων μηνυμάτων. Την διαχείριση αυτή την αναλαμβάνει συνήθως ένας ενδιάμεσος κόμβος που ονομάζεται broker (μεσάζοντας), στον οποίο αποστέλλονται τα μηνύματα και ο οποίος έχει γνώση για τα subscriptions που έχουν γίνει. Με αυτό τον τρόπο, ο broker είναι υπεύθυνος να αναμεταδώσει τα μηνύματα που του έρχονται σε όποιον παραλήπτη έχει κάνει subscribe στα αντίστοιχα



εικόνα 2.7: Πρωτόκολλο MQTT

topics. Η επικοινωνία προϋποθέτει την ύπαρξη ενός broker και άρα κάθε επικοινωνία αποτελείται από την αποστολή του μηνύματος από έναν client-publishser στο broker και από τον broker σε όποιον client-subscriber αφορά το μήνυμα. Όπως αναφέρθηκε παραπάνω, το pub/sub μοντέλο στηρίζεται στην απόζευξη αποστολέα και παραλήπτη, στην έλλειψη γνώσης δηλαδή του ενός για τον άλλον.

Αυτό μπορεί να αφορά:

1.Τοπική απόζευξη (space decoupling): Ο παραλήπτης και ο αποστολέας δεν γνωρίζουν ο ένας τον άλλον στην τοπολογία του δικτύου (π.χ. ip, port).

2.Χρονική απόζευξη (time decoupling): Ο παραλήπτης και ο αποστολέας δεν χρειάζεται να λειτουργούν την ίδια χρονική περίοδο.

3.Απόζευξη συγχρονισμού (synchronization decoupling): οι λειτουργίες publish και subscribe του παραλήπτη και του αποστολέα δεν συγκρούονται με την κανονική τους λειτουργία.

Όσον αφορά το MQTT, χρησιμοποιεί subject-based filtering με χρήση topics και μπορεί να υλοποιήσει και τις 3 παραπάνω περιπτώσεις απόζευξης. Συγκεκριμένα η διαμεσολάβηση ενός broker στην επικοινωνία ανάμεσα στους publishers και τους subscribers επιτυγχάνει την τοπική απόζευξη. Η χρονική απόζευξη μπορεί να υλοποιηθεί με αποθήκευση των μηνυμάτων στον broker για έναν offline client. Η συγκεκριμένη λειτουργία υλοποιείται από τους περισσοτέρους brokers, όπως και από τον Mosquitto. Τέλος, υπάρχουν πολλές βιβλιοθήκες για υλοποίηση των clients που δίνουν την δυνατότητα απόζευξης συγχρονισμού, όπως και η Paho.

2.5.2.1.2 Θέματα (Topics)

Στο MQTT τα topics είναι μια συμβολοσειρά (σε κωδικοποίηση UTF-8) μπορούν να δομούνται σε περισσότερα από ένα επίπεδα που χωρίζονται

μεταξύ τους με "/" (forward slash). Η δομή αυτή μοιάζει αρκετά με μια δομή φακέλων (file system) σε ένα λειτουργικό σύστημα. Για να κάνει ένας client publish ή subscribe δεν χρειάζεται να προϋπάρχει το topic. Αντίθετα κάθε topic δημιουργείται αυτόματα με την αποστολή ενός μηνύματος σε αυτό ή με το subscription σε αυτό. Αυτή είναι μια λειτουργικότητα του MQTT που το βοηθάει να είναι ελαφρύ χωρίς να επιβαρύνει τους clients με ελέγχους για το αν υπάρχουν τα topics κλπ. Κάθε συμβολοσειρά για να είναι έγκυρο όνομα θα πρέπει να έχει τουλάχιστον 1 χαρακτήρα, ενώ μπορεί να περιέχει και κενά, αριθμούς και σύμβολα (εκτός των δεσμευμένων /, +, # και δεν μπορεί να ξεκινάει με $). Επίσης διαχωρίζει κεφαλαία με πεζά γράμματα (case sensitive). Τα σύμβολα +, # είναι δεσμευμένα ως χαρακτήρες μπαλαντέρ επιπέδων. Συγκεκριμένα ο χαρακτήρας # χρησιμοποιείται για πολλαπλά επίπεδα και ο χαρακτήρας + χρησιμοποιείται για ένα επίπεδο.

Για παράδειγμα ένας client που έχει κάνει subscribe στο home/+/temperature, θα λαμβάνει μηνύματα από τα topics home/kitchen/temperature, home/livingRoom/temperature, home1/bedroom/temperature κλπ.

Ενώ ένας client που έχει κάνει subscribe στο antenna/# θα παίρνει μηνύματα από τα topics antenna/nw-attica/ilion, antenna/e-attica/sounio, antenna/thessaly κλπ καθώς και το antenna/. Μπορεί επίσης να γίνει και ο συνδυασμός δύο ή περισσοτέρων μπαλαντέρ από οποιοδήποτε είδος. Το σύμβολο $ στην αρχή ενός topic υποδηλώνει ένα topic που είναι δεσμευμένο από τον broker για την κατάστασή του π.χ αριθμός συνδεδεμένων clients, χρόνος λειτουργίας κ.α..

2.5.2.2 XMPP

Αποτελεί ένα πρωτόκολλο επικοινωνίας για middleware με γνώμονα τα μηνύματα που βασίζονται σε XML (Extensible Markup Language). Επιτρέπει την ανταλλαγή δεδομένων σε πραγματικό χρόνο μεταξύ δύο ή περισσοτέρων οντοτήτων δικτύου. Έχει χρησιμοποιηθεί και για συστήματα δημοσίευσης-εγγραφής, σηματοδότηση για VoIP, βίντεο, μεταφορά αρχείων, παιχνίδια, εφαρμογές Internet of Things (IoT) όπως το έξυπνο δίκτυο και υπηρεσίες κοινωνικής δικτύωσης. Επειδή το XMPP

είναι ένα ανοιχτό πρωτόκολλο, οι εφαρμογές μπορούν να αναπτυχθούν χρησιμοποιώντας οποιαδήποτε άδεια χρήσης λογισμικού και πολλές εφαρμογές διακομιστή, πελάτη διανέμονται ως ελεύθερο λογισμικό ανοιχτού κώδικα.

2.5.2.3 RESTful HTTP(Rest)

Ορίζει ένα σύνολο περιορισμών και ιδιοτήτων που βασίζονται στο HTTP. Οι υπηρεσίες Web που συμμορφώνονται με την αρχιτεκτονική του REST ή τις υπηρεσίες RESTful web, παρέχουν διαλειτουργικότητα μεταξύ συστημάτων υπολογιστών στο Διαδίκτυο. Οι υπηρεσίες ιστού που είναι συμβατές με το REST επιτρέπουν στα αιτούμενα συστήματα να έχουν πρόσβαση και να χειρίζονται τις αναπαραστάσεις των πόρων του διαδικτύου χρησιμοποιώντας ένα ομοιόμορφο και προκαθορισμένο σύνολο λειτουργιών. Χρησιμοποιείται συχνά σε εφαρμογές κινητών, ιστότοπους κοινωνικής δικτύωσης και αυτοματοποιημένες επιχειρηματικές διαδικασίες. Είναι ένα είδος επικοινωνίας μεταξύ τελικού χρήστη και των διακομιστών.

2.5.2.4 RabbitMQ:

Το RabbitMQ είναι ένα λογισμικό ανοιχτού κώδικα, το οποίο υλοποιεί το Πρωτόκολλο Queuing AMQP, που σχεδιάστηκε για να συνδέει τους διακομιστές μεταξύ τους (S2S). Οι βιβλιοθήκες για τη διεπαφή του πελάτη είναι διαθέσιμες για όλες τις κύριες γλώσσες προγραμματισμού. Προσφέρει μια σειρά από λειτουργίες που επιτρέπουν αξιόπιστη επικοινωνία, είναι εύκολο στη χρήση, διατίθεται σε όλα τα μεγάλα λειτουργικά συστήματα, υποστηρίζει έναν τεράστιο αριθμό πλατφορμών και υποστηρίζει την ανταλλαγή μηνυμάτων σε μια ποικιλία από πρωτόκολλα.

2.6 Narrow Band IoT

2.6.1 Εισαγωγή

Για να αντιμετωπίσει τις προκλήσεις του Διαδικτύου των πραγμάτων (IoT), ο οργανισμός προτυποποίησης τεχνολογιών ασύρματης επικοινωνίας 3GPP (3rd Generation Partnership Project) ανέπτυξε το Narrowband IoT (NB-IoT – μτφ. Επικοινωνία στενής ζώνης) ως μέρος της Release 13 (Ιούνιος 2016). Το NB-IoT έχει σχεδιαστεί για να προσφέρει βελτιωμένη εσωτερική κάλυψη, υποστήριξη ενός τεράστιου αριθμού συσκευών με χαλαρές απαιτήσεις καθυστέρησης και χαμηλή κατανάλωση ενέργειας. Το NB-IoT χρησιμοποιεί τις ήδη υφιστάμενες υποδομές των δικτύων LTE (Long Term Evolution) με απλοποιήσεις και βελτιστοποιήσεις.

Στους διακηρυγμένους στόχους της τεχνολογίας NB-IoT είναι :

- ***Βελτιωμένη εσωτερική κάλυψη:***

Ο στόχος είναι να επιτευχθεί εκτεταμένη κάλυψη 20 dB σε σύγκριση με τις παλιότερες συσκευές που βασίζονται στο GPRS. Αυτό αντιστοιχεί στην επίτευξη του μέγιστου στόχου απώλειας συζεύξεως (maximum coupling loss - MCL) 164 dB. Για αυτό το MCL, θα πρέπει να υποστηρίζετε ρυθμός δεδομένων τουλάχιστον 160 bps στο επίπεδο εφαρμογής (application layer) τόσο για την uplink όσο και για το downlink.

- ***Υποστήριξη μεγάλου αριθμού συσκευών χαμηλής ταχύτητας διαμεταγωγής (throughput):***

Ο στόχος είναι να υποστηρίξει τουλάχιστον 52547 συσκευές ανά cell/τομέα. Αυτός ο στόχος καθορίστηκε χρησιμοποιώντας 40 συσκευές ανά νοικοκυριό, με την πυκνότητα νοικοκυριού να βασίζεται σε υπόθεση εργασίας για το Λονδίνο

δηλαδή 1517 νοικοκυριά ανά τετραγωνικό χιλιόμετρο και απόσταση μεταξύ των τηλεπικοινωνιακών ιστών 1732 μ. .

- ***Μειωμένη πολυπλοκότητα:***

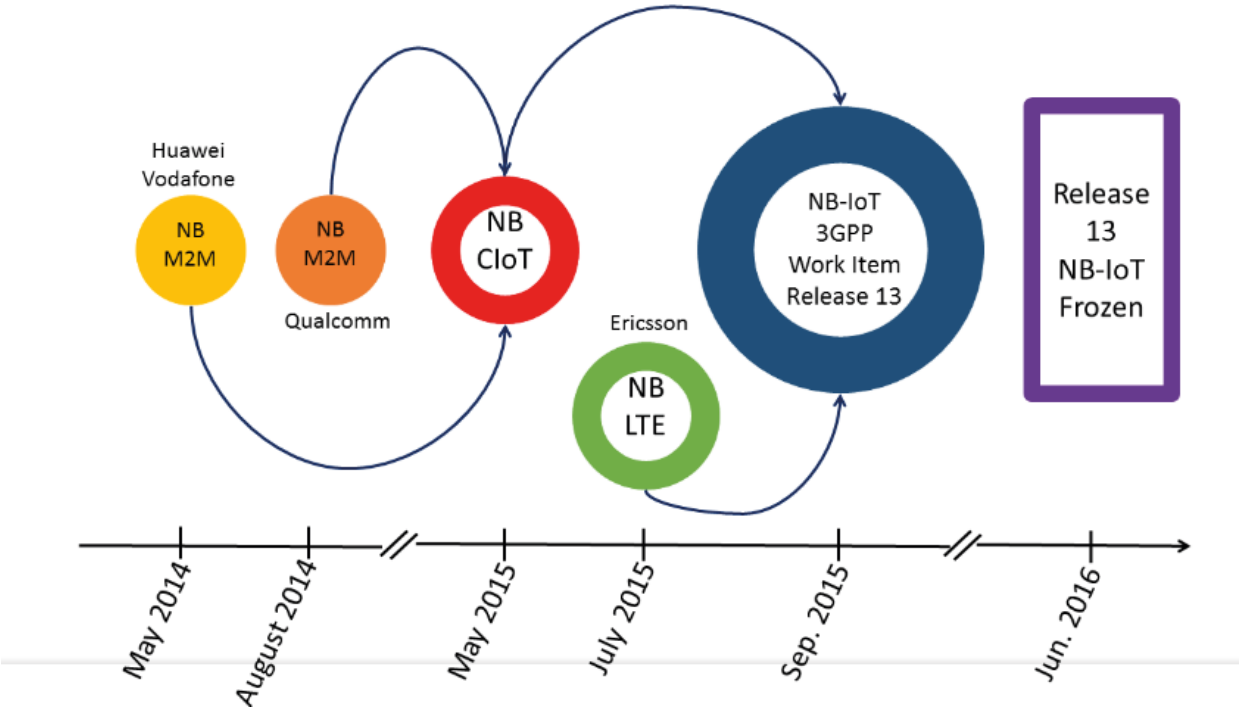
Ο στόχος είναι να μπορεί να υποστηριχθεί από συσκευές εξαιρετικά χαμηλής πολυπλοκότητας, με σκοπό την περαιτέρω ανάπτυξη και υποστήριξη IoT εφαρμογών.

- ***Βελτιωμένη απόδοση ισχύος:***

Ο στόχος είναι να έχουμε διάρκεια ζωής μπαταρίας δέκα ετών, με την μπαταρία αναφοράς να έχει χωρητικότητα 5 Wh.

- ***Καθυστέρηση (latency) :***

Η καθυστέρηση αναφοράς εξαιρέσεων να είναι στα 10 δευτερόλεπτα ή λιγότερο για το 99% των συσκευών.



εικόνα 2.8: Διαδικασία προτυποποίησης Narrow Band IoT

2.6.2 Εξέλιξη του NB-IoT

Τον Μάιο του 2014, η Huawei και η Vodafone πρότειναν το αντικείμενο έρευνας NB-M2M στο 3GPP GERAN (GSM/EDGE Radio Access Network) , το οποίο γρήγορα συγκέντρωσε την υποστήριξη και την αυξημένη προσοχή από άλλους σημαντικούς παρόχους κινητής τηλεφωνίας. Τον Οκτώβριο του ίδιου έτους, η Qualcomm υπέβαλε μια νέα πρόταση για ένα πρότυπο επικοινωνίας στενής ζώνης (narrow band) για συσκευές IoT με την ονομασία NB-OFDM. Τον Μάιο του 2015, οι δύο τεχνολογίες συγχωνεύτηκαν στο NB-CIoT (Narrow Band Cellular IoT). Εν τω μεταξύ, η Ericcson επιτάχυνε την έρευνά της για το Narrow Band IoT και πρότεινε το πρότυπο NB-LTE (Narrowband LTE) τον Αύγουστο του 2015. Το Σεπτέμβριο του 2015, το 3GPP αποδέχθηκε την ενσωμάτωση των τεχνολογιών στην Έκδοση 13 (Release 13). Το NB-CIoT είναι μια εκ του μηδενός προσέγγιση που προωθείται από τη Huawei.

3GPP Release	Features
Rel-13	New physical layer channels and signals for 180 kHz RF bandwidth, half-duplex only, 3 modes of operation (stand-alone, in-band, guard-band), single-tone and multi-tone transmission on uplink, RRC connection suspend/resume, data transmission via control plane, extended DRX, only idle-mode mobility
Rel-14	Positioning enhancements (E-CID requirements and OTDOA support), multicast support using SC-PTM, support higher data rates, allow paging and random access procedures on non-anchor carriers, new UE power class (14 dBm)
Rel-15	Reduced latency and power consumption, measurement accuracy improvement, random access reliability and range enhancements, small cell support, support for TDD, access barring enhancement, eDRX enhancements

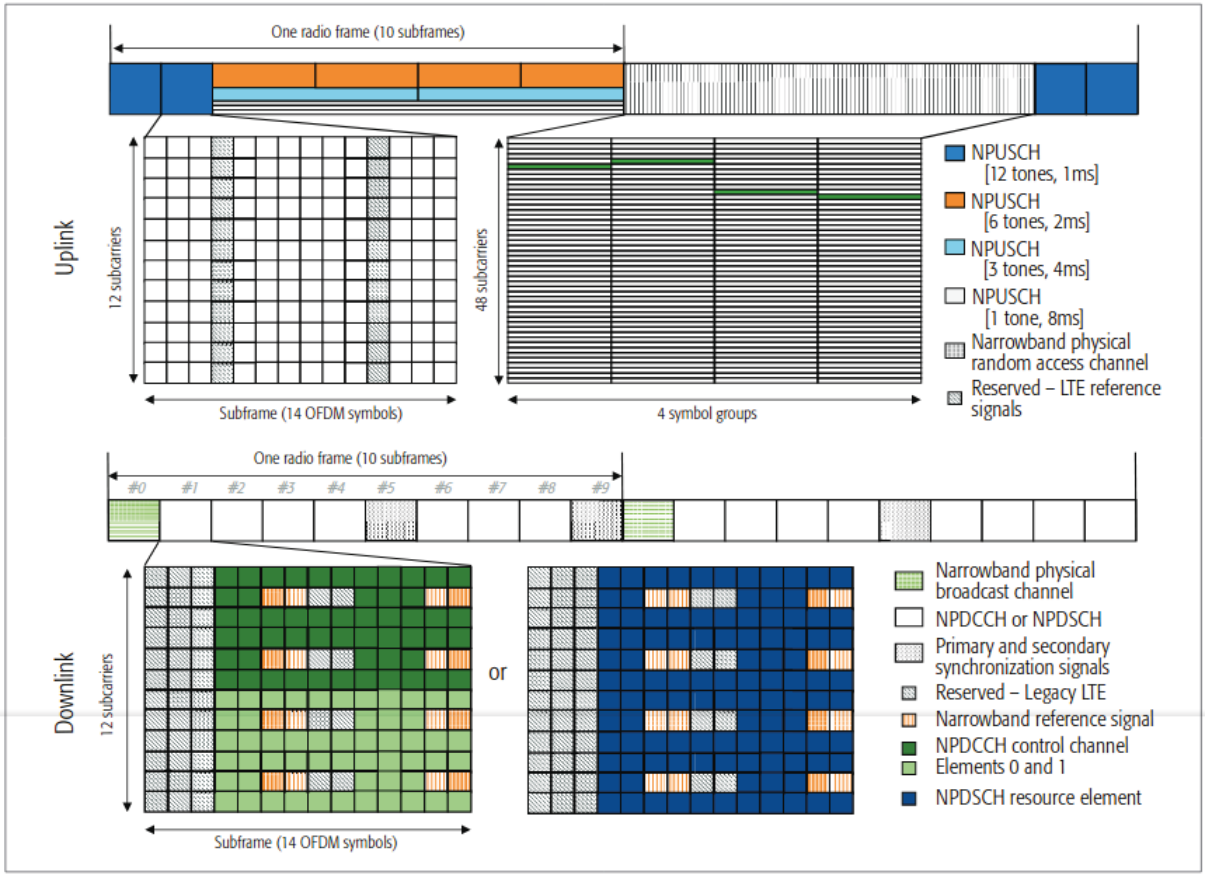
εικόνα 2.9: Εξέλιξη Narrow Band IoT για τα 3GPP Releases 13-15

Η διαφορά μεταξύ των NB-LTE και NB-CIoT έγκειται στο κατά πόσο μεγάλο μέρος των υφιστάμενων δικτύων LTE μπορεί να επαναχρησιμοποιηθεί. Το NB-CIoT απαιτεί νέα ολοκληρωμένα (chip-sets) και δεν φαίνεται να είναι συμβατό με οποιοδήποτε LTE δίκτυο παλαιότερο από το Release 13. Αντίθετα, η NB-LTE τεχνολογία μπορεί να ενσωματωθεί πλήρως στα υπάρχοντα δίκτυα LTE, λειτουργεί εντός των σημερινών ζωνών LTE και δεν χρειάζεται κάποιο ξεχωριστό δίκτυο. Τον Νοέμβριο του 2015, η 3GPP συμφώνησε ότι αυτές οι δύο πρωτοβουλίες εξελίσσονται σε ένα και μόνο πρότυπο με το όνομα Narrowband IoT (NB-IoT). Το NB-IoT καθορίζει ένα νέο τρόπο πρόσβασης για τις IoT συσκευές. Η NB-IoT είναι πλέον η κύρια πρόταση του οργανισμού 3GPP για τα Δίκτυα ευρείας κάλυψης και χαμηλής ισχύος (Low Power Wide Area Network - LPWAN) . Τον Ιούνιο του 2016 η 3GPP με το Release 13 ολοκλήρωσε την τυποποίηση της NB-IoT τεχνολογίας. Οι NB-IoT υποδομές (UE - user equipment) αναφέρονται ως Cat-NB1 (επίσης γνωστό ως Cat-M2). Η μείωση της πολυπλοκότητας τους, σε σύγκριση με την κατηγορία 1, είναι έως και 90%. Περαιτέρω βελτιώσεις του NB-IoT σε τομείς όπως ο ρυθμός μετάδοσης δεδομένων, ο γεωεντοπισμός, ο ρυθμός μετάδοσης δεδομένων, η καθυστέρηση (latency), η ενεργειακή κατανάλωση κ.α. πραγματοποιούνται στα 3GPP Releases 14 (Ιούνιος 2017) και 15 (Ιούνιος 2018). [10][11]

2.6.3 Σχεδίαση ραδιοσυχνοτήτων (radio design) του NB-IoT

Ο νέος σχεδιασμός διεπαφής ραδιοσυχνοτήτων NB-IoT προέρχεται από το παλιότερο πρότυπο LTE. Το φέρον σήμα του NB-IoT διαθέτει εύρος ζώνης 180 kHz με υποστήριξη για λειτουργία πολλαπλών φορέων (multi-carrier). Στην downlink εφαρμόζεται πρόσβαση πολλαπλής προσπέλασης συχνότητας (orthogonal frequency-di-vision multiple access - OFD-MA) με απόσταση 15kHz μεταξύ των 12 υπο-φορέων (subcarriers) με 14 σύμβολα που χρησιμοποιούνται για την κάλυψη ενός υποπλαισίου (sub-frame) 1 ms. Στο uplink, χρησιμοποιείται πολλαπλή πρόσβαση διαίρεσης

συχνότητας με ένα μόνο φορέα (single-carrier frequency-division multiple access - SC-FDMA), χρησιμοποιώντας απόσταση υποφορέων είτε 3.75 kHz, είτε 15 kHz. Τα φυσικά κανάλια και τα σήματα NB-IoT είναι κατ 'εξοχήν πολυπλεγμένα στο χρόνο. Η εικόνα 2.10 αποτυπώνει ένα παράδειγμα σχεδιασμού ενός υπο πλαισίου NB-IoT.



εικόνα 2.10: Παράδειγμα πλαισίου Narrow Band IoT

Το NB-IoT ορίζει τρεις τρόπους λειτουργίας για να προωθήσει την ευελιξία ανάπτυξης:

- ***Αυτόνομο (stand-alone):*** Γίνεται χρήση, έναν ή περισσότερων φορέων GSM αποκλειστικά για κίνηση NB-IoT .
- ***Εντός της ζώνη ασφαλείας του μεταφορέα LTE (guard-band) :*** Έχουμε χρήση μιας συχνότητας εύρους 180 Khz δεσμευμένη για IoT

κίνηση, που δεν χρησιμοποιείται για άλλες εμπορικές χρήσεις κινητής τηλεφωνίας (οι ζώνες συχνοτήτων guard-band χρησιμοποιούνται για τον διαχωρισμό και την αποτροπή παρεμβολών μεταξύ δύο μεγαλύτερων ομαδοποιήσεων συχνοτήτων) .

- ***Εντός ζώνης (in-band) :*** Ένα ή περισσότερα Μπλοκ Φυσικών Πόρων LTE (Physical Resource Blocks – PRBs) δεσμεύονται για NB-IoT κίνηση. Η συνολική ισχύς του eNB (δηλ. του σταθμού βάσης κατά την 3GPP ορολογία) διαμοιράζεται μεταξύ παραδοσιακής LTE κίνησης και NB-IoT με - τη δυνατότητα χρησιμοποίησης της φασματικής πυκνότητας ισχύος (power spectral density - PSD) για το NB-IoT. Η κοινή χρήση των PRBs μεταξύ NB-IoT και LTE επιτρέπει την αποτελεσματικότερη χρήση του φάσματος. Επιπλέον, αν και είναι δύο χωριστά συστήματα, μπορούν να υποστηριχθούν χρησιμοποιώντας το ίδιο εξοπλισμός σταθμού βάσης (eNB) [12][13]

2.6.4 Ενεργειακή αποδοτικότητα

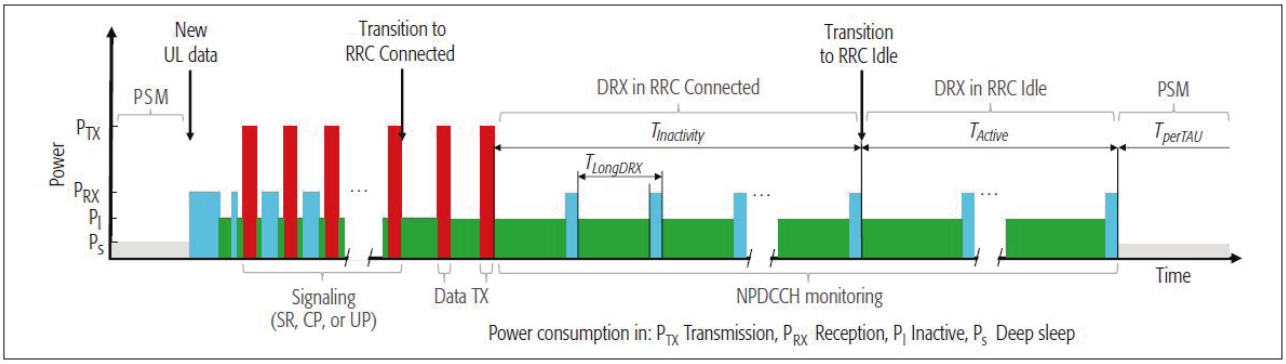
Το NB-IoT έχει σχεδιαστεί για συσκευές μακράς διάρκειας ζωής και στοχεύει σε διάρκεια ζωής μπαταρίας άνω των 10 ετών. Για το σκοπό αυτό, η NB-IoT επαναχρησιμοποιεί τους μηχανισμούς εξοικονόμησης ενέργειας του LTE, αλλά επεκτείνοντας τους χρονομετρητές (timers) για την επίτευξη μεγαλύτερης διάρκειας ζωής της μπαταρίας. Στο LTE, υπάρχουν δύο βασικοί μηχανισμοί εξοικονόμησης ενέργειας: Ασυνεχής Υποδοχή (Discontinuous Reception -DRX) και Λειτουργία Εξοικονόμησης Ενέργειας (Power Saving Mode – PSM). Και οι δύο μηχανισμοί τροποποιούν τον τρόπο με τον οποίο ο Τελικος Εξοπλισμός Χρήστη/ Τελική Συσκευή (User Equipment -UE) επικοινωνεί με το δίκτυο. Αυτή η επικοινωνία απαιτεί μια Σύνδεση Τηλεπικοινωνιακού Δικτύου (Radio Resource Connection – RRC) μεταξύ του εξοπλισμού του χρήστη και του Σταθμού Βάσης (eNB).

Υπάρχουν δύο πιθανές καταστάσεις σύνδεσης RRC: συνδεδεμένη (connected) και αδρανής (idle). Μια τελική συσκευή (UE) σε συνδεδεμένη κατάσταση (connected state) RRC έχει μια ενεργή σύνδεση RRC. Ως εκ τούτου, ο σταθμός βάσης μπορεί να προχωρήσει σε άμεση διάθεση πόρων στον τελικό χρήστη. Σε διαφορετική περίπτωση, το UE είναι σε RRC κατάσταση αναμονής (idle state). Ο τελικός χρήστης μπορεί να διέλθει από κατάσταση συνδεδεμένου RRC σε κατάσταση RRC αναμονής λόγω διαφόρων αιτιών όπως η αδράνεια (inactivity) ή η αποσύνδεση (detach) του τελικού χρήστη. Για την ανίχνευση της αδράνειας του UE, ο σταθμός βάσης χρησιμοποιεί ένα μετρητή χρόνου αδράνειας (inactivity timer) που μηδενίζει σαν μεταβλητή από την μεταφορά πακέτων δεδομένων.

Η εικόνα 2.11 απεικονίζει ένα παράδειγμα των μεταβάσεων της καταναλώσεως ισχύος που συμβαίνουν στην τελική συσκευή/χρήστη καθώς χρησιμοποιεί και τους δύο μηχανισμούς εξοικονόμησης ενέργειας. Περιγράφονται εν συντομία παρακάτω:

- ***DRX:***

Επιτρέπει στην τελική συσκευή να λαμβάνει διαδοχικά στο φυσικό κανάλι ελέγχου κατερχόμενης ζεύξης (physical downlink control channel -PDCCH). Το DRX ρυθμίζεται μέσω κύκλων DRX. Σε κάθε DRX κύκλο υπάρχουν δύο φάσεις. Αρχικά, η τελική συσκευή παρακολουθεί



εικόνα 2.11: Μετάβαση κατανάλωσης μεταξύ DRX (connected-idle) - PSM

το PDCCH για μια σύντομη περίοδο. Στην συνέχεια, το UE σταματά την παρακολούθηση του PDCCH για μεγάλο χρονικό διάστημα. Το DRX εξοικονομεί την μπαταρία του UE, αλλά ταυτοχρόνως επιτρέπει στο δίκτυο να φτάνει στο UE μέσω μηνυμάτων ειδοποίησης (paging) ή καναλιών ελέγχου κατερχόμενης (downlink) ζεύξης. Υπάρχουν βραχείς και πλατιοί τύποι κύκλων DRX. Στο LTE, μια τελική συσκευή μπορεί να χρησιμοποιήσει και τους δύο τύπους. Ωστόσο, στην NB-IoT, ένα UE χρησιμοποιεί μόνο μεγάλους κύκλους DRX.

- ***PSM:***

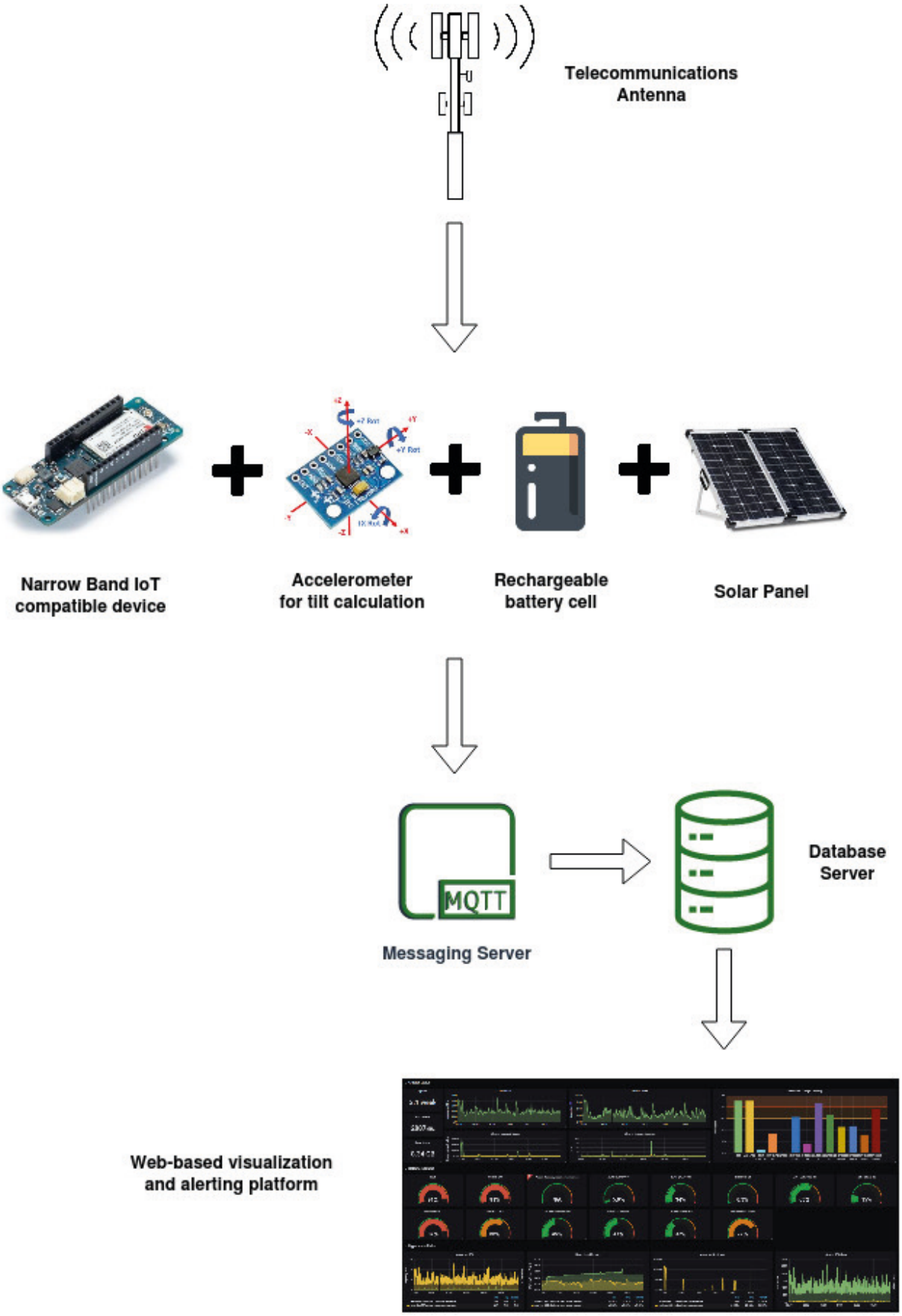
Επιτρέπει στο UE που βρίσκεται σε κατάσταση αναμονής RRC να εισέλθει σε βαθύ ύπνο (deepsleep) . Σε deepsleep, το UE είναι μη προσβάσιμο από το δίκτυο, αλλά εξακολουθεί να είναι καταχωρημένη. Η τελική συσκευή σε κατάσταση PSM παραμένει σε βαθύ ύπνο έως ότου η πλευρά της απαιτήσει την επανέναρξη επικοινωνίας με το δίκτυο. Ένα παράδειγμα είναι η περιοδική διαδικασία Ενημέρωσης Περιοχής Παρακολούθησης (Tracking Area Update – TAU) που ενεργοποιείται από τη λήξη ενός σχετικού χρονιστή ή η μετάδοση δεδομένων ανερχόμενης (uplink) κίνησης. Πριν από την είσοδό του σε κατάσταση PSM, το UE πρέπει να είναι προσβάσιμη από το δίκτυο για ένα χρονικό διάστημα. Κατά τη διάρκεια αυτής της περιόδου, το UE μπορεί χρησιμοποιήσει το DRX για πιθανή κίνηση downlink ενώ παράλληλα θα έχει εξοικονόμηση ενέργειας.

Το NB-IoT, υποστηρίζει ένα μεγάλο κύκλο DRX των 10,24 δευτερολέπτων στην κατάσταση συνδεδεμένου RRC . Στο αδρανές RRC, ο μέγιστος υποστηριζόμενος κύκλος DRX είναι 2,91 ώρες. Για το PSM, ο μέγιστος χρόνος deepsleep είναι 310 ώρες. Η επέκταση και των δύο μηχανισμών συνεπάγεται μεγαλύτερη καθυστέρηση (latency) καθώς το δίκτυο θα περιμένει μια μακρύτερη περίοδο έως ότου είναι σε θέση να φτάσει το UE. Ωστόσο, επιτυγχάνει την μείωση της κατανάλωσης ενέργειάς. [14][15][16]

Κεφάλαιο 3
Αρχιτεκτονική

3.1 Εισαγωγή

Στο κεφάλαιο αυτό θα πραγματοποιηθεί η αναλυτική περιγραφή των υλικών και άυλων στοιχείων που αποτελούν την λύση μας. Μέσω της παράλληλης περιγραφής των απαιτήσεων πάνω στις οποίες ήρθαν να δώσουν λύση, θα γίνει κατανοητή η επιλογή τους έναντι άλλων διαθέσιμων επιλογών. Στα πλαίσια της διπλωματικής κατασκευάστηκαν δυο διαφορετικές τελικές IoT συσκευές που ενσωματώνοντας ολοκληρωμένο κύκλωμα σύνδεσης στο Narrow Band IoT δίκτυο, κεραία, αισθητήρα καταγραφής της επιτάχυνσης/κλίσης, εξωτερική μπαταρία και φωτοβολταϊκό πάνελ καταγράφουν την κλίση τηλεπικοινωνιακών κεραιών και την προωθούν κατάλληλα μορφοποιημένη στο επόμενο επίπεδο. Η επιλογή της κατασκευής δυο διαφορετικών τελικών συσκευών, έγινε επιδιώκοντας την δυνατότητα πειραματικής σύγκρισης των δυνατοτήτων τους και των ορίων τους, ένα ζήτημα κομβικό καθώς αναφερόμαστε σε συσκευές σχετικά καινούργιες, ελάχιστα δοκιμασμένες. Στο επόμενο στάδιο εγκαταστάθηκε στην cloud υποδομή του εργαστηρίου ένας εξυπηρετητής MQTT που ανέλαβε το ρόλο του ενδιάμεσου ανάμεσα στα δεδομένα που αποστέλλονται από τις τελικές συσκευές και την βάση δεδομένων. Η βάση δεδομένων είναι υπεύθυνη για την αποδοτική, ασφαλή και εύκολα προσβάσιμη αποθήκευση των δεδομένων. Τέλος με την χρήση μιας ιδιαίτερα αποδοτικής αρχιτεκτονικής ο εξυπηρετητής της βάσης δεδομένων, ενσωματώνει και υποστηρίζει την διαδικτυακή πλατφόρμα οπτικοποίησης των δεδομένων. Η αποτύπωση της λύσης μας σε υψηλό επίπεδο φαίνεται στο παρακάτω σχήμα.



εικόνα 3.1: Αποτύπωση της λύσης σε υψηλό επίπεδο

3.2 Επιλογή πλατφόρμας IoT

Για την επιλογή της συσκευής που θα αποτελέσει την βάση της υλοποίησης μας, καθοριστικοί είναι οι ακόλουθοι παράγοντες.

- Η ύπαρξη ενσωματωμένου Modem σύνδεσης στο Narrow Band IoT δίκτυο
- Δυνατότητα εξοικονόμησης ενέργειας μέσω λειτουργίας deepsleep
- Δυνατότητα σύνδεσης ενεργειακού συσσωρευτή
- Η ύπαρξη ενσωματωμένου κυκλώματος επαναφόρτισης του ενεργειακού συσσωρευτή
- Δυνατότητα σύνδεσης φωτοβολταϊκού πάνελ
- Ύπαρξη ανεπτυγμένων βιβλιοθηκών διαμεσολάβησης και λειτουργίας των αισθητήρων επιτάχυνσης
- Σχετικά μικρό μέγεθος
- Η ευκολία προγραμματισμού με την χρήση γλώσσας προγραμματισμού υψηλού επιπέδου

Όλα τα παραπάνω είναι η αλήθεια πως περιορίζουν και υπαγορεύουν σε μεγάλο βαθμό τις επιλογές μας. Καθοριστικός παράγοντας είναι η μη ωριμότητα της τεχνολογίας NB-IoT, τουλάχιστον σε εμπορικό επίπεδο. Έτσι αυτή την περίοδο μπορούμε να βρούμε στην αγορά συσκευές που στοχεύουν κυρίως στην προτυποποίηση (prototyping), στις οποίες ο προγραμματισμός γίνεται σε χαμηλό επίπεδο. Χαρακτηριστικά η μόνη αλληλεπίδραση με το NB-IoT modem γίνεται μέσω AT commands χωρίς την χρήση κάποιας βιβλιοθήκης ενώ προφανώς δεν διαθέτουν δυνατότητες deepsleep ενεργειακής εξοικονόμησης ούτε δυνατότητες τροφοδοσίας μπαταριών. Χαρακτηριστικές τέτοιες συσκευές είναι οι SIMOCOM-EVB_V1.02 και ExpLoRer

Συσκευές οι οποίες σε πρώτη φάση και με βάση τα ονομαστικά χαρακτηριστικά τους καλύπτουν τις προδιαγραφές που θέσαμε είναι το

MKR1500 της Arduino, το Fipy της Pycom και εν μέρει το RAK8211. Επιλέγουμε τις δύο πρώτες στην βάση των ακόλουθων πλεονεκτημάτων τους και στην βάση κάποιων χαρακτηριστικών μειονεκτημάτων της τρίτης λύσης όπως καταγράφουμε παρακάτω. [16][17]

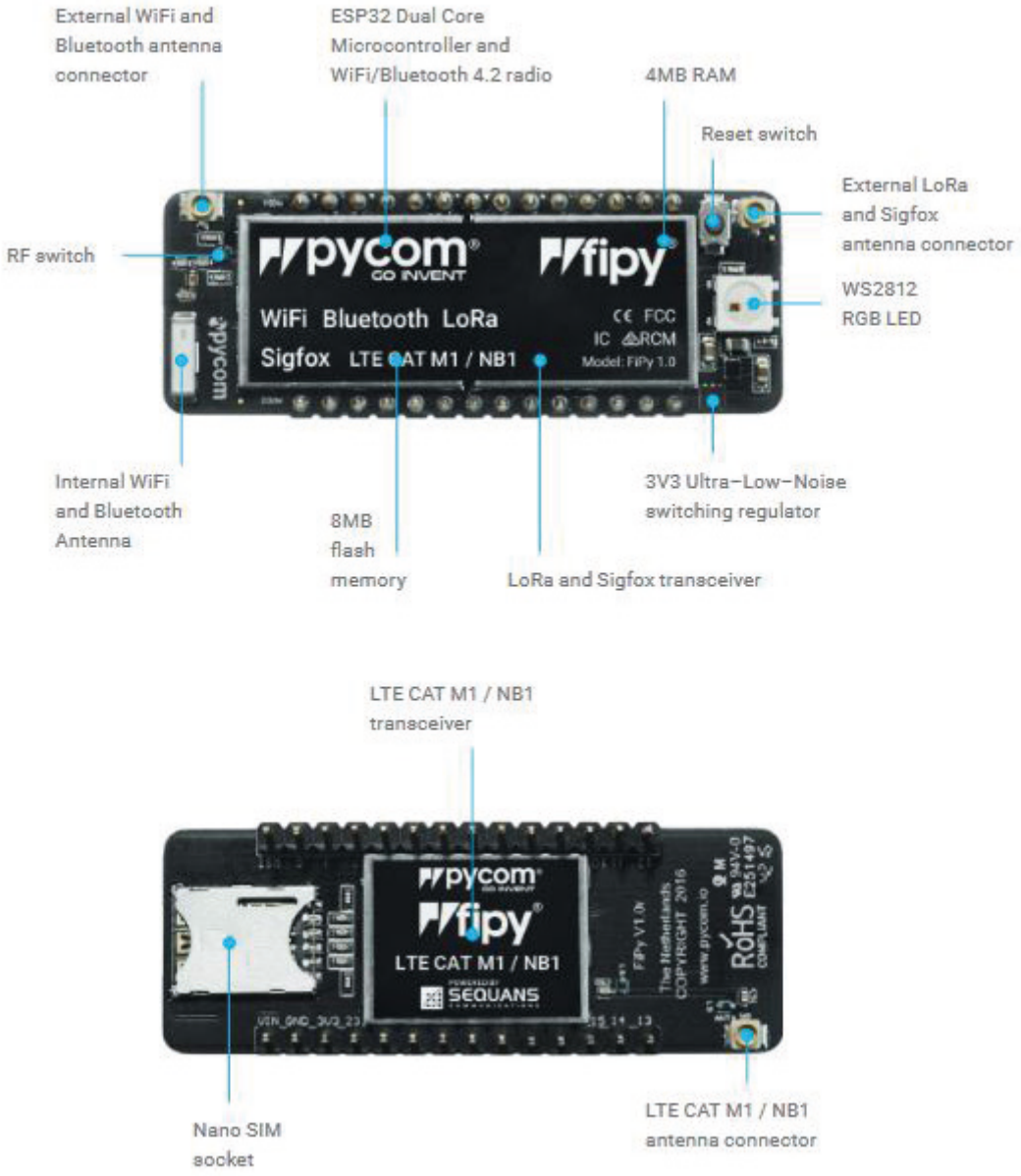
3.2.1 Πλεονεκτήματα Arduino MKR1500

εικόνα 3.2: Arduino MKR1500

- Ευκολία προγραμματισμού και διαχείρισης της συσκευής με την χρήση του προγράμματος Arduino IDE.
- Ύπαρξη μιας μεγάλης και ιδιαίτερα βοηθητικής κοινότητας χρηστών, που επί σειρά ετών χρησιμοποιούν τις συσκευές της Arduino δίνοντας συλλογική λύση σε μια σειρά τεχνικών ζητημάτων.
- Γλώσσα προγραμματισμού βασισμένη στην C με το ανάλογο διαθέσιμο documentation
- Ύπαρξη μιας τεράστιας βάσης βιβλιοθηκών που καλύπτουν σχεδόν κάθε πιθανό περιφερειακό αισθητήρα όπως τους αισθητήρες επιτάχυνσης που μας ενδιαφέρουν.

3.2.2 Πλεονεκτήματα Pycom Fipy

- Προγραμματισμός με την χρήση Micro Python που μας δίνει απλότητα στον προγραμματισμό και πολλές διαθέσιμες βιβλιοθήκες.
- Ευκολία προγραμματισμού και διαχείρισης της συσκευής με την χρήση GUI ή CLI εργαλείων (Atom, Rshell κ.α.)
- Ύπαρξη έτοιμων λύσεων με "πακέτα" αισθητήρων από την εταιρία Fipy (Pysense, Pytrack κ.α.)



εικόνα 3.3: Pycom Fipy

3.2.3 Μειονεκτήματα RAK8211

Προγραμματισμός με την χρήση Javascript, μια σχετικά καινούργια επιλογή για IoT συσκευές, χωρίς μεγάλη προγραμματιστική κοινότητα και εκτενές support που θα μπορούσε να βοηθήσει στην επίλυση προβλημάτων.

Έλλειψη μιας μεγάλης βάσης βιβλιοθηκών για αισθητήρες, για σύνδεση με το LTE δίκτυο και για διαδικασίες deepsleep.

Έλλειψη USB port.

Δυσκολία προγραμματισμού με την πλατφόρμα Espuino, που δεν είναι τόσο "ώριμη" όσο άλλες αντίστοιχες λύσεις.

εικόνα 3.4: RAK8211 iTracker

Χαρακτηριστικά		
	Fipy	**MKR1500**
Επεξεργαστής	ESP32	M0 32-bit SAMD21
Κύκλωμα δικτύου	Sequans Monarch SQN3330	SARA-R410M-02B
Γλώσσα προγραμματισμού	MicroPython	C/C++
Τάση λειτουργίας	3,3 V	3,3 V
Deepsleep Consumption (ονομαστικό)	~1-2 mA	~1,5 mA
Max Power Consumption (LTE NB1)	330 mA	140 mA
GPIOs	22	22
UART		
I2C		
SPI		
JST PHR2 battery connector		
USB input		
RGB onboard LED		X
Other supported networks	Wifi, BLE, LoRa, Sigfox, Cat M1	Cat M1

εικόνα 3.5: Χαρακτηριστικά MKP1500 και Fipy

3.3 Καταγραφή κλίσης με αισθητήρα επιτάχυνσης

Οι αισθητήρες επιτάχυνσης είναι ευαίσθητοι τόσο ως προς την γραμμική επιτάχυνση, όσο και ως προς το τοπικό βαρυτικό πεδίο. Το δεύτερο με την χρήση των κατάλληλων αλγορίθμων και προγραμματιστικών βιβλιοθηκών μας βοηθάει στον υπολογισμό της κλίσης.

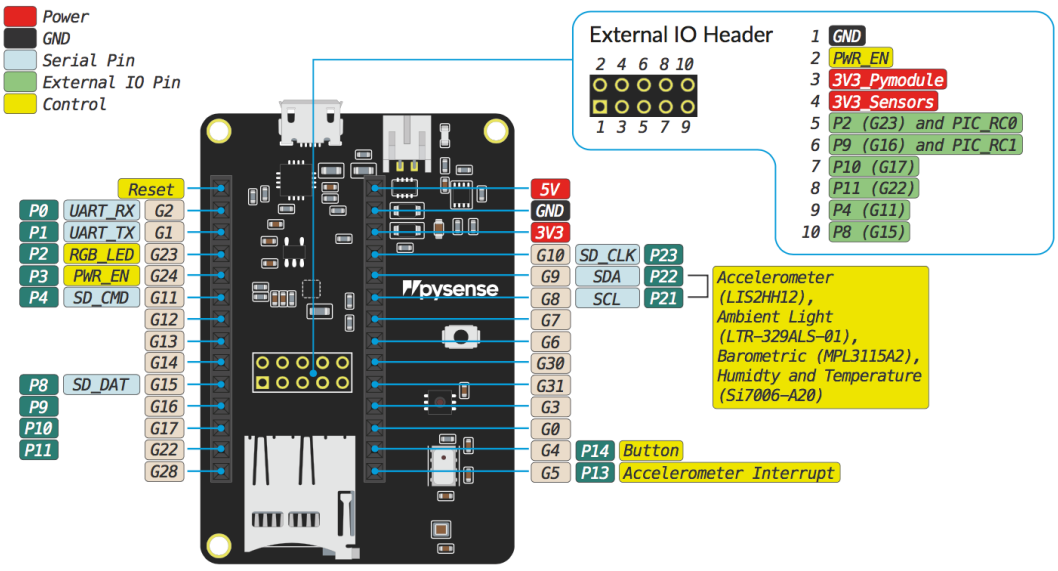
Για τους αισθητήρες επιτάχυνσης των δύο τελικών IoT συσκευών, απαιτούμε τα ακόλουθα χαρακτηριστικά.

- Μικρή κατανάλωση ενέργειας
- Ύπαρξη βιβλιοθηκών αλληλεπίδρασης και υπολογισμού κλίσης
- Ακρίβεια μετρήσεων (εξαρτάται σε μεγάλο βαθμό από τις κατάλληλες βιβλιοθήκες)

3.3.1 Pysense (LIS2HH12) - Αισθητήρας για το Fipy

Για το Fipy η αυτονόητη λύση είναι το ολοκληρωμένο πακέτο αισθητήρων Pysence. Η εν λόγω λύση είναι ανεπτυγμένη από την Pycom, εταιρία που αναπτύσσει και το Fipy παρέχοντας βιβλιοθήκες με απόλυτη συμβατότητα. Παράλληλα η λύση είναι ενσωματωμένη στην ευρύτερη αρχιτεκτονική του Fipy για μικρές καταναλώσεις σε κατάσταση deepsleep με την ονομαστική κατανάλωση για ένα Fipy μαζί με την πλακέτα Pysense να βρίσκεται στα 2,5 mA. Ο αισθητήρας επιτάχυνσης που περιλαμβάνεται στο Pysense είναι ο LIS2HH12 και με τις ανάλογες βιβλιοθήκες και τις κατάλληλη διαδικασία πολλαπλών μετρήσεων και εξαγωγής μέσου όρου μας δίνει μια ικανοποιητική ακρίβεια.

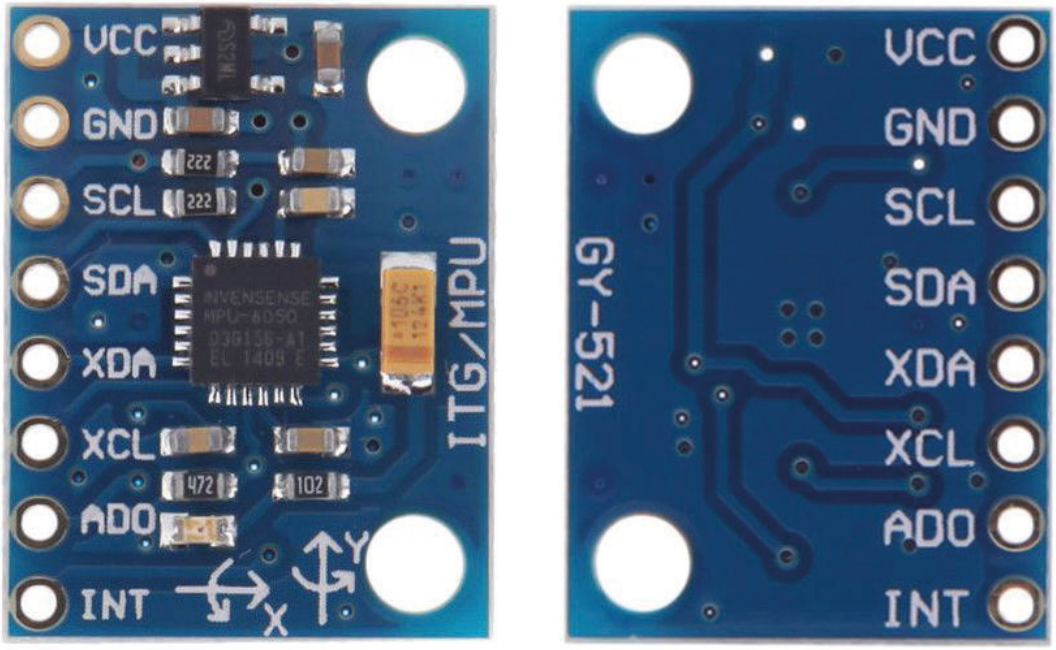
Παράλληλα το Pysense παρέχει ενσωματωμένο ελεγκτή φόρτισης μπαταρίας Li-ION/Li-Po και αισθητήρα μέτρησης θερμοκρασίας και υγρασίας. [18]



εικόνα 3.6: Pysense sensor bundle

3.3.2 MPU6050 - Αισθητήρας για το MKR1500

Για την περίπτωση του Arduino MKR1500 δεν υφίσταται ακόμη κάποια έτοιμη λύση στο επίπεδο των περιφερειακών αισθητήρων. Η λύση του MPU6050 μας καλύπτει στο κομμάτι της ενεργειακής κατανάλωσης (max active current 3.9 mA, nomimal deepsleep current < 0,1 mA). Προφανώς οι ελάχιστες καταναλώσεις δεν πλησιάζουν τις ονομαστικές καθώς για την διασύνδεση με το MKR1500 χρησιμοποιούμε breakout board όπως το GY-521. Παράλληλα με το MPU6050 έχουμε διάφορες ιδιαίτερα ανεπτυγμένες λύσεις βιβλιοθηκών.



εικόνα 3.7: πλακέτα GY-521 με τον αισθητήρα MPU6050

3.4 Συσσωρευτής ενέργειας

Και οι δύο λύσεις μας παρέχουν την δυνατότητα διασύνδεσης σε μπαταρίες τύπου Λιθίου Πολυμερών (Li-PO) και Λιθίου Ιόντων (Li-ION). Οι διαφορές μεταξύ τους καταγράφονται συνοπτικά στην εικόνα 3.8

εικόνα 3.9: σύγκριση των τεχνολογιών Li-Ion και Li-Po

Σαν λύση επιλέγουμε την χρήση μια μονής μπαταρίας 18650 χωρητικότητας 3325 mAh για τους παρακάτω λόγους :

- Μεγάλη Χωρητικότητα. Μια κοινή μπαταρία 18650 έχει χωρητικότητα ~3000 mAh.
- Κοινότυπος Σχεδιασμός. Οι εν λόγω μπαταρίες είναι το βάση σχεδίασης μπαταριών για σχεδόν το σύνολο των εμπορικών συσκευών μπαταριών πλην των Smartphones. Ενδεικτικά μπαταρίες 18650 χρησιμοποιούνται σε φορητούς υπολογιστές, οικιακούς συσσωρευτές για συστήματα ενεργειακής αυτονομίας, ηλεκτρικά οχήματα κ.λ.π.
- Μικρό Κόστος. Λόγω του μαζικής χρήσης και μαζικής παραγωγής της εν λόγω μπαταρίας το κόστος ανα milli AmperHour χωρητικότητας είναι χαμηλό σε σύγκριση με άλλες επαναφορτιζόμενες μπαταρίες.

- Δυνατότητα Εύκολης Αντικατάστασης. Αυτό μας διευκολύνει τόσο στην πιθανότητα φθοράς της μπαταρίας, ενώ μας δίνει την δυνατότητα αντικατάστασης με κάποια καινούργια μεγαλύτερης χωρητικότητας.

εικόνα 3.10: μπαταρία τύπου 18650

3.5 Φωτοβολταϊκό πάνελ

Το φωτοβολταϊκό πάνελ εξασφαλίζει την αυτονομία του συστήματός μας, παράλληλα με την χρήση μεθοδολογιών εξοικονόμησης ενέργειας. Λόγω της διαδικασίας deepsleep αλλά και των αραιών περιοδικών κύκλων ενεργοποίησης του αισθητήρα (μέγιστο 1-3 μετρήσεις ανά ημέρα), για την ενεργειακή αναπλήρωση και αυτονομία του αισθητήρα δεν απαιτείται ειδική διαστασιολόγηση του πάνελ πέρα από την απαίτηση για προσαρμογή του πάνω στην αδιάβροχη συσκευασία της τελικής συσκευής.

Μια άλλη απαίτηση είναι η παροχή της ενέργειας μέσω εξόδου USB, ώστε να γίνει χρήση των ελεγκτών φόρτισης που είναι ενσωματωμένοι στα MKR1500 και Fipy.

Τελικώς επιλέγουμε ένα φωτοβολταϊκό πάνελ OEM διαστάσεων 10x15 cm με έξοδο USB, ονομαστικής μέγιστης ισχύος 2,5 W (max 0,5 Amper @ 5V).

εικόνα 3.11: φωτοβολταϊκό πάνελ

3.6 Αδιάβροχη συσκευασία

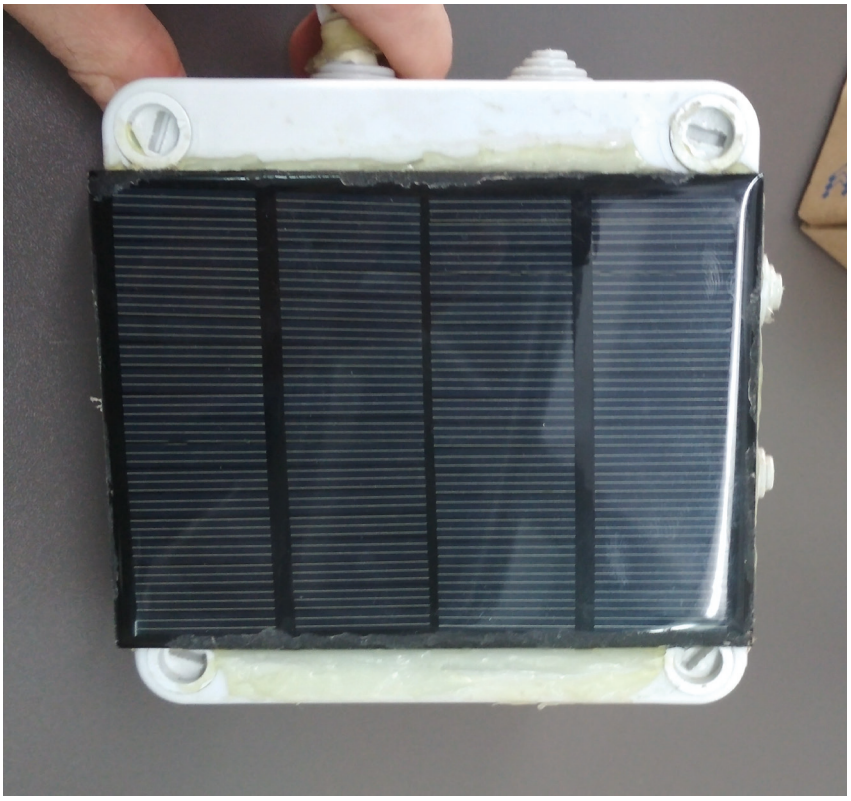
Για την συνολική συσκευασία των αισθητήρων και της μπαταρίας, επιλέγουμε ένα ηλεκτρολογικό κουτί εξωτερικού χώρου. Είναι η πιο άμεσα διαθέσιμη, οικονομική και πρακτική λύσης καθώς :

- Φέρει περιμετρικές φλάτζες στεγανοποίησης
- Κλίνει ερμητικά με σπείρωμα στις γωνίες
- Φέρει εισόδους για καλώδια, παραμετροποιήσιμου μεγέθους και με δυνατότητα στεγανοποίησης.

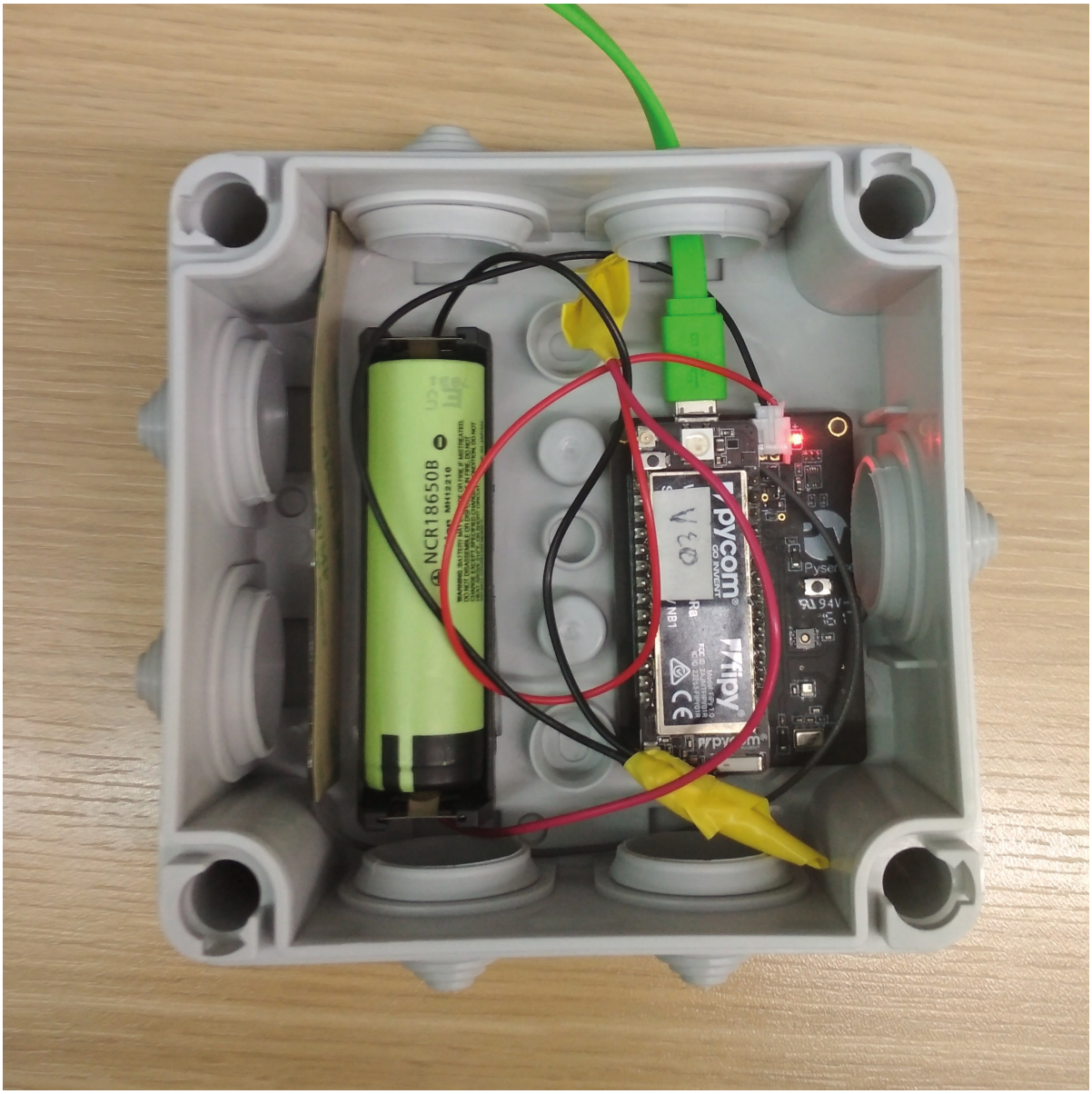
- Φέρει ευθεία επιφάνεια με δυνατότητα προσαρμογής φωτοβολταϊκού πάνελ
- Έχει έτοιμη πρόβλεψη, στην οπίσθια επιφάνεια, για προσαρμογή σε επιφάνειες (κεραίες, πάνελ κλπ)

εικόνα 3.12: τυπικό στεγανό ηλεκτρολογικό κουτί

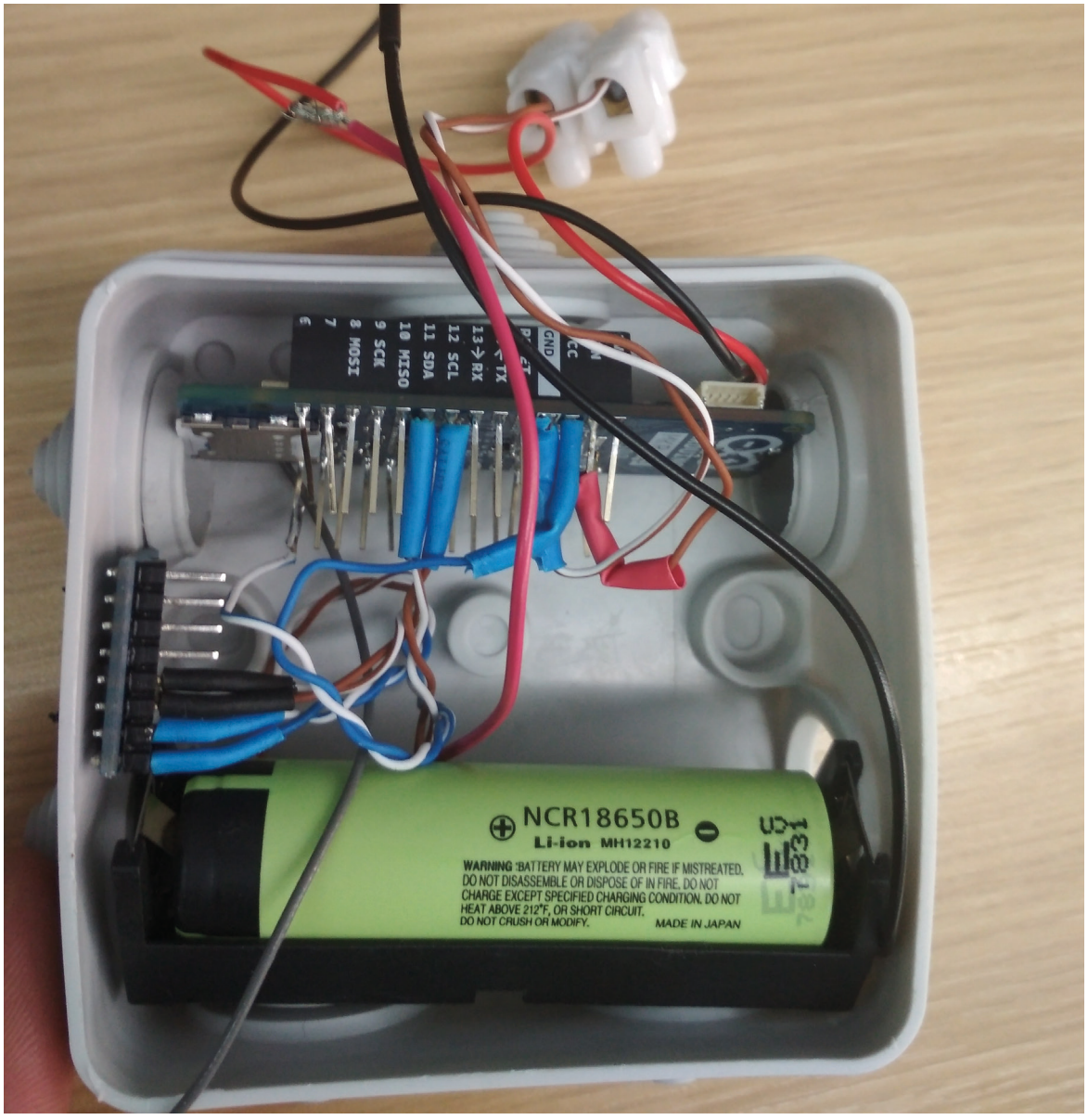
3.7 Συνολικός αισθητήρας



εικόνα 3.13: Η λύση με προσαρμοσμένο φωτοβολταϊκό πάνελ



εικόνα 3.14: Ο συνολικός αισθητήρας Fipy



εικόνα 3.15: Ο συνολικός αισθητήρας MKR1500

3.8 Η υποδομή υπολογιστικού νέφους (cloud computing)

Για την υποστήριξη της διπλωματικής σε επίπεδο υποδομών εξυπηρετητών, χρησιμοποιήθηκε η λύση ανοιχτού κώδικα Openstack.

Οι υποδομές cloud είναι εν πολύς το παρόν και το μέλλον των εξυπηρετητών. Μέσω της αρχιτεκτονικής Υποδομή σαν Υπηρεσία (Infrastructure as a Service – IaaS), μπορούμε να έχουμε βελτιστοποίηση της χρήσης πόρων, όπου με τους κατάλληλα διαστασιολογημένους φυσικούς εξυπηρετητές (bare metal) μπορούμε να δημιουργήσουμε πολλαπλούς εικονικούς εξυπηρετητές (virtual machines). Παράλληλα η μεταφορά των υποδομών από το φυσικό στο εικονικό επίπεδο του νέφους (cloud) μας προσφέρει αυξημένη ευελιξία στην αύξηση ή την μείωση πόρων βάση των πραγματικών αναγκών των εργασιών που θέλουμε να εκτελέσουμε.

Η λύση ανοιχτού λογισμικού Openstack βρίσκεται υπό ενεργό ανάπτυξη με εκδόσεις ανά 6μηνο . Παρέχει μεγάλες δυνατότητες λόγω της αρχιτεκτονικής αλλά και της κοινότητας και της υποστήριξης των συνεργατών της. Όλος ο κώδικας του Openstack είναι διαθέσιμος υπό την άδεια χρήσης Apache2. Ο αρχικός κώδικας του OpenStack αναπτύχθηκε από την NASA και την Rackspace Cloud με σκοπό την υποστήριξη υποδομών μεγάλης κλίμακας. Τα βασικά χαρακτηριστικά του OpenStack είναι :

- ***Επεκτασιμότητα (Scalable) :*** Η λύση χρησιμοποιείτε σε παγκόσμια κλίμακα, από εταιρίες των οποίων ο όγκος δεδομένων μετριέται σε petabytes, διανεμημένων σε αρχιτεκτονικές που περιλαμβάνουν μέχρι 1 εκατομμύριο φυσικές μηχανές, 60 εκατομμύρια εικονικά μηχανήματα και δισεκατομμύρια αποθηκευμένων αντικείμενων.
- ***Συμβατό και ευέλικτο (Compatible and Flexible) :*** Το OpenStack υποστηρίζει τις περισσότερες virtualization λύσεις της αγοράς: ESX, Hyper-V, KVM, LXC, QEMU, UML, Xen και XenServer.

- ***Ανοιχτό :*** Ολόκληρος ο κώδικας μπορεί να τροποποιηθεί και να προσαρμοστεί ανάλογα με τις ανάγκες μας. [19][20]

3.9 Εξυπηρετητής MQTT – Mosquitto

Ο εξυπηρετητής MQTT Mosquitto επιλέχθηκε γιατί είναι μια υλοποίηση ανοιχτού κώδικα, με πολύ χαμηλή κατανάλωση πόρων συστήματος, με μεγάλη κοινότητα χρηστών, πλατιά διαθέσιμες πληροφορίες παραμετροποίησης και επίλυσης προβλημάτων.

Το Mosquitto παρέχει έναν εξυπηρετητή που εναρμονίζεται με τα πρότυπα server / client του πρωτοκόλλου ανταλλαγής μηνυμάτων MQTT για τις εκδόσεις 5.0, 3.1.1, και 3.1 του πρωτοκόλλου. Το MQTT χρησιμοποιεί ένα μοντέλο δημοσίευσης / εγγραφής, έχει χαμηλό κόστος για το δίκτυο και μπορεί να εφαρμοστεί σε συσκευές χαμηλής ισχύος, όπως μικροελεγκτές που μπορούν να χρησιμοποιηθούν για την ανίχνευση αισθητήρων Internet of Things. Ως εκ τούτου, το Mosquitto προορίζεται για χρήση σε όλες τις περιπτώσεις όπου υπάρχει ανάγκη για ελαφριά μηνύματα, ειδικά σε περιορισμένες συσκευές με περιορισμένους πόρους. Η ανάπτυξη του Mosquitto πραγματοποιείται σαν κομμάτι του Eclipse Foundation.

Το Mosquitto περιλαμβάνει τρία μέρη:

- Τον κύριο εξυπηρετητή (server) mosquitto
- Τις διαδικασίες client mosquitto_pub και mosquitto_sub, βοηθητικά προγράμματα που φέρουν εις πέρας την επικοινωνία με τον εξυπηρετητή MQTT
- Μια βιβλιοθήκη πελάτη MQTT γραμμένη σε C

Για την εγκατάσταση του Mosquitto MQTT χρησιμοποιήθηκε ένα

Virtual Machine με διαθέσιμους πόρους 4 πυρήνων και 8 Gigabyte RAM. Το εν λόγω μηχάνημα παραμετροποιήθηκε με λειτουργικό ανοιχτού κώδικα Ubuntu 16.04 LTS. Η εγκατάσταση έγινε μέσω κονσόλας εντολών (CLI) με την παραμετροποίηση να βρίσκεται στην τοποθεσία /etc/mosquitto/mosquitto.conf . Τέλος χρειάστηκε να επιτρέψουμε στους κανόνες ασφαλείας (security groups) της υποδομής νέφους την πρόσβαση στις θύρες 1883 (για απλή / plaintext) και 31883 (για ασφαλή κρυπτογραφημένη κίνηση) από και προς τον εξυπηρετητή. [21]

3.10 Εξυπηρετητής Διεπαφής και Βάσης Δεδομένων

3.10.1 Στόχος

Ο εξυπηρετητής διεπαφής είναι ο μηχανισμός εκείνος που αναλαμβάνει να παραλάβει τα ακατέργαστα δεδομένα (raw data) από τον εξυπηρετητή μεταφοράς, εν προκειμένω του εξυπηρετητή Mosquitto MQTT και ύστερα από κατάλληλη πρώτη επεξεργασία να τα παραδώσει στον εξυπηρετητή Βάσης Δεδομένων.

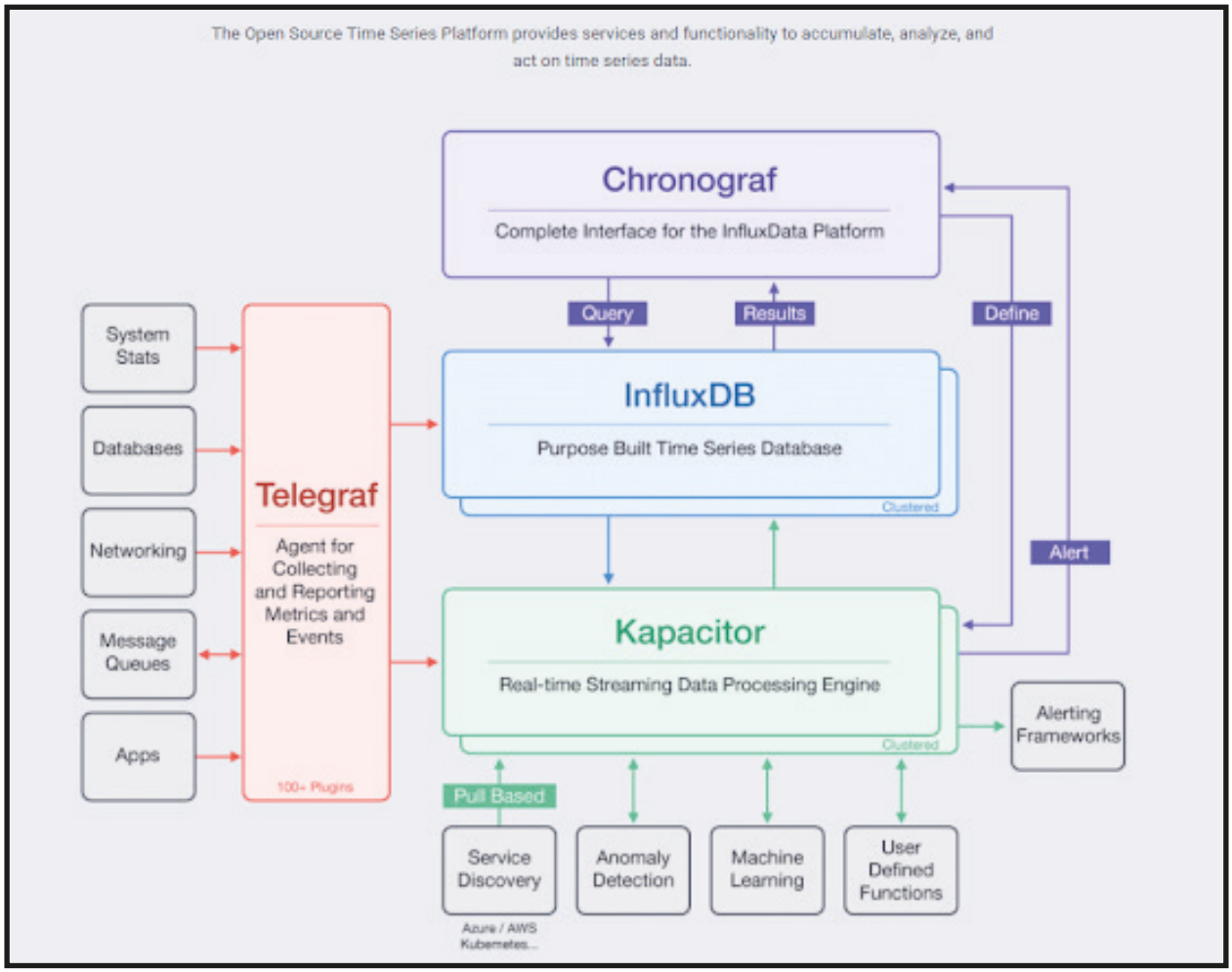
Η βάση δεδομένων οριστικοποιεί την επεξεργασία των δεδομένων, και τα αποθηκεύει με τρόπο που εξασφαλίζει την άμεση και οικονομική σε πόρους πρόσβαση.

3.10.2 Πλατφόρμα εξυπηρετητών TICK (Tick stack - Telegraf, InfluxDB, Chronograf and Kapacitor)

Την ενοποιημένη λύση έρχεται να την δώσει το TICK Stack.

Το TICK Stack είναι μια πλατφόρμα που αποτελείται από 4 προγράμματα ανοιχτού κώδικα, χαλαρά συζευγμένα μεταξύ τους αλλά με υψηλό βαθμό ολοκλήρωσης τα οποία καθιστούν εύκολη τη συλλογή, αποθήκευση, γραφική απεικόνιση και και τις ειδοποίηση πάνω σε χρονοσειρές δεδομένων. Τα συστατικά στοιχεία του TICK Stack

είναι: Telegraf, ένας μηχανισμός (agent) διακομιστή για τη συλλογή και αναφορά μετρήσεων. InfluxDB, μια βάση δεδομένων χρονοσειρών υψηλών επιδόσεων. Chronograf, μια διεπαφή χρήστη για την πλατφόρμα. και Kapacitor, ένας μηχανισμός επεξεργασίας δεδομένων που μπορεί να επεξεργαστεί, να μεταφέρει δεδομένα ροής από την InfluxDB.



εικόνα 3.16 : Σχεδιάγραμμα Tick Stack

Εμείς για την λύση μας θα χρησιμοποιήσουμε μόνο τα δύο πρώτα μέλη του TICK, δηλαδή τους εξυπηρετητές Telegraf και InfluxDB.

Η εγκατάσταση και η παραμετροποίηση και των δύο έγινε πάνω σε κοινό Virtual Machine, πάνω σε λειτουργικό Ubuntu 16.04 LTS με την χρήση κονσόλας εντολών. Για τις αυξημένες ανάγκες χώρου που

απαιτούνται από την βάση δεδομένων InfluxDB, παραμετροποιήθηκε και ενσωματώθηκε μέσω του Openstack υποδομή Cinder Volume.

3.10.3 Εξυπηρετητής διεπαφής – Telegraf

Το Telegraf είναι ένας εξυπηρετητής που με την χρήση πρόσθετων (plugins) φέρει εις πέρας τη συλλογή και την αναφορά μετρήσεων. Τα plugins του Telegraf μπορούν να εξάγουν ποικίλες μετρήσεις απευθείας από τα συστήματα όπου αυτές διακινούνται , είτε τραβώντας τες από διεπαφές αλληλεπίδρασης (API) με άλλες εφαρμογές ή ακόμα και παρακολουθώντας σε πραγματικό χρόνο τις μετρήσεις από υπηρεσίες όπως το StatsD και το Kafka. Διαθέτει επιπλέον plugins για την αποστολή μετρήσεων σε διάφορες άλλες βάσεις δεδομένων, υπηρεσίες και στοίβες μηνυμάτων, συμπεριλαμβανομένων των InfluxDB, Graphite, OpenTSDB, Datadog, Librato, Kafka, MQTT, NSQ και πολλών άλλων.

Στην περίπτωση μας με το κατάλληλο plugin, ο εξυπηρετητής Telegraf κάνει εγγραφή (subscribe) στο topic (θέμα) στο οποίο η τελική συσκευή μας δημοσιεύει (publish) δεδομένα. Ένα περαιτέρω βήμα είναι η αποδιαμόρφωση του μηνύματος που έχει αποσταλεί σε μορφή JSON (JavaScript Object Notation).

- ***JSON (JavaScript Object Notation)***: Το JSON είναι ένα "ελαφρύ" πρότυπο ανταλλαγής δεδομένων, κατανοητό στην σύνταξη και την ανάγνωσή του από άνθρωπο, και την ίδια στιγμή εύκολο για τις υπολογιστικές μηχανές να το δημιουργήσουν (generate) και να το αποκωδικοποιήσουν (parse)

Ο Telegraf με την δήλωση των κατάλληλων "tags" μπορεί να αποκωδικοποιήσει πληροφορία JSON, επιτρέποντας μας να απλοποιήσουμε σε τεράστιο βαθμό την αρχιτεκτονική του IoT οικοσυστήματος μας, καθώς

μπορούμε με την χρήση ενός και μόνο MQTT topic και την κατάλληλη JSON κωδικοποίηση να εξυπηρετήσουμε χιλιάδες συσκευές IoT χωρίς τα ζητήματα οργάνωσης που θα προέκυπταν με την χρήση ξεχωριστών topic ανά συσκευή με τα ανάλογα δεκάδες subtopics για τα δεδομένα που αυτή η συσκευή θα αποστέλλει.

3.10.3 Εξυπηρετητής βάσης δεδομένων χρονοσειρών – InfluxDB timeseries database

Το InfluxDB είναι μια βάση δεδομένων χρονοσειρών που έχει σχεδιαστεί για να χειρίζεται υψηλά φορτία εγγραφής και ερωτήσεων (queries) . Αποτελεί αναπόσπαστο τμήμα της στοίβας TICK. Το InfluxDB προορίζεται να χρησιμοποιηθεί ως υποδομή αποθήκευσης για εκείνες τις περιπτώσεις χρήσεων που περιλαμβάνουν μεγάλες ποσότητες δεδομένων με χρονική σήμανση, συμπεριλαμβανομένης της παρακολούθησης DevOps, μετρήσεων εφαρμογών, δεδομένων αισθητήρων IoT και αναλυτικών στοιχείων σε πραγματικό χρόνο. Τα βασικά χαρακτηριστικά της είναι τα ακόλουθα : Προσαρμοσμένη βάση δεδομένων υψηλής απόδοσης που έχει υλοποιηθεί ειδικά για δεδομένα χρονοσειρών. Η επεξεργαστική μηχανή TSM επιτρέπει υψηλή ταχύτητα εισόδου και συμπίεση δεδομένων Είναι γραμμένη εξ ολοκλήρου σε Go. Αυτό σημαίνει πως γίνεται compile από ένα μόνο δυαδικό αρχείο χωρίς περαιτέρω εξωτερικές εξαρτήσεις (dependencies). Διαθέτει απλές διεπαφές (API) HTTP με δυνατότητα εγγραφής και ερωτημάτων υψηλής απόδοσης. Υποστηρίζει πρόσθετα (plugins) για άλλα πρωτόκολλα λήψης δεδομένων όπως τα Graphite, collectd και OpenTSDB. Τα ερωτήματα προς την βάση γίνονται σε γλώσσα τύπου SQL (simple query language), προσαρμοσμένη για εύκολη αναζήτηση συγκεντρωτικών δεδομένων.

Η χρήση ετικετών (tags) επιτρέπουν την ευρετηρίαση των χρονοσειρών για γρήγορα και αποτελεσματικά ερωτήματα.

Τι είναι και για ποιο λόγο χρησιμοποιούμε βάση δεδομένων χρονοσειρών (timeseries database)

Χρονοσειρά είναι οποιοδήποτε σύνολο τιμών με timestamp όπου ο χρόνος είναι μια σημαντική συνιστώσα των δεδομένων. Το τυπικό παράδειγμα μιας χρονοσειράς είναι τα δεδομένα της τιμής ανταλλαγής μετοχών. Αν προβάλουμε τα δεδομένα που αφορούν την διακύμανση μιας μετοχής, την τιμής του bitcoin ή την κλίση μιας κεραίας σε ένα βάθος μηνών, εύκολα αντιλαμβανόμαστε ότι μιλάμε για έναν τεράστιο όγκο πληροφοριών, πληροφορίες που καλούμαστε διαρκώς να επεξεργαστούμε για διάφορους σκοπούς. Εκεί προκύπτει και η ανάγκη για βάσεις δεδομένων, εξειδικευμένες και βελτιστοποιημένες στον χειρισμό χρονοσειρών. Οι Βάσεις Δεδομένων Χρονοσειρών καλούνται συνήθως να επιλύσουν δύο απαιτήσεις: υψηλή απόδοση εγγραφής και υψηλά ποσοστά ερωτήσεων. Ας πάμε σε βάθος σε αυτά τα σημεία.

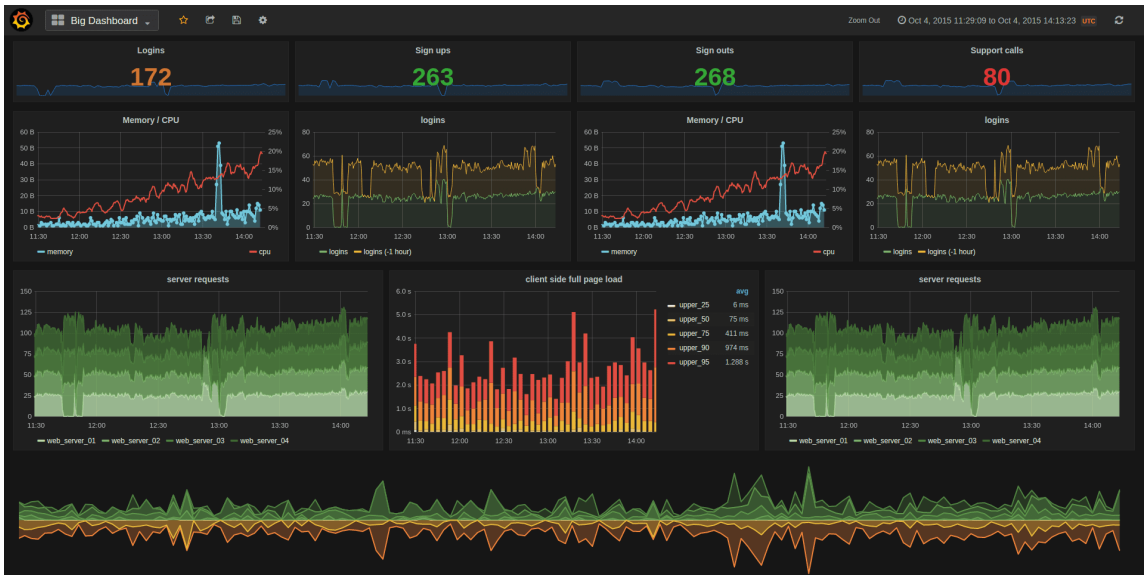
- ***Ρυθμός διεκπεραίωσης εγγραφών:*** Το ποσό των χρονικών σειρών που παράγονται από, για παράδειγμα, την παρακολούθηση ενός στόλου διακομιστών μπορεί γρήγορα να αυξηθεί από χιλιάδες νέες τιμές ανά δευτερόλεπτο σε εκατομμύρια νέες τιμές ανά δευτερόλεπτο. Η εισαγωγή αυτών των δεδομένων σε μια κανονική σχεσιακή βάση δεδομένων, όπως η MySQL και η PostgreSQL, επιβαρύνει πολύ γρήγορα τα περισσότερα συστήματα χωρίς προσεκτική εξειδικευμένη σχεδίαση. Ακόμα και τότε, υπάρχουν όρια στο ποσό των δεδομένων που μπορεί να δεχτεί ένα σύστημα ανά πάσα στιγμή.

- ***Ρυθμός διεκπεραίωσης ερωτημάτων:*** Μια πτυχή των δεδομένων χρονοσειρών είναι ότι τα νέα δεδομένα είναι σχεδόν πάντα πιο πολύτιμα από τα παλιά δεδομένα. Και πάλι, ας φέρουμε σαν παράδειγμα ένα στόλο διακομιστών. Το να γνωρίζεις ότι ένας διακομιστής έχει αρχίσει να χρησιμοποιεί όλη του

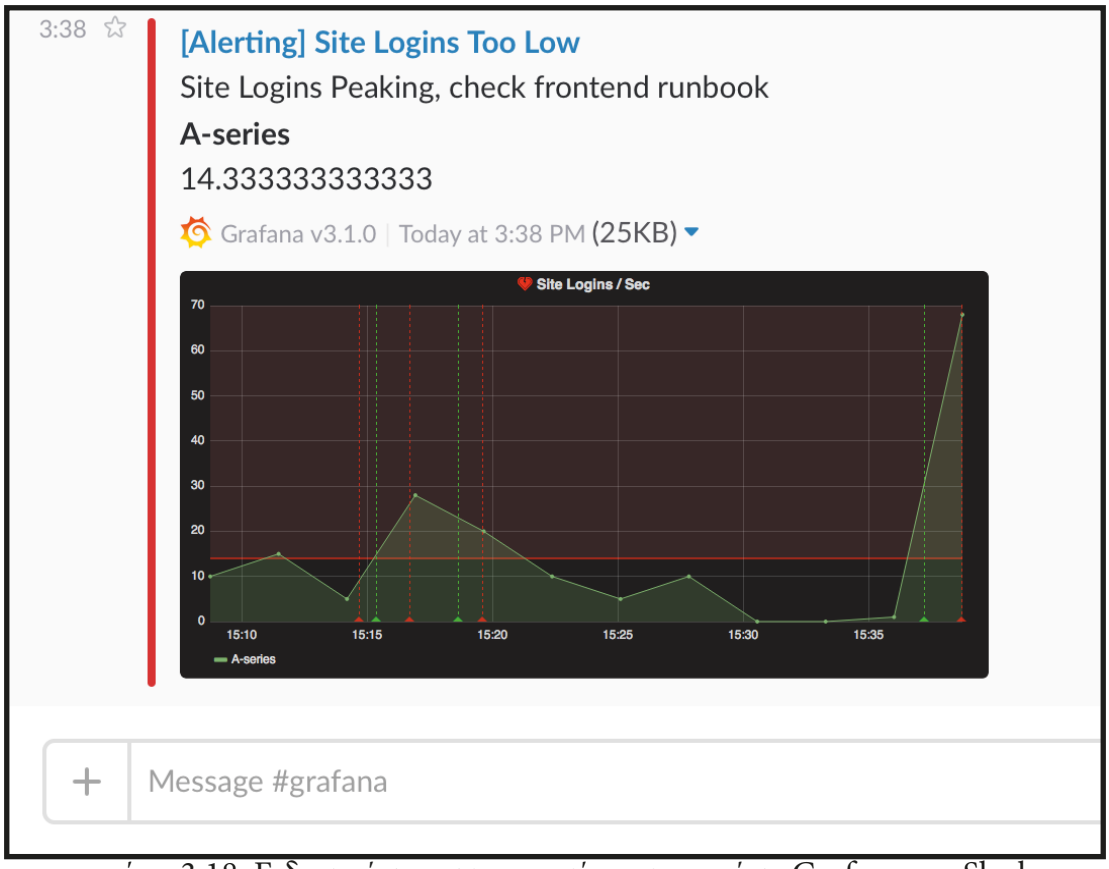
την CPU σε πραγματικό χρόνο, είναι πολύ πιο χρήσιμο από το να δει κανείς αυτά τα στοιχεία ακόμα και λίγα λεπτά πιο μετά. Η φύση των χρονοσειρών καθιστά αναγκαία την ταχύτερη δυνατή έκθεση των νέων δεδομένων σε ερωτήματα. Επειδή μια βάση δεδομένων χρονοσειρών ειδικεύεται στην επεξεργασία δεδομένων με χρονική σήμανση, είναι βελτιστοποιημένη κατάλληλα για την αντιμετώπιση των παραπάνω δύο θεμελιωδών ζητημάτων. [22][23]

3.11 Εξυπηρετητής Αναπαράστασης και Ειδοποιήσεων – Grafana

Το εργαλείο Grafana αποτελεί μία δημοφιλή πλατφόρμα ανοιχτού κώδικα για την ανάλυση δεδομένων και δημιουργία γραφικών παραστάσεων. Είναι γραμμένη στις γλώσσες Go, JavaScript και υποστηρίζει πολλές βάσεις δεδομένων όπως τις: Graphite, Elasticsearch, CloudWatch, InfluxDB, OpenTSDB, Prometheus, MySQL και την Postgres. Προσφέρει πολλά είδη γραφικής απεικόνισης στον χρήστη, δυνατότητα μεγέθυνσης σε συγκεκριμένη περιοχή καθώς και δυνατότητα απεικόνισης πολλών μετρήσεων σε μία γραφική παράσταση.



εικόνα 3.17: Απεικόνιση δεδομένων στην Grafana



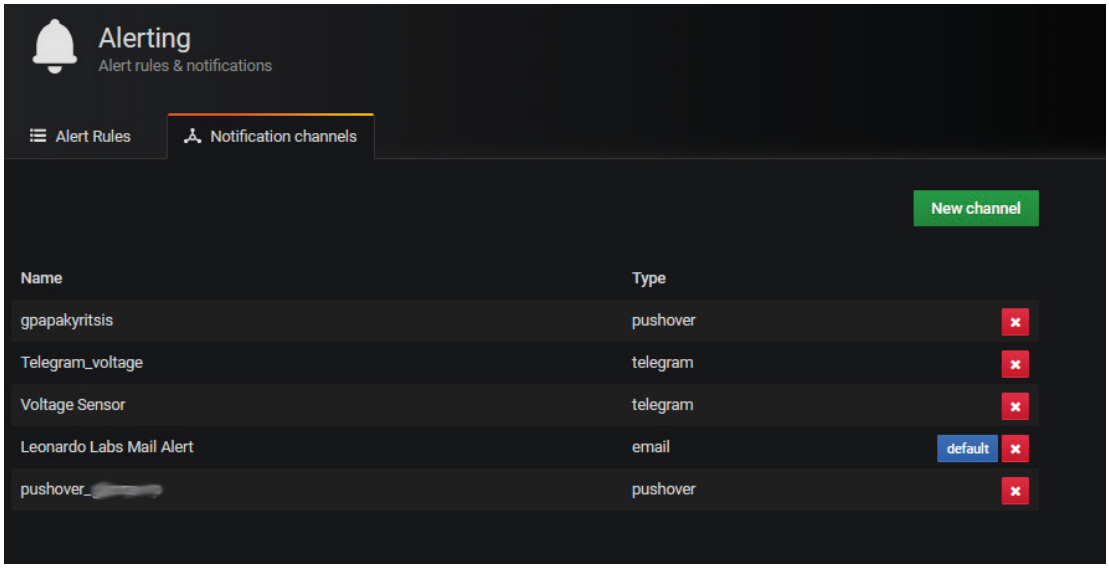
εικόνα 3.18: Ειδοποιήσεις στο κινητό με την χρήση Grafana και Slack

Μια άλλη ιδιαίτερα σημαντική δυνατότητα που παρέχει είναι η δημιουργία ειδοποιήσεων. Μέσω αυτών μπορούμε να λαμβάνουμε άμεση ενημέρωση για κρίσιμες τροποποιήσεις σε μία σειρά δεδομένων που παρακολουθούμε. Η Grafana διαθέσει APIs για την αποστολή ειδοποιήσεων σε μέηλ, Pushover, Pushbullet, Telegram κ.α.

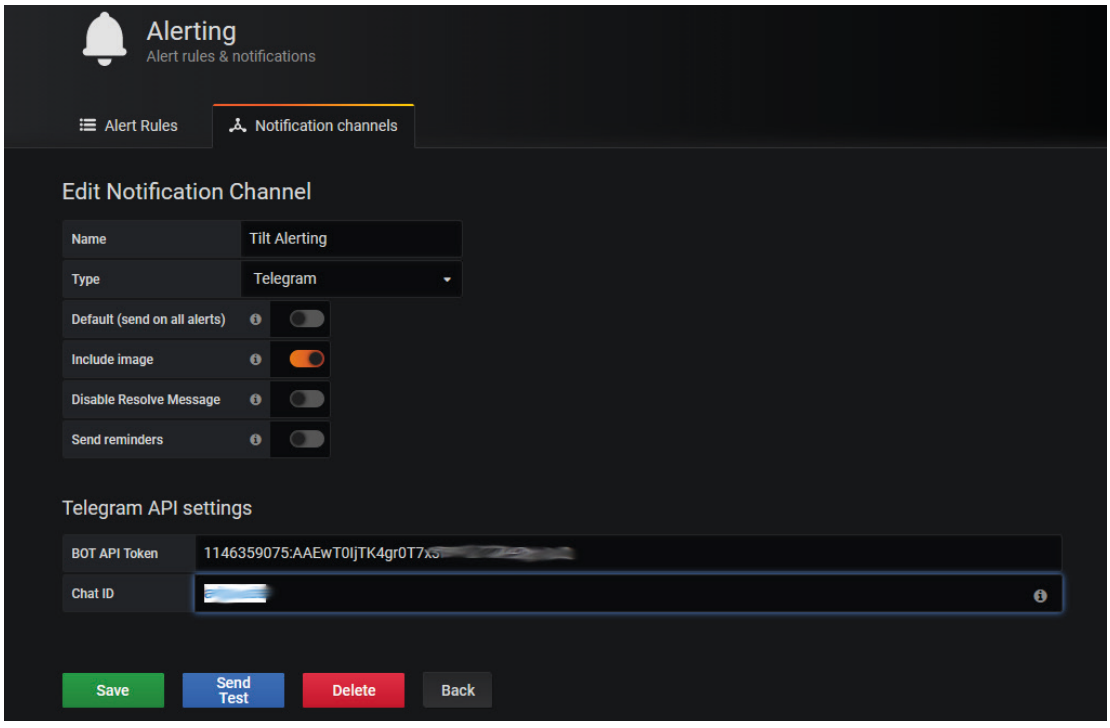
Επιλέχθηκε η εγκατάσταση του εξυπηρετητή Grafana στο ίδιο VM με την εγκατάσταση του Tick Stack. Ο λόγος που επιλέχθηκε η υλοποίηση της Grafana έναντι του Kapacitor που ενσωματώνεται στο Tick Stack, είναι οι διάφορες αυξημένες λειτουργίες γραφιστικής απεικόνισης που περιλαμβάνει.

3.12 Παραμετροποίηση ειδοποιήσεων

Με την χρήση του Grafana αρχικά δημιουργούμε ένα κανάλι ειδοποιήσεων (εικόνες 3.19-3.20). Για την περίπτωσή μας επιλέγουμε την χρήση του Telegram που μας δίνει την δυνατότητα να λαμβάνουμε ειδοποιήσεις σε μια σειρά συσκευών (κινητό, λάπτοπ, tablet).

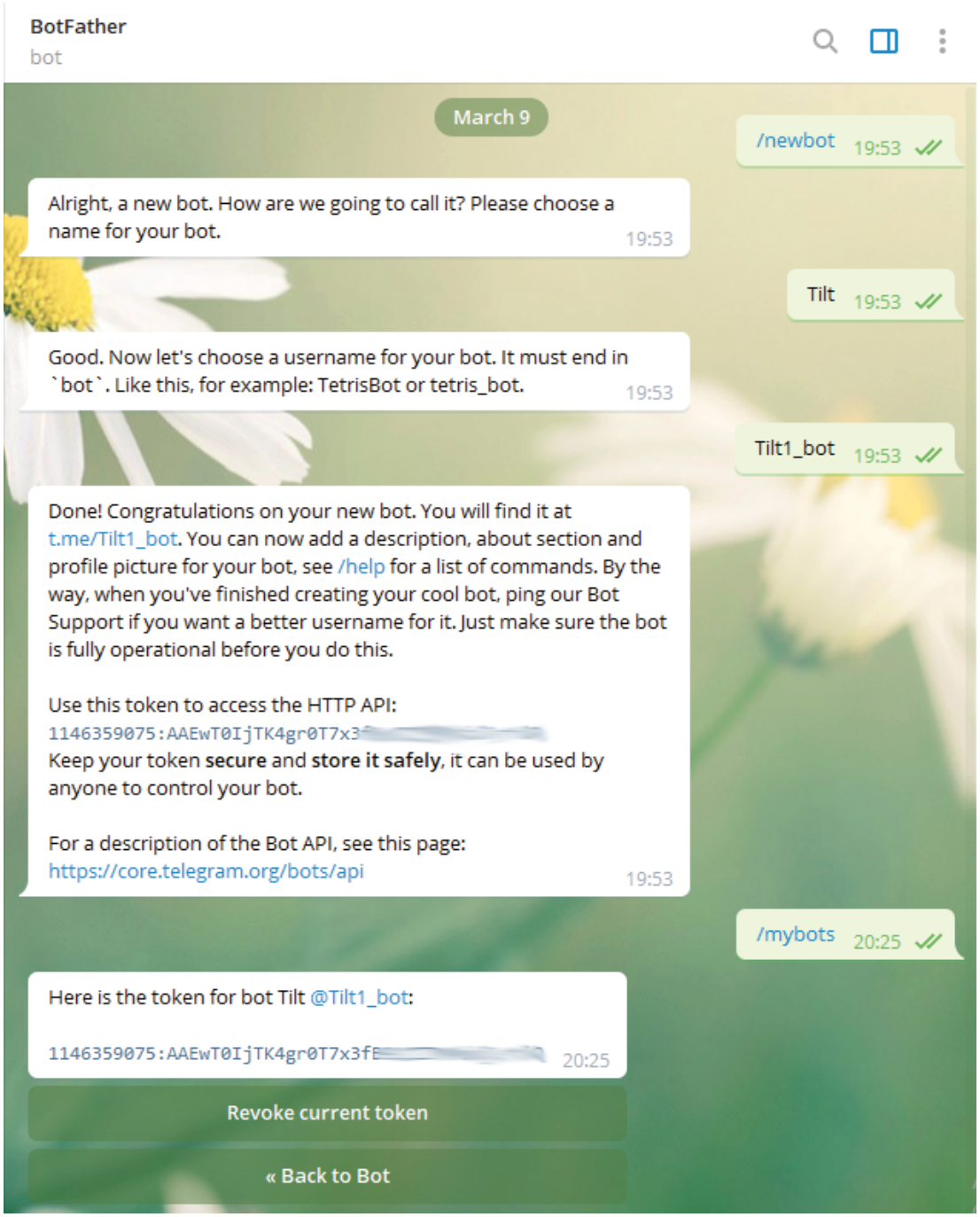


εικόνα 3.19: Μενού καναλιών ειδοποιήσεων στην Grafana



εικόνα 3.20: Κανάλι ειδοποίησης μέσω Telegram

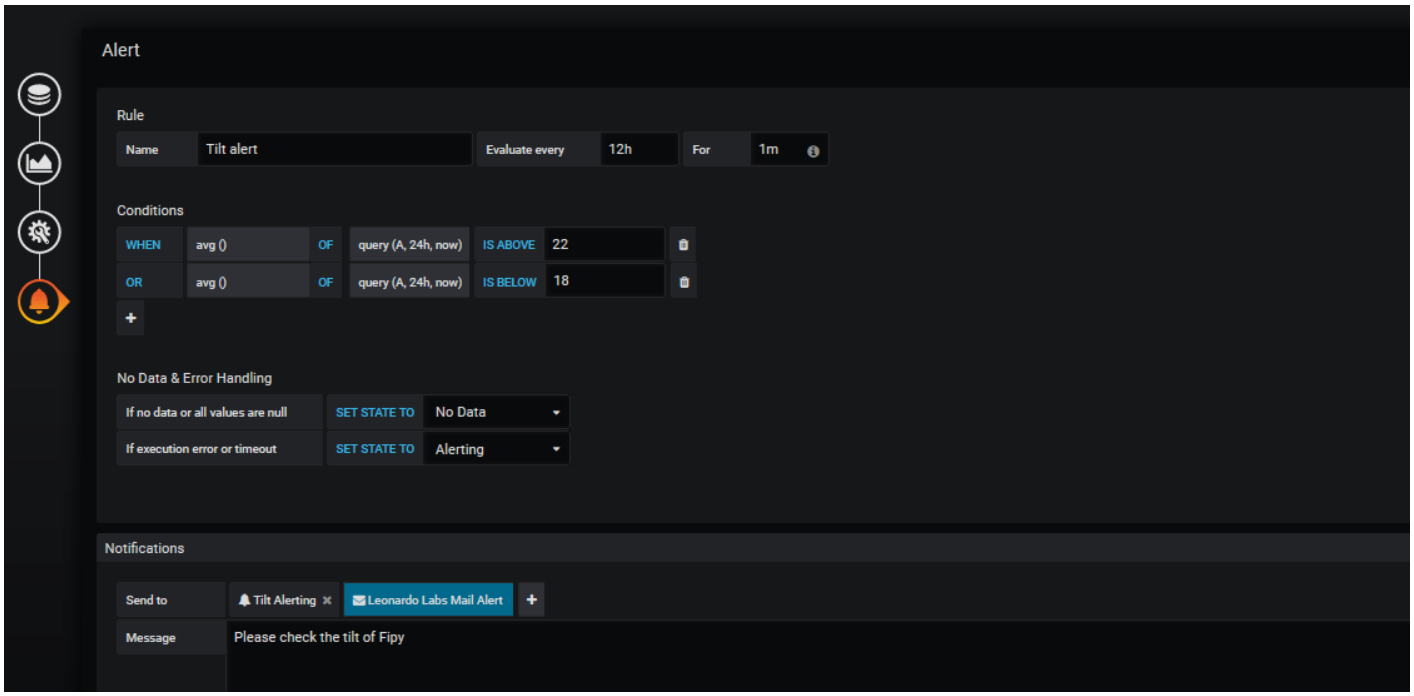
Στην συνέχεια εντός του Telegram δημιουργούμε το κατάλληλο bot που διασυνδέεται με την Grafana μέσω της κατάλληλης διεπαφής (API).



εικόνα 3.21: Δημιουργία Telegram alerting bot

Τέλος στην σελίδα στην Grafana όπου αποτυπώνονται τα δεδομένα κλίσης, δημιουργούμε τον κανόνα εκείνο που θα ενεργοποιεί το κανάλι ειδοποιήσεων και θα μας ενημερώνει στο κινητό για τυχόν απόκλιση.

Στο δικό μας παράδειγμα έχουμε ορίσει σαν φυσιολογική τιμή κλίσης τις 20 μοίρες και η Grafana θα μας αποστέλει ειδοποίηση όταν η τιμή αποκλείνει κατα +/- 2 μοίρες.



εικόνα 3.22: Δημιουργία ειδοποίσης μεταβολής κλίσης

Κεφάλαιο 4
Πειράματα

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιαστούν και θα σχολιαστούν τα παρακάτω πειράματα.

- Η απόκλιση (offset) των μετρήσεων κλίσης των δυο αισθητήρων από την πραγματική κλίση, όπως αυτή θα καταγράφεται παράλληλα από εξειδικευμένη συσκευή ακριβείας. Η σύγκριση θα πραγματοποιηθεί και για τους 2 άξονες (tilt, roll) για την εύρεση του άξονα εκείνου που δίνει την μεγαλύτερη ακρίβεια.
- Η επίδραση της θερμοκρασίας περιβάλλοντος στην καταγραφόμενη κλίση.
- Η κατανάλωση ενέργειας των αισθητήρων κατά την διάρκεια της εκτέλεσης του κώδικα μέτρησης κλίσης.
- Η κατανάλωση ενέργειας των αισθητήρων κατά την διάρκεια της εκτέλεσης του κώδικα μέτρησης κλίσης με την προσθήκη της υπορουτίνας μειωμένης κατάναλωσης ρεύματος (deepsleep).
- Η αποφόρτιση της μπαταρίας κατά την συνεχή λειτουργία χωρίς την παρουσία φωτοβολταϊκού πάνελ.
- Η αποφόρτιση της μπαταρίας κατά την συνεχή λειτουργία, με την παρουσία φωτοβολταϊκού πάνελ.
- Η χρήση της μπαταρίας κατά την διακοπτόμενη λειτουργία (12 μετρήσεις ανα ημέρα), με την παρουσία φωτοβολταικού πάνελ.

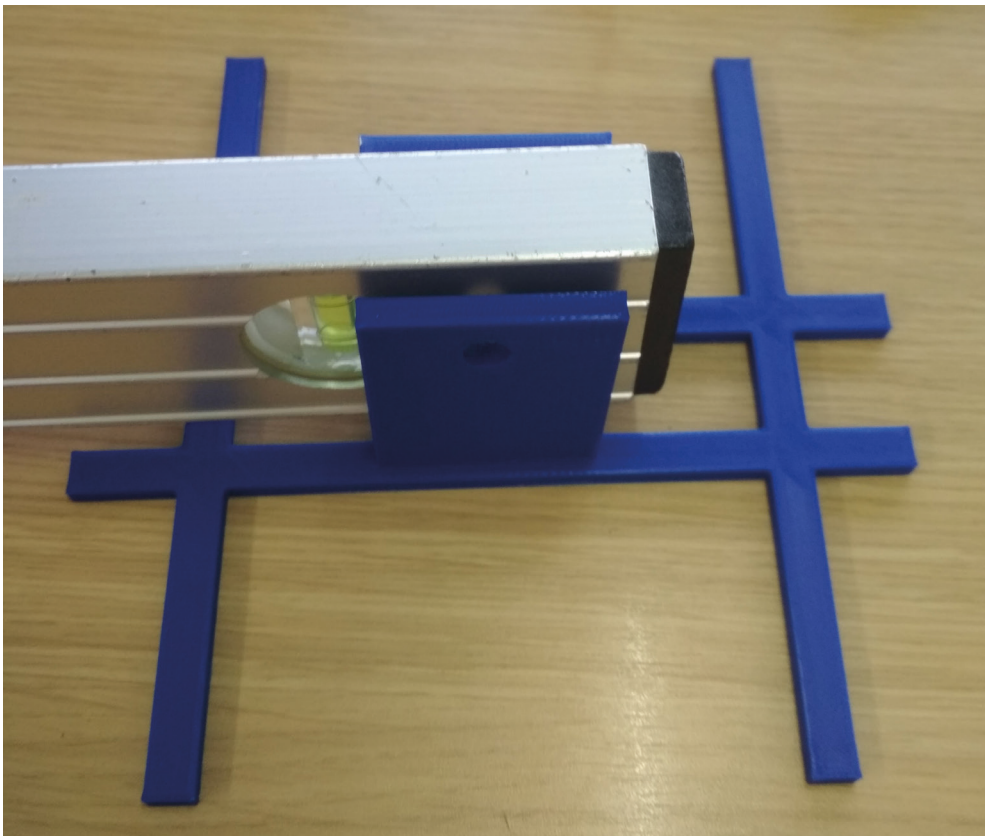
- Καταγραφές σχετιζόμενες με ζητήματα συνδεσιμότητας και σταθερότητας των IoT αισθητήρων καθώς και καταγραφή των πρακτικών επιλογών επίλυσής τους.

4.2 Μεθοδολογία πειραματικής διαδικασίας

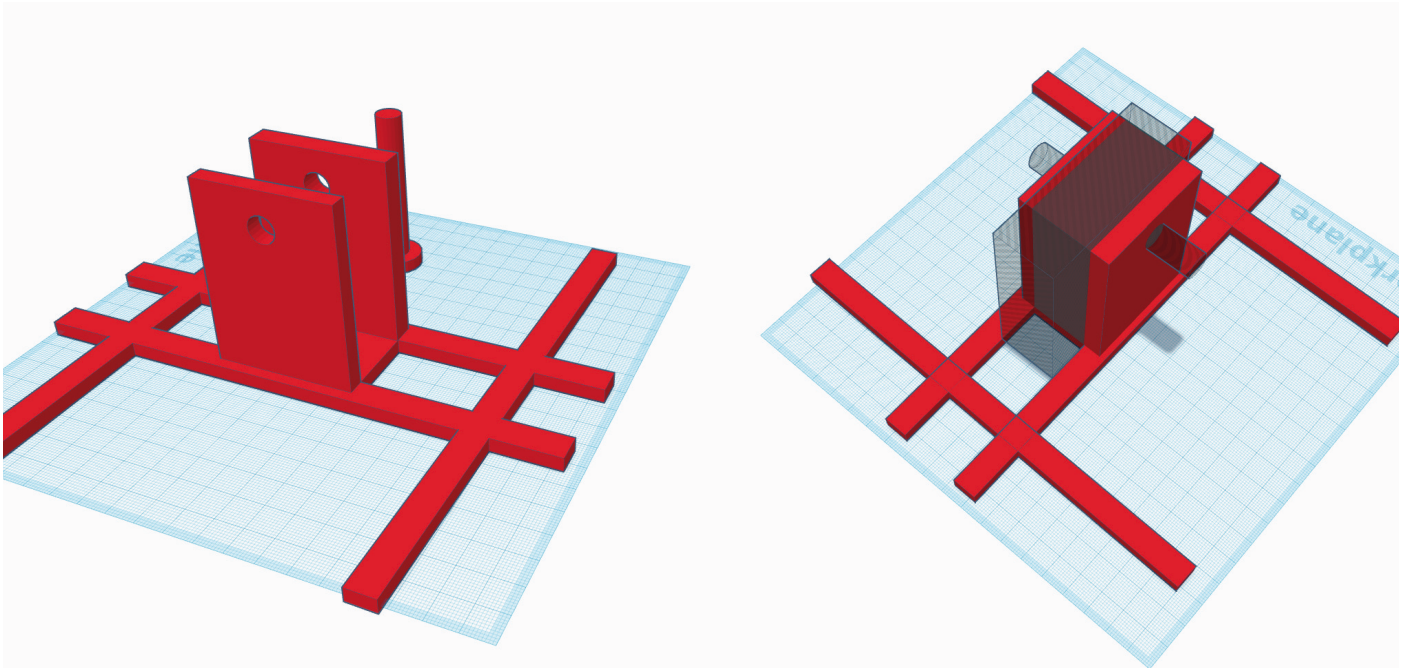
4.2.1 Πειράματα σχετιζόμενα με την απόκλιση πραγματικής και καταγραφόμενης κλίσης

Για την καταγραφή της πραγματικής κλίσης χρησιμοποιήθηκε το γωνιόμετρο BOSCH DNM 60L (εικόνα 4.1) προσαρμοσμένο σε μια κατάλληλη βάση (εικόνα 4.2) που μας επέτρεπε την ελεγχόμενη μεταβολή της κλίσης και την διατήρησή της σε σταθερό επίπεδο. Η βάση σχεδιάστηκε εκ του μηδενός σε τρισδιάστατο σχέδιο (εικόνα 4.3) με βάση τα χαρακτηριστικά του BOSCH DNM 60Lκαι υλοποιήθηκε με την χρήση 3D εκτύπωσης. Τέλος οι αισθητήρες προσαρμόστηκαν πάνω στο εργαλείο μέτρησης της κλίσης (εικόνα 4.4)

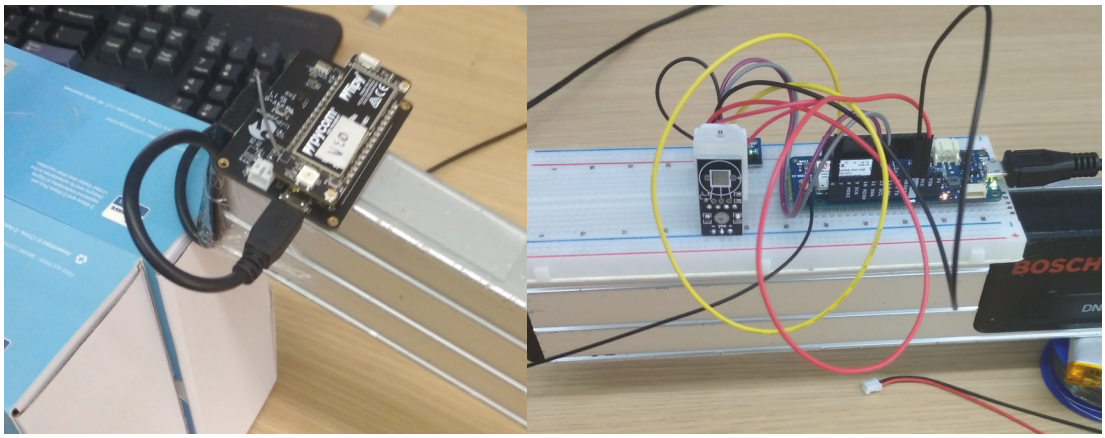
εικόνα 4.1: γωνιόμετρο ακριβείας BOSCH DNM60L



εικόνα 4.2: Βάση και γωνιόμετρο



εικόνα 4.3: Σχεδίαση βάσης

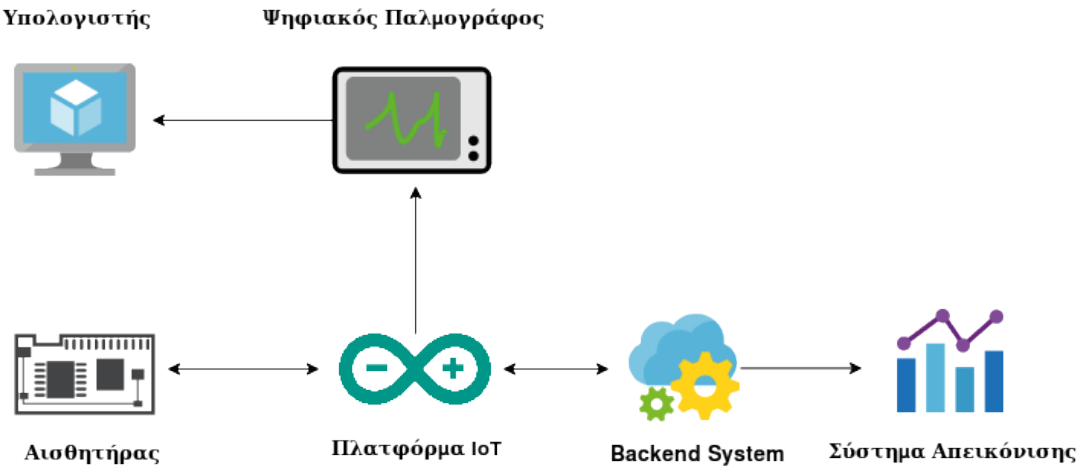


εικόνα 4.4: Προσαρμογή των αισθητήρων πάνω στο μετρητικό

4.2.2 Πειράματα σχετιζόμενα με τα ενεργειακά χαρακτηριστικά των αισθητήρων

Για την καταγραφή του ρεύματος κατανάλωσης χρησιμοποιήθηκε ο ψηφιακός παλμογράφος BK Predicion 2831E ο οποίος διαθέτει λογισμικό για την αποθήκευση των τιμών σε ένα αρχείο μορφής .csv στον υπολογιστή. Στη συνέχεια με τη χρήση των κατάλληλων custom python scripts απομακρύνθηκαν κάποια περιττά δεδομένα και δημιουργήθηκαν οι γραφικές παραστάσεις. Στην παρακάτω εικόνα φαίνεται αναλυτικά ο τρόπος που συνδέθηκε ο συσσωρευτής, η πλατφόρμα IoT και ο ψηφιακός παλμογράφος για να πραγματοποιηθεί κύκλωμα και να υλοποιηθούν οι μετρήσεις.

Όπως αποτυπώνεται και στο σχήμα μετρήσεις για την ενεργειακή



εικόνα 4.5: Μεθοδολογία καταγραφής των ενεργειακών μετρήσεων

λειτουργία των αισθητήρων (συγκεκριμένα για την στάθμη φόρτισης της μπαταρίας) κατεγράφησαν και μέσω του εξυπηρετητή γραφικής αναπαράστασης Grafana.

4.2.3 Καταγραφές σχετιζόμενες με ζητήματα λειτουργικότητας των αισθητήρων.

Οι καταγραφές αυτές πραγματοποιήθηκαν στο σειριακό περιβάλλον επικοινωνίας των αισθητήρων με τον υπολογιστή. Στην περίπτωση του Arduino MKR 1500 χρησιμοποιήθηκε το Serial Monitor του Arduino IDE, ενώ στην περίπτωση του Pycom Fipy για την σειριακή διεπαφή χρησιμοποιήσαμε το CLI πρόγραμμα Rshell.

4.3 Πειράματα

Στο κομμάτι αυτό απλώς αποτυπώνονται οι καταγραφές των πειραμάτων. Η ανάλυση των καταγραφών και των πειραμάτων γίνεται στο επόμενο κεφάλαιο, πριν τα εκάστωτε συμπεράσματα.

4.3.1 Απόκλιση (offset) μετρήσεων κλίσης

FIPY							
TEST 1 (pitch oriented)				TEST 2 (roll oriented)			
real	pitch	Pitch (offset)	roll	real	pitch	roll	roll (offset)
0	-0.91	0.91	-1.4	0	-1.48	0.25	-0.25
15	13.95	1.05	-2.36	15	0.06	15	0
30	29.13	0.87	-3.2	30	0.72	29.92	0.08
45	44.27	0.73	-2.16	45	0.14	44.74	0.26
60	59.49	0.51	-0.8	60	0.65	59.35	0.65
75	74.75	0.25	1	75	1.8	73.91	1.09
90	87.36	2.64	0.8	90	0.96	88.81	1.19

Pitch offset	
Min	0.25
Max	2.64
Max-Min	2.39
Median	0.87

Roll offset	
Min	-0.25
Max	1.19
Max-Min	1.44
Median	0.26

εικόνα 4.6: Απόκλιση γωνίας για το Fipy

MKR 1500							
TEST 1 (pitch oriented)				TEST 2 (roll oriented)			
real	pitch	Pitch (offset)	roll	real	pitch	roll	roll (offset)
0	-2.54	2.54	1.69	0	-1.96	0.99	-0.99
15	12.23	2.77	0.92	15	-2.13	12.57	2.43
30	25.66	4.34	0.29	30	-2.24	25.45	4.55
45	40.48	4.52	-0.07	45	-2.14	40.07	4.93
60	53.71	6.29	0.09	60	-2.32	54.3	5.7
75	68.2	6.8	-0.11	75	-1.26	70.79	4.21
90	83.5	6.5	-0.33	90	-1.51	84.97	5.03

Pitch offset	
Min	2.54
Max	6.8
Max-Min	4.26
Median	4.52

Roll offset	
Min	-0.99
Max	5.7
Max-Min	6.69
Median	4.55

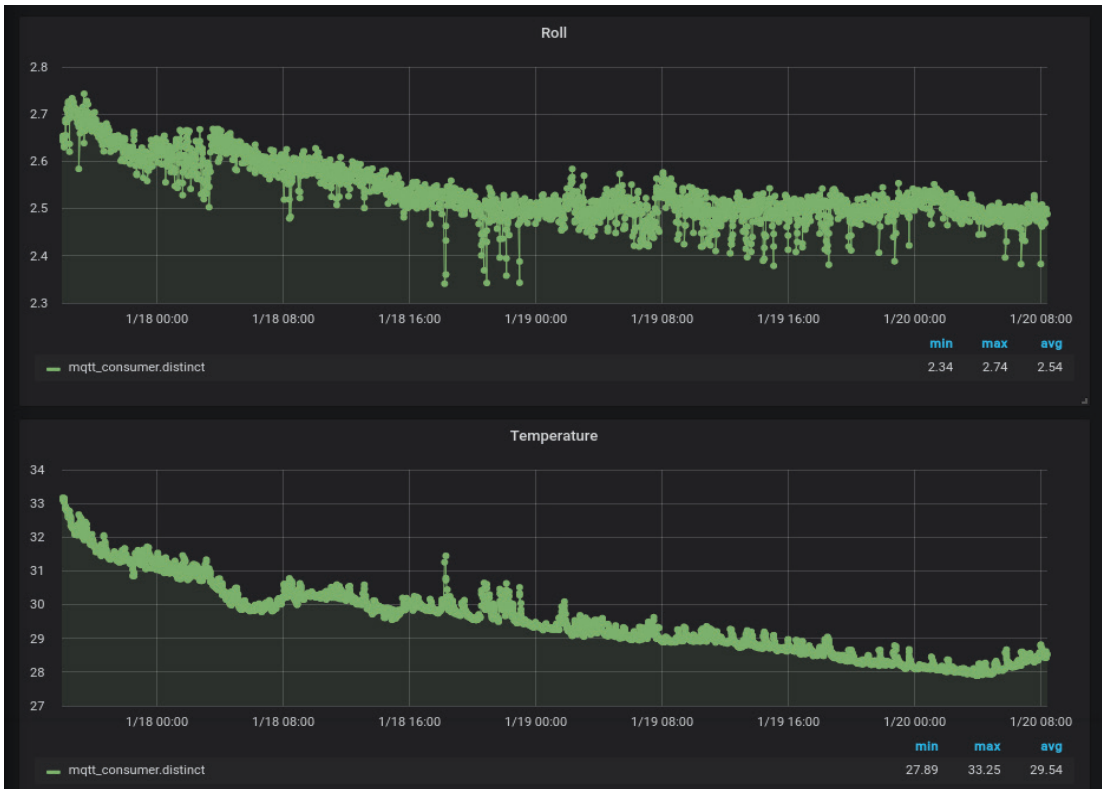
εικόνα 4.7: Απόκλιση γωνίας για το MKR1500

Παρατηρήσεις

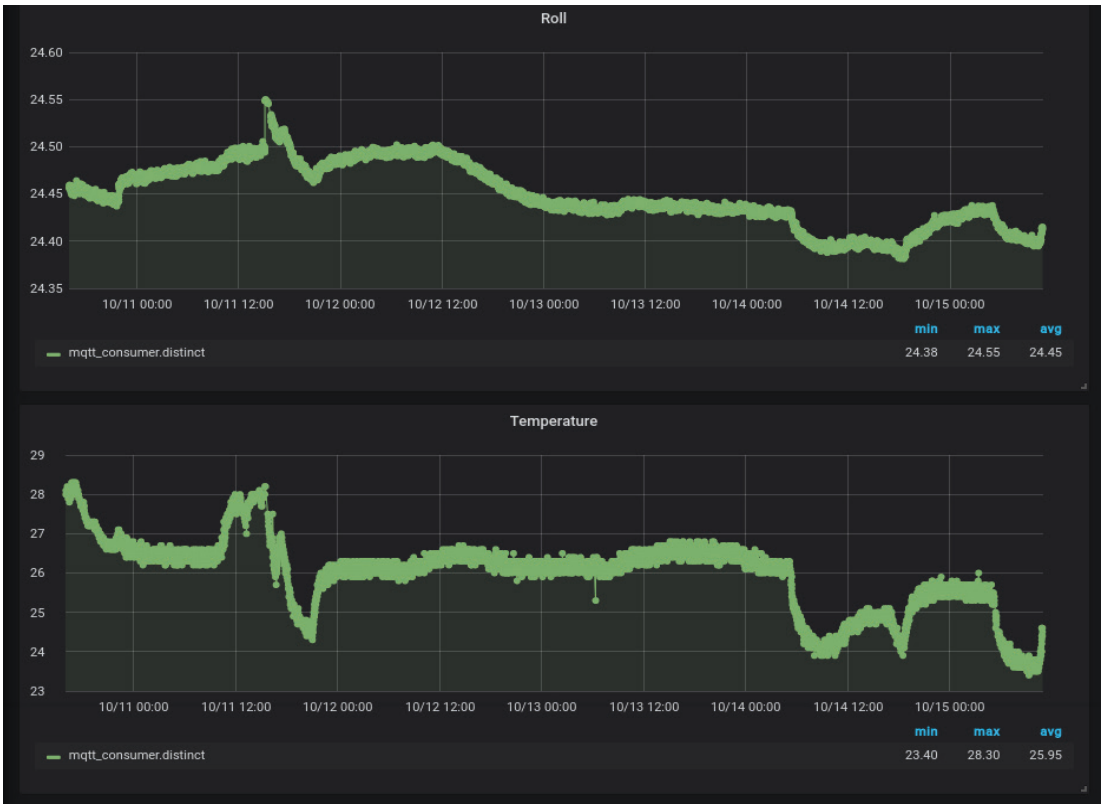
- Καταγράφουμε την απόκλιση και προς τους δύο πιθανούς προσανατολισμούς, ώστε να εντοπίσουμε ποιος μας δίνει το βέλτιστο αποτέλεσμα ακριβείας
- Δεν μας ενδιαφέρει το μέτρο της απόκλισης, καθώς αυτό οφείλεται στις αποκλίσεις που προκύπτουν κατά την προσαρμογή των αισθητήρων πάνω στο μετρητικό όργανο
- Μας ενδιαφέρει η διακύμανση της απόκλισης στις διάφορες μετρούμενες γωνίες, καθώς αυτή καθορίζει την ακρίβεια μέτρησης του αισθητήρα σε διαφορετικές γωνίες.
- Μέρος της απόκλισης μπορεί να οφείλεται σε διακυμάνσεις της γωνιάς στην μη-σημαντική - κατά περίπτωση – πλευρά. Δυστυχώς χωρίς την χρήση εξειδικευμένου εξοπλισμού που να μας εξασφαλίζει ακρίβειες <0.1° κάτι τέτοιο δεν μπορεί να εξακριβωθεί πειραματικά.

4.3.2 Επίδραση θερμοκρασίας

- ***Για το Fipy:*** Με βάση τα μέγιστα/ελάχιστα των δύο τιμών έχουμε διαφορά γωνίας 0.4° για διαφορά θερμοκρασίας 5,36° C. Με αναγωγή, η καταγραφή γωνίας αλλάζει κατά ~0.0746° μοίρες για κάθε 1 βαθμό Celsius μεταβολής της θερμοκρασίας.
-
- ***Για το MKR1500:*** Με βάση τα μέγιστα/ελάχιστα των δύο τιμών έχουμε διαφορά γωνίας 0.17° για διαφορά θερμοκρασίας 4.9° C. Με αναγωγή, η καταγραφή γωνίας αλλάζει κατά ~0.0347° μοίρες για κάθε 1 βαθμό Celsius μεταβολής της θερμοκρασίας.



εικόνα 4.8: Επίδραση της θερμοκρασίας για το Fipy



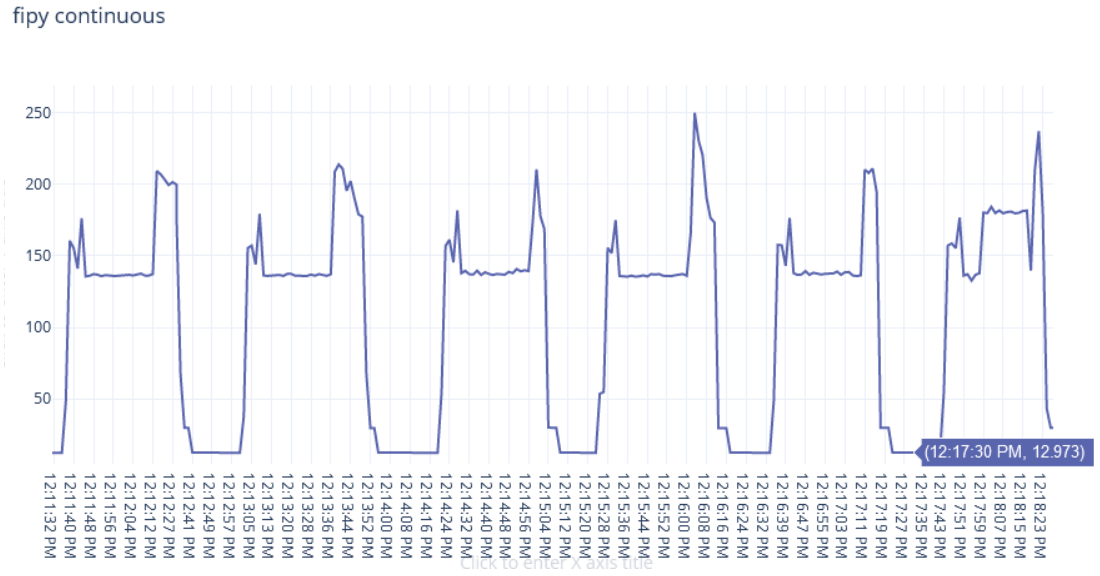
εικόνα 4.9: Επίδραση της θερμοκρασίας για το MKR1500

Παρατηρήσεις

- Ο αισθητήρας μας θα τοποθετηθεί σε τηλεπικοινωνιακά κεραιοσυστήματα που υπόκεινται σε ακραίες διακυμάνσεις θερμοκρασίας (~0ο-40ο C). Με βάση την μεταβολή κλίσης ανά μονάδα θερμοκρασίας θα έχουμε αποκλίσεις ±1,492 μοιρών για το Fipy και ±0,694 μοιρών για το MKR1500 στις ακραίες θερμοκρασίες σε σχέση με την κλίση που θα καταγράφεται σε μία ενδεικτική μέση θερμοκρασία 20ο. Αυτή η απόκλιση είναι αποδεκτή στις περισσότερες περιπτώσεις. Η παράλληλη καταγραφή της θερμοκρασίας από τον αισθητήρα θα επιτρέπει στους τεχνικούς που επιβλέπουν την τηλεπικοινωνιακή υποδομή να εκτιμούν σε πραγματικό χρόνο αν η μεταβολή κλίσης είναι πραγματική ή οφείλεται σε ακραία καιρικά φαινόμενα.
- Παρατίθεται η καταγραφή μόνο για τον Roll άξονα, καθότι για τον Tilt άξονα δεν καταγράφηκε κάποια αξιοσημείωτη ή καταγράψιμη συσχέτιση κλίσης-θερμοκρασίας.
- Τόσο η καταγραφή της κλίσης όσο και της θερμοκρασίας γίνεται με αισθητήρες διαθέσιμους στο εμπόριο χωρίς βαθμονόμηση εργαστηριακής ακριβείας με τις ανακρίβειες που αυτονόητα προκύπτουν από αυτή την συνθήκη.

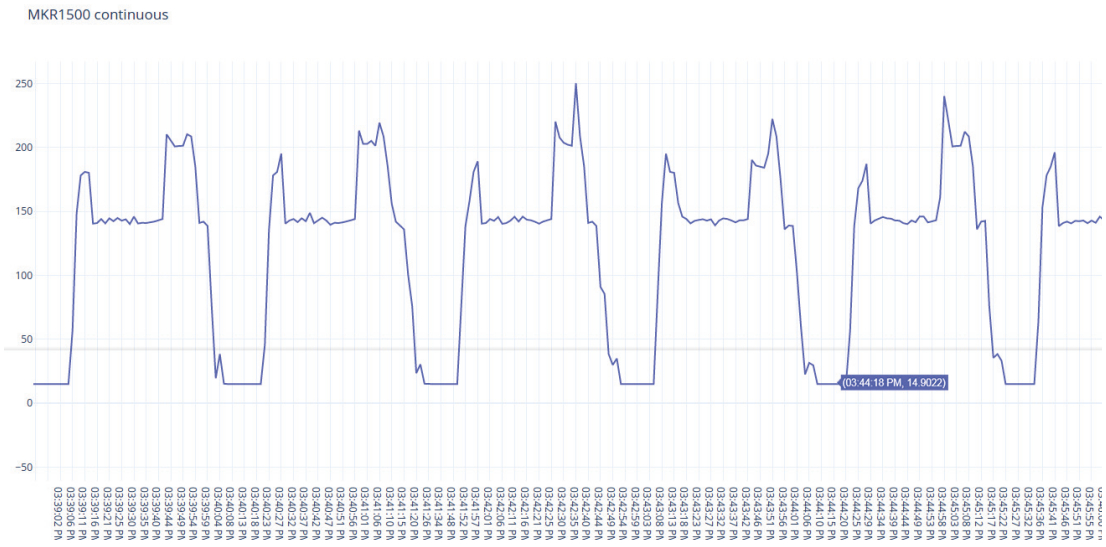
4.3.3 Καταγραφή έντασης ρεύματος συνεχούς λειτουργίας

- ***Για το Fipy:***



εικόνα 4.10: Κατανάλωση Fipy(mA) σε συνεχή λειτουργία

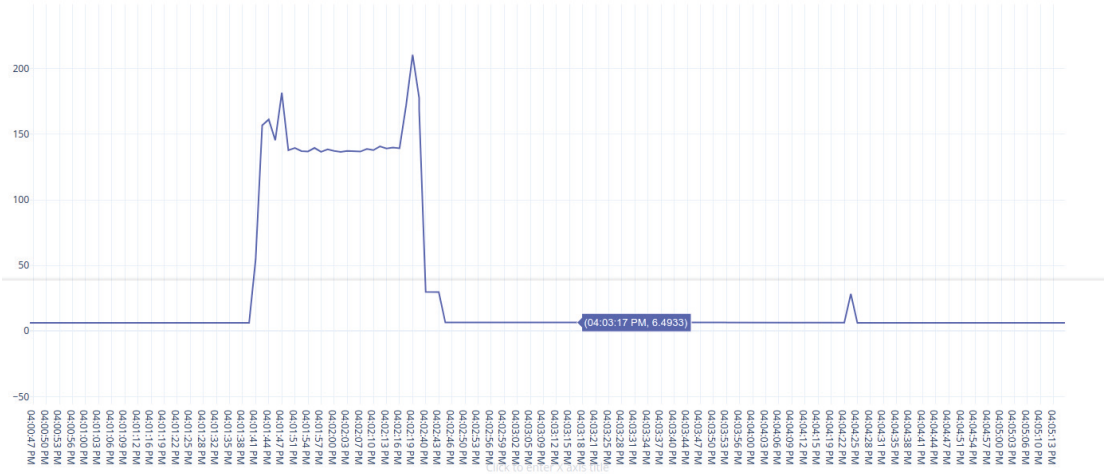
- ***Για το MKR1500:***



εικόνα 4.11: Κατανάλωση (mA) MKR1500 σε συνεχή λειτουργία

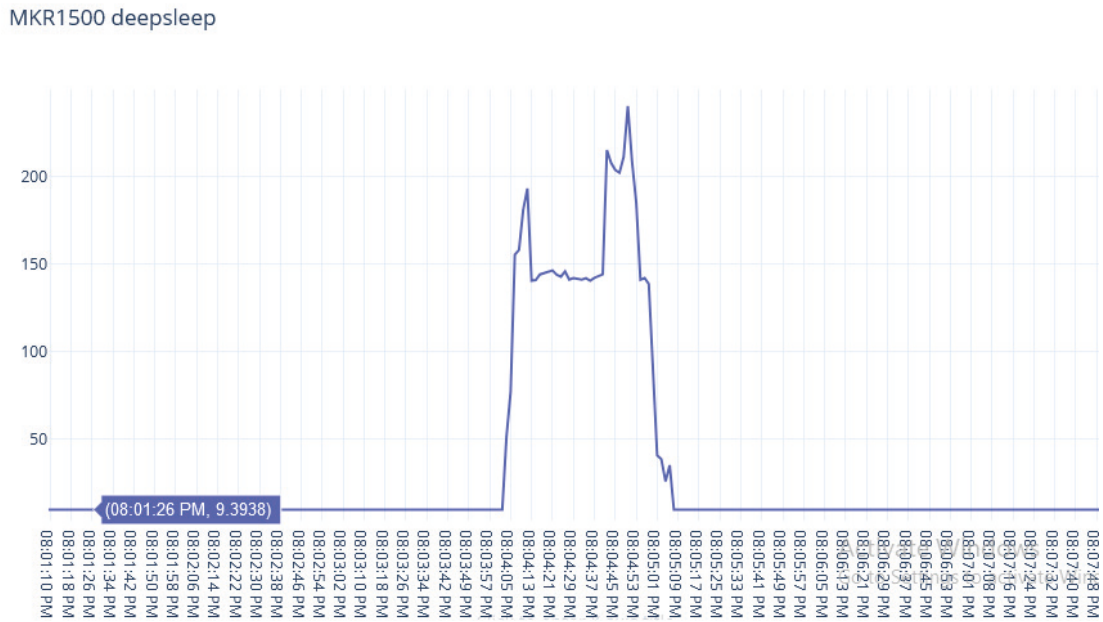
4.3.4 Καταγραφή έντασης ρεύματος λειτουργίας deepsleep

- ***Για το Fipy:***



εικόνα 4.12: Κατανάλωση(mA) Fipy σε λειτουργία deepsleep

- ***Για το MKR1500:***



εικόνα 4.13: Κατανάλωση(mA) MKR1500 σε λειτουργία deepsleep

Current (mA)	Fipy	MKR1500
start	~179 (peak)	~188 (peak)
plateu	~136	~144
network connection	~213 (peak)	~223
deepsleep	6,49	9,39
normal sleep	12,18	14,9

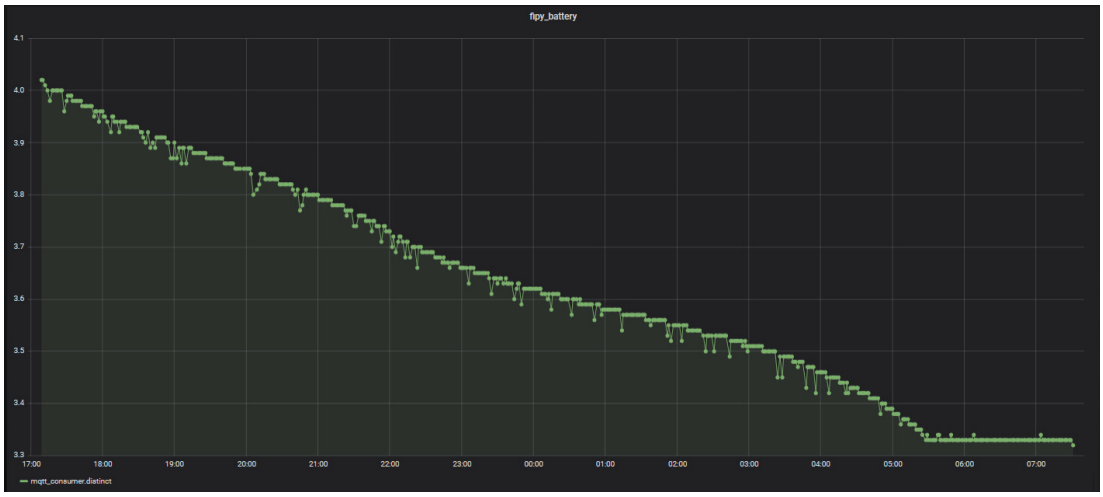
εικόνα 4.14: Προσέγγιση των ρευμάτων ανά κατάσταση λειτουργίας

Παρατηρήσεις

- Και στους δύο αισθητήρες παρατηρείται μια παρόμοια μορφή στην εξέλιξη της κατανάλωσης. Αρχικά μια κορυφή (peak) λόγω των αυξημένων ενεργειακών αναγκών εκκίνησης, στην συνέχεια ένα πλατό σχετικά ομοιόμορφης κατανάλωσης κατά την διάρκεια του επαναληπτικού προγραμματιστικού κύκλου (λούπα) υπολογισμού της κλίσης και άλλων στοιχείων (τάση μπαταρίας, θερμοκρασία) και στην συνέχεια ακόμη ένα peak κατά την διαδικασία ενεργοποίησης της Narrow Band IoT ζεύξης και αποστολής των δεδομένων.
- Η ρουτίνα ενεργοποίησης του NB-IoT modem και σύνδεσης στο δίκτυο, μπαίνει στο τελευταίο κομμάτι του κώδικα εξοικονομώντας ενέργεια.
- Θα μπορούσαμε να μειώσουμε την κατανάλωση ελαχιστοποιώντας τις επαναλήψεις υπολογισμού κλίσης. Αυτό όμως θα μείωνε την ακρίβεια μετρήσεων των αισθητήρων.
- Οι καταναλώσεις ηλεκτρικής ενέργειας δεν διαφέρουν ιδιαίτερα μεταξύ Fipy και MKR1500 με το δεύτερο να έχει λίγο υψηλότερες τόσο κατά την διάρκεια του κύκλου έναρξης-υπολογισμού-αποστολής όσο και κατά την διάρκεια του deepsleep (MKR1500: 9,39 mA έναντι Fipy: 6,49 mA).

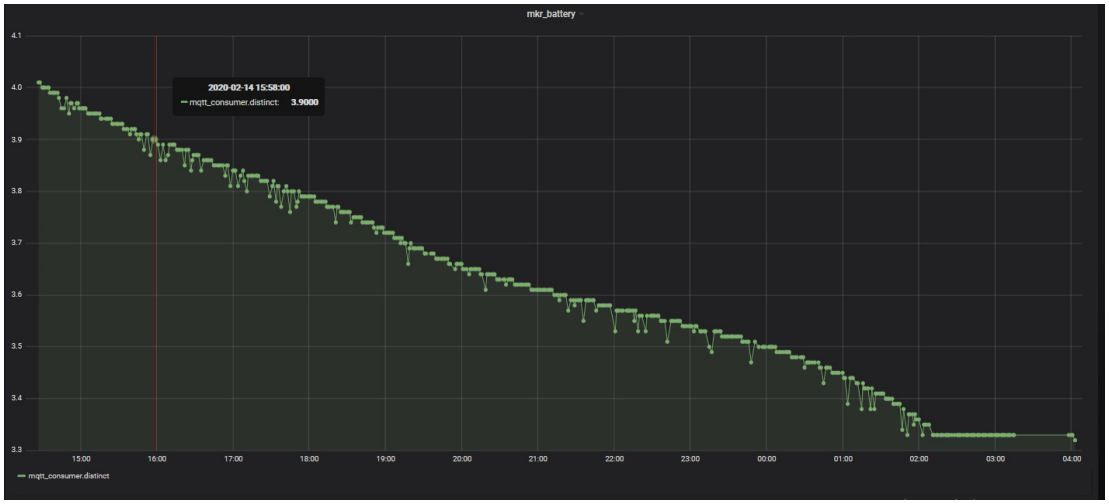
4.3.5 Αποφόρτιση μπαταρίας σε συνεχή λειτουργία χωρίς φωτοβολταϊκό πάνελ

- ***Για το Fipy:***



εικόνα 4.15: Αποφόρτιση μπαταρίας (V) για το Fipy

- ***Για το MKR1500:***



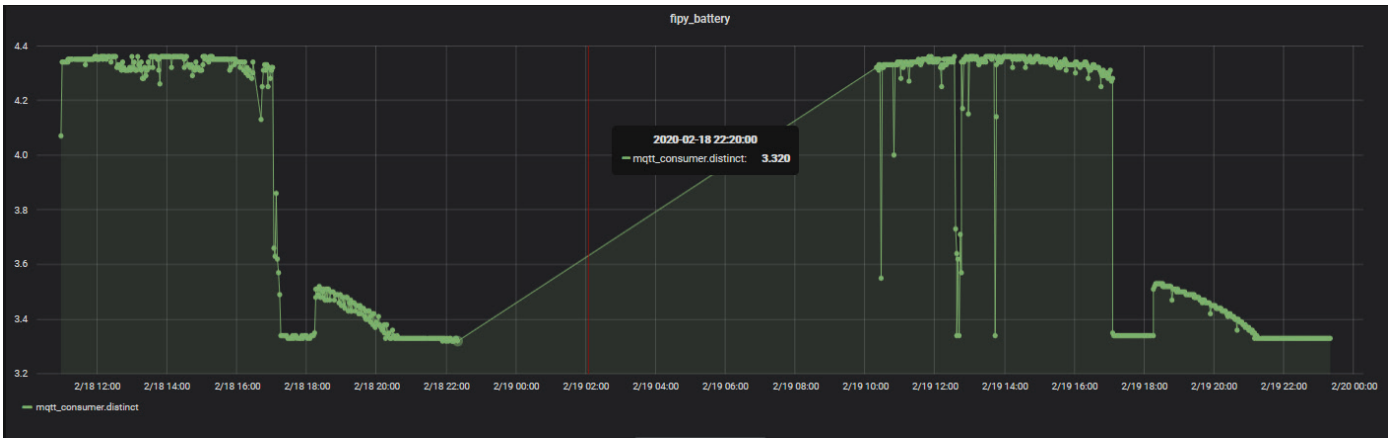
εικόνα 4.16: Αποφόρτιση μπαταρίας (V) για το MKR1500

Παρατηρήσεις

- Για το Fipy η πλήρης αποφόρτιση έγινε μέσα σε 14,5 ώρες. Για το MKR1500 η αποφόρτιση έγινε μέσα σε 13,5 ώρες, συνυπολογίζοντας όμως ότι όταν πλησίασε στην οριακή τάση λειτουργίας σταμάτησε να παίρνει μετρήσεις για μισή ώρα και στην συνέχεια επανήλθε.
- Κατά μέσο όρο έχουμε 1 μέτρηση ανά 78 δευτερόλεπτα, δηλαδή περίπου 46 μετρήσεις ανά ώρα κατά πολύ περισσότερες από τον στόχο για 1-3 μετρήσεις ανά ημέρα. Ουσιαστικά η μπαταρία σε πλήρη φόρτιση μας δίνει την δυνατότητα για ~600 με 670 συνεχείς μετρήσεις. Στην μη συνεχή λειτουργία εισέρχονται άλλοι παράγοντες όπως το ρεύμα αποφόρτισης της μπαταρίας και η κατανάλωση στην φάση deepsleep.

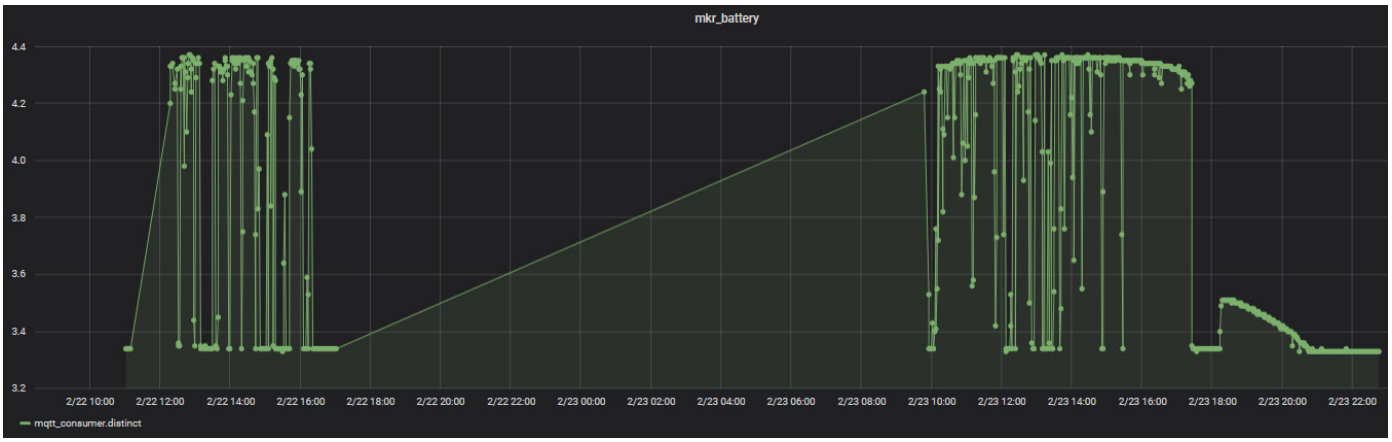
4.3.6 Φόρτιση - Αποφόρτιση μπαταρίας σε συνεχή λειτουργία με φωτοβολταϊκό πάνελ

- ***Για το Fipy:***



εικόνα 4.17: Συνεχής λειτουργία με φωτοβολταϊκό πάνελ για το Fipy

- ***Για το MKR1500:***



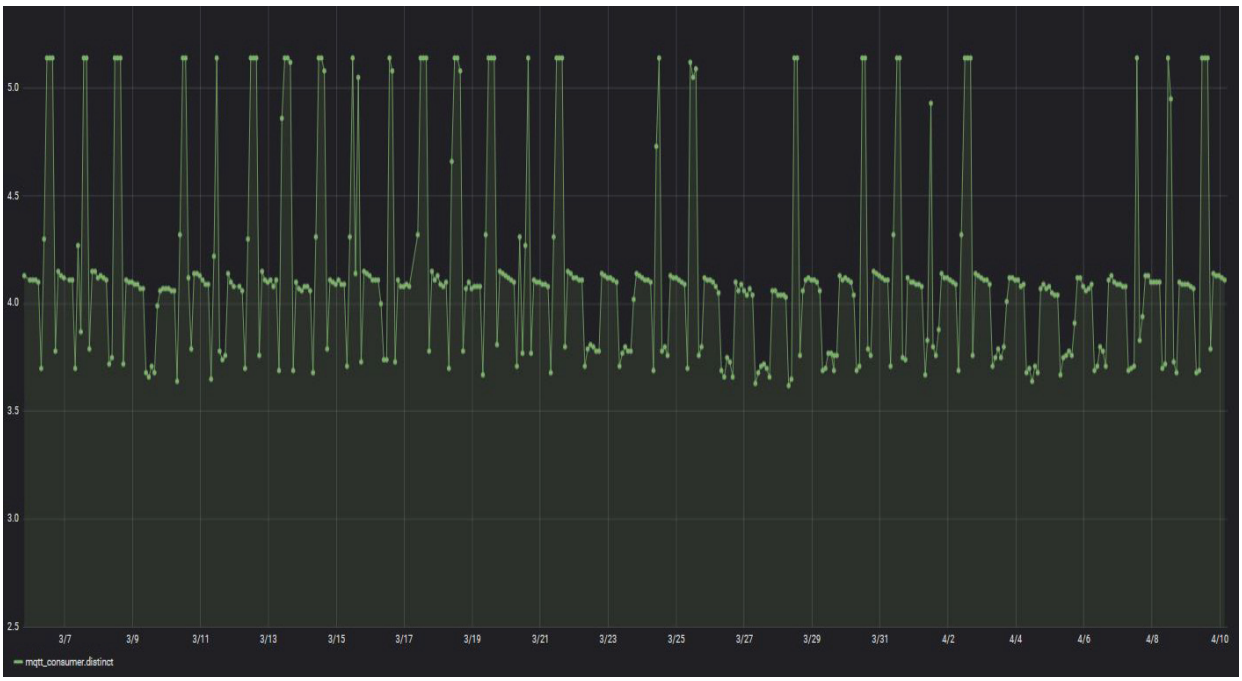
εικόνα 4.18: Συνεχής λειτουργία με φωτοβολταϊκό πάνελ για το MKR1500

Παρατηρήσεις

- Επιλέχθηκε η μέτρηση να γίνει με μη φορτισμένη μπαταρία, ώστε να μελετηθεί η δυνατότητα του φωτοβολταϊκού πάνελ να υποστηρίξει την λειτουργία του αισθητήρα, και να φορτίσει την μπαταρία.
- Παρατηρούμε ότι η τάση βρίσκεται σταθερά στα 4.3 V όσο το πάνελ δέχεται απευθείας ηλιακή ακτινοβολία (10:20 με 16:55).
- Όταν αυτή η απευθείας ηλιοφάνεια διακόπτεται από κάποιο παροδικό σύννεφο (μετρήσεις 19/02) είτε από έντονη συννεφιά (22-23/02) παρατηρούμε ότι η μετρούμενη τάση μπαταρίας πέφτει από την τάση φόρτισης (~4.3) στην πραγματική τάση της μπαταρίας εκείνη την στιγμή.

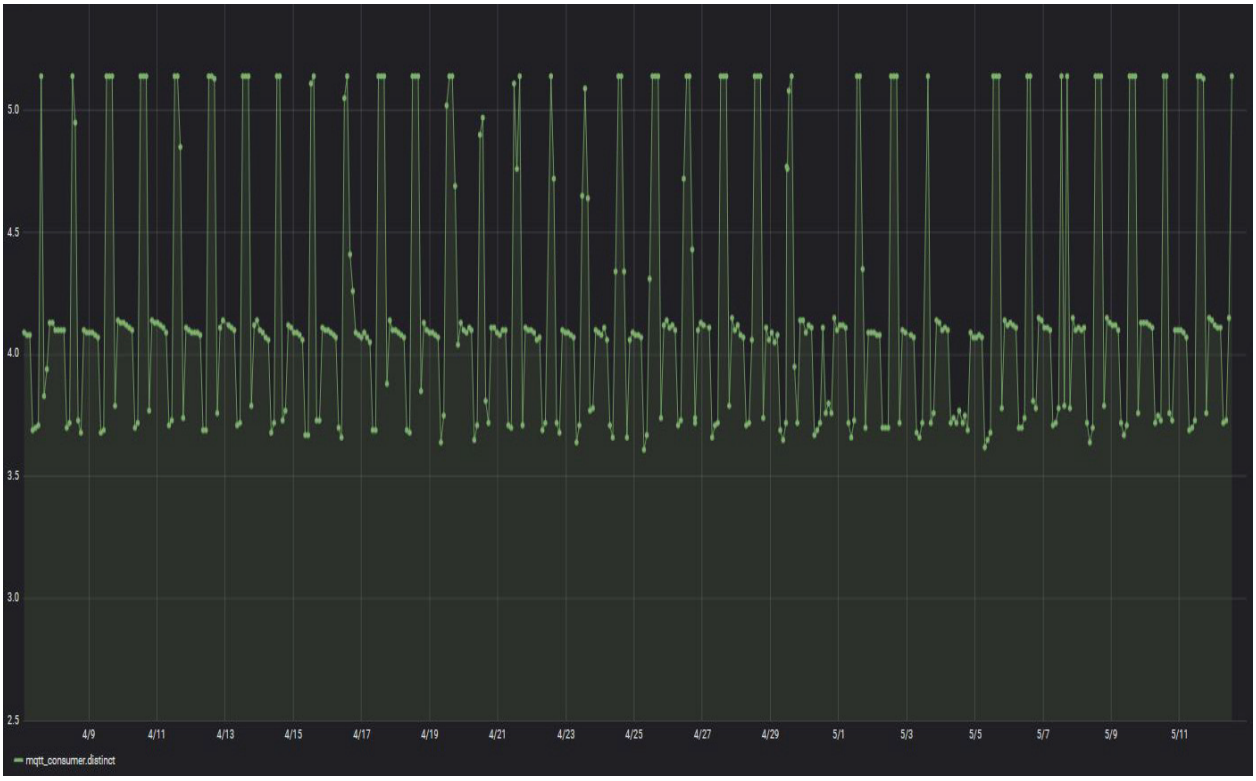
4.3.7 Φόρτιση - Αποφόρτιση μπαταρίας σε λειτουργία deepsleep με φωτοβολταϊκό πάνελ

- ***Για το Fipy:***



εικόνα 4.19: Λειτουργία deepsleep με φωτοβολταϊκό πάνελ για το Fipy

- ***Για το MKR1500:***



εικόνα 4.20: Λειτουργία deepsleep με φωτοβολταϊκό πάνελ για το MKR1500

Παρατηρήσεις

- Επιλέχθηκε η μέτρηση να γίνει σε συνθήκες μη ιδανικής ηλιοφάνειας (δεντροκάλυψη, κτίρια, κ.α.) ωστέ ο έλεγχος αποτελεσματικότητας του φωτοβολταικού πάνελ να γίνει σε πιο αντίξοες συνθήκες από αυτές στην κορυφή ενός κεραιοσυστήματος.
- Παρατηρούμε οτι με πλήρως φορτισμένη την μπαταρία, σε ημέρες και ώρε με ηλιοφάνεια, η φαινόμενη τάση μπαταρίας βρίσκεται στα ~5 Volt δηλαδή η τάση που δίνει το φωτοβολταϊκό πάνελ.
- Σε μέρες χωρίς ηλιοφάνεια, η τάση δεν κάνει τα αντίστοιχα peak, χωρίς όμως να επιρεάζεται η απρόσκοπτη λειτουργία του αισθητήρα.
- Ο ρυθμός μετρήσεων ήταν 1 μέτρηση / 2ώρες = 12 μετρήσεις ανά ημέρα, 12 φορές παραπάνω από την αναγκαία 1 μέτρηση την ημέρα.

Κεφάλαιο 5
Συμπεράσματα και προβλήματα

5.1 Απόκλιση (offset) μετρήσεων κλίσης

- Με βάση την διαφορά των τιμών μέγιστης-ελάχιστης απόκλισης, αλλά και της σχέσης τους με τον μέσο όρο της απόκλισης, διαπιστώνουμε ότι ο αισθητήρας Fipy είναι πιο ακριβής. Εκ των δύο προσανατολισμών, ο προσανατολισμός ως προς την μικρή πλευρά της πλακέτας (αλλαγή γωνίας ως προς το roll) παρέχει μικρότερη διακύμανση.

5.2 Επίδραση θερμοκρασίας

- Με βάση τα παραπάνω ο αισθητήρας που υλοποιείται με το Arduino MKR1500 είναι πιο ανθεκτικός στις μεταβολές της θερμοκρασίας. Όπως όμως καταγράφηκε και στην 1η παρατήρηση και ο αισθητήρας που υλοποιείται με το Fipy βρίσκεται εντός κανονικής λειτουργίας.

5.3 Μετρήσεις έντασης ρεύματος

- Με βάση τις καταναλώσεις που καταγράφονται για την φάση deepsleep και με μια ενδεικτική μπαταρία την Panasonic

NCR18650B – με 3.325 mAh, το MKR1500 μπορεί να παραμείνει σε κατάσταση deepsleep χωρίς κάποια πηγή τροφοδοσίας για 354 ώρες ή 14,75 ημέρες, ενώ το Fipy για 512 ώρες ή 21,3 ημέρες.

- Το Fipy έχει καλύτερη ενεργειακή λειτουργία από το MKR1500 γιατί υποστηρίζεται από μια ολοκληρωμένη λύση αισθητήρων, οικοδομημένη πάνω στην deepsleep λογική. Αντιθέτως ο αισθητήρας κλίσης του MKR1500 διασυνδέεται με I2C δίαυλο που χωρίς την ανάπτυξη των κατάλληλων βιβλιοθηκών, καταναλώνει ενέργεια ακόμη και στην συνθήκη deepsleep.

5.4 Αποφόρτιση μπαταρίας σε συνεχή λειτουργία χωρίς φωτοβολταϊκό πάνελ

- Όπως ήταν αναμενόμενο και από τις προηγούμενες καταγραφές κατανάλωσης, το Fipy έχει μικρότερη κατανάλωση και άρα μεγαλύτερη αυτονομία από το MKR1500.

5.5 Φόρτιση/Αποφόρτιση μπαταρίας σε συνεχή λειτουργία με φωτοβολταϊκό πάνελ.

- Το φωτοβολταϊκό πάνελ έχει την δυνατότητα να στηρίξει την αυτόνομη λειτουργία των αισθητήρων ακόμη και σε περιόδους χωρίς ηλιοφάνεια και ενώ λαμβάνουμε συνεχείς μετρήσεις.
- Ακόμη και στις περιπτώσεις ολικής αποφόρτισης της μπαταρίας (ένα σενάριο που το αντιμετωπίζουμε εδώ λόγω της συνεχούς λειτουργίας) το φωτοβολταϊκό πάνελ έχει την δυνατότητα να επαναφέρει τον αισθητήρα σε λειτουργία, διασφαλίζοντας περαιτέρω την αδιάλειπτη λειτουργία.

5.6 Φόρτιση/Αποφόρτιση μπαταρίας σε deepsleep λειτουργία με φωτοβολταϊκό πάνελ.

- Το φωτοβολταϊκό πάνελ έχει την δυνατότητα να στηρίξει την αυτόνομη λειτουργία των αισθητήρων ακόμη και σε περιόδους χωρίς ηλιοφάνεια και ενώ λαμβάνουμε ιδιαίτερα συχνές μετρήσεις.
- Οι αισθητήρες επιδικνύουν γενική σταθερότητα τόσο ως προς τους κύκλους φόρτισης αποφόρτισης, όσο και ως προς την γενική τους λειτουργία.

5.7 Προβλήματα συνδεσιμότητας

Τόσο το MKR1500 όσο και το Fipy εμφανίζουν σε μη περιοδικά διαστήματα ζητήματα αδυναμίας εγγραφής (register) στο δίκτυο NB-IoT. Αυτό οδηγεί το πρόγραμμα στην εκτέλεση μίας ατέρμονης λούπας χωρίς προοπτική σύνδεσης.

Επίλυση

Για την επίλυση υλοποιήθηκε στις δύο πλατφόρμες IoT μια υπορουτίνα χρονισμού επιτήρησης (WatchDog Timer – WDT). Η ρουτίνα WDT καλείτε με συγκεκριμένο χρονικό όρισμα, και πρέπει να ανατροφοδοτείται ανά τακτά διαστήματα. Αν αυτό δεν γίνει τότε επανεκκινεί την συσκευή μας. Έτσι ορίστηκε ότι σε περίπτωση που η διαδικασία σύνδεσης υπερβεί τα 2 λεπτά (ένα εύλογο και γενναιόδωρο χρονικό διάστημα) τότε η συσκευή επανεκκινεί.

Προβληματική (MKR1500)

Η σειρά συσκευών MKR της Arduino είναι σχετικά καινούργια και υλοποιείται με την χρήση επεξεργαστών αρχιτεκτονικής ARM (SAMD21 Cortex-M0+ 32bit low power). Αυτό συνεπάγεται ότι δεν υπάρχει

βιβλιοθήκη για WDT ειδικά προσαρμοσμένο στο MKR1500. Έτσι η γενική λύση κώδικα WDT που χρησιμοποιήθηκε δημιουργεί σύγκρουση (conflict) στην χρήση του εσωτερικού ρολογιού του επεξεργαστή, και αποτρέπει την παράλληλη χρήση της διαδικασίας WDT με την διαδικασία deepsleep.

Καταληκτικά

Με βάση τα συμπεράσματα που προκείπτουν από τα πειράματα, προκύπτει οτι και οι δύο αισθητήρες (Fipy και MKR1500)είναι κατάλληλοι για τον σκοπό της διπλωματικής. Παρόλα αυτά σε διάφορα πεδία, όπως η κατανάλωση (κανονική και deepsleep), η αποφόρτιση μπαταρίας και το γωνιακό offset το Fipy υπερτερεί. Παράλληλα η δυνατότητα του Fipy για λειτουργία Watchdog ταυτόχρονα με την λειτουργία Deepsleep, δίνει στο Fipy ένα σημαντικό πλεονέκτημα σε ένα καθοριστικό πεδίο, αυτό της σταθερότητας λειτουργίας. Έτσι στο ιστό θα επιλέξουμε την τοποθέτηση ενός αισθητήρα γωνίας βασιμένον στο Fipy.

Κεφάλαιο 6
Προεκτάσεις

Υποστήριξη IPv6

Η βασική σκέψη γύρω από μια μελλοντική επέκταση του συστήματος, αφορά την υποστήριξη για IPv6 συνδεσιμότητα. Καθώς οι IoT συσκεύες αυξάνονται εκθετικά, η δεξαμενή (pool) δημόσιων IPv4 διευθύνσεων (2^{32} = 4,294,967,296 διευθύνσεις) δεν μπορεί πλέον να καλύψει την ζήτηση. Το IPv6 με τις 2^{128} διευθύνσεις δίνει την απάντηση στο παραπάνω πρόβλημα. Άλλωστε όλη η λογική της εμπορικής διάθεσης του Narrow Band IoT κοιτάει προς αυτή την κατεύθυνση με τις εμπορικά διαθέσιμες NB-IoT SIM της Κοσμοτέ να υποστηρίζουν μόνο IPv6. Προς αυτή την κατεύθυνση χρειάζεται να γίνει προσπάθεια προς την εξέλιξη βιβλιοθηκών και firmware για τους αισθητήρες που να υποστηρίζουν IPv6, καθώς και σχεδιασμός της backend υποδομής (routers, mqtt, databases, κ.α.) ώστε να υποστηρίξει αυτή την μετάβαση.

Αξιοποίηση των δεδομένων/ Data Analytics

Προφανώς ένα σημαντικό τμήμα των μελλοντικών επεκτάσεων κοιτάει προς την αξιοποίηση των δεδομένων που συλλέγονται από τους αισθητήρες κλησης. Μελλοντική τοποθέτηση εκατοντάδων τέτοιων αισθητήρων, στην πλειοψηφία των τηλεπικοινωνιακών κεραιοσυστημάτων, θα μπορούσε να παρέχει χρήσιμα δεδομένα σε τεχνικούς και εταιρίες. Τα δεδομένα αυτά συλλεγόμενα από την υποδομή που χτίσαμε και με την κατάλληλη επεξεργασία θα μπορούσαν να οδηγήσουν στον

εξορθολογισμό και την βελτίωση των πρωτοκόλλων συντήρησης των κεραιοσυστημάτων εξοικονόμοντας πόρους και μειώνοντας τον κίνδυνο εργατικών ατυχημάτων. Παράλληλα θα οδηγούσαν στην διαμόρφωση τεχνικών προδιαγραφών (αντοχές υλικών, επιλογή προσανατολισμού κ.α.) στην βάση πραγματικών πειραματικών δεδομένων.

Υλικοτεχνικά

- ***Προσαρμογή του φωτοβολταϊκού πάνελ.*** Λόγω του σχήματος του πάνελ το οποίο δεν διαθέτει οπές για υποδοχή σπειρωμάτων, η στερέωση του φωτοβολταϊκού πάνελ πάνω στο κουτί αποδεικνύεται δύσκολη. Η λύση την οποία χρησιμοποιούμε με την χρήση κόλλας υψηλών προδιαγραφών, δεν μας εξασφαλίζει σταθερότητα σε ένα περιβάλλον όπου το πάνελ θα δέχεται διαρκώς τάσεις από διαφορετικές κατευθύνσεις.
- ***Στιβαρότητα του κουτιού.*** Η επιλογή του κουτιού έγινε με γνώμονα τα χαρακτηριστικά στεγανοποίησης αλλά και την ευκολία εύρεσης και αγοράς. Όταν επιδιώχθηκε να τοποθετηθεί σε πραγματικό περιβάλλον δοκιμής, η κατασκευή δεν άντεξε τις περιβαλλοντικές καταπονήσεις. Σημαντική μελλοντική προέκταση θα ήταν ο σχεδιασμός και η υλοποίηση μιας υδατοστεγούς συσκευασίας εκ του μηδενός. Αυτό θα έλυνε τα ζητήματα στιβαρότητας, αλλά θα μπορούσε να δώσει απαντήσεις και στα ζητήματα προσαρμογής της λύσης πάνω σε ιστούς και κεραίες διαφορετικών μεγεθών και σχημάτων.
- ***Προσαρμογή πάνω στην κεραία.*** Οι περισσότεροι τύποι κεραιών δεν έχουν προβλέψεις για την εύκολη προσαρμογή του συστήματός μας. Καταφέραμε να τοποθετήσουμε την λύση σε μια κεραία τετράγωνου σχήματος αλλά αυτό δεν είναι εφικτό για κάθε σχήμα και μέγεθος κεραίας.

Παράρτημα
Πηγαίος Κώδικας

Πηγαίος κώδικας Fipy

boot.py

```
# boot.py -- run on boot-up

import pycom
import machine
import time
import socket
import binascii
import ujson                                # for json

from mqtt import MQTTClient                 # for MQTT client
from network import LTE                     # for NB-IoT connectivity
from lib.LIS2HH12 import LIS2HH12           # for tilt
from pysense import Pysense
from SI7006A20 import SI7006A20             # for temperature
from machine import WDT                     # watchdog

pycom.heartbeat(False)                  # disabling the blue led ping
```

main.py

```
wdt = WDT(timeout=20000) # enabling watchdog with a timeout of 20 seconds

# Getting metrics
print("getting the metrics")
# for pitch
acc = LIS2HH12()
i = 0
pitch_total = 0
roll_total = 0

# for temperature
si = SI7006A20()
temp_total = 0

# for battery
py = Pysense()
batt_total = 0

while i < 50 :
  wdt.feed()
  i += 1
  pitch = acc.pitch()
  roll = acc.roll()
  temp = si.temperature()
  battery = py.read_battery_voltage()
  pitch_total = pitch_total + pitch
```

```
  roll_total = roll_total + roll
  temp_total = temp_total + temp
  batt_total = batt_total + battery
  time.sleep_ms(50)

pitch_final = pitch_total / 50
roll_final = roll_total / 50
temp_final = temp_total / 50
batt_final = batt_total / 50

battery_percentage = round(batt_final*100/4.2,1)
wdt.feed()

# Creating the JSON message

application_id = “nbiot_tilt”
site_id = “ote_academy”
room_id = “room_106”
device_id = “fipy_1”

dict = {}
dict[“application_id”] = application_id
dict[“site_id”] = site_id
dict[“room_id”] = room_id
dict[“device_id”] = device_id
dict[“pitch”] = pitch_final
dict[“roll”] = roll_final
dict[“battery”] = round(batt_final,2)
dict[“battery_percentage”] = battery_percentage
```

```
dict[“temperature”] = temp_final

encoded = ujson.dumps(dict)
print(encoded)

wdt.feed()

# Connecting to NB-IoT network
lte = LTE()
if not lte.isattached():
  lte.attach(band=20, apn=”iot”)
i = 0
while not lte.isattached():
  i += 1
  print(“waiting to attach ...”)
  time.sleep(0.25)
  pycom.rgbled(0x7f0000) # red
  time.sleep(0.25)
  pycom.rgbled(0x007f00) # green

pycom.rgbled(0x000000) # no led
wdt.feed()
if not lte.isconnected():
  lte.connect()    # start a data session and obtain an IP address
while not lte.isconnected():
  print(“is not connected”)
  time.sleep(0.25)
print(“is connected”)
time.sleep(1)
```

```
# Setting up MQTT
def sub_cb(topic, msg):
  print(msg)

client = MQTTClient("device_id", "*.*.*.*",port=41883, user="****" ,pass-
word="********")
client.set_callback(sub_cb)
client.connect()
client.subscribe(topic="**********")
wdt.feed()

# Sending JSON to MQTT

print("Sending to MQTT")
time.sleep(1)
client.publish(topic="**********", msg=encoded)
time.sleep(2)
print("Going to sleep")
time.sleep(1)
wdt.feed()

# De-attaching and deepsleep

lte.disconnect()
lte.dettach(reset=False)
while lte.isattached():
  print("is attached")
  time.sleep(0.25)
machine.deepsleep(2*60*60*1000)       # deepsleep for 2 hours
```

Πηγαίος κώδικας MKR1500

```
#include <ArduinoLowPower.h>

#ifndef _BV
#define _BV(n) (1<<(n))
#endif


#include <Arduino.h>
#include <MKRNB.h>
#include <WDTZero.h>
WDTZero MyWatchDoggy; // Define WDT

#include <ArduinoJson.h> // JSON library. Remember to change the
permmited message size inside PubSub library
#include <PubSubClient.h> //change the max packet size inside the Pub-
Sub library [#define MQTT_MAX_PACKET_SIZE 512]

#if defined(ARDUINO_ARCH_SAMD)
// for Zero, output on USB Serial console, remove line below if using pro-
gramming port to program the Zero!
  #define Serial SerialUSB
#endif

//######################### Temperature part #################
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN 6   // Digital pin connected to the DHT sensor
```

```
#define DHTTYPE  DHT22   // DHT 22 (AM2302)
DHT_Unified dht(DHTPIN, DHTTYPE);

//######################### Temperature part end ##############

// I2Cdev and MPU6050 must be installed as libraries, or else the .cpp/.h
files
// for both classes must be in the include path of your project
#include “I2Cdev.h”

#include “MPU6050_6Axis_MotionApps20.h”
//#include “MPU6050.h” // not necessary if using MotionApps include
file

// Arduino Wire library is required if I2Cdev I2CDEV_ARDUINO_
WIRE implementation
// is used in I2Cdev.h
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
  #include “Wire.h”
#endif

// class default I2C address is 0x68
// specific I2C addresses may be passed as a parameter here
// AD0 low = 0x68 (default for SparkFun breakout and InvenSense evalu-
ation board)
// AD0 high = 0x69
MPU6050 mpu;
//MPU6050 mpu(0x69); // <-- use for AD0 high

#define INTERRUPT_PIN 2 // use pin 2 on Arduino Uno & most boards

// MPU control/status vars
bool dmpReady = false; // set true if DMP init was successful
```

```
uint8_t mpuIntStatus;  // holds actual interrupt status byte from MPU
uint8_t devStatus;   // return status after each device operation (0 = success, !0 = error)
uint16_t packetSize;  // expected DMP packet size (default is 42 bytes)
uint16_t fifoCount;   // count of all bytes currently in FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer

// orientation/motion vars
Quaternion q;      // [w, x, y, z]     quaternion container
VectorInt16 aa;     // [x, y, z]accel sensor measurements
VectorInt16 aaReal;   // [x, y, z]gravity-free accel sensor measurements
VectorInt16 aaWorld;  // [x, y, z]world-frame accel sensor measurements
VectorFloat gravity;  // [x, y, z]gravity vector
float euler[3];     // [psi, theta, phi]  Euler angle container
float ypr[3];      // [yaw, pitch, roll]  yaw/pitch/roll container and gravity vector

// packet structure for InvenSense teapot demo
uint8_t teapotPacket[14] = { '$', 0x02, 0,0, 0,0, 0,0, 0,0, 0x00, 0x00, '\r', '\n' };

// ================================================
// ===                  INTERRUPT DETECTION ROUTINE  ===
// ================================================

volatile bool mpuInterrupt = false;   // indicates whether MPU interrupt pin has gone high
void dmpDataReady() {
  mpuInterrupt = true;
}

// ================================================
// ===                         MKR NB part          ===
```

```
// ================================================


#define PINNUMBER “”

// initialize the library instance
NB nbAccess(true); // NB access: include a ‘true’ parameter for debug
enabled
GPRS gprsAccess; // GPRS access
NBClient nb_client; // Client service for TCP connection
//initialize mqtt

PubSubClient client(nb_client); // more are set at setup()

const char* mqtt_server= “193.218.97.145”;
const char* mqtt_user=”test”;
const char* mqtt_pass=”@cosmote@”;
char msg[150];// in order to receive mqtt
int n = 0;  //counter to verify normal operation

// messages for serial monitor response
String oktext = “OK”;
String errortext = “ERROR”;

/* SENSORS PARAMETERS */

String site_id = “penteli_east”;
String room_id =”antenna”;
String device_id =”mkr1500_5”;
String application_id =”tilt”;

const int numReadings = 2500;
float average_pitch = 0;  // the average
```

```
float average_roll = 0;  // the average
float battery_absolute = 0;

/*
// ================================================
// ===          SIGNAL STRENGTH CALC        ===
// ================================================

void scan_net() {

 float rssi_total=0;
 float rssi_db_total=0;
 int k = 0;
 Serial.println("Getting signal metrics ");
 scannerNetworks.begin();
 while (k<100) {
  delay(100);
  k=k+1;
  String temp1 = scannerNetworks.getSignalStrength();
  rssi_total = rssi_total + temp1.toInt();
  Serial.print(".");
  if (k%20 ==0) {
   Serial.print(" ");
   Serial.print(k);
   Serial.println("%");
  }
 }
 rssi = rssi_total/100;
 rssi_db = 2*rssi - 113 ;

 Serial.println();
 Serial.print("Signal strength is : ");
 Serial.print(rssi_db);
```

```
 Serial.println(“ db”);
 Serial.println();
}
*/

// ================================================
// ===        BATTERY CALCULATION        ===
// ================================================


void battery_calc() {

 long battery_raw = 0;
 float battery_median;
 int i = 0;

 Serial.println(“Getting battery metrics “);

 while (i<100) {
  i = i + 1;
  battery_raw = battery_raw + analogRead(ADC_BATTERY);
  delay(100);
  Serial.print(“.”);
  if (i%20 ==0) {
   Serial.print(“ “);
   Serial.print(i);
   Serial.println(“%”);
  }

 }
 Serial.println();
 battery_median = battery_raw/100;
```

```
 battery_absolute = (battery_median * 4.3)/1023;
 Serial.print(“Battery is : “);
 Serial.print(battery_absolute);
 Serial.println(“ V”);
 Serial.println();


}

// ================================================
// ===          START NB MODULE       ===
// ================================================

void start_module() {

 Serial.print(“Connecting NB IoT / LTE Cat M1 network...”);
 boolean notConnected = true;
 // Start GSM shield
 while(notConnected)
 {
  if ((nbAccess.begin(PINNUMBER) == NB_READY) & (gprsAccess.
attachGPRS(“iot”) == GPRS_READY)) {
   notConnected = false;
  }
  else {
   Serial.println(“Not connected”);
   delay(1000);
  }
 }

 Serial.println(“Connected to GPRS network”);
}
```

```
// ================================================
// ===        CONNECT TO MQTT          ===
// ================================================

void reconnect() {
 // Loop until we're reconnected
 digitalWrite(A1, LOW);
 while (!client.connected()) {
  start_module();
  Serial.print("Attempting MQTT connection...");
  // Attempt to connect
  if (client.connect((device_id + String(random(1, 100000))).c_str(),
mqtt_user,mqtt_pass)) {
   Serial.println("connected");
   digitalWrite(A1, HIGH);

   //subscribe for update instructions
   client.subscribe("ota_updates");

  } else {
   Serial.print("failed, rc=");
   Serial.print(client.state());
   Serial.println(" try again in 5 seconds");
   // Wait 5 seconds before retrying
   delay(5000);
   }
 }
}

// ================================================
// ===        CALLBACK FUNCTION          ===
// ================================================
```

```
void callback(char* topic, byte* payload, unsigned int length) {
 Serial.println("Callback");
 int i;
 for (i = 0; i < length; i++) {
  msg[i] = (char)payload[i];
 }
 msg[i] = '\0';
}

// ================================================
// ===       PUBLISHING JSON TO MQTT          ===
// ================================================

void pub_json(int l, float m, float n, float o, float p, float q) {

 StaticJsonDocument <300> root;

 root ["site_id"] = site_id;
 root ["room_id"] = room_id;
 root ["device_id"] = device_id;
 root ["application_id"] = application_id;
 root ["counter"] = l;
 root ["roll"] = m;
 root ["pitch"] = n;
 root ["battery"] = o;
 root ["signal_db"] = p;
 root ["temperature"] = q;


 serializeJsonPretty(root, Serial);
 Serial.println();
 char JSONmessageBuffer[300];
 serializeJson(root, JSONmessageBuffer);
```

```
 Serial.println("Sending message to MQTT topic..");
 Serial.println(JSONmessageBuffer);

 if (client.publish("monitor_testing", JSONmessageBuffer) == true) {
  Serial.println("Success sending message");
 } else {
  reconnect();
  Serial.println("failed sending message");
  pub_json(l,m,n,o);
 }
}

// ================================================
// ===        MPU6050 INITIAL SETUP         ===
// ================================================

void tilt_init() {

 // join I2C bus (I2Cdev library doesn't do this automatically)
 #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
 Wire.begin();
 Wire.setClock(400000); // 400kHz I2C clock. Comment this line if hav-
ing compilation difficulties
 #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FAST-
WIRE
 Fastwire::setup(400, true);
 #endif

 // initialize device
 Serial.println(F("Initializing I2C devices..."));
 mpu.initialize();
 pinMode(INTERRUPT_PIN, INPUT);
```

```
// verify connection
Serial.println(F("Testing device connections..."));
Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") : F("MPU6050 connection failed"));

// load and configure the DMP
Serial.println(F("Initializing DMP..."));
devStatus = mpu.dmpInitialize();

// supply your own gyro offsets here, scaled for min sensitivity
mpu.setXGyroOffset(220);
mpu.setYGyroOffset(76);
mpu.setZGyroOffset(-85);
mpu.setZAccelOffset(1788); // 1688 factory default for my test chip

// make sure it worked (returns 0 if so)
if (devStatus == 0) {
// turn on the DMP, now that it's ready
Serial.println(F("Enabling DMP..."));
mpu.setDMPEnabled(true);

// enable Arduino interrupt detection
Serial.print(F("Enabling interrupt detection (Arduino external interrupt "));
Serial.print(digitalPinToInterrupt(INTERRUPT_PIN));
Serial.println(F(")..."));
attachInterrupt(digitalPinToInterrupt(INTERRUPT_PIN), dmpDataReady, RISING);
mpuIntStatus = mpu.getIntStatus();

// set our DMP Ready flag so the main loop() function knows it's okay to use it
Serial.println(F("DMP ready! Waiting for first interrupt..."));
```

```
 Serial.println(“”);
 Serial.println(“”);
 dmpReady = true;

 // get expected DMP packet size for later comparison
 packetSize = mpu.dmpGetFIFOPacketSize();
 } else {
 // ERROR!
 // 1 = initial memory load failed
 // 2 = DMP configuration updates failed
 // (if it’s going to break, usually the code will be 1)
 Serial.print(F(“DMP Initialization failed (code “));
 Serial.print(devStatus);
 Serial.println(F(“)”));
 }

}

// ================================================
// ===       MPU6050 CALCULATION LOOP        ===
// ================================================

void tilt_calc(int repeats) {
 int readIndex = 0;
 float total_pitch = 0;
 float total_roll = 0;
 while (readIndex < repeats) {
  // if programming failed, don’t try to do anything
  if (!dmpReady) return;

  // wait for MPU interrupt or extra packet(s) available
  while (!mpuInterrupt && fifoCount < packetSize) {
    // if (mpuInterrupt && fifoCount < packetSize) {
```

```
    // try to get out of the infinite loop
    fifoCount = mpu.getFIFOCount();
  }

  // reset interrupt flag and get INT_STATUS byte
  mpuInterrupt = false;
  mpuIntStatus = mpu.getIntStatus();

  // get current FIFO count
  fifoCount = mpu.getFIFOCount();

  // check for overflow (this should never happen unless our code is too
inefficient)
  if ((mpuIntStatus & _BV(MPU6050_INTERRUPT_FIFO_OFLOW_
BIT)) || fifoCount >= 1024) {
    // reset so we can continue cleanly
    mpu.resetFIFO();
    fifoCount = mpu.getFIFOCount();
    Serial.println(F("FIFO overflow!"));

  // otherwise, check for DMP data ready interrupt (this should happen
frequently)
  } else if (mpuIntStatus & _BV(MPU6050_INTERRUPT_DMP_INT_
BIT)) {
    // wait for correct available data length, should be a VERY short wait
    while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();

    // read a packet from FIFO
    mpu.getFIFOBytes(fifoBuffer, packetSize);

    // track FIFO count here in case there is > 1 packet available
    // (this lets us immediately read more without waiting for an interrupt)
    fifoCount -= packetSize;
```

```
   mpu.dmpGetQuaternion(&q, fifoBuffer);

   float x = q.x;
   float y = q.y;
   float z = q.z;
   float w = q.w;

   float roll = atan2(2*(z*y+w*x),1-2*(x*x+y*y))* 180/M_PI;
   float pitch = asin(2*(y*w-x*z))* 180/M_PI;

   total_pitch = total_pitch + pitch;
   total_roll = total_roll + roll;
   readIndex = readIndex + 1;
  }
 }
 average_pitch = (total_pitch / repeats);
 average_roll = (total_roll / repeats);

}

// ================================================
// ===          INITIAL SETUP          ===
// ================================================

void setup() {

 pinMode(5, OUTPUT); // using a digital pin as a controlled ground
 digitalWrite(5, HIGH);
 // initialize serial communication
 // (115200 chosen because it is required for Teapot Demo output, but it's
 // really up to you depending on your project)
 Serial.begin(115200);
```

```
 delay(20*1000);
 Serial.println("here it comes");
 //Serial.print("\nWDTZero-Demo : Setup Soft Watchdog at 2M inter-
val");
 //MyWatchDoggy.attachShutdown(myshutdown);
 //MyWatchDoggy.setup(WDT_SOFTCYCLE2M); // initialize
WDT-softcounter refesh cycle on 2m interval
 client.setServer(mqtt_server, 41883);
 client.setCallback(callback);
 tilt_init();
 //MyWatchDoggy.clear(); // refresh wdt - before it loops
 start_module();
 //MyWatchDoggy.clear(); // refresh wdt - before it loops
 Serial.end();
}

// ================================================
// ===        MAIN PROGRAM LOOP         ===
// ================================================

void loop() {

 pinMode(5, OUTPUT); // using a digital pin as a controlled ground
 digitalWrite(5, HIGH);
 Serial.begin(115200);
 tilt_calc(numReadings);
 battery_calc();
 scan_net()

 //MyWatchDoggy.clear(); // refresh wdt - before it loops
 Serial.print("Pitch : ");
 Serial.print(average_pitch);
```

```
Serial.print("\tRoll : ");
Serial.println(average_roll);
reconnect();
//MyWatchDoggy.clear(); // refresh wdt - before it loops
delay(500);
client.loop();

// Temperature calculation
sensors_event_t event;
dht.temperature().getEvent(&event);
temp=event.temperature;

if (n!=0) {
 pub_json(n,average_pitch,average_roll,battery_absolute, rssi_db,temp);
 Serial.println("going to sleep");
 Serial.end();
 n=n+1;
 digitalWrite(5, LOW);
 LowPower.deepSleep(2*60*60*1000);
 //MyWatchDoggy.clear(); // refresh wdt - before it loops
}
n=n+1;
//MyWatchDoggy.clear(); // refresh wdt - before it loops
delay(500);
n=n+1;
}

void myshutdown()
{
Serial.print("\nWe gonna shut down ! ...");
}
```

Βιβλιογραφία

[1] F. Athley and M. N. Johansson, “Impact of Electrical and Mechanical Antenna Tilt on LTE Downlink System Performance,” 2010 IEEE 71st Vehicular Technology Conference, Taipei, 2010, pp. 1-5.

[2] https://en.wikipedia.org/wiki/Internet_of_things

[3] A state of the art review on the Internet of Things (IoT) History, Technology and fields of deployment. P. Suresh J. Vijay Daniel ,Dr.V.Parthasarathy

[4] https://fardapaper.ir/mohavaha/uploads/2018/02/Fardapaper-A-state-of-the-art-review-on-the-Internet-of-Things-IoT-history-technology-and-fields-of-deployment.pdf

[5] Why the Internet of Things is called Internet of Things: Definition, history, disambiguation - https://iot-analytics.com/internet-of-things-definition/

[6] The Internet of Things:How the Next Evolution of the Internet Is Changing Everything. https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[7] https://steemit.com/iot/@jibi333/where-does-the-iot-go-next

[8] https://www.edureka.co/blog/iot-applications/

[9] https://www.quora.com/What-is-IoT-architecture

[10] R. Ratasuk, N. Mangalvedhe, Y. Zhang, M. Robert and J. Koskinen,

"Overview of narrowband IoT in LTE Rel-13," 2016 IEEE Conference on Standards for Communications and Networking (CSCN), Berlin, 2016, pp. 1-7, doi: 10.1109/CSCN.2016.7785170.

[11] R. Ratasuk, N. Mangalvedhe, Z. Xiong, M. Robert and D. Bhatoolaul, "Enhancements of narrowband IoT in 3GPP Rel-14 and Rel-15," 2017 IEEE Conference on Standards for Communications and Networking (CSCN), Helsinki, 2017, pp. 60-65, doi: 10.1109/CSCN.2017.8088599.

[12] R. Ratasuk, N. Mangalvedhe, J. Kaikkonen and M. Robert, "Data Channel Design and Performance for LTE Narrowband IoT," 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), Montreal, QC, 2016, pp. 1-5, doi: 10.1109/VTCFall.2016.7880951.

[13] X. Lin, A. Adhikary and Y. -. Eric Wang, "Random Access Preamble Design and Detection for 3GPP Narrowband IoT Systems," in IEEE Wireless Communications Letters, vol. 5, no. 6, pp. 640-643, Dec. 2016, doi: 10.1109/LWC.2016.2609914.

[14] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. Navarro-Ortiz and J. M. Lopez-Soler, "Narrowband IoT Data Transmission Procedures for Massive Machine-Type Communications," in IEEE Network, vol. 31, no. 6, pp. 8-15, November/December 2017, doi: 10.1109/MNET.2017.1700081.

[15] NarrowBand-IoT Performance Analysis for Healthcare Applications HassanMalik, Muhammad MahtabAlam, Yannick LeMoullec,Alar Kuusik

[16] https://www.arduino.cc/en/Guide/MKRNB1500

[17] https://docs.pycom.io/gitbook/assets/specsheets/Pycom_002_Specsheets_FiPy_v2.pdf

[18] https://docs.pycom.io/datasheets/boards/pysense/

[19] Omar SEFRAOUI, Mohammed AISSAOUI, Mohsine ELEULDJ; "OpenStack: Toward an Open-Source Solution for Cloud Computing", International Journal of Computer Applications (0975 -8887)Volume 55 -No. 03, October 2012

[20] Ken pepple july 2011.Deploying OpenStack. OReillyMedia, Inc., 1005 Gravenstein Highway North, Sebastopol,CA 95472

[21] Mosquitto: server and client implementation of the MQTT protocol – The Journal of Open Source Software, Roger A Light

[22] What is tick stack.
https://www.thoughtworks.com/radar/platforms/tick-stack

[23] Introduction to InfluxData's InfluxDB and TICK Stack
https://www.influxdata.com/blog/introduction-to-influxdatas-influxdb-and-tick-stack/