



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

**Πρόγραμμα Μεταπτυχιακών Σπουδών**

**Επιστήμη και Τεχνολογία της Πληροφορικής και των Υπολογιστών**

**Ειδίκευση Υλικού και Υπολογιστικών Συστημάτων**

**Διπλωματική Εργασία**

**Προγραμματιζόμενος Έλεγχος Ασύρματου**

**Δικτύου Αισθητήρων**

**Κωνσταντίνος Δ. Γιδαράκος**

A.M. 18015

**Δημήτριος Κ. Κυριακούλης**

A.M. 18016

Εισηγητής:

**Διονύσης Κανδρής,**

Καθηγητής

**Αθήνα, Ιούνιος 2021**



**UNIVERSITY OF WEST ATTICA**  
**FACULTY OF ENGINEERING**  
**DEPARTMENT OF INFORMATICS & COMPUTER ENGINEERING**  
**POSTGRADUATE COURSE: SCIENCE & TECHNOLOGY OF**  
**INFORMATICS AND COMPUTERS**

**Diploma Thesis**

**Programmable Control of a Wireless Sensor Network**

<b>Konstantinos D. Gidakos</b> Registration Number: 18015	<b>Dimitris K. Kyriakoulis</b> Registration Number: 18016
--	--

Supervisor:  
**Dionisis Kandris**  
Professor

**Athens, June 2021**



**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Τίτλος:**

**Προγραμματιζόμενος Έλεγχος Ασύρματου Δικτύου Αισθητήρων**

**Κωνσταντίνος Δ. Γιδαράκος**

**Δημήτριος Κ. Κυριακούλης**

**A.M. 18015**

**A.M. 18016**

**Εισηγητής:**

**Διονύσης Κανδρής**

**Εξεταστική Επιτροπή:**

**Διονύσης Κανδρής, Καθηγητής**

**Αντώνης Μπόγρης, Καθηγητής**

**Σταύρος Φατούρος, Αναπληρωτής Καθηγητής**

**Ημερομηνία εξέτασης: 29 Ιουνίου 2021**

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Διονύσης Κανδρής Καθηγητής	Αντώνης Μπόγρης Καθηγητής	Σταύρος Φατούρος Αναπληρωτής Καθηγητής
		



## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΩΝ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Οι κάτωθι υπογεγραμμένοι Κωνσταντίνος Γιδαράκος του Δημητρίου με αριθμό μητρώου 18015 και Δημήτριος Κυριακούλης του Κωνσταντίνου με αριθμό μητρώου 18016 φοιτητές του Προγράμματος Μεταπτυχιακών Σπουδών Επιστήμη και Τεχνολογία της Πληροφορικής και των Υπολογιστών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνουμε ότι:

«Είμαστε συγγραφείς αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχαμε για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες κάναμε χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνουμε ότι αυτή η εργασία έχει συγγραφεί από εμάς αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μας, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μας ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μας».

Οι Δηλούντες

Κωνσταντίνος Δ. Γιδαράκος

Δημήτριος Δ. Κυριακούλης



## Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα καθηγητή μας κ. Διονύσιο Κανδρή για την εμπιστοσύνη του, την καθοδήγηση του και την υποστήριξη του καθ' όλη την διάρκεια της εκπόνησης της παρούσας Διπλωματικής Εργασίας καθώς και την γενικότερη στήριξή του κατά την διάρκεια των σπουδών μας στο ΠΜΣ.

Ένα μεγάλο ευχαριστώ θα θέλαμε να απευθύνουμε σε όλους τους καθηγητές μας του Προγράμματος Μεταπτυχιακών Σπουδών για το εξαιρετικό κλίμα που βιώσαμε κατά την διάρκεια του προγράμματος. Είναι όλοι εξαιρετικοί άνθρωποι και επιστήμονες και έχουμε μόνο θετικά στοιχεία αποκομίσει από την επαφή μας κατά την διάρκεια υλοποίησης του ΠΜΣ.

Τέλος θα θέλαμε να ευχαριστήσουμε τις οικογένειες μας για την αμέριστη στήριξη τους σε αυτή την δύσκολη προσπάθεια και για την κατανόηση τους για τον χρόνο που τους στερήσαμε.

Κωνσταντίνος Δ. Γιδαράκος

Δημήτριος Κ. Κυριακούλης





## Περίληψη

Τις τελευταίες δεκαετίες ο άνθρωπος έχει υιοθετήσει την τεχνολογία ασυρμάτων επικοινωνιών σε ένα ευρύ φάσμα εφαρμογών εκμεταλλευόμενος τα συγκριτικά της πλεονεκτήματα σε σχέση με άλλες μορφές επικοινωνίας. Μια μορφή προηγμένης χρήσης ασυρμάτων τεχνολογιών είναι και τα Ασύρματα Δίκτυα Αισθητήρων (ΑΔΑ) που έχουν αλλάξει σε πολλές περιπτώσεις τον τρόπο με τον οποίο ο άνθρωπος παρακολουθεί φαινόμενα του περιβάλλοντος του. Χρησιμοποιούνται σε ένα διευρυνόμενο πεδίο εφαρμογών και καθημερινά υιοθετείται η χρήση τους σε νέες δραστηριότητες.

Η παρούσα πτυχιακή εργασία πραγματεύεται την ανάπτυξη και τον προγραμματιζόμενο έλεγχο των ΑΔΑ. Ειδικότερα, γίνεται εισαγωγή στα ΑΔΑ και στη συνέχεια ειδική αναφορά στην ηλεκτρονική πλατφόρμα BBC micro:bit η οποία και επιλέχτηκε στο πλαίσιο αυτής της μεταπτυχιακής διατριβής για την ανάπτυξη ενός ΑΔΑ. Με χρήση αυτού του ΑΔΑ εκπονήθηκαν πειραματικές ακολουθίες που οδήγησαν στην εξαγωγή χρήσιμων συμπερασμάτων.

## Abstract

In recent decades, man has adopted wireless communication technology in a wide range of applications, taking advantage of its comparative advantages over other forms of communication. One form of advanced use of wireless technologies is the Wireless Sensor Networks (WSNs) which have changed in many cases the way people monitor phenomena in their environment. They are used in an expanding field of applications and their use in new activities is adopted daily.

This dissertation deals with the development and the programmed control of WSNs. In particular, the WSNs are introduced and then a special reference is made to the BBC micro:bit electronic platform, which was selected in the context of this postgraduate dissertation for the development of a WSN. Using this WSN, experimental sequences were prepared that led to the drawing of useful conclusions.



# Περιεχόμενα

Ευχαριστίες .....	iii
Περίληψη .....	v
Abstract .....	v
Λίστα εικόνων .....	ix
Λίστα πινάκων .....	x
Λίστα συντομεύσεων .....	xi
Κεφάλαιο 1: Ασύρματα Δίκτυα Αισθητήρων .....	1
1.1 Εισαγωγή.....	1
1.2 Δομικά στοιχεία ενός ΑΔΑ - Κόμβοι.....	2
1.2.1 Δημοφιλείς πλατφόρμες κόμβων .....	6
1.2.2 Λειτουργικά Συστήματα για WSN nodes .....	10
1.3 Αρχιτεκτονική ενός Ασύρματου Δικτύου Αισθητήρων .....	13
1.3.1 Πρωτόκολλα Επικοινωνίας σε ένα WSN .....	16
1.3.1.1 Φυσικό Επίπεδο .....	17
1.3.1.2 Επίπεδο Ζεύξης Δεδομένων .....	18
1.3.1.3 Επίπεδο Δικτύου .....	18
1.3.1.4 Επίπεδο Μεταφοράς .....	20
1.3.1.5 Επίπεδο Εφαρμογής.....	22
1.3.2 Πρότυπα WSN .....	23
1.3.2.1 IEEE 802.15.4 .....	23
1.3.2.2 ZigBee .....	26
1.3.2.3 Πρωτόκολλο Bluetooth και Bluetooth Low Energy (BLE) .....	29
1.4 Εφαρμογές Ασύρματων Δικτύων Αισθητήρων .....	32
Κεφάλαιο 2: Η πλατφόρμα BBC micro:bit.....	37
2.1. Εισαγωγή.....	37
2.2 Τι ακριβώς είναι η πλατφόρμα BBC micro:bit .....	37
<b>2.3 Χρήση του micro:bit ως πλατφόρμα για την ανάπτυξη εφαρμογών WSN.....</b>	<b>38</b>
<b>2.3.1 Τεχνικά χαρακτηριστικά - Το υλικό του micro:bit.....</b>	<b>38</b>
<b>2.3.2 Υποστήριξη λογισμικού.....</b>	<b>41</b>
<b>2.3.3 Υποστηριζόμενο λογισμικό - Διαδικασίες προγραμματισμού.....</b>	<b>43</b>
Κεφάλαιο 3: Υλοποίηση πειραματικών σεναρίων .....	45
3.1. Εισαγωγή.....	45
3.2. Υλοποίηση βασικών τρόπων επικοινωνίας .....	45

3.2.1. Υλοποίηση επικοινωνίας Broadcast.....	45
3.2.2. Υλοποίηση επικοινωνίας Multicast.....	48
3.2.3. Υλοποίηση επικοινωνίας Unicast.....	50
3.2. Διαδικασίες Βαθμονόμησης κόμβων (Nodes Calibration)	54
3.2.1. Υλοποίηση σεναρίου Mode A (καταγραφή τιμών/μετρήσεων) – Button A (micro:bit).....	54
3.2.2. Υλοποίηση σεναρίου Mode B (προσδιορισμός άγνωστης θέσης κόμβου) – Button B (micro:bit).....	56
3.2.3. Ανάλυση – επεξηγήσεις του κώδικα στον cluster head.....	56
3.2.4. Αρχικοποιήσεις μεταβλητών / αναμονή επιλογής Mode A ή B.....	57
3.2.5. Επιλογή Mode A – Επαναλήψεις nodes/distances – REPL/led-display .....	60
3.2.6. Επιλογή Mode A – Εκπομπή από κόμβο – Κώδικας κόμβων.....	62
3.2.7. Επιλογή Mode A – Λήψη μηνυμάτων .....	64
3.2.8. Επιλογή Mode A – Αποθήκευση μέσου όρου μετρήσεων.....	67
3.2.9. Επιλογή Mode A – Δημιουργία αρχείου μετρήσεων για κάθε κόμβο.....	69
3.2.10. Επιλογή Mode A – Τέλος κόμβων – Κλείσιμο αρχείου .....	70
3.2.11. Επιλογή Mode B – Ενημέρωση πίνακα μέσων όρων από text αρχείο .....	70
3.2.12. Επιλογή Mode B – Λήψη και εκτύπωση μηνύματος.....	72
3.2.13. Επιλογή Mode B – Προσδιορισμός θέσης – Εμφάνιση σε REPL.....	73
3.2.14. Επιλογή Mode B – Εμφάνιση σε led-display .....	75
3.3. WSN – Γραφήματα.....	77
3.3.1. Μέσοι όροι Μετρήσεων.....	77
Κεφάλαιο 4: Επίλογος .....	83
4.1. Εισαγωγή.....	83
4.2. Σύνοψη εργασίας.....	83
4.3. Προβλήματα και αντιμετώπιση .....	84
4.4. Παρατηρήσεις – Συμπεράσματα.....	85
4.5. Προτάσεις – Βελτιώσεις .....	86
ΠΑΡΑΡΤΗΜΑ Α .....	89
Πηγαίος κώδικας cluster head .....	89
Βιβλιογραφία - Αναφορές.....	92

## Λίστα εικόνων

Εικόνα 1: Κόμβοι αισθητήρων διασκορπισμένοι σε ένα πεδίο ενδιαφέροντος [3] ...	1
Εικόνα 2: WSN για την ανίχνευση ηφαιστειακής δραστηριότητας [6] .....	2
Εικόνα 3: Τομή ενός κόμβου WSN #[8] .....	3
Εικόνα 4: Επιλογές αρχιτεκτονικής κόμβου WSN [5] .....	5
Εικόνα 5 : MicaZ.....	6
Εικόνα 6: Berkley / Crossbow Telos.....	6
Εικόνα 7: Eyes / EyesIFXv2.....	7
Εικόνα 8:Zolertia Firefly .....	7
Εικόνα 9: Arduino .....	7
Εικόνα 10: BBC micro:bit .....	7
Εικόνα 11: Stargate.....	9
Εικόνα 12: Imote.....	9
Εικόνα 13: Διαστρωματοποιημένη Αρχιτεκτονική πρωτοκόλλων επικοινωνίας και επίπεδα διαχείρισης[29] .....	17
Εικόνα 14:Κίνηση δεδομένων σε ένα WSN[45].....	21
Εικόνα 15: Τοπολογίες Star, Peer-to-Peer και Cluster-Tree [47] .....	25
Εικόνα 16: Δομή του υπέρ - πλαισίου MAC [50].....	26
Εικόνα 17: Διαστρωμάτωση προτύπου Zigbee με το IEEE 802.15.4[53] .....	27
Εικόνα 18: Συσκευές που ορίζονται στο πρότυπο ZigBee [56] .....	28
Εικόνα 19: Δείγμα τοπολογίας multi-hop του προφίλ BLE Mesh [62] .....	32
Εικόνα 20:Κατηγοριοποίηση με βάση την παρακολούθηση [7].....	33
Εικόνα 21: Δημοφιλείς κατηγορίες εφαρμογών WSN[66].....	33
Εικόνα 22: Υποκατηγορίες στρατιωτικών εφαρμογών WSN[66].....	34
Εικόνα 23: Κατηγορίες εφαρμογών για αστικά περιβάλλοντα[66] .....	35
Εικόνα 24: Ο οικιακός υπολογιστής BBC Micro .....	37
Εικόνα 25: BBC micro:Bit (front).....	37
<b>Εικόνα 26 To MU offline Python Editor.....</b>	<b>44</b>
Εικόνα 27: Broadcast .....	46
Εικόνα 28: Multicast .....	48
Εικόνα 29: Radio Packet.....	51

Εικόνα 30: Unicast .....	51
Εικόνα 31: Mode-A led display .....	61
Εικόνα 32: Repl mode-A μήνυμα αρχής μετρήσεων .....	62
Εικόνα 33: Repl mod-A ενδεικτικές μετρήσεις .....	66
Εικόνα 34: Mod-A περιεχόμενα αρχείου μέσωσν όρωνcalibrates.txt.....	68
Εικόνα 35: Repl mod-A περιληπτικό μήνυμα μετρήσεων.....	68
Εικόνα 36: Περιεχόμενα αρχείου μετρήσεων 1ου κόμβου .....	70
Εικόνα 37: Repl mode-B πίνακας μέσωσν όρων .....	71
Εικόνα 38: Repl mode-B μήνυμα από κόμβο .....	72
Εικόνα 39: Repl mode -B υπολογισμός απόστασης κόμβου.....	74
Εικόνα 40: Repl mode-B Υπολογισμός απόστασης με δύο κόμβους ενεργούς.....	74
Εικόνα 41: Repl mode-B Υπολογισμός απόστασης με τρεις κόμβους ενεργούς.....	75
Εικόνα 42: Mode-B led display .....	76
Εικόνα 43: Ενδεικτικές εικόνες led display mode - B .....	77
Εικόνα 44 Κλειστό Γήπεδο Μπάσκετ.....	78
Εικόνα 45: Κλειστή Αίθουσα Εκδηλώσεων.....	78
Εικόνα 46:Κόμβος 1 κλειστό γήπεδο μπάσκετ.....	79
Εικόνα 47: Κόμβος 1 κλειστή αίθουσα εκδηλώσεων .....	80
Εικόνα 48: Κόμβος 2 κλειστό γήπεδο του μπάσκετ .....	80
Εικόνα 49:Κόμβος 2 κλειστή αίθουσα εκδηλώσεων.....	81
Εικόνα 50: Κόμβος 3 κλειστό γήπεδο μπάσκετ .....	81
Εικόνα 51: Κόμβος 3 κλειστή αίθουσα εκδηλώσεων .....	82

## Λίστα πινάκων

Πίνακας 1: Χαρακτηριστικά γνωστών κόμβων WSN low-end[12],[13],[14] .....	8
Πίνακας 2: Χαρακτηριστικά high-end γνωστών πλατφόρμων WSN[12] .....	9
Πίνακας 3: Γνωστά Λειτουργικά Συστήματα WSN .....	13
Πίνακας 4:Πρωτόκολλα δρομολόγησης στα WSN .....	19
Πίνακας 5:Πρωτόκολλα επιπέδου Μεταφοράς WSN [46].....	22
Πίνακας 6:Σύγκριση Bluetooth με BLE [58], [62] .....	30
Πίνακας 7:Πίνακας μετρήσεων τρέχοντος κόμβου .....	65

## Λίστα συντομεύσεων

WSN	Wireless Sensor Network
DoD	Department of Defense USA
DARPA	Defense Advanced Research Project Agency
NASA	National Aeronautics and Space Administration
MEMS	Micro – Electro – Mechanical - Systems
A/D	Analog to Digital
ASIC	Application Specific Integrated Circuit
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
SoC	System on a Chip
RF	Radio Frequency
CPU	Central Processing Unit
6LoWPAN	IPv6 over Low -Power Wireless Personal Area Networks
RPL	Routing Protocol for Low-Power and Lossy Networks
CoAP	Constrained Application Protocol
MQTT	Message Queuing Telemetry Transport
LPWAN	Low-Power Wide Area Network
BLE	Bluetooth Low Energy
IEEE	Institute of Electrical and Electronics Engineers
PHY	Physical
MAC	Medium Access Control
NWK	Network Layer
APL	Application Layer
LR-WPAN	Low Rate Wireless Personal Area Networks
ED	Energy Detection
LQI	Link Quality Indication,
CCA	Clear Channel Assessment
BBC	British Broadcasting Corporation
IDE	Integrated Development Environment
VPL	Visual Programming Language





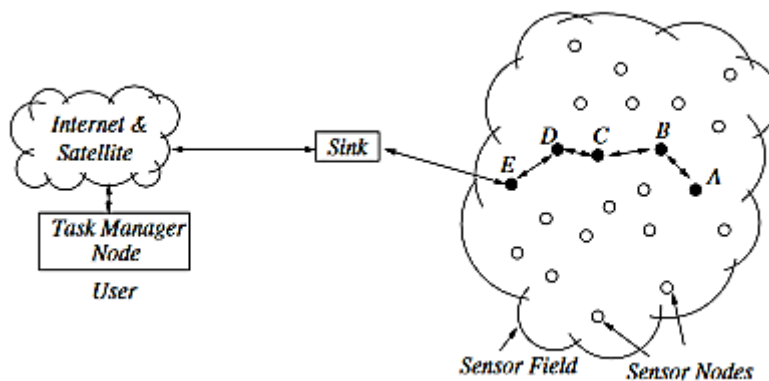
# Κεφάλαιο 1: Ασύρματα Δίκτυα Αισθητήρων

## 1.1 Εισαγωγή

Η ανάπτυξη των τεχνολογιών όλων των μορφών επικοινωνίας καθώς και οι σύγχρονες τεχνολογικές εξελίξεις σε διάφορους τομείς και ειδικότερα στον τομέα των μικρό ηλεκτρομηχανικών συστημάτων (MEMS) [1] έχουν καταστήσει εφικτή την δημιουργία ολοκληρωμένων κυκλωμάτων και συσκευών μικρού όγκου και χαμηλού κόστους παραγωγής που έχουν δυνατότητα ασύρματης επικοινωνίας σε μικρές αποστάσεις και χρησιμοποιούνται σε ένα μεγάλο εύρος εφαρμογών ανθρώπινης δραστηριότητας.

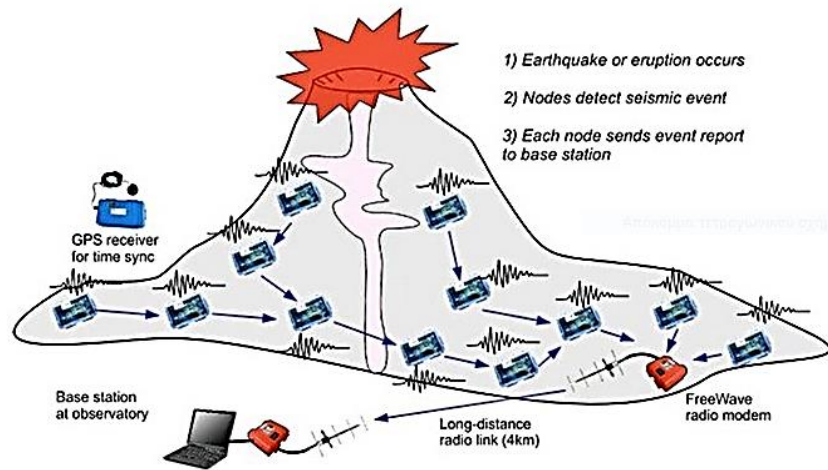
Αυτές οι συσκευές μπορεί να περιέχουν έναν ή περισσότερους αισθητήρες και ενεργοποιητές, μια διάταξη επεξεργασίας δεδομένων, μια διάταξη ασύρματης επικοινωνίας και κάποια διάταξη παροχής ενέργειας που συνήθως είναι μπαταρία ή/και άλλη διάταξη ανανεώσιμης πηγής ενέργειας. Μπορούν επίσης να επικοινωνούν μεταξύ τους δημιουργώντας ένα ασύρματο δίκτυο.

Ένα ασύρματο δίκτυο αισθητήρων- ΑΔΑ (Wireless Sensor Network - WSN) αποτελείται κατά κανόνα από έναν μεγάλο αριθμό κόμβων ασύρματων αισθητήρων που μπορούν να αναπτυχθούν σε μία περιοχή ενδιαφέροντος με σκοπό την παρακολούθηση, την ανίχνευση φυσικών μεγεθών και την συνεργατική μεταφορά της πληροφορίας σε κάποιο κεντρικό σημείο, κάποιο σταθμό βάσης. [3]



Εικόνα 1: Κόμβοι αισθητήρων διασκορπισμένοι σε ένα πεδίο ενδιαφέροντος [3]

Υπάρχουν υλοποιήσεις που η θέση του κάθε κόμβου είναι προσχεδιασμένη και συγκεκριμένη και παρομοίως συγκεκριμένη και η πυκνότητα του δικτύου. Αντίστοιχα υπάρχουν υλοποιήσεις που οι θέσεις των κόμβων έχουν μεγάλο βαθμό τυχαιότητας και μπορούν να αναπτυχθούν, για παράδειγμα με ρίψη από κάποιο ιπτάμενο μέσο, σε περιοχές δυσπρόσιτες και σε περιοχές όπου η ανθρώπινη παρουσία μπορεί να μην είναι δυνατή όπως για παράδειγμα παρακολούθηση ηφαιστειακής δραστηριότητας (volcano activity), διαχείριση καταστροφικής δραστηριότητας (disaster management) καθώς και άλλες παρόμοιες υλοποιήσεις.



Εικόνα 2: WSN για την ανίχνευση ηφαιστειακής δραστηριότητας [6]

Αυτό οδηγεί και στο συμπέρασμα ότι η λειτουργία τους, τα πρωτόκολλα επικοινωνίας και οι αλγόριθμοι τους πρέπει να διαθέτουν ένα ικανό βαθμό αυτοδιαχείρισης και αυτονομίας.

Από την άλλη αυτά τα χαρακτηριστικά ενός ασύρματου δικτύου δεδομένων είναι και η μεγαλύτερη πρόκληση στη σχεδίαση των τεχνολογιών λειτουργίας, επικοινωνιών, δρομολόγησης δεδομένων, διαχείρισης ενέργειας ώστε να είναι αυτές οι βέλτιστες δυνατές για την απόδοση και τη βιωσιμότητα ενός WSN.

Καταλαβαίνει κανείς με βάση τα προαναφερθέντα ότι το πεδίο εφαρμογής των ασύρματων δικτύων αισθητήρων είναι ευρύτατο.

Οι πρώτες εφαρμογές ασύρματων δικτύων αισθητήρων ήταν για στρατιωτική εφαρμογή αλλά στη συνέχεια χρησιμοποιήθηκαν σε περιβαλλοντικές εφαρμογές, στον τομέα υγείας, σε εφαρμογές για έξυπνα σπίτια- έξυπνες πόλεις, για βιομηχανικό έλεγχο, στις αγροτικές καλλιέργειες, στις μεταφορές και σε ένα συνεχώς διευρυνόμενο φάσμα εφαρμογών σε όλους τους τομείς ανθρώπινης δραστηριότητας.

Στο πρώτο αυτό κεφάλαιο γίνεται προσπάθεια περιγραφής των τεχνολογιών, των εφαρμογών, των πρωτοκόλλων που χρησιμοποιούνται καθώς και των πιο γνωστών πλατφορμών ανάπτυξης ασύρματων δικτύων αισθητήρων με ένα συνοπτικό αλλά και περιεκτικό τρόπο.

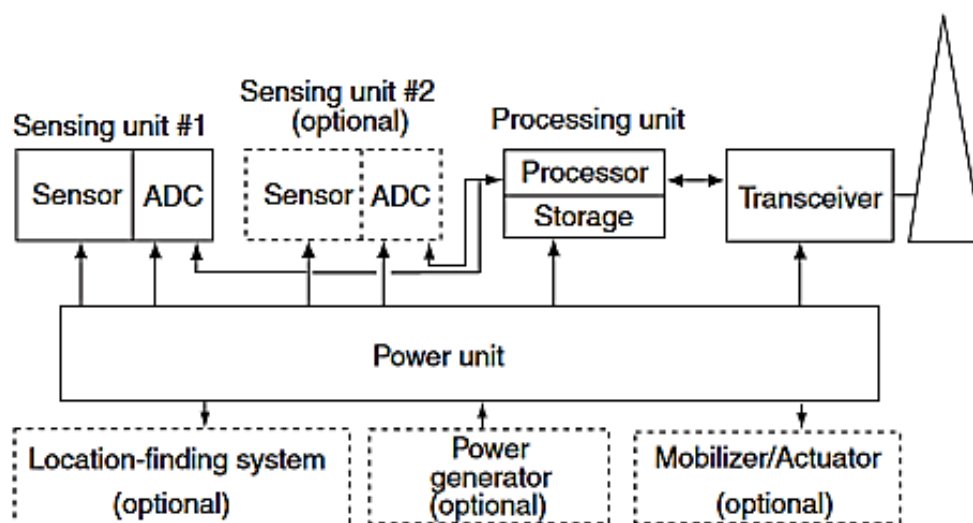
## 1.2 Δομικά στοιχεία ενός ΑΔΑ - Κόμβοι

Ένα ασύρματο δίκτυο αισθητήρων μπορεί να αποτελείται από μερικές δεκάδες έως και μερικές δεκάδες χιλιάδες κόμβους ασύρματων αισθητήρων. Παρόλο που γινόταν χρήση ασύρματων αισθητήρων για αρκετές δεκαετίες (από την δεκαετία '70) για στρατιωτικές εφαρμογές, παρακολούθηση ηφαιστειακής και σεισμικής δραστηριότητας κ.ο.κ, στην ανάπτυξη των ασύρματων κόμβων δικτύου αισθητήρων στη σύγχρονη μορφή της, ρόλο είχαν τα προγράμματα Smart Dust [4] της υπηρεσίας DARPA (Defense Advanced Research Projects

Agency) του DoD και Sensor Webs [5] της NASA στις ΗΠΑ στα τέλη της δεκαετίας '90, αρχές της δεκαετίας του 2000.

Ο κόμβος(node ή/και mote) είναι η συσκευή που αποτελεί το **δομικό** στοιχείο του ασύρματου δικτύου αισθητήρων. Πρόκειται για συσκευές που λόγω της φύσης των εφαρμογών WSN, πρέπει να έχουν περιορισμένο μέγεθος και κόστος και αντίστοιχα λόγω των χαρακτηριστικών αυτών περιορισμένους πόρους (υπολογιστική ισχύ, μνήμη, ενέργεια). [3]

Αναλαμβάνει την συλλογή την επεξεργασία και την αποστολή των δεδομένων/πληροφοριών που συλλέγει, λειτουργίες που επηρεάζουν άμεσα τους διαθέσιμους πόρους του.



Εικόνα 3: Τομή ενός κόμβου WSN #[8]

Στην εικόνα 3 φαίνεται το σχηματικό διάγραμμα ενός κόμβου WSN.

Παρατηρούμε:

- Την μονάδα αισθητήρων (sensing unit) και ενεργοποιητών (actuators) που μπορεί να περιλαμβάνει έναν ή περισσότερους αισθητήρες/ενεργοποιητές και έναν μετατροπέα αναλογικού σε ψηφιακό (A/D converter)
- Την μονάδα επεξεργασίας και αποθήκευσης δεδομένων (Processor / Storage Unit)
- Την μονάδα ασύρματης επικοινωνίας (Transceiver)
- Την μονάδα διαχείρισης ενέργειας (Power Unit)
- Επιπλέον υποσυστήματα που μπορεί να περιλαμβάνει προαιρετικά, όπως μια μονάδα GPS ή/και μια μονάδα κίνησης-κινητήρας και σύστημα κίνησης, ερπύστριες ή έλικες κλπ.- για κινητούς κόμβους (Mobilizer) ή ακόμα και κάποια διάταξη μετατροπής άλλης μορφής ενέργειας σε ηλεκτρική όπως ένα φωτοβολταϊκό (Power Generator).

Τα ακριβή χαρακτηριστικά του κάθε κόμβου καθορίζονται από τις απαιτήσεις της εφαρμογής για την οποία σχεδιάστηκε το WSN. Το είδος δηλαδή των αισθητήρων, τις δυνατότητες της μονάδας επεξεργασίας, την ενεργειακή απόδοση, το είδος της επικοινωνίας, το κόστος κ.ο.κ.

Η μονάδα αισθητήρων (*Sensing Unit*) μπορεί να περιλαμβάνει έναν ή περισσότερους αισθητήρες και ενεργοποιητές. Αισθητήρες χρησιμοποιεί ο άνθρωπος για πάρα πολλά χρόνια. Από ένα απλό θερμόμετρο που μετράει την θερμοκρασία περιβάλλοντος έως μικροαισθητήρες που βρίσκονται σε κάθε σύγχρονη ηλεκτρονική συσκευή.

Ο αισθητήρας είναι μία διάταξη που λαμβάνει ερεθίσματα από το περιβάλλον, ανιχνεύει τη μεταβολή ενός φυσικού μεγέθους και παράγει μια μετρήσιμη έξοδο με βάση αυτή την μεταβολή. (μετατρέπει μία μορφή ενέργειας σε μία άλλη). Στην περίπτωση μας μετατρέπει μία μη ηλεκτρική ποσότητα εισόδου  $E'$  σε μια ηλεκτρική ποσότητα εξόδου  $E$ . Αυτές οι μετρήσεις στέλνονται στην μονάδα επεξεργασίας που αναλαμβάνει την διαχείριση τους.

Αντιθέτως ένας ενεργοποιητής μπορεί να οριστεί ως μια συσκευή που μετατρέπει ένα ηλεκτρικό σήμα  $E$  σε μη ηλεκτρική ποσότητα  $E'$ . Ένας ενεργοποιητής δηλαδή μπορεί να εκτελέσει μια ενέργεια (π.χ. η ενεργοποίηση ενός φωτός, η ενεργοποίηση ενός συναγερμού, η απενεργοποίηση ενός συστήματος άρδευσης κ.λπ.) σε αντίδραση σε ένα συγκεκριμένο σήμα εισόδου.[2]

Στα [1], [2] γίνεται αναφορά στα διάφορα είδη αισθητήρων που μπορούν να χρησιμοποιηθούν για την παρατήρηση και καταγραφή ενός φαινομένου. Χαρακτηριστικά αναφέρονται ορισμένοι οι οποίοι είναι αρκετά δημοφιλείς σε υλοποιήσεις WSN: Αισθητήρες πίεσης, αδρανειακοί αισθητήρες, μαγνητόμετρα, θερμικοί αισθητήρες, χημικοί, αισθητήρες αερίου με βάση πολυμερές, ηλεκτροχημικοί αισθητήρες, βιοαισθητήρες, μοριακοί ειδικοί αισθητήρες, αισθητήρες με βάση τα κύτταρα, αισθητήρες για νευρικά συστήματα καθώς και αρκετοί ακόμα.

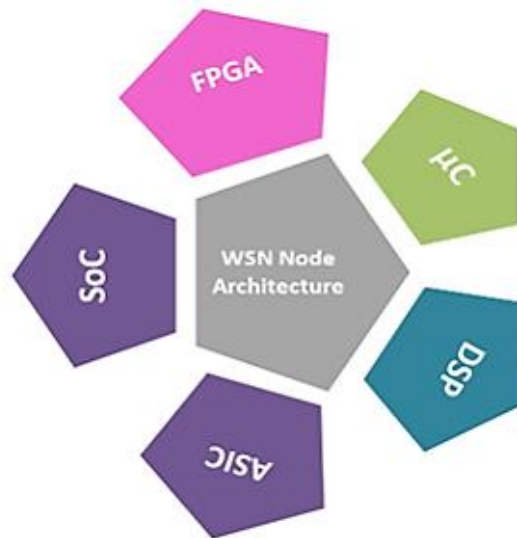
Η μονάδα επεξεργασίας δεδομένων (*Processing Unit*) συγκεντρώνει, επεξεργάζεται τα δεδομένα, καθώς και διαχειρίζεται όλα τα περιφερειακά υποσυστήματα. Η επιλογή της είναι κρίσιμη καθώς επηρεάζει την απόδοση, την κατανάλωση ενέργειας, το κόστος, το είδος της εφαρμογής.

Σύνηθες σε πληθώρα υλοποιήσεων είναι να χρησιμοποιούνται μικροελεγκτές διότι προσφέρουν αρκετά πλεονεκτήματα. Μερικά που θα μπορούσαν να αναφερθούν είναι οι δυνατότητες τους για διασύνδεση με μεγάλο αριθμό περιφερειακών, η χαμηλή κατανάλωση ενέργειας και το κόστος.

Επιπλέον, τμήμα του μικροελεγκτή μπορεί να τεθεί σε κατάσταση ύπνωσης (*sleep mode*), οπότε επιτυγχάνεται περαιτέρω μείωση της κατανάλωσης, ενώ χαρακτηρίζεται από γρήγορους χρόνους αφύπνισης

Υπάρχουν όμως υλοποιήσεις κόμβων με ASICs (*Application Specific Integrated Circuits*), DSPs (*Digital Signal Processing*), FPGAs (*Field Programmable Gate Arrays*), SoCs (*System on a Chip*), [9], [10]

Σε κάθε περίπτωση υπάρχουν πλεονεκτήματα και μειονεκτήματα και εξαρτάται από το είδος της εφαρμογής η επιλογή.



Εικόνα 4: Επιλογές αρχιτεκτονικής κόμβου WSN [5]

. Συνοπτικά θα μπορούσε να αναφερθεί πως οι μικροελεγκτές είναι επικρατέστεροι, αλλά υπόκεινται σε περιορισμούς στην υπολογιστική ισχύ και στη μνήμη, τα ASICs είναι αποδοτικότερα σε εφαρμογές που η βέλτιστη απόδοση είναι το ζητούμενο παράλληλα με χαμηλό ενεργειακό κόστος αλλά ανεβάζουν το κόστος παραγωγής, τα FPGAs είναι γρήγορα και υποστηρίζουν παράλληλο προγραμματισμό, αλλά έχουν αυξημένο κόστος και δυσκολία προγραμματισμού και οι DSPs έχουν καλύτερη απόδοση σε συγκεκριμένους αλγόριθμους αλλά είναι ακριβοί και μεγαλύτεροι σε μέγεθος. Περιπτώσεις συνδυαστικής επιλογής μεταξύ μικροελεγκτή και άλλων τεχνολογιών όπως FPGA επίσης μελετώνται. [9]

Για την αποθήκευση των δεδομένων χρησιμοποιείται μνήμη RAM, καθώς και μνήμες τεχνολογίας flash για λόγους που έχουν να κάνουν με το κόστος, την χωρητικότητα, την ταχύτητα και την ανθεκτικότητά της.

Ο πομποδέκτης (Transceiver) χρησιμοποιείται για την επικοινωνία των κόμβων. Η επιλογή του τρόπου μετάδοσης της πληροφορίας εξαρτάται από το μέσο που θα επιλεγεί για την υλοποίηση της ασύρματης επικοινωνίας, με συνηθέστερες επιλογές τις ραδιοσυχνότητες (RF), την οπτική επικοινωνία (infrared, laser) και τους υπερήχους (ultrasound). Επίσης μέσα όπως η μαγνητική επαγωγή χρησιμοποιούνται σε πολύ ειδικές περιπτώσεις συνήθως υπόγειων WSNs.

Από τις παραπάνω επιλογές κατά κύριο λόγο χρησιμοποιείται επικοινωνία ραδιοσυχνότητας (RF) ως πιο αποδοτική στις απαιτήσεις των περισσότερων εφαρμογών WSN. Η χρήση της εξασφαλίζει σχετικά μεγάλο εύρος χωρίς να απαιτείται οπτική επαφή αποστολέα και παραλήπτη, και είναι λιγότερο απαιτητική σε κατανάλωση πόρων που είναι βασικό για την βιωσιμότητα του δικτύου.

Ο πομποδέκτης μπορεί να έχει 4 καταστάσεις λειτουργίας:

- Κατάσταση Εκπομπής – Transmit State (ενεργοποίηση τμήματος εκπομπής και η κεραία εκπέμπει (ραδιοκύματα).
- Κατάσταση Λήψης – Receive State (ενεργοποίηση δέκτη και λήψη δεδομένων).
- Άερρη Κατάσταση – Idle State
- Κατάσταση Ύπνου – Sleep State

Κατά κανόνα την τροφοδοσία των κόμβων WSN αναλαμβάνουν υψηλής χωρητικότητας μακράς διάρκειας μπαταρίες (1-2 ή και περισσότερες), απλές ή επαναφορτιζόμενες. Ιδανικά κατά περίπτωση, μπορεί να συνδυαστεί με χρήση κάποιας μορφής παροχής ανανεώσιμης ενέργειας όπως φωτοβολταϊκή κυψέλη, ώστε να επεκταθεί η διάρκεια ζωής του WSN.

Σε μεγάλο αριθμό εφαρμογών, η ανάπτυξη ασύρματων δικτύων αισθητήρων γίνεται σε περιοχές και περιβάλλοντα που η ανθρώπινη παρουσία είναι δύσκολη ή ακόμα και αδύνατη, γεγονός που δεν αφήνει περιθώρια παρέμβασης σε περίπτωση κατανάλωσης των ενεργειακών αποθεμάτων των κόμβων.

Από τα παραπάνω γίνεται άμεσα αντιληπτό πως η βέλτιστη διαχείριση της ενεργειακής κατανάλωσης των κόμβων αποτελεί βασικό ζήτημα έρευνας της επιστημονικής κοινότητας, καθώς είναι κομβικής σημασίας για την διάρκεια ζωής του WSN.

### 1.2.1 Δημοφιλείς πλατφόρμες κόμβων

Το γεγονός ότι τα ΑΔΑ μπορούν να βρουν εφαρμογή σε μια πλειάδα εφαρμογών σχεδόν σε όλους τους τομείς ανθρώπινης δραστηριότητας καθώς και λόγω του ιδιαίτερου ερευνητικού ενδιαφέροντος που παρουσιάζουν οδήγησε στην ανάπτυξη ενός μεγάλου αριθμού υλοποιήσεων από πανεπιστήμια οργανισμούς καθώς και εταιρείες.



Εικόνα 5 : MicaZ



Εικόνα 6: Berkley / Crossbow Telos

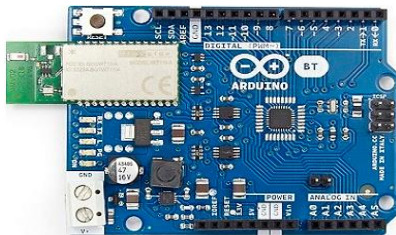




Εικόνα 7: Eyes / EyesIFXv2



Εικόνα 8: Zolertia Firefly



Εικόνα 9: Arduino



Εικόνα 10: BBC micro:bit

Οι πλατφόρμες για την υλοποίηση εφαρμογών WSN που έχουν αναπτυχθεί μέχρι τώρα θα μπορούσαν να καταταχθούν σε δύο κατηγορίες ανάλογα με τα χαρακτηριστικά τους [12]. Πλατφόρμες Χαμηλών προδιαγραφών (Low-End Platforms) και πλατφόρμες Υψηλών προδιαγραφών (High - End Platforms). Οι πλατφόρμες χαμηλού επιπέδου όπως γίνεται αντιληπτό έχουν χαμηλότερες τεχνικές δυνατότητες (μνήμη, υπολογιστική ισχύ, επικοινωνία) και σκοπός τους είναι να αναπτυχθούν σε μεγάλο πληθυσμό σε μια εφαρμογή WSN. Κάποιες από τις πλατφόρμες χαμηλών προδιαγραφών που έχουν χρησιμοποιηθεί σε μεγάλο αριθμό ερευνητικών, και όχι μόνο, εφαρμογών είναι:

- Η οικογένεια Mica (Mica, Mica2, MicaZ, Iris mote)
- Οι πλατφόρμες Telosb / Tmote με παρόμοια χαρακτηριστικά με αυτά των Mica,
- Οι πλατφόρμες Eyes / EyesIFXv2 που αναπτύχθηκαν από ένα 3ετες ευρωπαϊκό ερευνητικό πρόγραμμα και είναι παρόμοιας τεχνολογίας με το Telos
- και κάποιες ακόμα όπως Iris, Zigduino, Zolertia Firefly, SunSPOT αλλά και πλατφόρμες open – source hardware που βρίσκουν χρήση σε πολλούς τομείς, εκπαιδευτικούς και ερευνητικούς και όχι μόνο, όπως την πλατφόρμα Arduino, και πιο πρόσφατη την πλατφόρμα που αναπτύχθηκε από το BBC με όνομα micro:bit.

Στον πίνακα που ακολουθεί παρουσιάζονται κάποιες αρκετά γνωστές πλατφόρμες και γίνεται μια αναφορά στα κύρια χαρακτηριστικά του.

Τα στοιχεία έχουν αντληθεί από διαφορετικές πηγές [12], [13], [14]:



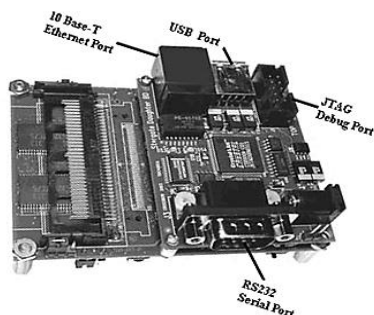
Πλατφόρμα	Μικροελεγκτής/ Επεξεργαστής	Πομποδέκτης	Συχνότητες Επικοινωνίας	Λειτουργικό Σύστημα
<b>MicaZ</b>	ATMega 128L, 8bit , 128KB program memory, 4KB SRAM	Chipcon CC2420	2.4 GHz	TinyOS, SOS, Mantis, Nano-PK, RETOS, LiteOS
<b>Mica2</b>	ATMega 128L, 8bit,128KB program memory, 4KB SRAM	Chipcon CC1000	315/433/868/916 MHz	TinyOS, SOS, Mantis
<b>IRIS</b>	ATmega 1281, 8bit, 128KB program memory, 8KB SRAM	Atmel AT86RF230	2.4 GHz	TinyOS, LiteOS
<b>Eyes</b>	MSP430F149, 16bit, 60KB program memory, 2KB SRAM	RFM TR1001	868 MHz	TinyOS, PEEROS
<b>EyesIFXv1</b>	MSP430F149, 16bit, 60KB program memory, 2KB SRAM	Infineon TDA5250	868 MHz	TinyOS
<b>EyesIFXv2</b>	MSP430F1611, 16bit, 48KB program memory, 10KB RAM	Infineon TDA5250	868 MHz	TinyOS
<b>BTnode</b>	ATmega 128L, 8bit , 128KB program memory, 4KB SRAM	Chipcon CC1000 and Bluetooth radio	433 – 915MHz and 2.4GHz	TinyOS
<b>Telos B</b>	TI MSP430F1611, 16bit, 48KB program memory, 10KB RAM	Chipcon CC2420	2.4 GHz	Contiki, TinyOS, SOS, RETOS
<b>Tmote Sky</b>	TI MSP430F1611, 16bit, 48KB program memory, 10KB RAM	Chipcon CC2420	2.4 GHz	Contiki, TinyOS, SOS, RETOS
<b>Sun SPOT</b>	AT91SAM9G20 processor, 32bit, 64KB ROM, 2x16KB SRAM	Chipcon CC2420	2.4 GHz	Java VM
<b>Shimmer</b>	TI MSP430 F1611, 16bit, 48KB program memory, 10KB RAM	Chipcon CC2420 and Bluetooth radio	2.4 GHz	TinyOS
<b>V-Link]</b>	Not specified	IEEE 802.15.4 compliant RF transceiver	2.4 GHz	Not specified
<b>TEHU- 1121</b>	Not specified	IEEE 802.15.4 compliant RF transceiver	2.4 GHz	Not specified
<b>NI WSN- 3202</b>	Not specified	IEEE 802.15.4 compliant RF transceiver	2.4 GHz	Not specified
<b>Arduino UNO</b>	ATmega328P, 8 bit AVR, 32KB Flash ROM	add-in modules IEEE 802.15.4 radio support, Zigbee and Bluetooth 5.1 with Bluetooth Low Energy(BLE)	433 – 915MHz and 2.4GHz	Not specified
<b>BBC micro:bit</b>	Nordic nRF51822, 32 bit ARM Cortex M0, 256KB Flash ROM, 128KB RAM	IEEE 802.15.4 radio support, Thread, Zigbee and Bluetooth 5.1 with Bluetooth Low Energy(BLE)	2.4 GHz	Mbed, Zephyr

Πίνακας 1: Χαρακτηριστικά γνωστών κόμβων WSN low-end[12],[13],[14]

Οι πλατφόρμες Υψηλού Επιπέδου(high-end) μπορούν να εκτελούν διαχειριστικές εργασίες ή εργασίες ενδιάμεσης πύλης δρομολόγησης (gateway) ή Σταθμού Βάσης και είναι εφοδιασμένες με καλύτερα τεχνικά χαρακτηριστικά.

Γνωστές πλατφόρμες που έχουν αναπτυχθεί για αυτό το σκοπό είναι οι:

- Stargate και Stargate NetBridge NB100 εφοδιασμένες με επεξεργαστές RISC της Intel, μεγάλη μνήμη RAM και πολλές διασυνδέσεις (PCMCIA, USB, Ethernet κα)
- Οι πλατφόρμες Imote/Imote2 που αναπτύχθηκαν από την Crossbow



Εικόνα 11: Stargate



Εικόνα 12: Imote

Στον παρακάτω πίνακα παρουσιάζονται τα χαρακτηριστικά των πλέον γνωστών πλατφορμών high-end για WSN.[12]

Πλατφόρμα	Επεξεργαστής	Μνήμη	Υποστηριζόμενες Διασυνδέσεις	Λειτουργικό Σύστημα
<b>Stargate</b>	Intel PXA255 Processor, 400MHz	64MB SDRAM, 32MB Flash	PCMCIA and compact flash connector, 51-pin expansion Connector for Mica2 Motes; Ethernet, RS232 Serial, JTAG, USB Connector via 51-pin Daughter Card Interface	Embedded Linux
<b>Netbridge NB-100</b>	Intel IXP420 Xscale Processor, 266MHz	32MB RAM, 8MB Flash, 2GB USB Flash Disk	Mica2, MicaZ, IRIS, Telos Connector Ethernet, USB connector	Debian Linux
<b>Imote</b>	ARM7 processor, 12MHz	64KB SRAM, 512KB Flash	I2C, UART, USB, JTAG connector	TinyOS
<b>Imote2</b>	Marvell PXA271 Xscale Processor, 13 – 416MHz	256KB SRAM, 32MB SDRAM, 32MB Flash Memory	Integrated 802.15.4 radio, support for external radio through SDIO, and UART; USB client and host, 2xSPI, 3xUART, Camera, I2C, I2S, GPIO, AC97 connector	TinyOS, Linux, SOS

Πίνακας 2: Χαρακτηριστικά high-end γνωστών πλατφόρμων WSN[12]

Σε αυτή την εργασία χρησιμοποιήθηκε ως εργαλείο για την υλοποίηση ασύρματων δικτύων αισθητήρων η πλατφόρμα **BBC micro:bit** στην οποία γίνεται αναλυτική αναφορά σε επόμενο κεφάλαιο.

## 1.2.2 Λειτουργικά Συστήματα για WSN nodes

“Θα μπορούσαμε να πούμε ότι ένα λειτουργικό σύστημα μπορεί να ειπωθεί ως ιδεατή μηχανή ή ως διαχειριστής πόρων” (Tanenbaum, 2009)

Σε ένα τυπικό υπολογιστικό σύστημα ένα Λειτουργικό Σύστημα αναλαμβάνει να διαχειριστεί πόρους που γενικά είναι σε αφθονία όπως τον επεξεργαστή, την μνήμη, δίσκους, πληκτρολόγιο, ποντίκι, διασυνδέσεις, να ελέγχει την κατανομή τους στους χρήστες και στα προγράμματα του συστήματος, διαχειριζόμενο την χωρική και χρονική πολυπλεξία τους. Οι περιορισμοί που υπάρχουν σε ένα κόμβο WSN είναι τέτοιοι που επιβάλλουν διαφορετικό τρόπο σκέψης στην σχεδίαση ενός Λ.Σ για WSN. [15][16]

Η λειτουργικότητα ενός WSN εκτείνεται σε δύο επίπεδα. Σε επίπεδο κόμβου έχει σημασία η διαχείριση ενέργειας, η ΚΜΕ, η διαχείριση μνήμης, η διαχείριση των αισθητήριων, η διαχείριση των επικοινωνιών κα, ενώ σε επίπεδο δικτύου έχει σημασία η επικοινωνία των κόμβων, η δρομολόγηση των δεδομένων, τα πρωτόκολλα επικοινωνίας, τα χαρακτηριστικά του σήματος.

Ο σχεδιασμός ενός Λ.Σ. για WSN θα πρέπει αντιμετωπίζει ικανοποιητικά αρκετές προκλήσεις. Από τον περιορισμό στους πόρους ενός κόμβου αλλά και θέματα όπως:

- την ανάγκη φορητότητας τους σε διαφορετικές πλατφόρμες μιας και υπάρχουν αρκετές διαφορετικού υλικού πλατφόρμες,
- την ανάγκη οι κόμβοι να λειτουργούν πολυδιεργασιακά εφόσον καλούνται σε πολλές περιπτώσεις να εκτελούν ταυτόχρονα εργασίες όπως χρήση των αισθητήριων τους και δρομολόγηση μηνυμάτων από διπλανούς κόμβους ή συνάθροιση δεδομένων ανάλογα με τις ανάγκες της εφαρμογής,
- Θέματα που προκύπτουν από την κατανεμημένη φύση αυτών των συστημάτων όπως:
  - την ελαχιστοποίηση του κόστους επικοινωνίας μεταξύ των κόμβων
  - την ετερογένεια των κόμβων, μιας και σε μεγάλο μέγεθος WSN με μεγάλο πληθυσμό κόμβων μπορούν να συνυπάρχουν κόμβοι με διαφορετικό υλικό και δυνατότητες που θα επιτελούν διαφορετικής φύσης εργασίες
  - την αλλαγή στο δίκτυο που προκύπτει από την αποτυχία κόμβων είτε επειδή τελείωσε η ενέργεια τους είτε λόγω ειδικών περιβαλλοντικών συνθηκών που δεν επιτρέπουν πάντα την επικοινωνία από όλους τους κόμβους
  - να είναι προσαρμόσιμο και επεκτάσιμο σε διαφορετικά πεδία εφαρμογών ανάλογα με τις διαφορετικές ανάγκες της κάθε εφαρμογής.

Τα χαρακτηριστικά και οι απαιτήσεις που διαφοροποιούν τα υπάρχοντα ΛΣ για WSN σύμφωνα με τα [15][16] είναι η αρχιτεκτονική του ΛΣ, το προγραμματιστικό μοντέλο, οι δυνατότητες χρονοπρογραμματισμού και διαχείρισης πραγματικού χρόνου, το αποτύπωμα μνήμης, η συνδεσιμότητα δικτύου και υποστήριξη πρωτοκόλλων, η ενεργειακή απόδοση, η υποστήριξη υλικού.

Σε επίπεδο αρχιτεκτονικής ΛΣ, ο τρόπος σχεδίασης επηρεάζει τον τρόπο με τον οποίο το Λ.Σ. διαχειρίζεται τις υπηρεσίες που παρέχει στις εφαρμογές καθώς και το μέγεθος του πυρήνα. Βιβλιογραφικά γνωρίζουμε τις εξής αρχιτεκτονικές σχεδίασης λειτουργικών συστημάτων: η μονολιθική (monolithic) σχεδίαση, η αρθρωτή αρχιτεκτονική ή και

αρχιτεκτονική μικροπυρήνα (microkernel), η αρχιτεκτονική εικονικής μηχανής και η στρωματοποιημένη αρχιτεκτονική.

Κατά την μονολιθική αρχιτεκτονική ο πυρήνας είναι μια ενιαία μεγάλη διεργασία που εκτελείται εξ ολοκλήρου σε έναν ενιαίο χώρο διευθύνσεων. Οι υπηρεσίες της εφαρμόζονται ξεχωριστά και κάθε υπηρεσία παρέχει μια διεπαφή για άλλες διεργασίες. Όλες οι υπηρεσίες συστήματος συνδυάζονται σε μία εικόνα συστήματος.

Η αρχιτεκτονική μικροπυρήνα έχει μινιμαλιστικό πυρήνα ο οποίος προσφέρει την ελάχιστη δυνατή λειτουργικότητα και το μεγαλύτερο μέρος των υπηρεσιών παρέχεται από ξεχωριστές διεργασίες, γνωστές ως διακομιστές. Ορισμένοι από αυτούς τους διακομιστές εκτελούνται στον πυρήνα και ορισμένοι εκτελούνται στον χώρο χρήστη. Λόγω αυτής η σχεδίασης το μέγεθος του πυρήνα μειώνεται σημαντικά. Παρέχει καλύτερη αξιοπιστία, ευκολία επέκτασης και προσαρμογή αλλά μπορεί να έχει χειρότερες επιδόσεις σχετικά με την μονολιθική σχεδίαση.

Μια άλλη αρχιτεκτονική επιλογή είναι αυτή της εικονικής μηχανής. Η κύρια ιδέα είναι η εξαγωγή εικονικών μηχανών που αναπαριστούν το υλικό σε προγράμματα χρηστών προσφέροντας όλες τις δυνατότητες του κανονικού υλικού. Το βασικό πλεονέκτημα είναι η φορητότητά του και ένα μειονέκτημα μπορεί να είναι οι μειωμένες επιδόσεις του συστήματος.

Μια στρωματοποιημένη (layered) αρχιτεκτονική OS εφαρμόζει υπηρεσίες με τη μορφή επιπέδων. Τα πλεονεκτήματα που προσφέρει είναι η ευκολία διαχείρισης, ευκολία στην κατανόηση και αξιοπιστία. Κύριο μειονέκτημα είναι ότι δεν είναι πολύ ευέλικτη από την πλευρά του σχεδιασμού του λειτουργικού συστήματος.

Το υποστηριζόμενο προγραμματιστικό μοντέλο καθορίζει τον τρόπο με τον οποίο ένας προγραμματιστής μοντελοποιεί το πρόγραμμα. Τα τυπικά μοντέλα προγραμματισμού για λειτουργικά συστήματα WSN χωρίζονται σε συστήματα χειρισμού γεγονότων ή συμβάντων και προγραμματισμό πολυνημάτωσης. Το μοντέλο προγραμματισμού πολυνημάτωσης δίνει την ευκαιρία να εκτελείται κάθε εργασία στο δικό της νήμα και η επικοινωνία μεταξύ διεργασιών γίνεται με API και είναι το προτιμητέο μοντέλο από τους προγραμματιστές εφαρμογών. Σε ένα σύστημα χειρισμού συμβάντων, κάθε εργασία ενεργοποιείται από ένα εξωτερικό συμβάν, όπως μια διακοπή. Συνοδεύεται συχνά από ένα μοντέλο κοινόχρηστης στοίβας και έναν απλό βρόχο συμβάντος.

Ένα άλλο κρίσιμο στοιχείο οποιουδήποτε λειτουργικού συστήματος είναι ο μηχανισμός χρονοπρογραμματισμού, που καθορίζει τη σειρά με την οποία εκτελούνται οι εργασίες σε μια CPU. Ο τρόπος λειτουργίας του επηρεάζει άλλες ιδιότητες όπως η ενεργειακή απόδοση, οι δυνατότητες σε πραγματικό χρόνο, η απόδοση του συστήματος, το μοντέλο προγραμματισμού και η καθυστέρηση. Οι δυνατότητες για αντιδράσεις σε στενά περιθώρια πραγματικού χρόνου (real-time) που απαιτούνται από κάποιες εφαρμογές πρέπει να ικανοποιούνται από τον ανάλογο μηχανισμού χρονοπρογραμματισμού.

Ένα ΛΣ για WSN πολύ περισσότερο από ένα παραδοσιακό σύστημα θα πρέπει να έχει το ελάχιστο δυνατό αποτύπωμα στη μνήμη του συστήματος μια και οι συσκευές αυτές είναι εφοδιασμένες με λίγα KB μνήμης πολλές φορές μικρότερη από την μνήμη ενός τυπικού συστήματος.

Η κατανομημένη φύση των δικτύων WSN κάνει την επικοινωνία μεταξύ των κόμβων απαραίτητη οπότε ένα ΛΣ θα πρέπει να υποστηρίζει εγγενώς την επικοινωνία μεταξύ των κόμβων και μάλιστα οι πιο πολλές από αυτές τις συσκευές αλλά και η διαφορετικότητα του περιβάλλοντος των WSN εφαρμογών κάνουν αναγκαία την υποστήριξη περισσότερων από έναν τρόπους επικοινωνίας (IEEE 802.15.4, BLE (Bluetooth Low Energy), αλλά και Ethernet) και πολλών πρωτοκόλλων όπως 6LoWPAN, ZigBee, RPL, CoAP, MQTT, LPWAN (LoRa) κ.α.

Οι δυνατότητες δυναμικού επαναπρογραμματισμού του ΛΣ είναι επίσης σημαντική ανάγκη λόγω της δυσκολίας έως και αδυναμίας πρόσβασης στους κόμβους μετά την ανάπτυξη τους στο πεδίο της εφαρμογής.

Πρέπει να υποστηρίζονται τεχνικές απομακρυσμένης μετάδοσης και διαχείρισης του κώδικα της εφαρμογής, τεχνικές για την προσθήκη, διαγραφή η/και τροποποίηση του κώδικα που εκτελείται στον κόμβο κατά την διάρκεια εκτέλεσης (runtime), τεχνικές διασφάλισης ακεραιότητας και ελέγχου εκδόσεων λογισμικού

Η διαχείριση της ενέργειας είναι το κύριο θέμα ζωτικής σημασίας για έναν κόμβο WSN αλλά και για την ίδια την εφαρμογή εφόσον τις περισσότερες φορές δεν υπάρχει η δυνατότητα να τροφοδοτηθούν με επιπλέον ενέργεια κόμβοι που εξαντλήσαν την ενέργεια των μπαταριών τους.

Ο πομποδέκτης ραδιοσυχνοτήτων καθώς και η λειτουργία του μικροελεγκτή είναι συνήθως οι πιο απαιτητικές διαδικασίες σε έναν κόμβο. Για αυτό είναι σύνηθες να υπάρχουν μεγάλα διαστήματα κατάστασης ύπνου - αναστολής του μικροελεγκτή, καθώς το ίδιο και τα πρωτόκολλα που χρησιμοποιούνται, υποστηρίζουν και για τον πομποδέκτη ραδιοσυχνοτήτων.

Από άποψη υποστήριξης υλικού για συσκευές WSN ο βαθμός ετερογένειας είναι πολύ μεγάλος με πολλές διαφορετικές πλατφόρμες, που ένα γενικό λειτουργικό σύστημα πρέπει να υποστηρίζει. Η μεγάλη ποικιλία εφαρμογών για WSN οδήγησε στην ανάπτυξη μιας μεγάλης ποικιλίας τεχνολογιών υλικού και επικοινωνιών. Οι πιο δημοφιλείς συσκευές βασίζονται σε διάφορες αρχιτεκτονικές και οικογένειες μικροελεγκτών, όπως αρχιτεκτονική επεξεργαστή 8-bit πχ Intel 8051 και Atmel AVR, αρχιτεκτονική 16-bit πχ TI MSP430, αρχιτεκτονική 32-bit όπως ARM Cortex-M και MIPS32 και αρχιτεκτονικές 64-bit μελετώνται ήδη για μελλοντική χρήση.[17]

Κάποια από τα δημοφιλή ΛΣ που έχουν χρησιμοποιηθεί σε εφαρμογές WSN είναι: TinyOS[21], Contiki[22], RIOT[24], mbedOS[23], FreeRTOS, MantisOS, nano-RK, ZephyrOS[25].

Με βάση το πλαίσιο που ορίστηκε πιο πάνω, παρατίθεται παρακάτω πίνακας με αναφορά στα βασικά χαρακτηριστικά δημοφιλών λειτουργικών συστημάτων για πλατφόρμες υλοποίησης WSN:

WSN OS	Architecture	Programming Model	Scheduling	Real-Time Support	Network Protocols	Development Language	Hardware Support
<b>TinyOS</b>	Monolithic	Event Driven, Threads	Cooperative, FIFO, EDF	Not Supported	802.15.4, Zigbee, 6LowPan, RPL, COAP, MQTT	NesC	AVR, MSP430, px27ax
<b>Contiki</b>	Modular	Protothreads and Event Driven	Preemptive	Partial Support	802.15.4, BLE, 6LowPan, RPL, COAP, MQTT	C	AVR, MSP430, ARM7, ARM Cortex-M, PIC32, x86, 6502
<b>RIOT</b>	Microkernel	Multi-threading	Preemptive	Supported	802.15.4, Zigbee, BLE, LPWAN, 6LowPan, RPL, COAP, MQTT	C	AVR, MSP430, ARM7, ARM Cortex-M, x86, TI MSP430 devices
<b>FreeRTOS</b>	Microkernel	Multi-threading	Preemptive	Supported	802.15.4, Zigbee, 6LowPan, RPL, COAP, MQTT	C	AVR, MSP430, ARM, x86, 8052, Renesasc
<b>MantisOS</b>	Layered	Multi-threading	Priority Based, RR	Partial Support		C	Atmel AVR, MSP430
<b>Mbed Os</b>	Microkernel	Multi-threading	Preemptive	Supported	802.15.4, Zigbee, BLE, LPWAN, 6LowPan, RPL, COAP, MQTT	C	ARM Cortex-M
<b>Nano-RK</b>	Monolithic	Multi-threading	Preemptive, Fixed Priority Based	Supported		C	
<b>Zephyr OS</b>	Monolithic	Multi-threading	Preemptive, Cooperative	Supported	802.15.4, Zigbee, BLE,, 6LoWPan, RPL, COAP, MQTT	C	ARM (Cortex-M, Cortex-R, Cortex-A53), x86, x86-64, ARC, RISC-V, Nios II, Xtensa

Πίνακας 3: Γνωστά Λειτουργικά Συστήματα WSN

### 1.3 Αρχιτεκτονική ενός Ασύρματου Δικτύου Αισθητήρων

Κατά την υλοποίηση ενός ασύρματου δικτύου αισθητήρων σημαντικός αριθμός, που μπορεί να κυμαίνεται από μερικές δεκάδες έως πολλές χιλιάδες, ασύρματων κόμβων αισθητήρων, στατικών ή κινούμενων, μπορεί να έχουν αναπτυχθεί με προσχεδιασμένο ή τυχαίο τρόπο σε μια περιοχή, λαμβάνοντας μετρήσεις και αποστέλλοντας τα δεδομένα σε κεντρικό σημείο δηλαδή σε έναν ή περισσότερους σταθμούς βάσης (Base station ή Sink).

Ανάλογα με την εφαρμογή, τα WSN εμφανίζουν διαφορετική δομή, πυκνότητα και μπορεί οι κόμβοι να έχουν προκαθορισμένες ή τυχαίες θέσεις. Απαραίτητη σε κάθε περίπτωση είναι η συνεργασία των κόμβων καθώς αναλόγως με την τοπολογία, η επικοινωνία μέχρι τον BS μπορεί να πραγματοποιείται με ένα άλμα (one hop) ή με πολλαπλά άλματα (multiple hops).

Οι κόμβοι δηλαδή συνήθως έχουν διπλό ρόλο:

- Μπορεί να αποτελούν πηγή δεδομένων (source), καθώς συλλέγουν μετρήσεις μέσω των αισθητήριων οργάνων, τις επεξεργάζονται και τις αποστέλλουν ασύρματα στον κοντινότερο γείτονα τους ώστε τελικά να καταλήξουν μέσω μιας διαδρομής πολλαπλών αλμάτων στον σταθμό συγκέντρωσης (BS).
- Και μπορούν να έχουν ρόλο Δρομολογητή Δεδομένων (gateway), που αναμεταδίδει πακέτα δεδομένων από έναν κόμβο σε άλλον και ως τον σταθμό βάσης, όπου θα γίνει περαιτέρω επεξεργασία, ανάλυση και μετέπειτα διαχείριση τους (εφαρμογή, αποστολή στο νέφος κλπ.) εφόσον για λόγους χαμηλότερης κατανάλωσης η επικοινωνίες έχουν μικρό εύρος μετάδοσης.

Σε πολλές περιπτώσεις ο σταθμός βάσης δεν είναι ένας μόνο αλλά υπάρχουν πολλοί σταθμοί βάσης και οι κόμβοι προωθούν την πληροφορία στον πιο εύκολα προσβάσιμο.

Ο σταθμός βάσης (base station - BS) βιβλιογραφικά αναφέρεται και ως συλλέκτης ή νεροχύτης (sink), αφού αποτελεί τον αποδέκτη όλων των δεδομένων συλλογής. Μπορεί να είναι ένας κόμβος σαν τους άλλους εντός του δικτύου αλλά σύνηθες είναι να είναι και μία μονάδα εκτός WSN όπως ένας φορητός υπολογιστής ή και κάποια gateway για την αποθήκευση και επεξεργασία των δεδομένων στο νέφος (cloud).

Ένα σημαντικό χαρακτηριστικό των WSNs είναι η αυτόνομη λειτουργία τους. Ακριβώς λόγω της φύσης των εφαρμογών στις οποίες χρησιμοποιούνται που μπορεί να είναι περιοχές αλλά και περιβάλλοντα δυσπρόσιτα έως και απρόσιτα για τον άνθρωπο, τα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται θα πρέπει να έχουν χαρακτηριστικά αυτο-ίασης καθώς και δυνατότητες λειτουργίας σε καταστάσεις αλλαγών της τοπολογίας. Ταυτόχρονα η βέλτιστη διαχείριση της ενέργειας των κόμβων του δικτύου αποτελεί τον βασικότερο περιορισμό τους και ίσως το πιο προκλητικό ερευνητικό θέμα.

Η ασύρματη επικοινωνία αποτελεί μια από τις κύριες δραστηριότητες κατανάλωσης ενέργειας. Γι' αυτό ο σχεδιασμός ενεργειακά αποδοτικών πρωτοκόλλων επικοινωνίας και διαδικασιών λειτουργίας τους καθίσταται επιτακτική ανάγκη για την βιωσιμότητα ενός WSN. [5]

Σε ερευνητικό επίπεδο σχετικά με το ζήτημα της μεγιστοποίησης της διάρκειας λειτουργίας ενός WSN γίνεται μελέτη από διάφορες πλευρές:

- Ενεργειακή βελτιστοποίηση των διαδικασιών δρομολόγησης των προς μετάδοση δεδομένων,
- τεχνικές ομαδοποίησης των κόμβων σε συστοιχίες (clusters),
- διαδικασίες συνάθροισης δεδομένων (data aggregation),
- τεχνικές συγχώνευσης

είναι μερικές από τις τεχνολογίες που μελετώνται και εφαρμόζονται για την μεγιστοποίηση της διάρκειας ζωής ενός WSN.

Η διαχείριση της **τοπολογίας** του δικτύου είναι ένα θέμα με πολλές ιδιαιτερότητες σε ένα WSN το οποίο ανάλογα με το λόγο για τον οποίο έχει αναπτυχθεί μπορεί να οδηγήσει είτε σε προεπιλεγμένη θέση των κόμβων και της πυκνότητας του δικτύου είτε σε θέσεις με

μεγάλο βαθμό τυχαιότητας. Όπως προκύπτει ο σχεδιασμός της τοπολογίας ενός δικτύου αισθητήρων θα μπορούσε να διαχωριστεί σε τρεις φάσεις[3] :

- Διαδικασίες πριν και κατά την διάρκεια της ανάπτυξης του δικτύου λόγω του τρόπου τοποθέτησης των κόμβων που μπορεί να περιλαμβάνει την ρίψη, διασπορά, ή προμελετημένη τοποθέτησή τους
- Διαδικασίες μετά την ανάπτυξη που μπορεί να προκύψουν από την θέση, τις περιβαλλοντικές συνθήκες ή πιθανές δυσλειτουργίες κόμβων
- Διαδικασίες ανάπτυξης επιπλέον κόμβων σε περιπτώσεις που δεν υπάρχει ικανοποιητική κάλυψη του πεδίου ενδιαφέροντος, που πρέπει να αντικατασταθούν κόμβοι που δεν έχουν επικοινωνία είτε λόγω θέσης είτε λόγω απώλειας της ενέργειάς τους,

Η επιλογή του ασυρμάτου μέσου μετάδοσης εξαρτάται από το είδος της εφαρμογής για την οποία υλοποιήθηκε το ασύρματο δίκτυο αισθητήρων και είναι πολύ σημαντική για την αξιοπιστία της επικοινωνίας καθώς αναλόγως το μέσο εισάγονται διαφορετικοί περιορισμοί. Οι επιλογές είναι ανάμεσα στις ραδιοσυχνότητες RF, την οπτική επικοινωνία (υπέρυθρες και όχι μόνο), τα ηχητικά κύματα (ακουστική επικοινωνία) και την μαγνητική επαγωγή.

Συνηθέστερη επιλογή αποτελεί αυτή των ραδιοσυχνοτήτων ως πιο αποδοτική στις απαιτήσεις των περισσότερων εφαρμογών WSN. Η χρήση της εξασφαλίζει σχετικά μεγάλο εύρος και υψηλά ποσοστά δεδομένων, χωρίς να απαιτείται οπτική επαφή αποστολέα και παραλήπτη. Παράλληλα είναι λιγότερο απαιτητική σε κατανάλωση πόρων που είναι βασικό για την βιωσιμότητα του WSN.

Δημοφιλής επιλογή για τις συνδέσεις ραδιοσυχνοτήτων είναι η χρήση του ISM (Industrial Scientific and Medical Bands) φάσματος με προφανή πλεονεκτήματα την ελεύθερη (δωρεάν) χρήση συχνοτήτων και την διαθεσιμότητα του παγκοσμίως. Χρησιμοποιούνται συνήθως συχνοτήτες της τάξης των 433MHz στην Ευρώπη και 915MHz στις ΗΠΑ και 2.4GHz παγκοσμίως.

Η οπτική, υπέρυθρη, επικοινωνία υλοποιείται στη ζώνη που βρίσκεται στα κατώτερο άκρο του ορατού φάσματος με μήκος κύματος 750 έως 1000 nm. Ένας πομπός που αποτελείται από μια δίοδο εκπομπής φωτός (LED) ή μια δίοδο λέιζερ (LD) χρησιμοποιείται για να εκπέμψει υπέρυθρο φως, το οποίο μπορεί να συλληφθεί από μια φωτοδίοδο [26]. Ο πιο συχνά χρησιμοποιούμενος τύπος τέτοιας επικοινωνίας είναι η επικοινωνία από σημείο σε σημείο, όπου ο πομπός και ο δέκτης πρέπει να έχουν οπτική επαφή. Υπάρχουν επίσης περιπτώσεις που έχει χρησιμοποιηθεί ζώνη φάσματος, διαφορετική της υπέρυθρης, για επικοινωνία μεταξύ κόμβων αισθητήρων. Τέτοιο παράδειγμα είναι η εφαρμογή Smart Dust, το οποίο είναι ένα αυτόνομο σύστημα ανίχνευσης, υπολογισμού και επικοινωνίας που χρησιμοποιεί το οπτικό μέσο για μετάδοση. [4]

Σε πολλές περιπτώσεις τα ιδιαίτερα χαρακτηριστικά του περιβάλλοντος στο οποίο αναπτύσσεται το WSN καθορίζουν το είδος επικοινωνίας. Μια τέτοια περίπτωση είναι τα υποβρύχια δίκτυα αισθητήρων που προτιμώνται τα ηχητικά κύματα έναντι των υπολοίπων μορφών ασύρματης επικοινωνίας. Η διαφορά μεταξύ των επίγειων και υποβρυχίων δικτύων



αισθητήρων είναι το υποβρύχιο περιβάλλον που οδηγεί σε διαφορετικά χαρακτηριστικά διάδοσης και εξασθένησης του σήματος από ό, τι στον αέρα.

Έρευνες στον τομέα αυτό που έχουν ξεκινήσει από τον Β΄ Παγκόσμιο Πόλεμο στις ΗΠΑ για την ασύρματη επικοινωνία με τα πολεμικά υποβρύχια καταδεικνύουν ότι τα ηλεκτρομαγνητικά κύματα διαδίδονται σε μεγάλες αποστάσεις μέσω αγωγίμου θαλάσσιου νερού μόνο σε πολύ χαμηλές συχνότητες (30–300 Hz) [27]. Αυτό απαιτεί μεγάλες κεραιές και υψηλή ισχύ μετάδοσης, συνθήκες ακατάλληλες για την ανάπτυξη δικτύων αισθητήρων.

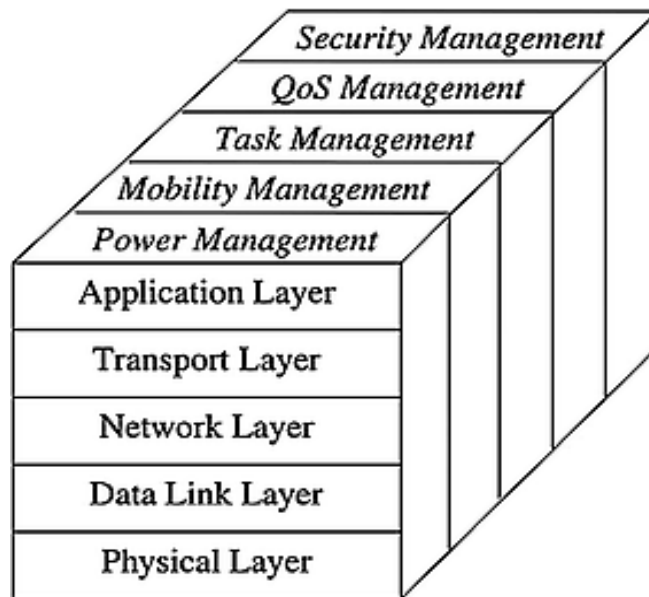
Επομένως, η επικοινωνία με ηλεκτρομαγνητικά κύματα δεν είναι ο καλύτερος τρόπος για υποβρύχια επικοινωνία. Αν επιλεγούν τα οπτικά κύματα που δεν πάσχουν από τόσο υψηλή εξασθένηση, το πρόβλημα που δημιουργείται είναι η σημαντική σκέδαση κατά την επικοινωνία των οπτικών κυμάτων κάτω από το νερό. Επιπλέον, η μετάδοση οπτικών σημάτων απαιτεί υψηλή ακρίβεια στην κατεύθυνση των στενών ακτινών λέιζερ, με αποτέλεσμα τα οπτικά κύματα να χρησιμοποιούνται μόνο για επικοινωνία μικρής εμβέλειας στο υποβρύχιο περιβάλλον.

Τα παραπάνω καθιστούν την ακουστική επικοινωνία την πιο κατάλληλη τεχνολογία φυσικού επιπέδου για υποβρύχια δίκτυα. Για την αποτελεσματική επικοινωνία απαιτούνται κατάλληλα μικρόφωνα και δέκτες για τη δημιουργία υποβρύχιων ασύρματων συνδέσεων. Η ποιότητα της επικοινωνίας μπορεί να επηρεαστεί επίσης από απώλεια διαδρομής, θόρυβο, πολλαπλή διαδρομή, φαινόμενο Doppler και καθυστέρηση διάδοσης και εξαρτάται από τις ιδιότητες του νερού, όπως η θερμοκρασία, το βάθος και η σύνθεση.

Σε περιπτώσεις ανάπτυξης WSN εφαρμογών σε υπόγεια περιβάλλοντα η επικοινωνία επηρεάζεται από υπόγεια εμπόδια όπως χώμα, βράχια και σήραγγες. Οι παραδοσιακές τεχνικές μετάδοσης ηλεκτρομαγνητικών σημάτων αντιμετωπίζουν πολλαπλά προβλήματα όπως υψηλά επίπεδα εξασθένησης και απώλεια διαδρομής. Για να αντιμετωπιστούν ικανοποιητικά οι προκλήσεις της υπόγειας επικοινωνίας μια εναλλακτική τεχνική διάδοσης σήματος για υπόγεια ασύρματη επικοινωνία είναι η μαγνητική επαγωγή καθώς έχει παρατηρηθεί πως πυκνά μέσα όπως το έδαφος και το νερό προκαλούν μικρή μεταβολή στον ρυθμό εξασθένησης των μαγνητικών πεδίων από εκείνο του αέρα, καθώς η μαγνητική διαπερατότητα καθενός από αυτά τα υλικά είναι παρόμοια. [28]

### 1.3.1 Πρωτόκολλα Επικοινωνίας σε ένα WSN

Στο σχήμα που παρατίθεται παρακάτω φαίνεται η διαστρωματοποιημένη αρχιτεκτονική των πρωτοκόλλων επικοινωνίας σε ένα WSN. Στο σχεδιασμό των πρωτοκόλλων αυτών πρέπει πάντα να ληφθούν υπόψη τα ιδιαίτερα επιμέρους χαρακτηριστικά του δικτύου με κυριότερο την αποτελεσματική διαχείριση της ενέργειας και την περιορισμένη επεξεργαστική δυνατότητα των κόμβων, αλλά και χαρακτηριστικά όπως η ανάγκη δυναμικής δρομολόγησης αλλά και ανά-δρομολόγησης, η αλλαγές στην τοπολογία αφού κάποιος κόμβος μπορεί να χάσει τα αποθέματα ενέργειας και να τεθούν εκτός δικτύου, καθώς και χαρακτηριστικά όπως κινητικότητα των κόμβων ή ποιότητα υπηρεσίας και η ασφάλεια.[30]



Εικόνα 13: Διαστρωματοποιημένη Αρχιτεκτονική πρωτοκόλλων επικοινωνίας και επίπεδα διαχείρισης[29]

Παρατηρούμε ότι αποτελείται από πέντε επίπεδα: φυσικό, ζεύξης δεδομένων, δικτύου, μεταφοράς και εφαρμογής, καθώς και πέντε διαχειριστικά επίπεδα που αναλαμβάνουν να βελτιώσουν την λειτουργία ενός WSN με βάση διαφορετικές μετρικές. Αυτά είναι τα : διαχείρισης ενέργειας, διαχείρισης κινητικότητας, διαχείρισης εργασιών, διαχείρισης ποιότητας υπηρεσίας και διαχείριση ασφάλειας.

Στα διαχειριστικά επίπεδα το επίπεδο διαχείρισης ενέργειας είναι υπεύθυνο για την αποτελεσματική διαχείριση της κατανάλωσης ενέργειας, το επίπεδο διαχείρισης κινητικότητας για την διατήρηση των διαδρομών δρομολόγησης ακόμα και στην περίπτωση που έχουμε κινούμενους κόμβους, το επίπεδο διαχείρισης εργασιών αναλαμβάνει την καταμέριση εργασιών δρομολόγησης, χρήσης αισθητήριων οργάνων και επεξεργασίας των δεδομένων στους κόμβους, το επίπεδο διαχείρισης QoS φροντίζει για την βελτιστοποίηση της απόδοσης με βάση συγκεκριμένες μετρικές και το επίπεδο διαχείρισης ασφάλειας αναλαμβάνει θέματα που άπτονται στην ασφαλή μετάδοση των δεδομένων.

### 1.3.1.1 Φυσικό Επίπεδο

Το φυσικό επίπεδο ασχολείται με τις ανάγκες των τεχνικών μετατροπής των ροών δεδομένων σε κατάλληλο για μετάδοση και λήψη σήμα. Είναι υπεύθυνο για την επιλογή της κατάλληλης ζώνης συχνότητας μετάδοσης, την επιλογή της φέρουσας, την ανίχνευση σήματος, την διαμόρφωση του σήματος καθώς και την κρυπτογράφηση δεδομένων. Το ασύρματο κανάλι παραμορφώνει το σήμα που μεταδίδεται. Φαινόμενα εξασθένησης σήματος, ανάκλασης, διάθλασης και διασποράς πρέπει να αντιμετωπιστούν αποτελεσματικά με τις τεχνικές που υιοθετούνται στο φυσικό επίπεδο. Η σημαντικότερη όμως παράμετρος είναι η διαχείριση της ενέργειας για τη βιωσιμότητα του WSN. Εάν για παράδειγμα χρησιμοποιηθεί επικοινωνία πολλαπλών αλμάτων σε ένα WSN μπορεί να αντιμετωπιστούν και προβλήματα σκίασης και απώλειας διαδρομής και να καταναλωθεί μικρότερο ενεργειακό φορτίο.

Σχετικά με την μελέτη των τεχνικών διαμόρφωσης για WSN στα [32] [33], γίνεται μια αναλυτική περιγραφή των τεχνικών διαμόρφωσης σε σχέση με την κατανάλωση ενέργειας σε ένα WSN.

Κατά τον σχεδιασμό των πρώτων πλατφόρμων όπως οι Berkeley Xbow Mica2 χρησιμοποιήθηκε ψηφιακή διαμόρφωση συχνότητας FSK δύο συμβόλων (Binary) ενώ στο πρότυπο 802.15.4 χρησιμοποιήθηκε OQPSK πολλαπλών συμβόλων (M-ary).

### 1.3.1.2 Επίπεδο Ζεύξης Δεδομένων

Στο επίπεδο ζεύξης δεδομένων λαμβάνουν χώρα οι διαδικασίες της δημιουργίας πλαισίων δεδομένων, ο έλεγχος σφαλμάτων, ο έλεγχος ροής δεδομένων και ο έλεγχος πρόσβασης στο κοινό μέσο. Είναι υπεύθυνο για τον ισότιμο και αποτελεσματικό διαμοιρασμό του ασύρματου καναλιού στους κόμβους και την κατασκευή της δομής των δικτυακών συνδέσεων ώστε να επιτευχθεί η επικοινωνία σημείου προς σημείο.

Για τον έλεγχο σφαλμάτων (Bit Error Rate -BER) χρησιμοποιούνται τεχνικές Forward error correction –FEC με την χρήση κατάλληλων τεχνικών διόρθωσης σφαλμάτων που γενικά προτιμούνται λόγω της δυνατότητας των κόμβων να εκτελούν επεξεργασία, Automatic repeat Request – ARQ και υβριδικές ARQ ανάλογα με το κόστος επανεκπομπής καθώς και το κόστος της αύξησης της επικεφαλίδας (Overhead).

Ο έλεγχος πρόσβασης στο μέσο γίνεται στο υποεπίπεδο MAC, ώστε να αποφεύγονται οι συγκρούσεις και αναμεταδόσεις μηνυμάτων κατά την μετάδοση που τελικά οδηγούν σε σπατάλη των ενεργειακών αποθεμάτων των κόμβων. [34][35] Εδώ σε σχέση με τα παραδοσιακά δίκτυα σε ένα πρωτόκολλο για WSN λαμβάνεται πάντα υπόψη η αποτελεσματική διαχείριση ενέργειας, οι συχνές αλλαγές στην τοπολογία και η πυκνότητα του δικτύου που μπορεί να οδηγήσει πολλοί κόμβοι να λαμβάνουν πακέτα που προορίζονται για άλλους καθώς και ο μεγάλος αριθμός των κόμβων.

Τα περισσότερα πρωτόκολλα MAC που λειτουργούν στα WSN βασίζονται για τον έλεγχο πρόσβασης στο μέσο, είτε σε τεχνικές Πολλαπλής Πρόσβασης με Ακρόαση Φέροντος (CSMA) είτε σε τεχνικές Πολλαπλής πρόσβασης με πολυπλεξία διαίρεσης χρόνου (TDMA) ενώ συνήθως αποφεύγονται τεχνολογίες βασισμένες σε FDMA για λόγους κατανάλωσης ενέργειας.

### 1.3.1.3 Επίπεδο Δικτύου

Η βασική λειτουργία του επιπέδου δικτύου είναι η δρομολόγηση των δεδομένων. Οι εγγενείς περιορισμοί που υπάρχουν από την φύση των WSN κάνουν τα παραδοσιακά πρωτόκολλα που είναι βασισμένα στο IP να μην είναι κατάλληλα για χρήση σε WSN. Η πάντα περιορισμένη ισχύ επεξεργασίας, μετάδοσης και ενέργειας σε συνδυασμό με θέματα όπως την πλεονασματικότητα της μεταδιδόμενης πληροφορίας σε πυκνά δίκτυα, την ροή από πολλές διαφορετικές πηγές σε ένα σημείο (Σταθμό βάσης ή sink) καθώς και την αδυναμία δημιουργίας ενός κοινού σχήματος διευθυνσιοδότησης για μεγάλους πληθυσμούς κόμβων

αισθητήρων είναι περιορισμοί που επιβάλουν διαφορετική διαχείριση της δρομολόγησης των δεδομένων.[30], [40], [41]

Στα WSN για τον σχεδιασμό των πρωτοκόλλων για την αποτελεσματική δρομολόγηση των πακέτων δεδομένων σε ένα περιβάλλον πολλαπλών συνήθως αλμάτων μεταξύ κόμβων λαμβάνονται υπόψη:

- Η ενεργειακή διαχείριση για την μέγιστη διάρκεια ζωής των κόμβων
- Η ανάγκη μείωσης της πολυπλοκότητας καθώς και την ανάγκη επεκτασιμότητας
- Το γεγονός ότι τα WSN είναι συνήθως εστιασμένα στα δεδομένα (data-centric networks)
- Διαδικασίες συνάθροισης των δεδομένων (data aggregation techniques) μπορούν να εκτελεστούν από γειτονικούς κόμβους
- Η διαχείριση της τοπολογίας που μπορεί να ποικίλει αναλόγως με τον τρόπο ανάπτυξης του WSN

Διάφοροι αλγόριθμοι έχουν αναπτυχθεί για την δρομολόγηση σε WSN και μπορούν να κατηγοριοποιηθούν σε τέσσερις κατηγορίες: δεδομένο-κεντρικοί (data-centric), ιεραρχικοί (hierarchical), γεωγραφικοί (location-based) και αλγόριθμοι για παροχή ποιότητας υπηρεσίας (Quality of Service - QoS).

<b>Data-centric (Flat –Based)</b>	<b>Hierarchical</b>	<b>Location-based (Geographical)</b>	<b>QoS-based</b>
Flooding	LEACH	MECN	SAR
Gossiping	PEGASIS	SMECN	Min-Cost Path
SPIN	TEEN	PRADA	SPEED
Directed Diffusion	APTEEN	GAF	Maximum lifetime energy routing
Energy-aware routing	Energy-aware routing for cluster - based sensor networks	GEAR	Maximum lifetime data gathering
Gradient-based routing	Self-organizing protocol		Energy-aware QoS routing protocol
COUGAR			
ACQUIRE			

**Πίνακας 4:Πρωτόκολλα δρομολόγησης στα WSN**

Στα δεδομένο-κεντρικά (Data-centric or Flat–Based) πρωτόκολλα όλοι οι κόμβοι αντιμετωπίζονται ισότιμα από το δίκτυο και η δρομολόγηση βασίζεται σε ερωτήματα σχετικά με τα δεδομένα (data queries) και αποστολή προς τον Σταθμό βάσης μόνο από τους κόμβους των οποίων τα δεδομένα ικανοποιούν τα κριτήρια του ερωτήματος, με βασικό στόχο τον περιορισμό των πολλών περιττών μεταδόσεων.

Τα ιεραρχικά πρωτόκολλα (Hierarchical or Clustering) ως στόχο έχουν την ομαδοποίηση των κόμβων σε συστάδες (ομάδες), στις οποίες οι κόμβοι έχουν ρόλους, σε αυτούς που κάνουν παρατήρηση του φαινομένου (sensing) και σε επικεφαλής ομάδας

(cluster-heads) που αναλαμβάνουν να κάνουν συνάθροιση και επεξεργασία των πρωτογενών δεδομένων πριν την προώθηση τους στον Σταθμό Βάσης .

Τα γεωγραφικά πρωτόκολλα χρησιμοποιούν την πληροφορία θέσης των κόμβων για να μεταδώσουν τα δεδομένα στις επιθυμητές περιοχές και όχι σε ολόκληρο το δίκτυο. Η επιλογή της χρήσης GPS αποφεύγεται για τον καθορισμό της θέσης του κόμβου για ενεργειακούς λόγους. Προτιμώνται μέθοδοι τριγωνισμού. Δεδομένου ότι οι κόμβοι του δικτύου κατανέμονται χωρικά σε ένα πεδίο και δεν χρησιμοποιείται σχήμα διευθυνσιοδότησης οι πληροφορίες τοποθεσίας του κάθε κόμβου μπορούν να χρησιμοποιούνται για την δρομολόγηση των δεδομένων με στόχο την βελτίωση της ενεργειακής απόδοσης και την μείωση της καθυστέρησης.

Τα περισσότερα πρωτόκολλα δρομολόγησης σχεδιάζονται με βάση την ελαχιστοποίηση της κατανάλωσης ενέργειας. Υπάρχουν όμως περιπτώσεις που για συγκεκριμένες εφαρμογές WSN λαμβάνονται υπόψη και άλλες μετρικές για την παροχή ποιότητας υπηρεσιών (QoS based). Τα πρωτόκολλα αυτά βασίζονται σε γενικές παραμέτρους ροής δικτύου και προσπαθούν να ικανοποιήσουν ορισμένες απαιτήσεις QoS μαζί με τη λειτουργία δρομολόγησης.

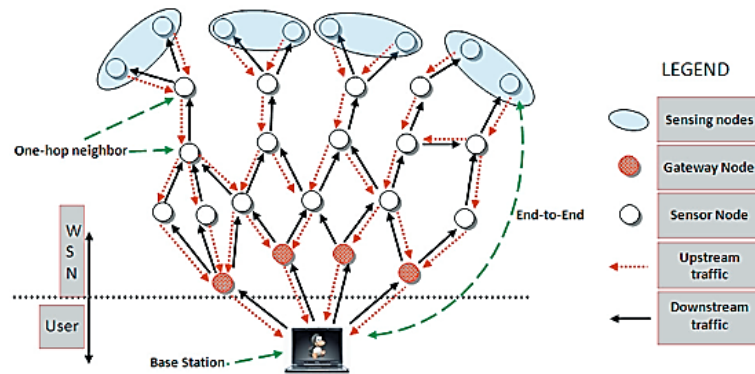
Μια άλλη κατηγοριοποίηση που μπορεί να γίνει είναι αναλόγως με το αν οι διαδρομές δρομολόγησης είναι προσχεδιασμένες ή υπολογίζονται κατ' απαίτηση αντιδρώντας σε κάποιο γεγονός. Έτσι κάποια πρωτοκολλά δρομολόγησης λειτουργούν προδραστικά (proactively) με βάση προϋπάρχουσες διαδρομές που ανανεώνονται περιοδικά ή αντιδραστικά (reactively) δημιουργώντας την διαδρομή την στιγμή που θα την χρειαστούν. Υπάρχουν και πρωτόκολλα που λειτουργούν υβριδικά κάνοντας συνδυασμό των δύο τεχνικών. [40]

#### 1.3.1.4 Επίπεδο Μεταφοράς

Στο επίπεδο μεταφοράς στα WSN τα υπάρχοντα πρωτόκολλα όπως το TCP και το UDP δεν είναι κατάλληλα αφού κατά τον σχεδιασμό τους δεν λήφθηκαν υπόψη τους τα ιδιαίτερα χαρακτηριστικά των δικτύων WSN. Για τα υπάρχοντα πρωτόκολλα η από άκρο σε άκρο αξιόπιστη μεταφορά δεδομένων ακολουθεί τις σχεδιαστικές αρχές που έχουν καθιερωθεί από το TCP οι οποίες βασίζονται σε σχήμα καθολικής διευθυνσιοδότησης, σε αναγνώριση σφαλμάτων και αναμεταδόσεις από άκρο σε άκρο. Τα πρωτόκολλα του επιπέδου μεταφοράς υλοποιούνται στους τερματικούς σταθμούς του δικτύου και εξασφαλίζουν την από άκρο σε άκρο αξιόπιστη μετάδοση των δεδομένων στις εφαρμογές άσχετα με τα πρωτόκολλα επιπέδου δικτύου. Οι τεχνικές αυτές είναι πολύ επιβαρυντικές για WSN και στο θέμα διαχείρισης ενέργειας αλλά και σε θέματα που έχουν να κάνουν με την διαχείριση της μνήμης των κόμβων αφού απαιτούν να παραμένουν τα μεταδιδόμενα πακέτα στον buffer μέχρι να επιβεβαιώσει ο κόμβος λήψης ότι παρέλαβε το πακέτο. [44], [45]

Στα WSN η ροή της πληροφορίας μπορεί να είναι είτε κυρίως προς τον Σταθμό Βάσης (upstream – ανοδική ροή) από τους κόμβους που παρακολουθούν – καταγράφουν το φαινόμενο στο πεδίο (sensing nodes) είτε πολύ αραιότερα από τον Σταθμό Βάσης προς τους

κόμβους(downstream– καθοδική ροή) για περιπτώσεις που θα χρειαστεί να αποσταλούν στους κόμβους από το Σταθμό Βάσης ενημερώσεις λογισμικού, ασφαλείας κοκ.



Εικόνα 14:Κίνηση δεδομένων σε ένα WSN[45]

Στον σχεδιασμό των πρωτοκόλλων μεταφοράς για WSN λαμβάνεται μέριμνα για την αξιόπιστη παράδοση δεδομένων και τον έλεγχο συμφόρησης ειδικά στις περιπτώσεις των Σταθμών Βάσης και των γειτονικών τους κόμβων, στους οποίους καταλήγει ο κύριος όγκος των μηνυμάτων. Μετάδοση χωρίς την εγκατάσταση σύνδεσης ή με πολύ απλές διαδικασίες εγκατάστασης σύνδεσης βελτιώνουν την συνολική απόδοση και την καθυστέρηση μετάδοσης ειδικά στις εφαρμογές που οι κόμβοι στέλνουν δεδομένα αντιδρώντας σε κάποιο γεγονός (reactively). Για να επιτευχθεί η απαιτούμενη αξιοπιστία μετάδοσης χρησιμοποιούνται μηχανισμοί είτε από Άκρο σε Άκρο (end to end) είτε από κόμβο σε κόμβο (hop – by – hop) όπου αναλόγως την μετάδοση και το είδος της εφαρμογής επιλέγεται η καταλληλότερη τεχνική ή και ανταλλαγή μεταξύ τους. Για τις downstream μεταδόσεις που θα πρέπει να εξασφαλίζεται αξιόπιστη μετάδοση για το σύνολο των εν λειτουργία κόμβων (ενημερώσεις λογισμικού, εφαρμογής, ασφάλειας) χρησιμοποιούνται συνήθως μηχανισμοί hop – by – hop με τεχνικές αποστολής μηνύματος αρνητικής αναγνώρισης (nack) ενώ στους μηχανισμούς end – to – end χρησιμοποιούνται τεχνικές αποστολής μηνύματος θετικής αναγνώρισης (ack).[45]

Σχετικά με τον έλεγχο της συμφόρησης τα πρωτόκολλα του επιπέδου μεταφοράς μπορεί να υλοποιούν μηχανισμούς ειδοποίησης συμφόρησης στους γειτονικούς κόμβους ή και μηχανισμούς αποφυγής συμφόρησης είτε με την προσαρμογή του ρυθμού μετάδοσης (rate adjustment) και την ανακατεύθυνση της κυκλοφορίας των δεδομένων από διαδρομές που δεν τελούν υπό συμφόρηση (traffic redirection) είτε πιο σπάνια με ένα μηχανισμό περιορισμού των εκπομπών του κόμβου σε σχέση με τους γειτονικούς του όταν ανιχνεύσει ότι αποστέλλονται ίδια δεδομένα με τα δικά του (polite gossip policy)[46]

#### Αξιόπιστη παράδοση δεδομένων και έλεγχος συμφόρησης:

TRCCIT(2010) Tunable Reliability With Congestion Control For Information Transport

CRRT(2009) Congestion Aware And Rate Controlled Reliable Transport

CTCP(2008) Collaborative Transport Control Protocol

RT2 (2008) Real-Time And Reliable Transport  
ART(2007) Asymmetric And Reliable Transport  
RCRT(2007) Rate-Controlled Reliable Transport  
Flush (2007) Flush  
DST (2006) Delay Sensitive Transport  
Port (2005) Price-Oriented Reliable Transport Protocol  
ESRT(2005) Event-To-Sink Reliable Transport  
STCP(2005) (Sensor Transmission Control Protocol)

**Αξιοπιστία παράδοσης μόνο:**

---

ERTP(2009) (Energy-efficient and reliable transport protocol)  
Garuda(2008) Garuda  
DTSN (2007) Distributed Transport For Sensor Networks  
RBC (2005) Reliable Bursty Convergecast  
DTC(2004) Distributed TCP Caching  
RMST(2003) Reliable Multi-Segment Transport  
PSFQ (2002) Pump Slowly Fetch Quickly

**Έλεγχος Συμφόρησης μόνο:**

---

PHTCCP(2008) Prioritized Heterogeneous Traffic-Oriented Congestion Control Protocol  
PCCP(2006) Priority-Based Congestion Control Protocol  
Siphon (2005) Siphon  
Fusion (2004) Fusion  
CCF (2004) Congestion Control And Fairness  
Trickle(2004) Trickle  
CODA(2002) Congestion Detection And Avoidance  
ARC(2001) Adaptive Rate Control

**Πίνακας 5: Πρωτόκολλα επιπέδου Μεταφοράς WSN [46]**

Γνωστά πρωτόκολλα επιπέδου μεταφοράς που υποστηρίζουν είτε αξιόπιστη παράδοση είτε έλεγχο συμφόρησης είτε και τα δύο φαίνονται στον ανωτέρω πίνακα[46]

### 1.3.1.5 Επίπεδο Εφαρμογής

Το επίπεδο εφαρμογής σε ένα WSN λειτουργεί αφαιρετικά σχετικά με την υποκείμενη υποδομή, την τοπολογία και τα πρωτοκολλα ώστε να παρέχει στην εφαρμογή και τον χρήστη την διεπαφή να μπορέσει να διαδράσει με το WSN. Περιλαμβάνει την εφαρμογή την οποία εξυπηρετεί το WSN καθώς και διαχειριστικές λειτουργίες.



Στα WSN παρότι υιοθετείται η αρχιτεκτονική των επιπέδων, λόγω της φύσης των δικτύων που βασίζονται σε κόμβους με πολύ περιορισμένους πόρους, στο ασύρματο κανάλι που είναι ευάλωτο σε μεταβολές από πολλά διαφορετικά αίτια καθώς και στο γεγονός έχουμε δίκτυα που είναι οδηγούμενα από την εφαρμογή που εξυπηρετούν τελικά οδηγούν σε λύσεις που κάνουν χρήση μιας πιο ευέλικτης αρχιτεκτονικής που συνδυάζει τεχνικές μεταξύ επιπέδων. Πετυχαίνεται με αυτό τον τρόπο καλύτερη αποδοτικότητα και λειτουργικότητα των περιορισμένων πόρων του δικτύου.

### 1.3.2 Πρότυπα WSN

Σε ένα τόσο ευρύ πεδίο ενδιαφέροντος όπως είναι τα WSN που μπορούν να χρησιμοποιηθούν σε ένα πολύ μεγάλο φάσμα εφαρμογών ανθρώπινης δραστηριότητας για παρακολούθηση κάποιου φαινομένου, ανίχνευση συμβάντων, παρακολούθηση θέσης (tracking), ανίχνευση ταυτότητας γεγονότων καθώς και ενεργοποίησης μηχανισμών ανάλογα με το συμβάν, της ετερογένειας του υλικού σε συνάρτηση με το γεγονός ότι μπορεί να αναπτυχθούν σε πολλά διαφορετικά περιβάλλοντα είτε υπέργεια, υπόγεια, υποβρύχια κοκ θα πρέπει να υπάρχει κάποιος βαθμός προτυποποίησης των πρωτοκόλλων που χρησιμοποιούνται. Παρακάτω γίνεται μια αναφορά στα γνωστότερα πρότυπα που έχουν υλοποιηθεί για τον σκοπό αυτό.

#### 1.3.2.1 IEEE 802.15.4

Το πρωτόκολλο IEEE 802.15.4 (LR-WPAN) δημιουργήθηκε αρχικά το 2003 για να προσδιορίσει την ασύρματη επικοινωνία, χαμηλής ροής και περιορισμένης εμβέλειας μεταξύ σχετικά απλών συσκευών, με ελάχιστη κατανάλωση ενέργειας που η λειτουργία τους ορίζεται σε μια προσωπική περιοχή λειτουργίας. Είναι σχεδιασμένο για χαμηλού ρυθμού ασύρματα προσωπικά δίκτυα (Low Rate Wireless Personal Area Networks, LR-WPAN) και προδιαγράφει τα χαρακτηριστικά του φυσικού επιπέδου (PHY) και του υποεπιπέδου Ελέγχου Προσπέλασης στο Μέσο Μετάδοσης MAC (Medium Access Control) [47][49][50].

Σε αυτό στηρίχθηκαν αρκετά άλλα πρότυπα που επέκτειναν την λειτουργία του καθορίζοντας τα υπόλοιπα επίπεδα που δεν προδιαγράφονται στο IEEE 802.15.4, με μερικά από αυτά να είναι τα δημοφιλή Zigbee, 6LowPan, WirelessHart, ISA100.11 .

Στα χαρακτηριστικά του είναι η ευκολία στην εγκατάσταση, η λειτουργία περιορισμένης έκτασης, η αξιόπιστη μετάδοση δεδομένων και η λογική διάρκεια ζωής της μπαταρίας. Στην αρχική του έκδοση του 2003 ορίζονται δύο φυσικά επίπεδα βασισμένα στην τεχνική διαμόρφωσης διεύρυνσης φάσματος απευθείας ακολουθίας (Direct Sequence Spread Spectrum- DSSS) που προκαθορίζονται τρεις ζώνες συχνοτήτων στο ISM (Industrial Scientific Medical) φάσμα :

- Το ένα επίπεδο ορίζεται στη ζώνη των 2.4 GHz για καθολική (global) χρήση με ρυθμούς δεδομένων 250kbps, και
- Το άλλο επίπεδο ορίζεται στη ζώνη των 915 MHz για την Αμερική με ρυθμό δεδομένων 40kbps και των 868 MHz για την Ευρώπη με ρυθμό δεδομένων 20 kbps. Στην αναθεώρηση του 2006 αυξήθηκε ο ρυθμός μετάδοσης στις ζώνες 868 Mhz στα 100kbps και 915 Mhz στα 250kbps



Προδιαγράφει τα παρακάτω κανάλια επικοινωνίας:

- 16 κανάλια αντιστοιχούν στη ζώνη των 2.4 GHz,
- 10 κανάλια στη ζώνη των 915 MHz που στην αναθεώρηση του 2006 επεκτάθηκαν σε 30 και
- 1 κανάλι στη ζώνη των 868 MHz.

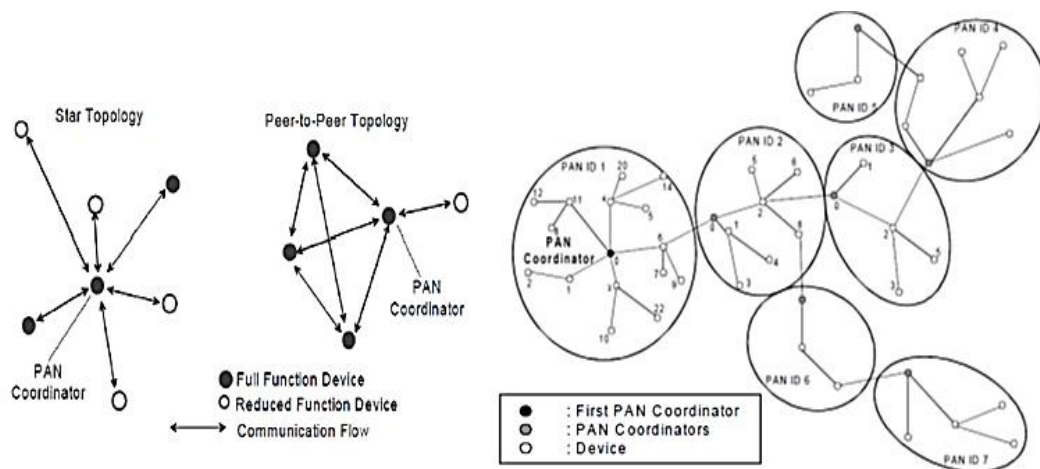
Σχετικά με τα θέματα διευθυνσιοδότησης ενός κόμβου, υπάρχει δυνατότητα χρήσης είτε μιας 64-bit IEEE διεύθυνσης ή μιας 16-bit σύντομης διεύθυνσης. Ένα μεμονωμένο IEEE 802.15.4 δίκτυο μπορεί να εξυπηρετήσει μέχρι  $65535(2^{16})$  συσκευές με την διεύθυνση 0xFFFF να είναι δεσμευμένη για μεταδόσεις εκπομπής.

Το φυσικό επίπεδο προσφέρει δύο υπηρεσίες στο υποεπίπεδο MAC μέσω των PHY data service και PHY management service. Είναι υπεύθυνο για:

- την διαχείριση του πομποδέκτη (transceiver) θέτοντας τον είτε σε κατάσταση εκπομπής, λήψης ή sleep,
- την ανίχνευση ενέργειας (Energy Detection, ED) καναλιού, που δίνει ένδειξη της ισχύος του λαμβανόμενου σήματος στο εύρος ζώνης του καναλιού
- την ένδειξη ποιότητας ζεύξης (Link Quality Indication, LQI) για κάθε πακέτο που λαμβάνεται. Χρησιμοποιεί την ανίχνευση ενέργειας ληφθέντος σήματος (ED), ένα ποσοστό σήματος προς θόρυβο ή ένα συνδυασμό αυτών για τη μέτρηση της ισχύος της ποιότητας μιας ζεύξης, μέσω της οποίας λαμβάνεται ένα πακέτο
- την εκτίμηση ελεύθερου καναλιού (Clear Channel Assessment-CCA) για πολλαπλή πρόσβαση με ανίχνευση φέροντος σήματος και αποφυγή συγκρούσεων (Carrier Sense Multiple Access with Collision Avoidance, CSMA/CA). Εκτελείται χρησιμοποιώντας την ανίχνευση ενέργειας (ED), ανίχνευση φέροντος σήματος ή συνδυασμό και των δύο.
- την επιλογή συχνότητας καναλιού εφόσον οι ασύρματες ζεύξεις που υποστηρίζει το πρότυπο 802.15.4 μπορούν να λειτουργήσουν σε πολλά, 27 αρχικά περισσότερα στις αναθεωρημένες εκδόσεις, διαφορετικά κανάλια.
- και την αποστολή και λήψη δεδομένων, στοιχειώδη διαδικασία του επιπέδου που αναλαμβάνει την διαμόρφωση και διάδοση του σήματος. Στο επίπεδο της συχνότητας 2,4GHz ακολουθείται διαμόρφωση 16-ary O-QPSK (orthogonal quadrature phase shift keying) ενώ στο επίπεδο των συχνοτήτων 868/915 ορίζεται διαμόρφωση BPSK (Binary Phase Shift Keying)

Για την τοπολογία των κόμβων στο IEEE 802.15.4 ορίζονται δύο τύποι συσκευών δικτύου. Οι συσκευές πλήρους λειτουργίας (Full Function Device - FFD) και οι συσκευές περιορισμένης λειτουργίας (Reduced Function Devices - RFD). Οι πρώτες οι οποίες λειτουργούν είτε ως συσκευές συντονισμού του δικτύου (Coordinators) είτε ως απλές συσκευές και μπορούν να αναμεταδίδουν μηνύματα σε άλλες FFD ή σε RFD. Οι συσκευές RFD ορίζονται απλές, με μικρές απαιτήσεις πόρων και επικοινωνίας και μπορούν να επικοινωνήσουν μόνο με FFD, ενώ δεν μπορούν να λειτουργήσουν ως συντονιστές.

Υποστηρίζονται τοπολογίες δικτύωσης είτε τύπου αστέρα είτε ομότιμου δικτύου (peer to peer). Απαιτείται και στις δύο περιπτώσεις τουλάχιστον μια συσκευή FFD για να λειτουργεί ως συντονιστής του δικτύου



Εικόνα 15: Τοπολογίες Star, Peer-to-Peer και Cluster-Tree [47]

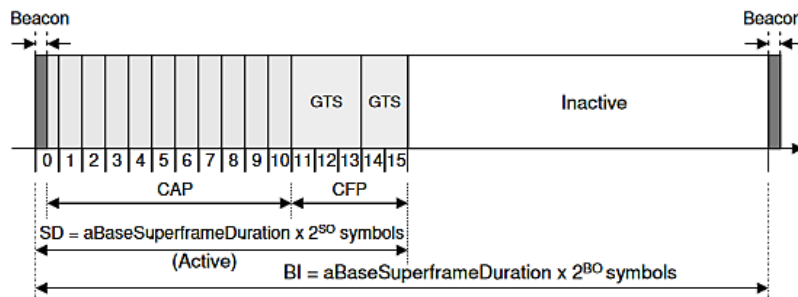
Το υποεπίπεδο ελέγχου πρόσβασης στο μέσο (MAC) εξασφαλίζει την διασύνδεση των ανώτερων επιπέδων με το φυσικό με την αποστολή των πλαισίων. Η δομή του πλαισίου (frame) έχει σχεδιαστεί ώστε μην είναι πολύπλοκο και να καθιστά αποτελεσματική την αναμετάδοση τους ακόμη και σε κανάλια με θόρυβο

Όπως το φυσικό προφέρει και αυτό δύο υπηρεσίες την υπηρεσία δεδομένων MAC και την υπηρεσία διαχείρισης MAC. Μια από τις διαδικασίες για τις οποίες είναι ο έλεγχος της πρόσβασης στο φυσικό μέσο με την χρήση τεχνικής Πολλαπλής Πρόσβασης με Ανίχνευση Φέροντος και Αποφυγή Συγκρούσεων (Carrier Sense Multiple Access with Collision Avoidance / CSMA-CA), μέσω της οποίας καθορίζεται η χρονική στιγμή που θα εκπέμψει μια συσκευή ή θα τεθεί σε αναμονή για την λήψη ενός πακέτου. Επίσης υλοποιούνται οι παρακάτω διαδικασίες σε αυτό:

- Δημιουργία των beacons (φάροι) στην περίπτωση που η συσκευή είναι συντονιστής. Το πρότυπο καθορίζει την προαιρετική χρήση υπέρ-πλασίων (superframes). Ένας συντονιστής επιλέγει αν θα ενεργοποιήσει τους beacons που μπαίνουν στην αρχή και στο τέλος του πλαισίου και καθορίζουν ότι ακολουθεί υπέρ-πλαίσιο
- Συγχρονισμός των συσκευών στο εισερχόμενο beacon ώστε να συγχρονίσουν οι συνδεδεμένες στον Coordinator συσκευές
- Υποστήριξη σύνδεσης και αποσύνδεσης προσωπικού δικτύου περιοχής (PAN) για να υποστηρίξει την αποδιάρθρωση. Ενσωματώνει την χρήση λειτουργιών σύνδεσης και αποσύνδεσης ώστε να ρυθμίζει αυτόματα είτε τοπολογία αστεριού, είτε αυτορυθμιζόμενου, ομότιμου δικτύου.
- Παροχή αξιόπιστης σύνδεσης μεταξύ δύο ομότιμων οντοτήτων MAC. Αναπτύσσονται μηχανισμοί όπως η αναγνώριση λήψης πλαισίου και αναμετάδοση, η επαλήθευση δεδομένων χρησιμοποιώντας CRC 16-bit,

καθώς και το CSMA-CA για την ενίσχυση της αξιοπιστίας της σύνδεσης μεταξύ δύο ομότιμων,

- Χειρισμός και συντήρηση του μηχανισμού εγγυημένου χρονικού διαστήματος (Guaranteed Time Slot - GTS): Όταν λειτουργεί σε κατάσταση ενεργού beacon, ένας coordinator μπορεί να ελέγξει τα τμήματα GTS του ενεργού υπέρ - πλαισίου μιας συσκευής που καθορίζουν την περίοδο χωρίς συγκρούσεις (Contention Free Period - CFP)
- Υποστήριξη των παραμέτρων ασφαλείας



Εικόνα 16: Δομή του υπέρ - πλαισίου MAC [50]

Το πρότυπο 802.15.4 υπόκειται σε συνεχείς αναθεωρήσεις και βελτιώσεις από την IEEE. Χαρακτηριστικά αναφέρεται η αναθεώρηση του 2006 στην οποία πέρα από την βελτίωση του ρυθμού μετάδοσης δεδομένων στις συχνότητες των 868/915 MHz, ορίζονται τέσσερα επίπεδα ανάλογα τη μέθοδο διαμόρφωσης που ακολουθείται. Στα τρία χρησιμοποιείται η τεχνική διαμόρφωσης DSSS ενώ για στις συχνότητες των 868/915MHz μπορεί να επιλεγεί ένας συνδυασμός παράλληλης και όχι απευθείας ακολουθίας διεύρυνσης φάσματος (binary keying και amplitude shift keying - PSSH)

Με την αναθεώρηση του 2007, κυκλοφόρησε η έκδοση 802.15.4a. Οι αλλαγές που ορίστηκαν αφορούν στην επέκταση των τεσσάρων ζωνών συχνοτήτων της αναθεώρησης 2006 σε έξι, όπου η μία κάνει χρήση τεχνικής Direct Sequence Ultra Wideband (UWB) και η άλλη τεχνικής Chirp Spread Spectrum (CSS). Η UWB κάνει χρήση τριών ζωνών συχνοτήτων με την πρώτη να είναι κάτω του 1 GHz, η επόμενη μεταξύ 3 GHz και 5 GHz και η τρίτη στο φάσμα 6 GHz και 10 GHz. Η CSS κάνει χρήση συχνοτήτων 2.4 GHz.

Αυτή την στιγμή είναι ενεργή η αναθεώρηση 802.15.4z του 2020 η οποία αντικατέστησε την 802.15.4n του 2017.

### 1.3.2.2 ZigBee

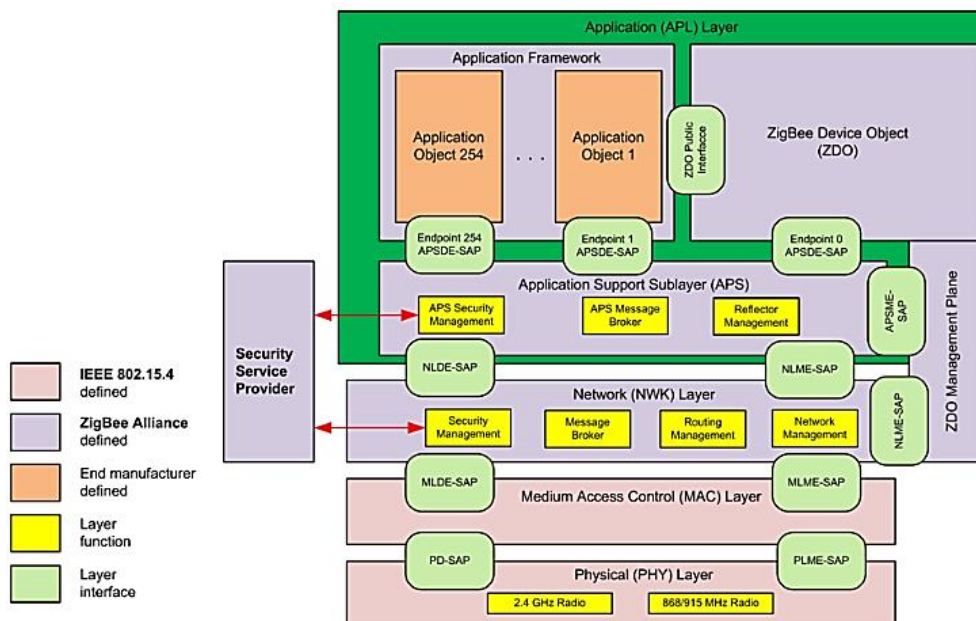
Το πρότυπο ZigBee έχει αναπτυχθεί από την ZigBee Alliance που είναι όμιλος από διεθνείς συνεργαζόμενες εταιρείες, και βασίζεται - επεκτείνει τις λειτουργίες του προτύπου IEEE 802.15.4[47] αναπτύσσοντας την τεχνολογία των επιπέδων που δεν ορίζονται σε αυτό. Είναι ένα ανοιχτό παγκόσμιο πρότυπο ασύρματης επικοινωνίας πολύ χαμηλού κόστους, πολύ χαμηλής κατανάλωσης ισχύος, αμφίδρομης, ασύρματης επικοινωνίας για εφαρμογές

χωρίς μεγάλες απαιτήσεις στους ρυθμούς μετάδοσης δεδομένων. Κύρια χρήση του είναι σε εφαρμογές έξυπνων πόλεων, όπως τα δίκτυα υγείας, υπηρεσίες αποθήκευσης, υπηρεσίες παρακολούθησης και υπηρεσίες ασφαλείας. [52],[53],[54]

Τον Δεκέμβριο 2004 επικυρώθηκε το προσχέδιο της έκδοσης 1.0 και αυτή τη στιγμή είναι ενεργή η αναθεώρηση r21 του προτύπου από τον Αυγουστο 2015.

Η αρχιτεκτονική του πρωτοκόλλου ZigBee ακολουθεί την διαστρωματοποιημένη αρχιτεκτονική των επιπέδων, με κάθε επίπεδο να προσφέρει το σύνολο των υπηρεσιών του προς τα ανώτερα επίπεδα. Χρησιμοποιεί οντότητες δεδομένων (data entity) που παρέχουν υπηρεσίες μετάδοσης δεδομένων και οντότητες διαχείρισης (management entity) για τις υπόλοιπες υπηρεσίες. Οι οντότητες παρέχουν μια διασύνδεση προς στο ανώτερο επίπεδο μέσω ενός σημείου πρόσβασης υπηρεσίας (Service Access Point - SAP), και κάθε σημείο πρόσβασης υπηρεσίας υποστηρίζει έναν αριθμό από βασικές λειτουργίες ώστε να επιτευχθεί η απαιτούμενη λειτουργικότητα.

Το IEEE 802.15.4 ορίζει τις τεχνικές προδιαγραφές των δύο κατώτερων επιπέδων του πρωτοκόλλου: του φυσικού επιπέδου (Physical - PHY) και του επιπέδου ελέγχου προσπέλασης μέσου (Medium Access Control - MAC). Τα ανώτερα επίπεδα της στοίβας, του επιπέδου δικτύου (Network Layer - NWK) και του πλαισίου για το επίπεδο εφαρμογών (Application Layer - APL) ορίζονται από την ZigBee Alliance[53].

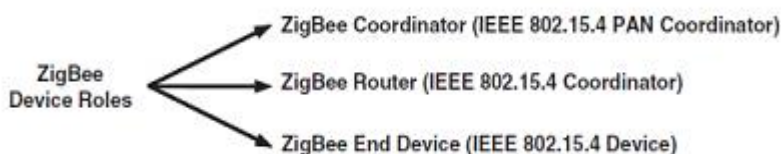


Εικόνα 17: Διαστρωμάτωση προτύπου Zigbee με το IEEE 802.15.4[53]

Ακολουθεί κατά βάση τα χαρακτηριστικά που προσδιορίζει το πρότυπο 802.15.4 Υποστηρίζει τοπολογίες αστέρα (star), δέντρου (tree) και πλέγματος (mesh) όπου οι κόμβοι συνεργάζονται και δρομολογούν δεδομένα μεταξύ τους, με αποτέλεσμα να επιτυγχάνονται υψηλού επιπέδου επικοινωνίες δεδομένων.

Χρησιμοποιούνται οι συσκευές που ορίζονται στο πρότυπο IEEE 802.15.4 δηλαδή συσκευές πλήρους λειτουργίας (FFD) που μπορούν να έχουν το ρόλο του συντονιστή ή και του δρομολογητή, και περιορισμένης λειτουργίας (RFD) που έχουν την δυνατότητα να

επικοινωνήσουν μόνο με FFD, ενώ υπάρχει ο ρόλος του συντονιστή δικτύου (Pan Coordinator) που χρησιμοποιείται για τον έλεγχο και την διαχείριση των δεδομένων που διακινούνται.



Εικόνα 18: Συσκευές που ορίζονται στο πρότυπο ZigBee [56]

Το επίπεδο δικτύου παρέχει κατάλληλη διεπαφή στο επίπεδο εφαρμογών για την λειτουργία του με τις οντότητες υπηρεσίας δεδομένων (NLDE) μέσω του σημείου πρόσβασης NLDE-SAP και την οντότητα υπηρεσίας διαχείρισης (NLME) μέσω του σημείου πρόσβασης NLME-SAP. Η οντότητα NLDE χρησιμοποιείται από την NLME για την παροχή μέρους των διαχειριστικών της λειτουργιών. Η NLME διατηρεί και διαχειρίζεται την βάση δεδομένων των πληροφοριών του δικτύου (Network Information Base – NIB).

Μερικές από τις αρμοδιότητες του επιπέδου δικτύου είναι όπως καθορίζονται είναι: να δημιουργεί ένα νέο δίκτυο, να επιτρέπει τη σύνδεση ή την αποχώρηση συσκευής σε υπάρχον δίκτυο, να ορίζει τις διευθύνσεις των συσκευών που συνδέονται στο δίκτυο ενός συντονιστή (coordinator), να καθορίζει το ρόλο μιας νέας συσκευής που συνδέεται στο δίκτυο, να ανακαλύπτει γειτονικές συσκευές δηλαδή με αυτές που μπορεί να επικοινωνήσει με μια μόνο μετάβαση, να αλλάζει μηχανισμό δρομολόγησης των δεδομένων ,να ανακαλύπτει τις πιο σύντομες διαδρομές.

Το επίπεδο εφαρμογής είναι το ανώτερο επίπεδο που ορίζεται από το πρότυπο ZigBee. Αποτελείται από το πλαίσιο εφαρμογής ZigBee (ZigBee Application Framework), τα αντικείμενα ZigBee (ZigBee Device Objects - ZDO) και το υπό-επίπεδο υποστήριξης (Application Support SubLayer – APS), όπως φαίνεται και στην Εικόνα 17. Το πλαίσιο εφαρμογής ZigBee μπορεί να περιλαμβάνει έως 254 Application Objects (APO), τμήματα λογισμικού της εφαρμογής, που ελέγχουν μία μονάδα υλικού που βρίσκεται στη συσκευή και αποτελούν μέρος της εφαρμογής. Τα αντικείμενα ZDO παρέχουν υπηρεσίες που επιτρέπουν στα APO να αναγνωρίζονται και να οργανώνονται σε μια κατανεμημένη εφαρμογή. [53],[54]

Το υπό-επίπεδο υποστήριξης εφαρμογών (APS) παρέχει διεπαφή μεταξύ του επιπέδου δικτύου (NWK) και του επιπέδου εφαρμογών (APL) μέσω ενός συνόλου υπηρεσιών που χρησιμοποιούνται από τα ZDO και από τα αντικείμενα εφαρμογών του σχεδιαστή. Οι υπηρεσίες παρέχονται από τις οντότητες, δεδομένων υπό-επιπέδου υποστήριξης (Application Support Sub-layer Data Entity – APSDE) μέσω του σημείου πρόσβασης υπηρεσιών APSDE (APSDE-SAP) και διαχείρισης υπό-επιπέδου Υποστήριξης (Application Support Sub-layer Management Entity - APSME) μέσω του σημείου πρόσβασης υπηρεσιών APSME (APSME-SAP). (Εικόνα 17)

Αρμοδιότητες του επιπέδου εφαρμογών αποτελούν η ανίχνευση συσκευών ZigBee στο δίκτυο, η διαχείριση του ρόλου των συσκευών, η διαχείριση των διευθύνσεών τους, η αξιόπιστη μεταφορά δεδομένων.

### 1.3.2.3 Πρωτόκολλο Bluetooth και Bluetooth Low Energy (BLE)

Το Bluetooth είναι ένα πρωτόκολλο που ορίζει ασύρματες επικοινωνίες μικρής εμβέλειας μεταξύ πολλών διαφορετικών τύπων συσκευών. Αρχικά αναπτύχθηκε από την Ericsson Mobile Communications το 1994, για χρήση μεταξύ κινητών τηλεφώνων και των αξεσουάρ τους. Το 1998, η IBM, η INTEL, η Nokia και η Toshiba συνεργάστηκαν με την σουηδική εταιρεία και δημιούργησαν την Ομάδα Ειδικού Ενδιαφέροντος (Bluetooth Special Interest Group – SIG) που διευρύνθηκε με την άφιξη επιπλέον πολλών μεγάλων εταιρειών όπως οι 3Com, Microsoft, Motorola καθώς και πολλές ακόμα που σήμερα φτάνουν περίπου τις 35 χιλ. εταιρείες και οργανισμούς.

Είναι ανοιχτή βιομηχανική προδιαγραφή και τυποποιήθηκε από τον IEEE με το πρότυπο IEEE 802.15.1 το οποίο όμως από τις Μάιο του 2018 το έχει ανακαλέσει. Καθορίζει ασύρματες επικοινωνίες σε μικρή εμβέλεια στο φάσμα των ISM συχνοτήτων από 2,402GHz έως 2,480GHz και την δημιουργία ασύρματων δικτύων προσωπικής περιοχής (WPANs). Μεταδίδει δεδομένα σε πακέτα σε κάθε κανάλι από τα 79 Bluetooth κανάλια με εύρος 1MHz καθένα. Αρχικά υιοθετήθηκε διαμόρφωση σήματος (Gaussian frequency shift keying –GFSK) με ρυθμούς μετάδοσης μέχρι 1Mbps με ονομασία Βασικός Ρυθμός Μετάδοσης (Basic Rate – BR), αλλά από την έκδοση Bluetooth 2.0 με την χρήση σχημάτων διαμόρφωσης  $\pi/4$ -DPSK και 8-DPSK έγιναν εφικτοί ρυθμοί μετάδοσης της τάξης των 2 και 3 Mbps αντίστοιχα με ονομασία Βελτιωμένος Ρυθμός Μετάδοσης (Enhanced Data Rate – EDR) . [55], [57]

Χρησιμοποιείται ευρέως σε υπολογιστικές συσκευές όπως κινητά τηλέφωνα, ταμπλέτες, φορητούς ηλεκτρονικούς υπολογιστές για την επικοινωνία με περιφερειακές συσκευές όπως φορητά ηχεία, ακουστικά κ.α.

Το πρωτόκολλο Bluetooth στην έκδοση BR/EDR δεν είναι προσαρμοσμένο για χρήση σε συσκευές με τα χαρακτηριστικά των κόμβων WSN και δεν θεωρείται κατάλληλο για τέτοια δίκτυα.

Παράλληλα με την εξέλιξη του πρωτοκόλλου Bluetooth BR/EDR αναπτύχθηκε τα τελευταία χρόνια το πρότυπο Bluetooth Low Energy – BLE. Ξεκίνησε αρχικά από την εταιρεία Nokia το 2006 με το όνομα Wibree. Στην εξέλιξη του υιοθετήθηκε από το Bluetooth SIG και τον Δεκέμβριο 2009 ενσωματώθηκε στο Bluetooth 4.0 με την τρέχουσα ονομασία του Bluetooth Low Energy - BLE.

Το BLE είναι μια ασύρματη τεχνολογία που προορίζεται για επικοινωνίες μικρής εμβέλειας χαμηλής ισχύος, χαμηλής πολυπλοκότητας και χαμηλού κόστους που αποτελούν μια ενδιαφέρουσα λύση για εφαρμογές δικτύων WSN.

Στο [62] επιχειρείται μια μελέτη σχετικά με την κατανάλωση του BLE συγκριτικά με το πρωτόκολλο ZigBee/802.15.4, και συμπερασματικά καταλήγει ότι το BLE έχει πραγματικά πάρα πολύ μικρή κατανάλωση και πετυχαίνει ένα πολύ καλό λόγο κατανάλωσης για κάθε μεταδιδόμενο bit.

Για την αντιμετώπιση παρεμβολών και εξασθένησης χρησιμοποιείται και στο BLE όπως στο κλασικό Bluetooth τεχνολογία αναπήδησης συχνότητας (Frequency Hopping Spread Spectrum - FHSS) οπότε οι συσκευές για να επικοινωνήσουν συγχρονίζονται στο ίδιο κανάλι συχνότητας την ίδια χρονική στιγμή. Η διαφορά είναι ενώ στο κλασικό Bluetooth η



αλλαγές συχνότητας συμβαίνουν για κάθε πακέτο με συχνότητα 1600 φορές το second στο BLE οι αλλαγές συμβαίνουν ανά γεγονός σύνδεσης(Connection Event) με μέγιστο μήκος 4 sec. Το εύρος συχνοτήτων που χρησιμοποιείται είναι και εδώ 2400 - 2485 MHz αλλά η κατανομή καναλιών είναι 40 κανάλια με εύρος με 2 MHz το καθένα. [62]

Property	Classic Bluetooth	Bluetooth Low Energy
Optimized for	Continuous data streaming [22]	Short burst data transmission [22]
Frequency Range	2400 . . . 2483.5 MHz	
Channel Allocation	79 channels $f = 2402 + k$ MHz ( $k=0\dots 78$ )	40 channels $f = 2402 + k*2$ MHz ( $k=0\dots 39$ )
Modulation	GFSK (BR) $\pi/4$ DQPSK (EDR)-8 DPSK (EDR)	GFSK
Data Rate	up to 723.2 kbps (BR) up to 2178.1 kbps (EDR)	up to 236.7 kbps
TDMA Units	Fix sized (time) slots	Variable sized events
FHSS	Per packet	Per event
Connectionless Communication	No	Yes
Device Roles	Inquiring device, inquiry scanner, paging device, page scanner, master- slave	Advertiser, scanner, initiator, master-slave
Explicit Low Power Modes/State	Yes	No
Piconet Communication Topology	Star	
Scatternet Support	Yes	Yes
Slave Role: Number of Connections $\geq 1$	Yes	Yes
Dual Role	Yes	Yes
Role Switch	Yes	No

Πίνακας 6: Σύγκριση Bluetooth με BLE [58], [62]

Παρότι το πρωτόκολλο BLE αρχικά υποστήριζε μόνο τοπολογίες αστέρα όπως και το κλασικό BR/EDR Bluetooth και μεταδόσεις μονού βήματος(single hop) μεταξύ ενός master –slave συσκευών το Bluetooth SIG υιοθέτησε Ιούλιο 2017 τις προδιαγραφές Mesh Profile και Mesh Model για την υλοποίηση δικτύωσης πλέγματος πολλαπλών βημάτων.

Το Mesh Profile BLE βασίζεται σε μια πλήρη στοίβα BLE και χρησιμοποιεί τις βασικές δυνατότητες ασύρματης επικοινωνίας του BLE. Παρέχει μια επιλογή τοπολογίας για συσκευές BLE όπου όλες οι συσκευές είναι σε θέση να επικοινωνούν μεταξύ τους υπό την προϋπόθεση ότι το επιτρέπει η φυσική τοπολογία. Σχεδιάστηκε για να υποστηρίξει multi- hop δίκτυα μεγάλης κλίμακας και, ως εκ τούτου, είναι ιδανικό για δίκτυα αισθητήρων, παρακολούθηση στοιχείων και οποιαδήποτε άλλη περίπτωση χρήσης που απαιτεί την ασφαλή επικοινωνία λίγων έως μερικών χιλιάδων συσκευών [62].

Οι συσκευές ενός δικτύου πλέγματος BLE ονομάζονται κόμβοι και συσχετίζονται μεταξύ τους σε ένα δίκτυο πλέγματος με την διαδικασία του “provisioning”. Στο τέλος αυτής της διαδικασίας ένας κόμβος έχει ένα κλειδί δικτύου που αντιπροσωπεύει την ιδιότητα μέλους του κόμβου στο συγκεκριμένο δίκτυο πλέγματος.

Ένας κόμβος πλέγματος BLE μπορεί να αποτελείται από πολλά στοιχεία. Τα στοιχεία μπορούν να βρίσκονται σε διάφορες καταστάσεις που αντιπροσωπεύονται από τιμές

κατάστασης. Ένα στοιχείο που εκθέτει μια κατάσταση είναι ένας διακομιστής ενώ ένα στοιχείο που αποκτά πρόσβαση σε μια κατάσταση είναι ένας πελάτης.

Η επικοινωνία στα δίκτυα πλέγματος BLE είναι προσανατολισμένη στα μηνύματα. Τα μηνύματα χρησιμοποιούνται για την ενεργοποίηση λειτουργιών σε μια τιμή κατάστασης ή σε ένα σύνολο τιμών κατάστασης. Υπάρχουν τρεις τύποι μηνυμάτων: GET, SET και STATUS. Τα GET μηνύματα χρησιμοποιούνται για να ζητήσουν την τιμή μιας δεδομένης κατάστασης. Τα μηνύματα STATUS αποστέλλονται ως απάντηση στα μηνύματα GET και τα μηνύματα SET χρησιμοποιούνται για την τροποποίηση της τιμής μιας κατάστασης.

Στο προφίλ BLE Mesh καθορίζονται επικοινωνίες unicast και δύο τύποι διευθύνσεων ομάδας. Οι διευθύνσεις ομάδας είναι διευθύνσεις multicast που αντιπροσωπεύουν ένα ή περισσότερα στοιχεία που ανήκουν σε έναν ή περισσότερους κόμβους. Τα μηνύματα αποστέλλονται συνήθως σε διευθύνσεις ομάδας.

Το μοντέλο επικοινωνίας σε δίκτυα πλέγματος BLE είναι τύπου publish /subscribe: Ο κόμβος που αποστέλλει ένα μήνυμα το δημοσιεύει και οι κόμβοι που λήψης αποτελούν τους συνδρομητές οι οποίοι έχουν ρυθμιστεί ώστε να λαμβάνουν μηνύματα που αποστέλλονται σε καθορισμένες διευθύνσεις.

Οι κόμβοι είναι δυνατόν να στέλνουν και να λαμβάνουν μηνύματα. Υπάρχει όμως περίπτωση ένας κόμβος να παρέχει πρόσθετες υπηρεσίες. Καθορίζονται 4 τύποι υπηρεσιών από το προφίλ BLE Mesh: Relay, Proxy, Friend και Low Power.

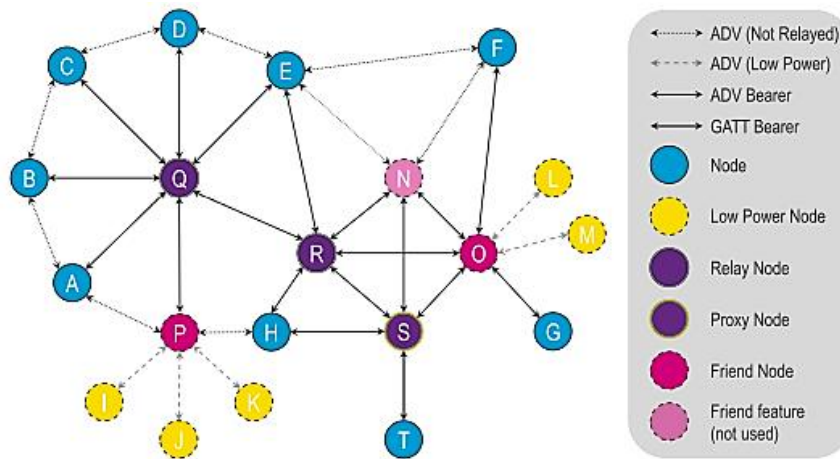
Ο Relay κόμβος μπορεί να προωθεί τα μηνύματα που λαμβάνει και να ενεργοποιεί με αυτό τον τρόπο την Multi-hop επικοινωνία.

Οι Proxy κόμβοι παρέχουν μια διεπαφή γενικών χαρακτηριστικών για την διασύνδεση κόμβων BLE που δεν υποστηρίζουν το προφίλ Mesh, να διασυνδεθούν σε ένα τέτοιο δίκτυο.

Οι υπηρεσίες Friend και Low Power λειτουργούν παράλληλα σε δύο κόμβους. Ο κόμβος Low Power λειτουργεί σε έναν κύκλο χαμηλής ισχύος για να εξοικονομήσει όσο το δυνατόν περισσότερη ενέργεια. Ωστόσο, πρέπει να μπορεί να λαμβάνει μηνύματα ανά πάσα στιγμή. Ο κόμβος Friend αναλαμβάνει την λήψη των μηνυμάτων που προορίζονται για τον κόμβο Low Power και τα αποθηκεύει προσωρινά. Σε μια φάση ενεργούς λειτουργίας, ο κόμβος χαμηλής ισχύος επικοινωνεί με τον κόμβο Friend για χαμένα μηνύματα. Η σχέση μεταξύ των δύο κόμβων ονομάζεται Friendship στο πλαίσιο του προφίλ BLE Mesh.

Στην εικόνα που ακολουθεί αναπαριστάται ένα χαρακτηριστικό δείγμα της τοπολογίας που περιγράφεται από το προφίλ BLE Mesh με όλες τις πιθανές υπηρεσίες που ενδέχεται να παρέχει ένας κόμβος.





Εικόνα 19: Δείγμα τοπολογίας multi-hop του προφίλ BLE Mesh [62]

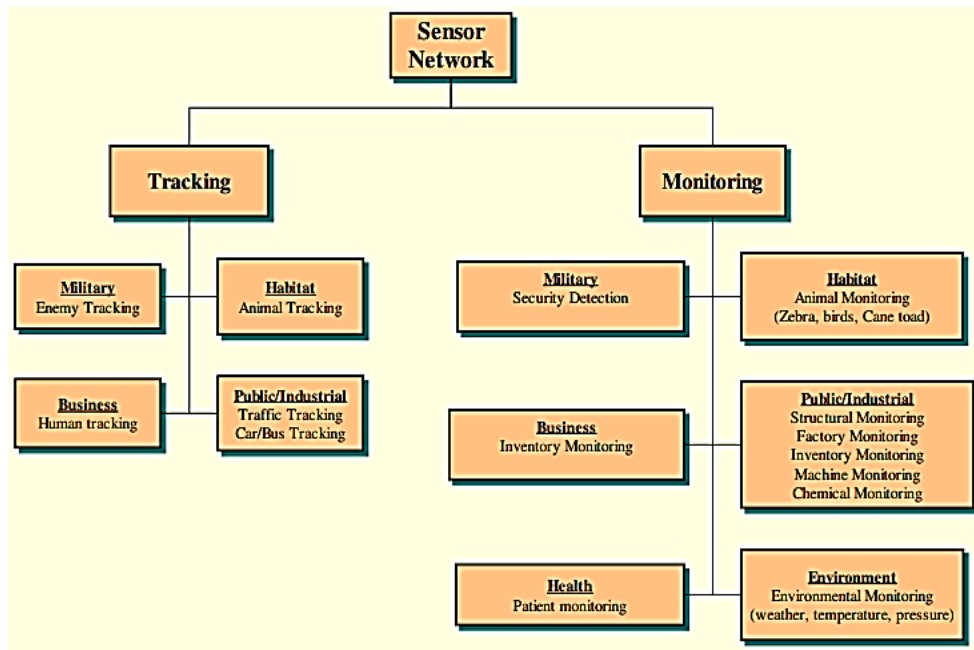
## 1.4 Εφαρμογές Ασύρματων Δικτύων Αισθητήρων

Από την πρώτη εμφάνιση των Ασύρματων Δικτύων Αισθητήρων που όπως και άλλες τεχνολογίες ξεκίνησαν ως στρατιωτική εφαρμογή μέχρι σήμερα έχει περάσει μεγάλο χρονικό διάστημα και σε αυτό ωρίμασε και η τεχνολογία τους. Πρότυπα που καθορίζουν αρχές επικοινωνίας και λειτουργίας έχουν αναπτυχθεί και σταδιακά υλοποιούνται εφαρμογές σε μεγάλο εύρος, σε όλα σχεδόν τα πεδία ανθρώπινης δραστηριότητας. Αν έπρεπε κάποιος να απαριθμήσει τις σύγχρονες εφαρμογές WSN θα ήταν πρακτικά αδύνατο αφού είναι ένας τομέας που διαρκώς εμπλουτίζεται με νέες.

Όπως ήδη έχει αναφερθεί στο πρώτο κεφάλαιο ένα WSN μπορεί να αποτελείται από κόμβους που έχουν τη δυνατότητα να παρακολουθούν ένα ευρύ φάσμα φαινομένων που μπορεί να περιλαμβάνει την κίνηση ενός αντικειμένου, τα επίπεδα θορύβου, το φως, την θερμοκρασία, την υγρασία, την παρουσία ή απουσία ουσιών όπως χημικών, βιολογικών, πυρηνικών καθώς και αντικειμένων ή ακόμα και την μηχανική καταπόνηση στην οποία υπόκειται κάποιο αντικείμενο και ανάλογα να αποστέλλουν τα δεδομένα τους για ενημέρωση ή/και περαιτέρω επεξεργασία ή να ενεργοποιούν κάποια διάταξη με την χρήση κατάλληλων ενεργοποιητών. Για τον σκοπό αυτό χρησιμοποιούν μια ευρεία γκάμα αισθητήριων οργάνων και διατάξεων ενεργοποίησης.

Γίνεται εύκολα αντιληπτό ότι οι δυνατότητες για ανάπτυξη εφαρμογών WSN είναι ουσιαστικά απεριόριστες σε όλα τα πεδία. Στην βιβλιογραφία εντοπίζονται αναφορές για ένα μεγάλο τμήμα αυτών είτε πρόκειται για ώριμες τεχνολογίες που χρησιμοποιούνται για μεγάλο χρονικό διάστημα είτε για τεχνολογίες που βρίσκονται στα σπάργανα.

Συνήθως γίνεται διαχωρισμός των εφαρμογών σε κατηγορίες με βάση διάφορα χαρακτηριστικά τους και οι κατηγορίες ποικίλλουν. Κατηγοριοποίηση εφαρμογών με βάση το περιβάλλον και το είδος των WSN που αναπτύσσονται δηλαδή επίγεια, υπόγεια, υποβρύχια, πολυμεσικά και κινητά αλλά αναφέρεται και κατηγοριοποίηση ανάλογα με το αν είναι επιφορτισμένες με την παρακολούθηση - επίβλεψη της εξέλιξης ενός φαινομένου (monitoring) ή τον εντοπισμό - ανίχνευση (tracking) [7].



Εικόνα 20:Κατηγοριοποίηση με βάση την παρακολούθηση [7]

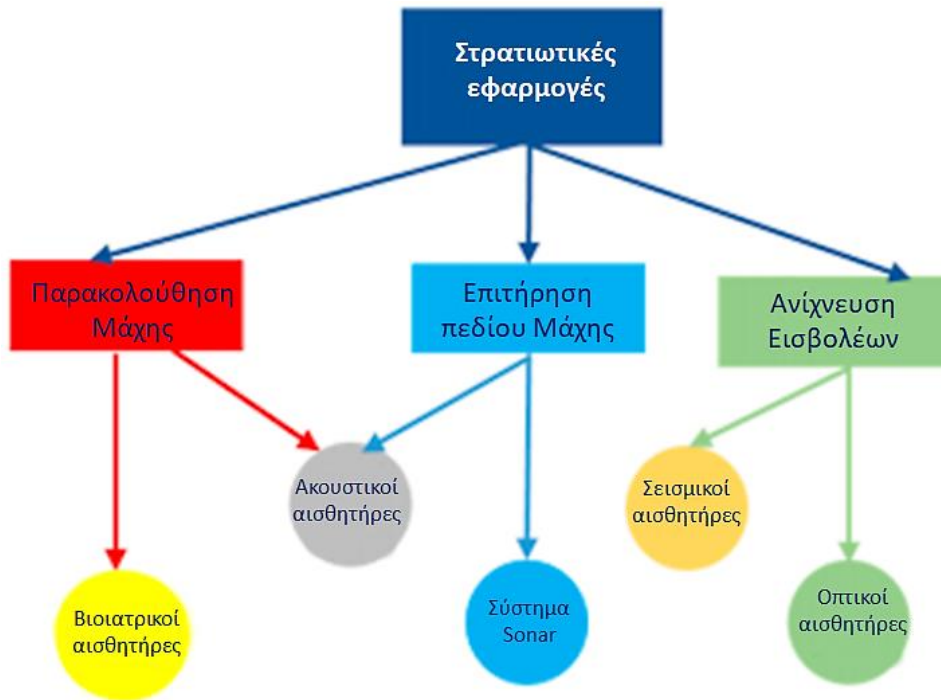
Στο [66] γίνεται μια αναλυτική περιγραφή και κατηγοριοποίηση των περισσότερων περιπτώσεων. Αναφέρονται οι πιο δημοφιλείς κατηγορίες με βάση το είδος της χρήσης τους: στρατιωτικών εφαρμογών, αστικών εφαρμογών, βιομηχανικών εφαρμογών, σχετικές με την χλωρίδα και την πανίδα, περιβαλλοντικές και εφαρμογές σχετικές με την υγεία.



Εικόνα 21: Δημοφιλείς κατηγορίες εφαρμογών WSN[66]

Το πεδίο των στρατιωτικών εφαρμογών ήταν το πρώτο εστίασαν οι ερευνητικές προσπάθειες από την δεκαετία του 90 με το project Smart Dust [4] με στόχο την ανάπτυξη κόμβων σε μέγεθος κυβικών χιλιοστών για την ανάπτυξή τους σε εχθρικά περιβάλλοντα. Το συγκεκριμένο project έχει χρησιμοποιηθεί σε μια πληθώρα στρατιωτικών εφαρμογών από παρακολούθηση μεταφορών και παρακολούθηση πεδίου μάχης μέχρι την παρακολούθηση της θερμοκρασίας και της υγρασίας τροφών όπως γαλακτοκομικών προϊόντων και κρέατος. Έκτοτε η τεχνολογική εξέλιξη οδήγησε στην ανάπτυξη WSN για πολλών διαφορετικών τύπων στρατιωτικές εφαρμογές που πολλές από αυτές λόγω του ιδιαίτερα ευαίσθητου τομέα τους

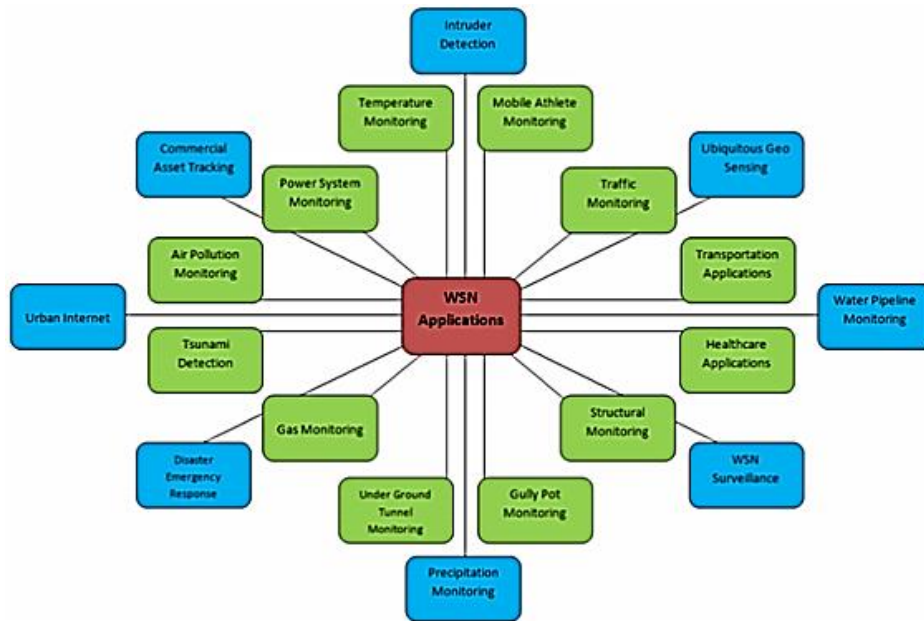
μπορεί να παραμείνουν απόρρητες για πάντα. Με την χρήση κατάλληλων αισθητήριων οργάνων μπορούν να αναπτυχθούν πολλών διαφορετικών ειδών εφαρμογές για στρατιωτική χρήση με μερικές δημοφιλείς να είναι: ανίχνευση και αναγνώριση επιθέσεων, παρακολούθηση επικοινωνιών, επιτήρηση πεδίου μάχης, παρακολούθηση κίνησης στρατευμάτων, παρακολούθηση εξοπλισμών, αναγνώριση και στοχεύσεων. Στο [66] κατηγοριοποιούνται οι στρατιωτικές εφαρμογές καθώς και οι αισθητήρες που χρησιμοποιούνται κατά περίπτωση και στην εικ.20



Εικόνα 22: Υποκατηγορίες στρατιωτικών εφαρμογών WSN[66]

Στα [68], [69], [70], [71] αναφέρονται ενδεικτικά κάποιες στρατιωτικές εφαρμογές. Στο [68] περιγράφεται το σύστημα ανίχνευσης ελεύθερων σκοπευτών Boomerang που μπορεί να ανιχνεύσει με ακρίβεια την θέση ελεύθερων σκοπευτών και χρησιμοποιείται από στρατό και δυνάμεις τήρησης της τάξης.

Στο [69] και στο [70] περιγράφονται παρόμοιας φιλοσοφίας συστήματα παρακολούθησης, το VigilNet μια εφαρμογή παρακολούθησης στόχων σε δύσκολα περιβάλλοντα που ανιχνεύουν το μαγνητικό πεδίο από την κίνηση μεταλλικών οχημάτων, με μαγνητικούς αισθητήρες και μια εφαρμογή επιτήρησης εδάφους με την χρήση ακουστικών και μαγνητικών αισθητήρων. Στο [71] μια εφαρμογή για την παρακολούθηση της υγείας των στρατευμάτων με την χρήση ενός Δικτύου Αισθητήρων Σώματος (Body Sensor Network – BSN).



Εικόνα 23: Κατηγορίες εφαρμογών για αστικά περιβάλλοντα[67]

Οι εφαρμογές που είναι σχετικές την αντιμετώπιση θεμάτων σχετικών με τις δραστηριότητες του ανθρώπου σε αστικές περιοχές και μπορούν να βελτιώσουν την ποιότητα ζωής του ανθρώπου περιγράφονται αναλυτικά στο [67] από όπου και η εικόνα. Εφαρμογές που έχουν αναπτυχθεί μέχρι σήμερα παρουσιάζονται καθώς και λύσεις που μπορούν να δοθούν σε προβλήματα αστικού περιβάλλοντος που ενδεχομένως έχουν αντιμετωπιστεί με τρόπους κοστοβόρους και ασύμφορους. Γίνεται αναλυτική μελέτη σε εφαρμογές που έχουν αναπτυχθεί ή μπορούν να αναπτυχθούν σχετικά θέματα όπως τα συστήματα και τα δίκτυα ηλεκτροδοτήσεως, τα δίκτυα ύδρευσης και λυμάτων, τις αστικές μεταφορές, σε εφαρμογές διαχείρισης καταστροφών και ασφαλείας, στην παρακολούθηση της μόλυνσης του αστικού περιβάλλοντος καθώς και στην παρακολούθηση της δομικής ακεραιότητας κτιρίων και άλλων δομικών κατασκευών (Structural monitoring).

Στον βιομηχανικό τομέα μπορούν να αναπτυχθούν εφαρμογές WSN για να διαχειριστούν θέματα σχετικά με τα logistics, την ρομποτική και της εφαρμογές της στην βιομηχανία και την επιτήρηση υγείας μηχανών που σκοπό έχει την παρακολούθηση λειτουργίας του τεχνικού εξοπλισμού και την πρόληψη δυσλειτουργιών και σφαλμάτων.

Στον τομέα της διαχείρισης της χλωρίδας και πανίδας που είναι πολύ σημαντικός για την οικονομία αλλά και για το βιοτικό επίπεδο των κατοίκων μιας χώρας χρησιμοποιούνται εφαρμογές WSN για την παρακολούθηση των καλλιεργειών είτε πρόκειται για υπαίθριες καλλιέργειες είτε για καλλιέργειες θερμοκηπίων καθώς και στην κτηνοτροφία.

Σημαντική εφαρμογή βρίσκουν τα WSN στην κατηγορία της παρακολούθησης περιβαλλοντικών φαινομένων ειδικά σε ακραίες συνθήκες και σε απομακρυσμένες περιοχές. Οι κατηγορίες των εφαρμογών εδώ είναι παρακολούθηση υδάτινων πόρων είτε πόσιμων είτε θαλασσίων, η παρακολούθηση της ποιότητας του ατμοσφαιρικού αέρα και η παρακολούθηση φυσικών φαινομένων για την πρόληψη η και καλύτερη διαχείριση φυσικών καταστροφών. Εδώ εφαρμογές έχουν υπάρξει στο θέμα της παρακολούθησης σεισμικής δραστηριότητας, της παρακολούθησης ηφαιστειακής δραστηριότητας, της πρόληψης των

δασικών πυρκαγιών καθώς και της παρακολούθησης και έγκαιρης ειδοποίησης σε περιπτώσεις παλιρροϊκών κυμάτων (tsunami).

Τέλος στον τομέα της υγείας έχουν αναπτυχθεί εφαρμογές που κατηγοριοποιούνται σχετικά με την παρακολούθηση ασθενών εντός κάποιας νοσοκομειακής μονάδας, την παρακολούθηση και υποβοήθηση ασθενών στο σπίτι και την παροχή δεδομένων σε πραγματικό χρόνο με την χρήση συστημάτων που μπορεί ο ασθενής να φέρει επάνω του (wearable monitoring).

## Κεφάλαιο 2: Η πλατφόρμα BBC micro:bit

### 2.1. Εισαγωγή

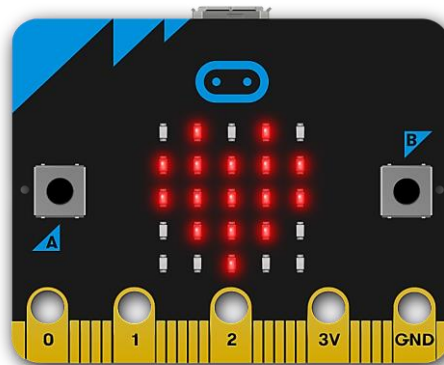
Στην εργασία αυτή έχει επιλεγεί το BBC micro:bit V1.5, ως μια πλατφόρμα η οποία θα μπορούσε να χρησιμοποιηθεί για την ανάπτυξη WSN εφαρμογών. Η συγκεκριμένη πλατφόρμα έχει δημιουργηθεί ως εκπαιδευτικό εργαλείο εκμάθησης βασικών δεξιοτήτων προγραμματισμού και ψηφιακών επικοινωνιών και δεν αποτελεί αμιγώς μια πλατφόρμα ανάπτυξης WSN. Παρά ταύτα το μέγεθός της, το υλικό της, ο συνδυασμός των ενσωματωμένων αισθητήρων της με το γεγονός ότι μπορεί πολύ εύκολα επεκταθούν με πρόσθετους καθώς και το υποστηριζόμενο λογισμικό και η στοίβα των υποστηριζόμενων πρωτοκόλλων την καθιστούν κατάλληλη για ένα πλήθος εφαρμογών σε πολλά διαφορετικά πεδία δραστηριότητας.

### 2.2 Τι ακριβώς είναι η πλατφόρμα BBC micro:bit

Το 2015 ο οργανισμός BBC (British Broadcasting Corporation) στα πλαίσια της καμπάνιας Make It Digital που σκοπό της έχει να εισάγει τα νέα παιδιά στις ψηφιακές δεξιότητες και να τα εμπνεύσει να ασχοληθούν με τον προγραμματισμό και την ψηφιακή τεχνολογία συνεργάστηκε με άλλους 28 εταίρους, πολύ γνωστούς οργανισμούς αλλά και εταιρείες, και δημιούργησαν το BBC micro:bit. Η καμπάνια εμπνεύστηκε από την επιτυχία του *BBC Micro* ενός 8bit home computer που την δεκαετία του '80 βοήθησε στην εκπαίδευση των πολλών νέων ανθρώπων στον ψηφιακές δεξιότητες σε συνδυασμό με σειρές μαθημάτων που προβαλλόταν από την εκπαιδευτική τηλεόραση του BBC. Στα πλαίσια της καμπάνιας Make It Digital από το 2016 και μετά μοιράστηκαν δωρεάν πάνω από ένα εκατομμύριο micro:bits σε μικρούς μαθητές στο Ηνωμένο Βασίλειο.



Εικόνα 24: Ο οικιακός υπολογιστής BBC Micro



Εικόνα 25: BBC micro:Bit (front)

Η βασική φιλοσοφία για την δημιουργία της πλατφόρμας ήταν να βοηθήσει τους νέους να εμπλακούν πιο ενεργά στην διαδικασία ανάπτυξης λογισμικού παρά στην απλή χρήση. Το BBC micro:bit ακολουθεί την σχεδιαστική φιλοσοφία του open source hardware καθώς και των single board microcontrollers που έγινε ιδιαίτερα δημοφιλής σε πολλές εφαρμογές με την πλατφόρμα Arduino.

Ο σχεδιασμός της πλατφόρμας είναι ελκυστικός με αρκετά παιγνιώδη χαρακτηριστικά. Για την διασύνδεση της με υπολογιστή αρκεί ένα τυπικό καλώδιο micro USB και προσαρμόζεται ως εξωτερική μονάδα δίσκου χωρίς την ανάγκη λογισμικού προγράμματος οδήγησης.

## 2.3 Χρήση του micro:bit ως πλατφόρμα για την ανάπτυξη εφαρμογών WSN

Το micro:bit σχεδιάστηκε και αναπτύχθηκε για εκπαιδευτική χρήση, χωρίς να έχει ένα αμιγώς προσανατολισμένο χαρακτήρα για εφαρμογές WSN. Μετά από την περιγραφή του υλικού, του λογισμικού που υποστηρίζει και των ευρύτερων δυνατοτήτων του, παρατηρούμε ότι έχει όλα τα χαρακτηριστικά να χρησιμοποιηθεί ως μια πλατφόρμα για την ανάπτυξη εφαρμογών WSN και IoT.

### 2.3.1 Τεχνικά χαρακτηριστικά - Το υλικό του micro:bit

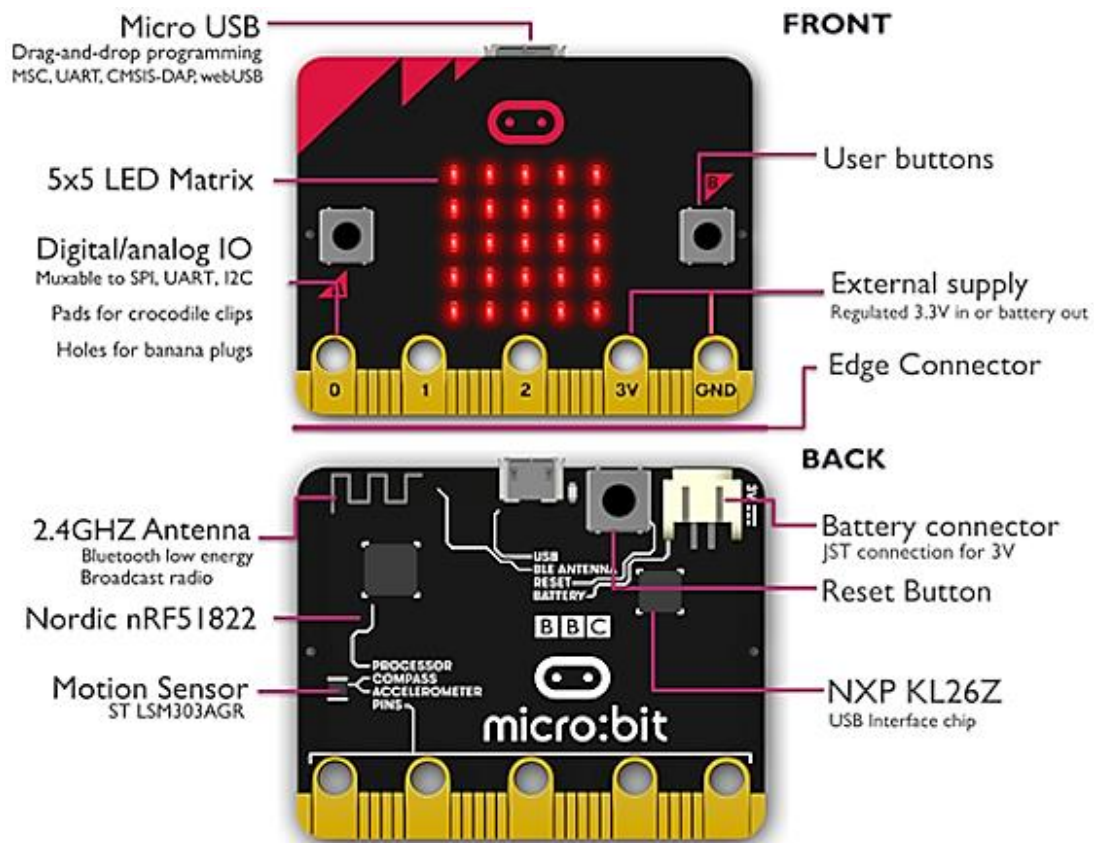
Το BBC micro:bit παράγεται σε συσκευασία πλακέτας τυπωμένου κυκλώματος (pcb) με μέγεθος 4 x 5 εκατοστά, που είναι περίπου όσο το μισό μιας πιστωτικής κάρτας.

Μπορεί να τροφοδοτηθεί από μια ποικιλία πηγών όπως δύο μπαταρίες AAA, μια μπαταρία μεγέθους κουμπιού, σύνδεση micro USB ή οποιαδήποτε άλλη πηγή 3 βολτ.

Είναι εξοπλισμένο, στην μπροστινή του όψη, με δύο προγραμματιζόμενα buttons ελέγχου και στην πίσω όψη είναι εφοδιασμένο με ένα ακόμα button με λειτουργία Reset. Τα μπροστινά κουμπιά A και B μπορούν να προγραμματιστούν στην εφαρμογή χρήστη για οποιονδήποτε σκοπό. Λειτουργούν με μια τυπική ανεστραμμένη ηλεκτρική λειτουργία, όπου μια αντίσταση pull-up εξασφαλίζει ένα λογικό «1» κατά την απελευθέρωση του κουμπιού και ένα λογικό «0» όταν πατηθεί το κουμπί.

Στην μπροστινή όψη βρίσκεται τοποθετημένο και ένα display αποτελούμενο από εικοσιπέντε led επιφανειακής στήριξης τα οποία είναι δομημένα σε πλέγμα 5x5. Τα led του display προγραμματίζονται το καθένα ανεξάρτητα, και επιτρέπουν την εμφάνιση μηνυμάτων και υποτυπωδών γραφικών. Το ίδιο display χρησιμοποιείται επίσης και ως αισθητήρας φωτεινότητας που μετράει το φως περιβάλλοντος. Δίνεται η δυνατότητα μέσω κατάλληλων εντολών να οδηγείται από τον χρήστη ως έξοδος ή να το διαβάζει ως είσοδο. [72]





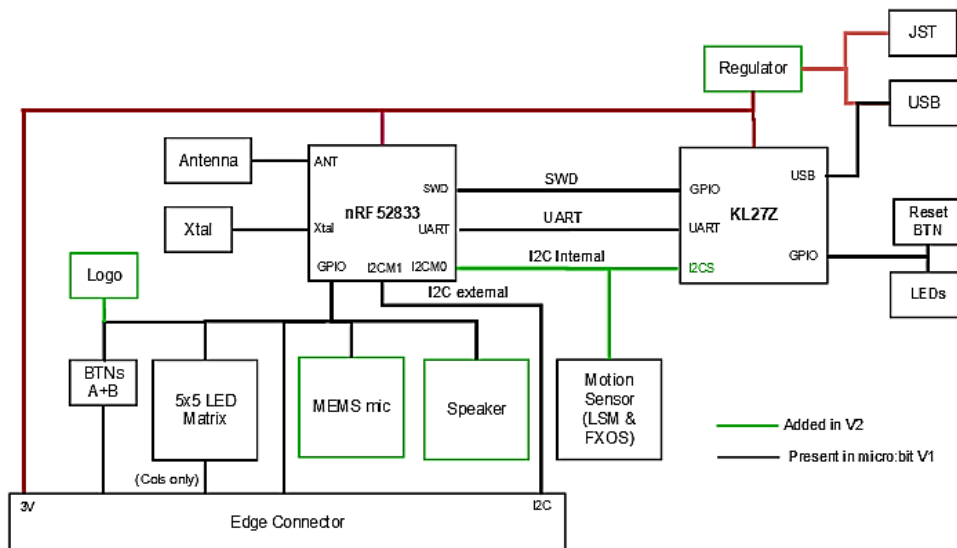
Εικόνα 26: Το BBC micro:bit [72]

Πέραν του αισθητήρα φωτός περιλαμβάνει επίσης ενσωματωμένους αισθητήρες θερμοκρασίας, επιταχυνσιόμετρο και μαγνητόμετρο (πιξίδα). Ο ενσωματωμένος αισθητήρας θερμοκρασίας μπορεί να χρησιμοποιηθεί για την μέτρηση θερμοκρασίας περιβάλλοντος από -25C έως 75C. Με την χρήση του ενσωματωμένου

Για την διασύνδεση του με πρόσθετους αισθητήρες ή ενεργοποιητές μπορεί να χρησιμοποιηθεί ακροδέκτης εικοσιπέντε επαφών ο οποίος περιέχει επαφές και για την διασύνδεση με παροχή ρεύματος 3V, για την γείωση καθώς και 3 επαφές για αναλογικό/ψηφιακό I/O. Και τα δύο buttons ελέγχου A και B συνδέονται σε επαφές GPIO που είναι επίσης προσβάσιμες μέσω του ακροδέκτη επαφών.

Επιπλέον του μικροελεγκτή εφαρμογών, περιέχει επεξεργαστή διασύνδεσης για την διαχείριση της επικοινωνίας μέσω USB και την υποστήριξη της διαδικασίας αναδίπλωσης κώδικα, μεταφοράς και απόθεσης, τον Freescale KL27 για το micro:bit V2 ή τον KL26 για το micro:bit V1, συνδεδεμένο με τον επεξεργαστή εφαρμογών, nRF52833 για το micro:bit V2 ή τον nRF51822 micro:bit V1. Οι δύο μικροελεγκτές συνδέονται μέσω δυο σειριακών διεπαφών, SWD και UART όπως φαίνεται και στην εικόνα 27, οι οποίες στην περίπτωση του micro:bit V2 είναι τρεις. Η σειριακή διεπαφή SWD – Serial Wire Debug χρησιμοποιείται για τον προγραμματισμό του μικροελεγκτή εφαρμογών ενώ η UART χρησιμοποιείται για την αποστολή μηνυμάτων από τον επεξεργαστή εφαρμογών μέσω του επεξεργαστή διασύνδεσης και της θύρας USB στον υπολογιστή με τον οποίο είναι συνδεδεμένο το micro:bit για την εμφάνιση μηνυμάτων στον χρήστη - προγραμματιστή.



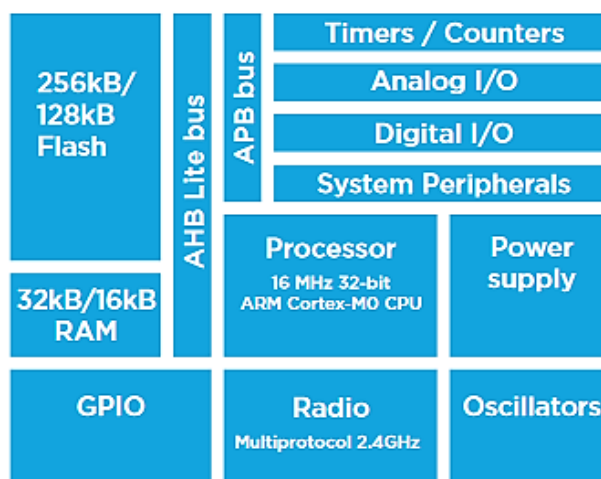


Εικόνα 27: Μπλοκ διάγραμμα υλικού με διαφορές micro:bit V1 και V2 [72]

Η τρίτη διασύνδεση που έχει προστεθεί στο micro:bit V2 έχει να κάνει με τον διαφορετικό τρόπο που αυτό χειρίζεται τα motion sensors καθώς και την τροφοδοσία του από μπαταρίες σε σχέση με το V1.

Το micro:bit V1.5, που χρησιμοποιήθηκε για τις ανάγκες της παρούσης εργασίας, είναι εφοδιασμένο με το nRF51822 SoC της Nordic Semiconductors, ένα γενικού σκοπού και εξαιρετικά χαμηλής κατανάλωσης SoC με μια ποικιλία περιφερειακών (on-chip) καθώς και ορισμένα περιφερειακά εκτός της συσκευασίας του ολοκληρωμένου, ενώ η έκδοση micro:bit V2.0 που είναι εμπορικά διαθέσιμη αυτή τη στιγμή εφοδιάζεται με το nRF52833 SoC με ελαφρώς καλύτερα χαρακτηριστικά.

Έχει κατασκευαστεί γύρω από την 32bit ARM Cortex-M0 κεντρική μονάδα επεξεργασίας και περιλαμβάνει 256kB/128kB μνήμης flash καθώς και 32kB/16kB μνήμης RAM.



Εικόνα 28: Διάγραμμα του nRF51822 SoC του micro:bit [74]

Ο ενσωματωμένος πομποδέκτης 2.4GHz υποστηρίζει επικοινωνίες Bluetooth Low Energy το οποίο επιτρέπει την επικοινωνία με ένα ευρύ φάσμα συσκευών που το υποστηρίζουν. Υποστηρίζει επίσης πολλά άλλα πρότυπα ραδιοεπικοινωνιών συμπεριλαμβανομένου του Nordic Gazel. Αυτό το πρωτόκολλο παρέχει μια πολύ απλή διεπαφή μετάδοσης ραδιοσυχνοτήτων μικρών πακέτων μεταξύ άλλων συσκευών που υποστηρίζουν αυτό το ιδιόκτητο πρωτόκολλο, όπως άλλες συσκευές micro:bit, μέσω της χρήσης της διεπαφής “radio” που μπορεί να χρησιμοποιηθεί σε πολλές από τις υποστηριζόμενες γλώσσες προγραμματισμού.

### 2.3.2 Υποστήριξη λογισμικού

Το micro:bit έχει δημιουργηθεί ως εκπαιδευτική πλατφόρμα για την ενεργή εμπλοκή αρχάριων και ειδικότερα νέων με την διαδικασία ανάπτυξης λογισμικού και ως τέτοια υποστηρίζει ένα σύνολο γλωσσών προγραμματισμού υψηλού επιπέδου που είτε προ-υπήρχαν όπως η C/C++, η Python (MicroPython) και η JavaScript, είτε αναπτύχθηκαν ειδικά για την συγκεκριμένη πλατφόρμα όπως το ανοιχτού κώδικα περιβάλλον οπτικού προγραμματισμού (VPL – Visual Programming Language) MakeCode από την Microsoft.

Για τον προγραμματισμό της συσκευής μπορεί να χρησιμοποιηθεί κάποιο περιβάλλον ανάπτυξης από την πληθώρα επιλογών που υπάρχουν διαθέσιμες, συμβατό με την εκάστοτε γλώσσα προγραμματισμού που θα επιλεγεί . Για όλες τις υποστηριζόμενες γλώσσες υπάρχουν περιβάλλοντα τα οποία είναι διαθέσιμα υπό την μορφή επισκέψιμης online πλατφόρμας ανάπτυξης καθώς και εργαλεία που μπορούν να χρησιμοποιηθούν ως λογισμικό που εγκαθίσταται στον υπολογιστή διασύνδεσης και μπορούν να χρησιμοποιηθούν offline.

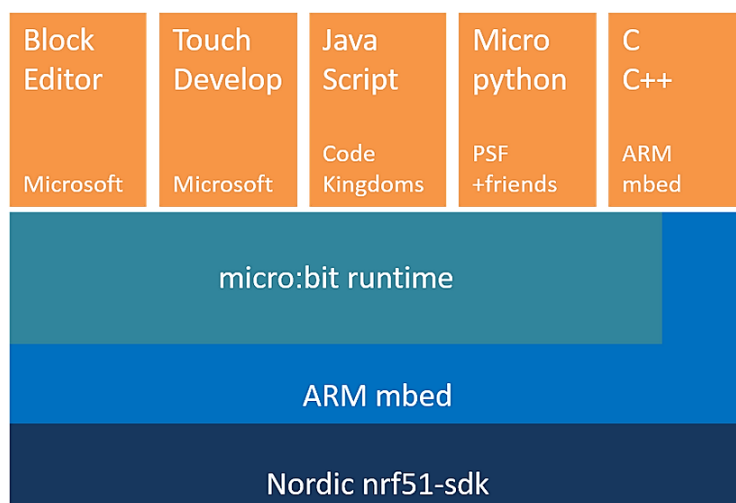
Έτσι κάποιος που θα επιλέξει να αναπτύξει εφαρμογές σε C/C++ μπορεί να επιλέξει να χρησιμοποιήσει είτε τον [Mbed Online Compiler](#) είτε την έκδοση του [Mbed Studio](#) που υπάρχει για τοπική εγκατάσταση, είτε ακόμα και το [Arduino IDE](#) (C++). Υποστηρίζεται επίσης η χρήση της γλώσσας προγραμματισμού [Python](#) μέσω του διερμηνευτή της [MicroPython](#), που είναι μια λιτή αλλά αποτελεσματική υλοποίηση της γλώσσας προγραμματισμού Python 3. Για την ανάπτυξη κώδικα έχει αναπτυχθεί online η πλατφόρμα <https://python.micro:bit.org> για χρήση μέσω φυλλομετρητή ή μπορεί να χρησιμοποιηθεί κάποιο υποστηριζόμενο Ολοκληρωμένο Περιβάλλον Ανάπτυξης(IDE) το οποίο προϋποθέτει πριν την εγκατάσταση του στο σύστημα του χρήστη. Επίσης αν επιλεγεί κάποια οπτική γλώσσα προγραμματισμού για την ανάπτυξη εφαρμογών με την χρήση πλακιδίων, υποστηρίζεται μια πληθώρα εργαλείων όπως [MakeCode](#) ή και [Scratch](#) ή μπορεί να χρησιμοποιήσει την δυνατότητα που δίνει η πλατφόρμα [MakeCode](#) για να προγραμματίσει με την χρήση [Javascript](#). Και φυσικά υπάρχουν πολλές ακόμα γλώσσες προγραμματισμού που υποστηρίζονται και ανάλογες πλατφόρμες ανάπτυξης. [72]

Για την διασύνδεση του micro:bit με κάποιο υπολογιστικό σύστημα και τον προγραμματισμό του χρησιμοποιείται μια απλή usb. Ο υπολογιστής αναγνωρίζει τον εσωτερικό χώρο αποθήκευσης του micro:bit ως εξωτερική αποσπώμενη συσκευή αποθήκευσης την οποία αναγνωρίζει με το όνομα MICRO:BIT και μπορεί να χρησιμοποιήσει όπως οποιαδήποτε εξωτερική συσκευή αποθήκευσης.

Σε όλα τα προαναφερόμενα περιβάλλοντα ανάπτυξης κώδικα παράγεται ένα αρχείο σε μορφή .hex. Το αρχείο αυτό μέσω της διασύνδεσης USB αποτίθεται στον εμφανιζόμενο στον Η/Υ χώρο αποθήκευσης με όνομα MICRO:BIT. Αυτό διευκολύνει τις διαδικασίες προγραμματισμού του και διασφαλίζει ότι ακόμα και αν ο εκτελέσιμος κώδικας περιέχει σφάλματα που θα οδηγήσουν ακόμα και στην δυσλειτουργία της συσκευής θα μπορεί πάντα να επαναπρογραμματιστεί.

Αυτό γίνεται εφικτό όπως αναφέρθηκε και στην παράγραφο 2.2.1 με την χρήση του επεξεργαστή διασύνδεσης, KL26 ή KL27 όπως φαίνεται και στην εικ. 27, που αναλαμβάνει την διαχείριση της διασύνδεσης USB, και τις διαδικασίες προγραμματισμού και εκσφαλμάτωσης, ενώ ο κώδικας εκτελείται από τον μικροελεγκτή εφαρμογών (nRF52833 V2 ή nRF51822 V1).

Για να είναι δυνατή η χρήση γλωσσών υψηλού επιπέδου όπως η Javascript υπάρχουν πολλά επίπεδα λογισμικού που εκτελούνται στο micro:bit. Μερικά από αυτά, όπως το Arm Mbed και η MicroPython προ-υπήρχαν του έργου micro:bit του BBC και άλλα, όπως ο micro:Bit runtime γράφτηκαν ειδικά για αυτό.



Εικόνα 29: Διάγραμμα δόμησης κώδικα [73]

Το Arm Mbed παρέχει ένα Hardware Abstraction Layer (HAL) για ολοκληρωμένα που χρησιμοποιούν επεξεργαστές Arm Cortex. Επειδή το υλικό του micro:bit βασίζεται στο Mbed HDK και το λογισμικό στο Mbed SDK αυτό σημαίνει ότι οι προγραμματιστές που χρησιμοποιούν το micro:bit έχουν ήδη πρόσβαση σε μια τεράστια βιβλιοθήκη στοιχείων που μπορούν να χρησιμοποιήσουν. Παρέχεται εύκολη πρόσβαση σε περιφερειακές διασυνδέσεις όπως SPI, I2C και σειριακές για χρήση από περιβάλλοντα υψηλότερου επιπέδου.

Το micro:bit runtime δημιουργήθηκε από το Lancaster University ως μέρος της προσπάθειας της υιοθέτησης της Πληροφορικής από τα σχολεία. Παρέχει ένα Device Abstraction Layer (DAL), το οποίο είναι κατασκευασμένο με το Arm Mbed. Ουσιαστικά παρέχει ένα εύκολο στη χρήση περιβάλλον για τον προγραμματισμό της πλατφόρμας σε C/C++. Περιέχει οδηγούς συσκευών για όλες τις δυνατότητες υλικού του micro:bit και μια σουίτα από λειτουργίες για να είναι ο προγραμματισμός ευκολότερος. Αυτά επιτρέπουν τον έλεγχο της μήτρας LED για το display, τον έλεγχο των ραδιοεπικοινωνιών και τις υπηρεσίες Bluetooth Low Energy.

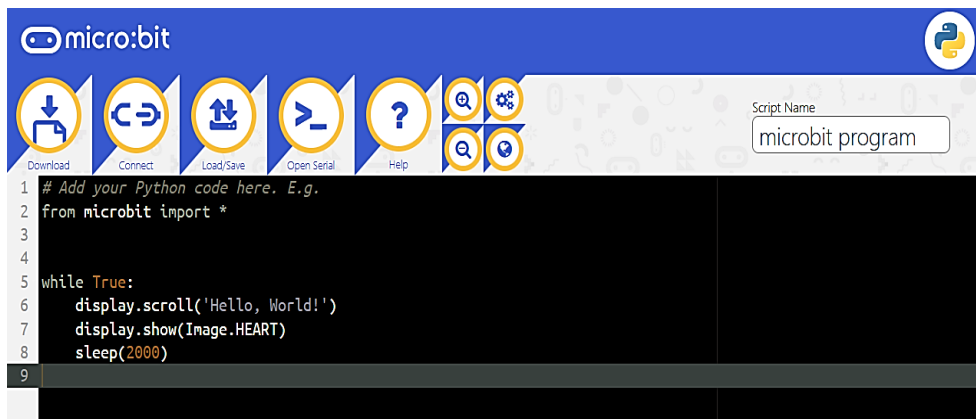
### 2.3.3 Υποστηριζόμενο λογισμικό - Διαδικασίες προγραμματισμού

Όπως ήδη αναφέρθηκε το micro:bit ως μια πλατφόρμα προσανατολισμένη στην εκπαίδευση στον προγραμματισμό προσφέρει πολλές επιλογές σχετικά με την δυνατότητα προγραμματισμού της. Αναλόγως την γλώσσα προγραμματισμού που θα επιλέξει ο χειριστής της υπάρχουν και οι ανάλογες πλατφόρμες για τον προγραμματισμό του.

Στην παρούσα εργασία επιλέχθηκε ως εργαλείο ανάπτυξης η χρήση της γλώσσας προγραμματισμού [Python](#).

Το micro:bit υποστηρίζει τον διερμηνευτή της [MicroPython](#), που είναι μια λιτή αλλά αποτελεσματική υλοποίηση της γλώσσας προγραμματισμού Python 3. Περιλαμβάνει ένα μικρό υποσύνολο της τυπικής βιβλιοθήκης Python και είναι βελτιστοποιημένη για εκτέλεση σε μικροελεγκτές και σε περιορισμένα περιβάλλοντα.

Για την ανάπτυξη κώδικα με την συγκεκριμένη υλοποίηση έχει αναπτυχθεί η πλατφόρμα <https://python.micro:bit.org> για χρήση μέσω φυλλομετρητή. Μπορεί εναλλακτικά να χρησιμοποιηθεί κάποιο από τα υπόλοιπα υποστηριζόμενα λογισμικά για ανάπτυξη το οποίο προϋποθέτουν την προηγούμενη εγκατάστασή τους στο σύστημα χρήσης. Τέτοια είναι το [Mu offline Python editor](#), το γνωστό IDE για την Python [PyCharm \(έχει ειδικό πρόσθετο για την επικοινωνία με micro:bit\)](#) καθώς και ένα μικρού μεγέθους αλλά πολύ αποτελεσματικό IDE για Python με όνομα [Thonny](#).

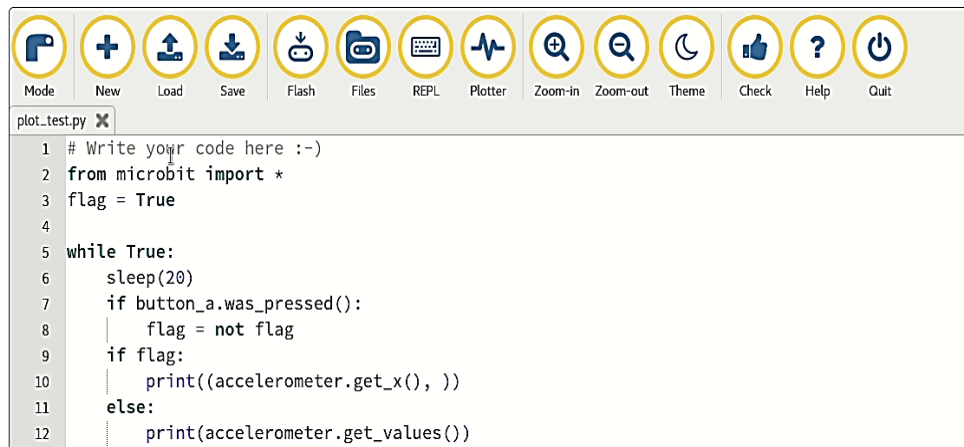


Εικόνα 30: Online περιβάλλον ανάπτυξης <https://python.micro:bit.org>

Το [Mu offline Python editor](#) ξεκίνησε ως ένα project από το Python Software Foundation ως συμβολή στην προσπάθεια ανάπτυξης της πλατφόρμας του micro:bit μαζί με τον online editor. Επιτρέπει την χρήση δυνατοτήτων που δεν είναι διαθέσιμες μέσω της πλατφόρμας για φυλλομετρητές είναι στην διάθεση του χρήστη και με μια εύχρηστη λειτουργικότητα.

Με την χρήση των κατάλληλων λειτουργιών ο χρήστης μπορεί να αντιγράψει τον κώδικα του στην διασυνδεδεμένη micro:bit συσκευή μέσω του button Flash, να έχει πρόσβαση στο σύστημα αρχείων του micro:bit μέσω του button Files, είτε να χρησιμοποιήσει την λειτουργικότητα του button REPL(read, Evaluate, Print, Loop) για να έχει την δυνατότητα να βλέπει άμεσα την λειτουργικότητα του κώδικα του σε μια σύγχρονη επικοινωνία με την διασυνδεδεμένη συσκευή micro:bit. Τέλος με την χρήση του button Plotter δίνεται η

δυνατότητα να σχεδιάζει άμεσα γραφήματα από τα αριθμητικά δεδομένα που παρέχει η διασυνδεδεμένη micro:bit συσκευή.



Εικόνα 26 Το MU offline Python Editor

## Κεφάλαιο 3: Υλοποίηση πειραματικών σεναρίων

### 3.1. Εισαγωγή

Αφού παρουσιάστηκε η εκδοχή του micro:bit ως πλατφόρμα για υλοποίηση ασύρματων δικτύων (WSN) στη συνέχεια θα παρουσιαστούν διάφορες υλοποιήσεις, πειραματισμοί αλλά και σεσάρια ώστε έτσι να αποτυπωθούν πρακτικά πλέον, οι δυνατότητες, τα τυχόν προβλήματα αλλά και τα πιθανά σεσάρια που μπορούν να αναπτυχθούν με τη χρήση της συγκεκριμένης πλατφόρμας.

Για τον προγραμματισμό των micro:bits αλλά και την επικοινωνία τους με Η/Υ, χρησιμοποιήθηκε MicroPython σε συνδυασμό με τον Mu editor.

Επειδή η συγκεκριμένη πλατφόρμα είναι σχετικά καινούργια στη αγορά και άγνωστη η «συμπεριφορά» της αλλά και η αξιοπιστία της του σε δίκτυο WSN, αρχικά έγινε προσπάθεια να υλοποιηθούν οι τρεις βασικοί τρόποι επικοινωνίας που συναντάμε στα ασύρματα δίκτυα:

- i. Broadcast ( Εκπομπή μηνύματος σε όλους )
- ii. Multicast ( Εκπομπή μηνύματος σε πολλούς – συγκεκριμένους )
- iii. Unicast ( Εκπομπή μηνύματος σε έναν )

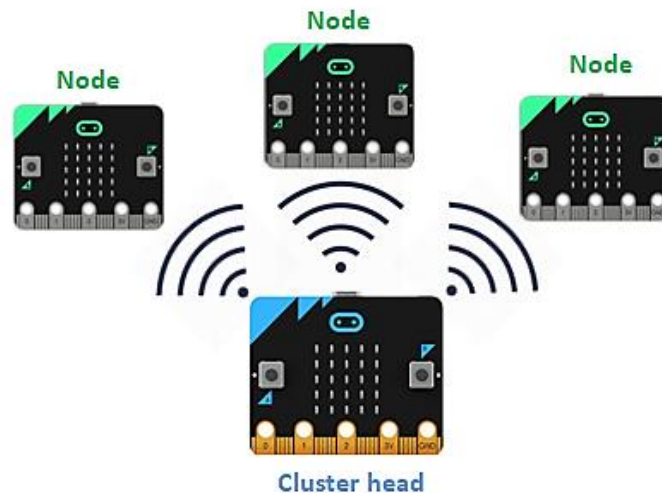
Πρέπει να σημειωθεί ότι το περιβάλλον που υλοποιήθηκαν τα σεσάρια ήταν τυχαίο και μη εργαστηριακό, με τα micro:bits να βρίσκονται μεταξύ τους σε μικρή απόσταση μερικών εκατοστών.

### 3.2. Υλοποίηση βασικών τρόπων επικοινωνίας

#### 3.2.1. Υλοποίηση επικοινωνίας Broadcast

Στην broadcast επικοινωνία έχουμε επικοινωνία ενός κόμβου-micro:bit με όλους τους υπόλοιπους κόμβους-micro:bits του ασύρματου δικτύου.

Για να γίνει πιο κατανοητό το εγχείρημα θεωρούμε ότι έχουμε το παράδειγμα ενός WSN, το οποίο αποτελείται από έναν (1) cluster head (CH) και τρεις (3) κόμβους (nodes), όπως φαίνεται στην ακόλουθη εικόνα.



Εικόνα 27: Broadcast

Το σενάριο είναι, κάθε φορά που θα πατιέται στον cluster head το button A, θα στέλνεται προς όλους (broadcast) ένα μήνυμα «Hello» το οποίο θα τυπώνεται (scrolling) στο led-display όλων των nodes.

Για να υλοποιηθεί αυτό θα πρέπει στον cluster head να εκτελείται κατάλληλος κώδικας εκπομπής (broadcast) του μηνύματος «Hello» και σε όλα τα nodes αντίστοιχος κώδικας λήψης (receive) του μηνύματος.

Στην microPython του micro:bit, το βασικό module για την υλοποίηση ασύρματης επικοινωνίας είναι το **radio** (<https://micro:bit-micropython.readthedocs.io/en/latest/radio.html>).

Ο κώδικας που τρέχει στο cluster head micro:bit, είναι ο ακόλουθος:

```
# Import everything from the micro:bit code library.
from micro:bit import *
# The radio module allows devices to work together via simple
wireless networks.
import radio

# Turns the radio on.
radio.on()

# Runs endlessly!
while True:
    # Show X (NO) on display.
    display.show(Image.NO)
    # Press button A to send a message.
    if button_a.is_pressed():
        # Send "Hello" message via wireless.
        radio.send('Hello')
        # Show tick (YES) on display.
        display.show(Image.YES)
        # Wait 1 sec.
        sleep(1000)
```

Η λειτουργία του κώδικα είναι η εξής: Εμφανίζεται στο display του cluster head (CH) το σύμβολο ☒ (όχι αποστολή μηνύματος), έως ότου πατηθεί το πλήκτρο A. Μόλις πατηθεί το πλήκτρο A στέλνεται ασύρματα (broadcast) το μήνυμα «HELLO» και για ένα δευτερόλεπτο εμφανίζεται στο display του CH το σύμβολο ☑ (αποστολή μηνύματος). Στη συνέχεια ξαναεμφανίζεται το σύμβολο ☒, έως ότου ξαναπατηθεί το πλήκτρο A στον CH. Αυτή η λειτουργία στον CH εκτελείται συνέχεια (ατέρμονα).

Ο κώδικας που τρέχει από τη μεριά όλων των κόμβων (nodes) είναι ο ακόλουθος:

```
# Import everything from the micro:bit code library.
from micro:bit import *

# The radio module allows devices to work together via simple
wireless networks.
import radio

# Turns the radio on.
radio.on()

# Runs endlessly!
while True:
    # Show sad face on display.
    display.show(Image.SAD)
    # Listen for a message.
    message = radio.receive()
    # If message received, scroll it on display.
    if message is not None:
        display.scroll(message)
```

Η λειτουργία του κώδικα είναι η εξής: Στο display όλων των nodes εμφανίζεται το σύμβολο ☹ (SAD), ένδειξη ότι δεν έχουν λάβει ακόμα κάποιο μήνυμα. Μόλις λάβουν κάποιο μήνυμα το εμφανίζουν στο display τους (scrolling) – στο συγκεκριμένο σενάριο το «HELLO» – και στη συνέχεια εμφανίζεται πάλι το σύμβολο ☹ έως ότου λάβουν το επόμενο μήνυμα «HELLO».

Η λειτουργικότητα του σεναρίου ήταν επιτυχής.

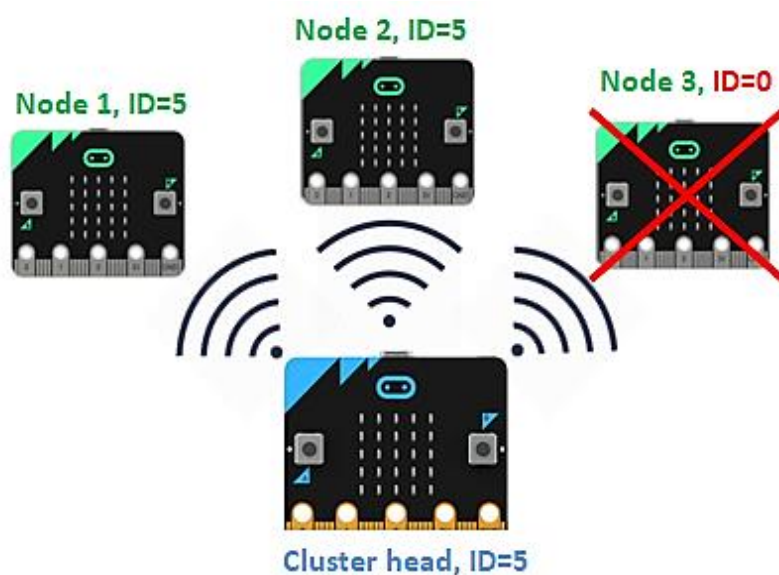


### 3.2.2. Υλοποίηση επικοινωνίας Multicast

Μία βασική δυνατότητα που μας δίνει το module radio είναι να δημιουργούμε groups. Για να μπορούν μια ομάδα micro:bits να επικοινωνούν μεταξύ τους, πρέπει να ανήκουν στο ίδιο group. Το group ουσιαστικά καθορίζεται από έναν αριθμό (ID) από 0 έως 255. Ο εξ' ορισμού (default) αριθμός του group που ανήκει ένα micro:bit είναι το μηδέν (0). Συνεπώς στην προηγούμενη επικοινωνία (broadcast) που παρουσιάστηκε, όλα τα micro:bit (CH και nodes) ανήκαν στο ίδιο (default) group μηδέν (0). Σημείωση: Όταν χρησιμοποιείται το default group 0 δε χρειάζεται κάποια σχετική οδηγία/εντολή στον κώδικα.

Για την υλοποίηση του σεναρίου multicast, θα χρησιμοποιηθεί η ίδια τοπολογία WSN με το παράδειγμα του broadcast. Θα αποτελείται από έναν (1) cluster head (CH) και τρεις (3) κόμβους (nodes). Σε αυτή την περίπτωση όμως ο σκοπός είναι, πολλοί κόμβοι να λαμβάνουν το μήνυμα «Hello», όχι όμως όλοι. Πιο συγκεκριμένα, να το λαμβάνουν οι δύο (2) από τους τρεις (3) συνολικά κόμβους, όπως φαίνεται στην ακόλουθη εικόνα.

Δηλαδή, κάθε φορά που θα πατιέται στον cluster head το button A, θα στέλνεται ένα μήνυμα «Hello», το οποίο όμως θα λαμβάνεται μόνο από τα nodes 1 και 2. Στη συνέχεια το μήνυμα θα τυπώνεται (scrolling) στα led-display τους.



Εικόνα 28: Multicast

Για να υλοποιηθεί αυτό θα πρέπει ο cluster head και τα nodes 1 και 2 να βρίσκονται στο ίδιο group, δηλαδή να έχουν το ίδιο group ID.

Πιο αναλυτικά, όπως έχει ήδη αναφερθεί, στον CH εκτελείται κατάλληλος κώδικας εκπομπής (send) του μηνύματος «Hello» και σε όλα τα nodes αντίστοιχος κώδικας λήψης (receive) του μηνύματος.

Ο κώδικας που τρέχει στο cluster head micro:bit, είναι ο ακόλουθος:

```
# Import everything from the micro:bit code library.
from micro:bit import *
# The radio module allows devices to work together via simple
wireless networks.
import radio

# Turns the radio on.
radio.on()
# Set your group ID
radio.config(group = 5)

# Runs endlessly!
while True:
    # Show X (NO) on display.
    display.show(Image.NO)
    # Press button A to send a message.
    if button_a.is_pressed():
        # Send "Hello" message via wireless.
        radio.send('Hello')
        # Show tick (YES) on display.
        display.show(Image.YES)
        # Wait 1 sec.
        sleep(1000)
```

Η λειτουργία του κώδικα είναι ίδια με αυτή του CH στο broadcast, με τη μόνη αλλά και βασική διαφορά ότι ορίζεται το group ID του CH σε 5 (function radio.config).

Ο κώδικας που τρέχει από τη μεριά των κόμβων (nodes) 1 και 2 είναι ο ακόλουθος:

```
# Import everything from the micro:bit code library.
from micro:bit import *
# The radio module allows devices to work together via simple
wireless networks.
import radio
# Turns the radio on.
radio.on()
# Set your group ID
radio.config(group = 5)
# Runs endlessly!
while True:
    # Show sad face on display.
    display.show(Image.SAD)
    # Listen for a message.
    message = radio.receive()
    # If message received, scroll it on display.
    if message is not None:
        display.scroll(message)
```

Η λειτουργία του κώδικα είναι ίδια με αυτή των nodes στο broadcast, με τη μόνη αλλά και βασική διαφορά ότι ορίζεται το group ID των δύο nodes (1 και 2), σε 5 (function `radio.config`). Αυτό επιτυγχάνεται γιατί στα nodes 1 και 2 τρέχει ακριβώς ο ίδιος κώδικας.

Ο κώδικας που τρέχει από τη μεριά του κόμβου (node) 3 είναι ο ακόλουθος:

```
# Import everything from the micro:bit code library.
from micro:bit import *
# The radio module allows devices to work together via simple
wireless networks.
import radio

# Turns the radio on.
radio.on()
# Runs endlessly!
while True:
    # Show sad face on display.
    display.show(Image.SAD)
    # Listen for a message.
    message = radio.receive()
    # If message received, scroll it on display.
    if message is not None:
        display.scroll(message)
```

Ο κώδικας στο node 3 είναι ακριβώς ίδιος με τον κώδικα των nodes στο σενάριο broadcast. Όπως όμως έχει ήδη αναφερθεί, όταν δεν έχει καθοριστεί το group ID σε ένα `micro:bit`, η εξ' ορισμού τιμή είναι μηδέν (0). Άρα το node 3, βρίσκεται σε διαφορετικό group ID από όσους συμμετέχουν στην τοπολογία (CH και nodes 1,2).

Συγκεντρωτικά στο multicast η λειτουργικότητα του σεναρίου είναι η εξής:

Ο cluster head θα κάνει αποστολή του μηνύματος «HELLO», με τη βασική διαφορά ότι, το μήνυμα θα το λάβουν μόνο τα nodes που βρίσκονται στο ίδιο group ID με το CH. Στο συγκεκριμένο παράδειγμα, θα το λάβουν μόνο τα nodes 1 και 2 που βρίσκονται στο ίδιο group ID 5 με τον CH.

Συνεπώς, το node 3 δε θα λάβει ποτέ μήνυμα, γιατί βρίσκεται σε διαφορετικό group ID από τον CH και αυτό είναι το 0 (default group ID).

Η λειτουργικότητα του σεναρίου ήταν επιτυχής.

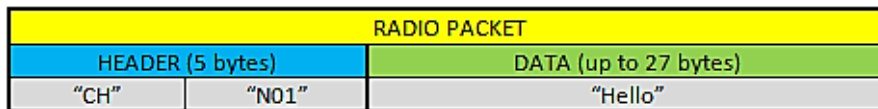
### 3.2.3. Υλοποίηση επικοινωνίας Unicast

Σε αυτό το σενάριο θα επιχειρηθεί κάτι διαφορετικό προγραμματιστικά. Για την επικοινωνία του cluster head με συγκεκριμένο nodes, θα δημιουργείτε ένα πακέτο δεδομένων (radio packet), το οποίο θα στέλνετε προς όλους τους κόμβους αλλά θα αφορά συγκεκριμένο αποδέκτη (κόμβο).

Το radio packet θα αποτελείται από δύο τμήματα: την κεφαλίδα (header) και τα δεδομένα (data). Το μήκος της κεφαλίδας θα είναι τέσσερα (4) bytes εκ' των οποίων, τα δύο

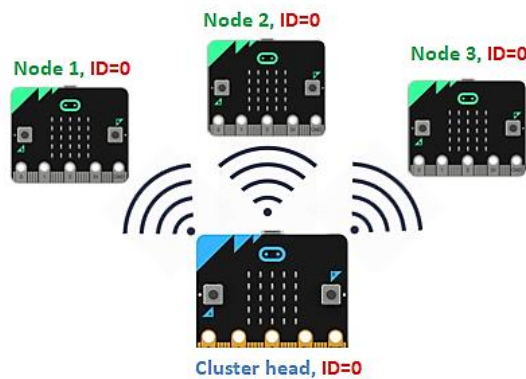
(2) πρώτα θα είναι το όνομα/διεύθυνση του αποστολέα και τα τρία (3) επόμενα το όνομα/διεύθυνση του παραλήπτη. Επειδή, το εξ' ορισμού μήκος του πακέτου είναι 32 bytes (radio module – default length), τα υπόλοιπα 27 bytes που απομένουν – από το πακέτο – θα περιλαμβάνουν τα δεδομένα του μηνύματος.

Στο ακόλουθο σχήμα φαίνεται το περιεχόμενο ενός radio packet το οποίο θα σταλεί από τον cluster head (CH), θα αφορά το node 1 (N01) και τα δεδομένα/μήνυμα θα είναι το «Hello»:



Εικόνα 29: Radio Packet

Για την υλοποίηση του σεναρίου unicast, θα χρησιμοποιηθεί η ίδια τοπολογία WSN με το παράδειγμα του broadcast. Θα αποτελείται από έναν (1) cluster head (CH) και τρεις (3) κόμβους (nodes). Σημείωση: Στην ακόλουθη εικόνα τονίζεται, ότι όλα τα micro:bits θα βρίσκονται στο ίδιο default group (ID=0). Όπως έχει ήδη αναφερθεί σε αυτή την περίπτωση, στον κώδικα, δε χρειάζεται κάποια συγκεκριμένη οδηγία.



Εικόνα 30: Unicast

Το σενάριο αποτελείται από τρεις εκπομπές unicast, που θα επιλέγονται από τα buttons του cluster head:

1. Button A: Όταν θα πατιέται στον cluster head το button A, θα στέλνεται προς όλους (broadcast) ένα μήνυμα «Hello», αλλά επειδή θα αφορά μόνο τον κόμβο 1, θα εμφανίζεται (scrolling) μόνο στο led-display του κόμβου 1.
2. Button B: Στην περίπτωση που θα πατηθεί στον cluster head το button B, θα στέλνεται προς όλους (broadcast) ένα μήνυμα «Hello», αλλά επειδή θα αφορά μόνο τον κόμβο 2, θα εμφανίζεται (scrolling) μόνο στο led-display του κόμβου 2.
3. Button A+B: Τέλος, στην περίπτωση που θα πατηθούν τον cluster head ταυτόχρονα τα button A και B, θα στέλνεται προς όλους (broadcast) ένα μήνυμα «Hello», αλλά επειδή θα αφορά μόνο τον κόμβο 3, θα εμφανίζεται (scrolling) μόνο στο led-display του κόμβου 3.

Σημείωση: Όσο αφορά τη λειτουργία του led-display στον CH, είναι ίδιας λογικής με τα προηγούμενα σενάρια. Συνοπτικά:  όχι αποστολή μηνύματος και  αποστολή μηνύματος (buttons A, B, A+B).

Ο κώδικας που τρέχει στον cluster head είναι ο ακόλουθος:

```
from micro:bit import *
import radio
# Turn radio on
radio.on()
# Sender and receivers constants
sender_id = 'CH'
receiver_01 = 'N01'
receiver_02 = 'N02'
receiver_03 = 'N03'
# Message constant
data = 'Hello'
# Runs endlessly!
while True:
    # Show X (NO) on display.
    display.show(Image.NO)
    # Press buttons A to set the receiver.
    if button_a.is_pressed():
        receiver_id = receiver_01
        # Wait 1/10 sec.
        sleep(100)
        # Press buttons A+B to set the receiver.
        if button_b.is_pressed():
            # Set receiver
            receiver_id = receiver_03
    # Press buttons B to set the receiver.
    elif button_b.is_pressed():
        receiver_id = receiver_02
        # Wait 1/10 sec.
        sleep(100)
        # Press buttons B+A to set the receiver.
        if button_a.is_pressed():
            # Set receiver
            receiver_id = receiver_03
    else:
        # Next while.
        continue
    # Create the header
    header = sender_id + receiver_id
    # Create the radio packet
    radio_packet = header + data
    # Send radio packet via wireless.
    radio.send(radio_packet)
    # Show tick (YES) on display.
    display.show(Image.YES)
    # Wait 1 sec.
    sleep(1000)
```

Η λειτουργία του κώδικα, όπως αναφέρθηκε, εξαρτάται από το ποιο ή ποια button πατιούνται. Έτσι ανάλογα την επιλογή A, B ή A+B (ταυτόχρονα) δημιουργείται η κατάλληλη

επικεφαλίδα (header), η οποία στη συνέχεια ενώνεται με το μήνυμα «Hello» (data) και αποστέλλεται ασύρματα (radio\_packet) προς όλα τα nodes της τοπολογίας μας.

Το περιεχόμενο του header αποτελείται από δυο (2) χαρακτήρες «CH» (sender\_id, 2 bytes) και από τρεις (3) χαρακτήρες «Nxx» (receiver\_id, 3 bytes), όπου xx ο αριθμός του node (N01, N02, N03). Στο receiver\_id αποδίδεται τιμή ανάλογα του/των buttons που πατήθηκαν (receiver\_01, receiver\_02, receiver\_03).

Σημείωση: Επειδή πρακτικά το ταυτόχρονο πάτημα των δυο buttons είναι δύσκολο να επιτευχθεί, υλοποιείται μια προγραμματιστική τεχνική κατά την οποία, μόλις πατηθεί το button A μετά από 1/10 του δευτερολέπτου (sleep 100) ελέγχεται αν πατήθηκε και το button B. Σε αυτή την περίπτωση θεωρείται ταυτόχρονο το πάτημα των δύο πλήκτρων. Με την ίδια τεχνική υλοποιείται και η περίπτωση που πατηθεί πρώτα το button B και μετά το A.

Από τη μεριά των κόμβων, ακολουθεί ο κώδικας του node 1 (receive\_id='N01'). Η μόνη διαφορά με τους άλλους δύο κόμβους είναι η δήλωση του receive\_id, δηλαδή receive\_id='N02' και receive\_id='N03' για τον κόμβο 2 και 3 αντίστοιχα.

```
from micro:bit import *
import radio

# Turn radio on
radio.on()

# Receiver constant
receiver_id = 'N01'

# Runs endlessly!
while True:
    # Show node number on display.
    display.show(receiver_id[2:])
    # Listen for a radio packet.
    radio_packet = radio.receive()
    # If radio packet received then check for length and
    receiver_id
    if radio_packet is not None:
        # Display data if length is 5 or greater and
        # receiver_id is for current node
        if len(radio_packet) >= 5 and radio_packet[2:5] ==
receiver_id:
            data = radio_packet[5:]
            display.scroll(data)
```

Η λειτουργία του κώδικα διαφοροποιείται σε σχέση με τα δύο προηγούμενα σενάρια. Στην οθόνη του κάθε micro:bit/node εμφανίζεται ο αντίστοιχος αριθμός του receive\_id. Για τον κόμβο N01 εμφανίζεται ο αριθμός 1 και για τον N02, N03, ο αριθμός 2 και 3 αντίστοιχα.

Στη συνέχεια μόλις το κάθε node λάβει κάποιο radio packet ελέγχονται δυο συνθήκες: α) αν το μήκος του πακέτου είναι μεγαλύτερο ή ίσο των 5 bytes, πράγμα που

σημαίνει ότι ήρθε η ελάχιστη πληροφορία από το πακέτο, που είναι το HEADER και β) αν το πακέτο αφορά το συγκεκριμένο node, συγκρίνοντας τα bytes 3-5 του header (περιέχει το id του παραλήπτη) με το receive\_id του κόμβου.

Αν τελικά ισχύουν και οι δύο συνθήκες (α και β) εμφανίζεται (scrolling) στο led-display του κόμβου το τμήμα DATA του radio packet που έλαβε. Στο συγκεκριμένο σενάριο το μήνυμα «Hello».

Σημείωση: Επειδή έχουμε μόνο έναν αποστολέα (CH), παρόλο που στο radio packet περιλαμβάνεται το id του (sender\_id='CH'), στη λήψη δεν υλοποιείται κάποιος έλεγχος για το ποιος αποστέλλει το πακέτο.

Η λειτουργικότητα του σεναρίου για όλες τις πιθανές επιλογές (button A, B, A+B) ήταν επιτυχής.

## 3.2. Διαδικασίες Βαθμονόμησης κόμβων (Nodes Calibration)

Μία από τις βασικότερες λειτουργίες στη μελέτη και ανάλυση των ασύρματων δικτύων, είναι η βαθμονόμηση των κόμβων τους (Nodes Calibration). Ο όρος βαθμονόμηση αναφέρεται στις διαδικασίες που πρέπει να ακολουθηθούν έτσι ώστε να δοθεί η δυνατότητα σε ένα αυτόνομο ασύρματο δίκτυο, στο να προσδιορίζει τη θέση των διαφόρων κόμβων του, οι οποίοι μπορεί να βρίσκονται διάσπαρτοι στην εμβέλεια του.

Ουσιαστικά για να θεωρηθεί μια βαθμονόμηση ολοκληρωμένη και στη συνέχεια να μπορέσει να εκτιμηθεί η επιτυχία της, θα πρέπει να παρέχεται η δυνατότητα δύο βασικών λειτουργιών (mode):

- A. Καταγραφή από σταθερό σταθμό βάσης κατάλληλων τιμών/μετρήσεων, βάσει προκαθορισμένων μηνυμάτων-εκπομπών από κόμβους, τοποθετημένους σε προκαθορισμένες θέσεις (Mode A).
- B. Προσδιορισμός της άγνωστης θέσης/απόστασης ενός κόμβου, από το σταθμό βάσης, βάσει των καταγεγραμμένων τιμών/μετρήσεων (Mode B).

Σημείωση: Όπως θα αναφερθεί και στην ανάλυση του κώδικα, η ενεργοποίηση του mode A ή B, γίνεται αντίστοιχα, με το button A ή B του micro:bit.

### 3.2.1. Υλοποίηση σεναρίου Mode A (καταγραφή τιμών/μετρήσεων) – Button A (micro:bit)

Στην περίπτωση των micro:bits, θα επιχειρηθεί η βαθμονόμηση τους βάσει της ισχύς του σήματός τους, έτσι ώστε στη συνέχεια να δίνεται η δυνατότητα προσδιορισμού της απόστασης τους από έναν σταθερό σταθμό βάσης (cluster head). Αυτό που θα επιχειρηθεί να επαληθευτεί είναι ότι, όσο πιο κοντά το node στο cluster head, τόσο πιο ισχυρό το σήμα και όσο πιο μακριά τόσο πιο ασθενές.

Για να υλοποιηθεί αυτό το πειραματικό σενάριο – το οποίο θα περιλαμβάνει και μετρήσεις – θα χρησιμοποιηθεί η function `radio.receive_full()`, η οποία επιστρέφει λεπτομερείς πληροφορίες σε σχέση με την `radio.receive()`, που χρησιμοποιήθηκε ως τώρα.

Η function `radio.receive_full()` ουσιαστικά επιστρέφει μια python tuple (πλειάδα) με τις εξής πληροφορίες:

- i. Το μήνυμα. Η μορφή του είναι σε bytes.
- ii. Την ισχύ του σήματος (Received Signal Strength Indicator). Είναι σε dBm με τιμές από 0 (Ισχυρό) έως -255 (Αδύναμο). Από αυτή την τιμή/μέτρηση θα προσδιοριστεί και η θέση του κόμβου.
- iii. Χρονosφραγίδα (timestamp) σε msec, κατά τη λήψη κάθε μηνύματος.

Το σενάριο της βαθμονόμησης που θα υλοποιηθεί, θα περιλαμβάνει έναν σταθμό βάσης (cluster head) και τρεις (3) κόμβους (nodes). Οι κανόνες της βαθμονόμησης θα είναι οι ακόλουθοι:

- i. Το περιβάλλον των μετρήσεων όσο το δυνατόν χωρίς φυσικά ή τεχνητά εμπόδια. Συγκεκριμένα έγιναν σε κλειστό γήπεδο μπάσκετ και επαναλήφθηκαν σε κλειστή αίθουσα εκδηλώσεων.
- ii. Τοποθέτηση του cluster head σε σταθερό σημείο στο έδαφος.
- iii. Τοποθέτηση του κάθε node σε προκαθορισμένες αποστάσεις: 0.5, 1, 1.5, 2, 4, 8, 16 (αποστάσεις σε μέτρα).
- iv. Εκπομπή από το κάθε node προκαθορισμένο μήνυμα. Ουσιαστικά θα εκπέμπει το ID του (1, 2 ή 3) βάσει του οποίου θα αναγνωρίζεται από τον cluster head.
- v. Λήψη πεπερασμένων μετρήσεων από τον cluster head, για κάθε απόσταση ανά node: 15 μετρήσεις.
- vi. Υπολογισμός του μέσου όρου των 15 μετρήσεων.
- vii. Αποθήκευση των μέσων όρων αλλά και όλων των μετρήσεων, σε αντίστοιχα αρχεία κειμένου (text files).

Άρα ο cluster head θα πραγματοποιήσει, για 3 (κόμβους) x 7 (αποστάσεις) x 15 (μετρήσεις), συνολικά 315 μετρήσεις.

Επειδή από τις μετρήσεις που θα συλλεχθούν, θα γίνεται στη συνέχεια και ο προσδιορισμός της άγνωστης θέσης/απόστασης του κάθε node από το cluster head, αναζητήθηκε τρόπος μόνιμης αποθήκευσης – των μετρήσεων – για επαναλαμβανόμενη χρήση και ενδεχομένως περαιτέρω επεξεργασία τους.

Η λύση βρέθηκε με την υλοποίηση ενός πολύ χρήσιμου ίσως και μοναδικού χαρακτηριστικού του micro:bit, όπως έχει αναφερθεί και στο κεφ. 2, της υποστήριξης filesystem (δυνατότητα διαχείρισης αρχείων). Ουσιαστικά κατά την διαδικασία της βαθμονόμησης δημιουργούνται text αρχεία - στη flash memory του micro:bit - στα οποία αποθηκεύονται «μόνιμα» όλες οι μετρήσεις.

Να τονιστεί ότι κάθε φορά που επιλέγεται το mode A, ξεκινάει μια καινούργια βαθμονόμηση και όλες οι προηγούμενες μετρήσεις διαγράφονται.



### 3.2.2. Υλοποίηση σεναρίου Mode B (προσδιορισμός άγνωστης θέσης κόμβου) – Button B (micro:bit)

Βάσει των αποθηκευμένων μετρήσεων στο micro:bit, ο cluster head έχει πλέον τη δυνατότητα να προσδιορίσει την τυχαία θέση των nodes. Στην πράξη θα τοποθετηθούν σε τυχαίες θέσεις, έτσι ώστε να επαληθευτεί η απόσταση τους από τον cluster head.

Ουσιαστικά θα γίνεται σύγκριση την αποθηκευμένης μέτρησης (Mode A), με την μέτρηση από την τρέχουσα/τυχαία θέση (Mode B). Με όποια τιμή αποθηκευμένης μέτρησης, έχει μικρότερη απόλυτη διαφορά η τρέχουσα/τυχαία μέτρηση, τότε βρίσκεται και πιο κοντά σε αυτήν.

Για να γίνει πιο κατανοητό, δίνεται ένα παράδειγμα για node σε τυχαία θέση με ενδεικτικές μετρήσεις: Μέσος όρος μέτρησης στα 4 μέτρα, 70% της μέγιστης ισχύς του σήματος (Mode A) και μέσος όρος μέτρησης στα 8 μέτρα, 40% της μέγιστης ισχύς του σήματος (Mode A). Όταν θα μετρηθεί (Mode B) ισχύ σήματος 60%, αυτό θα σημαίνει ότι ο κόμβος βρίσκεται πιο κοντά στα 4 μέτρα, γιατί στα 4 μέτρα έχουμε απόλυτη διαφορά 10% ( $70\% - 60\% = 10\%$ ) και στα 8 μέτρα απόλυτη διαφορά 20% ( $40\% - 60\% = -20\%$ ).

Όπως γίνεται αντιληπτό, θα πρέπει να προηγηθεί το mode A τουλάχιστον μία φορά και στη συνέχεια να ενεργοποιηθεί το mode B. Το mode B θα επαναλαμβάνεται με τα ίδια αρχεία/μετρήσεις, μέχρις ότου επιλεγθεί εκ νέου το mode A.

### 3.2.3. Ανάλυση – επεξηγήσεις του κώδικα στον cluster head

Αρχικά πρέπει να επισημανθεί ότι οι μετρήσεις από τον cluster head μπορούν να πραγματοποιηθούν με δύο τρόπους:

1. Με χρήση Η/Υ. Ο cluster head συνδεδεμένος με Η/Υ (θύρα USB), έτσι ώστε να εμφανίζονται σχετικά μηνύματα στο περιβάλλον του Mu editor/REPL.
2. ΧΩΡΙΣ τη χρήση Η/Υ. Στο led-display του micro:bit ενεργοποιούνται/ απενεργοποιούνται συγκεκριμένα led, βάσει των οποίων προσδιορίζεται επακριβώς η κατάσταση στην οποία βρίσκεται ο cluster head.

Ποιος τρόπος θα επιλεγεί εξαρτάται αποκλειστικά από το αν θα είναι συνδεδεμένο το cluster head με Η/Υ (Mu editor/REPL) ή όχι. Για λόγους πληρότητας της επεξήγησης του κώδικα, θα θεωρηθεί ότι ακολουθείται ο 1<sup>ος</sup> τρόπος (Με χρήση Η/Υ). Σε κάθε περίπτωση υπάρχει πάντα καθοδήγηση από το led-display του cluster head.

Συνεπώς κάθε εντολή `print` στον κώδικα, όταν ακολουθείται ο 1<sup>ος</sup> τρόπος, έχει οπτικό αποτέλεσμα στο περιβάλλον του Mu editor/REPL. Το Read-Evaluate-Print-Loop (REPL) – όπως έχει αναφερθεί και στο κεφ. 2 – είναι η δυνατότητα live-code στο περιβάλλον του Mu editor, δηλαδή παρακολούθησης των εντολών που εκτελούνται στο micro:bit, σε πραγματικό χρόνο.

Κατόπιν όλων αυτών των εισαγωγικών επεξηγήσεων ακολουθεί η ανάλυση, των σχετικών εντολών που «τρέχουν» στον cluster head. Αν και ο κώδικας έχει αρκετά σχόλια, για την καλύτερη κατανόηση του, επιλέχθηκε η επεξήγηση να γίνει τμήμα-τμήμα. Επίσης λόγω

του μεγέθους του, τα τμήματα του θα παρουσιάζονται σε μορφή screenshot έτσι ώστε να είναι πιο ευδιάκριτος.

Στο ΠΑΡΑΡΤΗΜΑ Α αποτυπώνεται ολόκληρος ο πηγαίος κώδικας του cluster head έτσι ώστε να υπάρχει και «εικόνα» της συνολικής του λειτουργίας.

### 3.2.4. Αρχικοποιήσεις μεταβλητών / αναμονή επιλογής Mode A ή B

```
from microbit import *
import radio
print ()
print ()
print ('R E S E T is pressed')
print ()
print ('Press the button "A" for calibration or "B" to calculate distance.')
print ()
# Number of nodes.
rows = 3
# Number of distances.
cols = 7
# Create a list of measurement distance names.
distances = [0.5, 1, 1.5, 2, 4, 8, 16]
# Number of measurements per distance.
measurements = 15
# Initial rejected measurements.
rejected = 0
# Create a list of size rows x cols, filled with zeros.
nodes_averages = [[0] * cols for i in range(rows)]
# List with differences between current and calibrated measurements.
dif_mes = [0] * cols

while True:
    # Show ? on display.
    display.show('?')
    # Press buttons A to calibrate.
    if button_a.is_pressed():
        ...
        ...
    # Press buttons B to load calibrates from microbit.
    elif button_b.is_pressed():
        ...
        ...
```

Ο κώδικας ξεκινάει με απλές εντολές `print` οι οποίες εμφανίζουν στο τμήμα REPL του Mu editor σχετικά μηνύματα. Παρατηρούμε ότι πρώτο εμφανίζεται το μήνυμα 'R E S E T is pressed'.

Για να διασφαλιστεί η σωστή εκκίνηση του κώδικα στο cluster head/micro:bit αλλά και η σωστή επικοινωνία με το REPL, θα πρέπει, αφού επιλεγεί η λειτουργία REPL (Mu editor) στη συνέχεια να πατήσουμε το RESET button του micro:bit. Η εμφάνιση του πρώτου μηνύματος, από την «εκτέλεση» του κώδικα στο τμήμα REPL, επιβεβαιώνει την επιτυχή εκκίνηση/επικοινωνία.

Με το επόμενο μήνυμα ('Press the button "A" for calibration or "B" to calculate distance.') γίνεται αντιληπτό ότι, θα πρέπει να πατηθεί ένα από τα δυο button του micro:bit έτσι ώστε να ενεργοποιηθεί το αντίστοιχο mode A ή B του cluster head.

Επειδή ο κώδικας επιχειρήθηκε να γίνει όσο το περισσότερο παραμετρικός, οι επόμενες εντολές είναι οι αρχικοποιήσεις απαραίτητων μεταβλητών και πινάκων. Πιο αναλυτικά:

Στη μεταβλητή rows δηλώνεται το πλήθος των nodes και στη μεταβλητή cols το πλήθος των αποστάσεων που θα γίνουν οι μετρήσεις. Για τις συγκεκριμένες πειραματικές μετρήσεις – όπως έχει αναφερθεί – έχουν δηλωθεί αντίστοιχα 3 και 7 γιατί θα χρησιμοποιηθούν τρία (3) nodes/micro:bit σε επτά (7) αποστάσεις. Ποιες είναι αυτές οι αποστάσεις, δηλώνονται στην επόμενη μεταβλητή (distances).

Στη μεταβλητή distances αποθηκεύεται η λίστα/πίνακας των λεκτικών των αποστάσεων [0.5, 1, 1.5, 2, 4, 8, 16].

Στη measurements δηλώνεται το πλήθος των μετρήσεων για κάθε απόσταση, που είναι δεκαπέντε (15).

Σημαντικό: Στις ασύρματες επικοινωνίες υπάρχουν αστάθμητοι παράγοντες, όπως άλλες ασύρματες συσκευές/επικοινωνίες, οι οποίες μπορούν να δημιουργούν παρεμβολές μέχρι και διακοπή της επικοινωνίας. Για να περιοριστούν αυτά τα προβλήματα, δεν πραγματοποιείται μόνο μια μέτρηση αλλά ένα πλήθος μετρήσεων, από το οποίο θα προσδιοριστεί με μεγαλύτερη ακρίβεια η πραγματική τιμή – της μέτρησης –.

Στη περίπτωση που θέλουμε να απορρίψουμε κάποιες αρχικές μετρήσεις – για κάθε απόσταση –, αυτό δηλώνεται στη rejected. Η αρχική τιμή αυτής της μεταβλητής είναι 0, δηλαδή δεν απορρίπτεται καμιά μέτρηση. Ο ουσιαστικός ρόλος της θα σχολιαστεί στη συνέχεια πιο αναλυτικά.

Από τις τιμές των rows και cols μεταβλητών δημιουργείται μία δισδιάστατη λίστα/πίνακας, η nodes averages, στην οποία οι γραμμές είναι όσες το πλήθος των nodes (3) και οι στήλες όσες είναι οι αποστάσεις (7). Επειδή όμως για κάθε απόσταση γίνονται δεκαπέντε (15) μετρήσεις, στη θέση κάθε node/απόστασης θα αποθηκεύεται ο μέσος όρος των δεκαπέντε (15) μετρήσεων. Άρα αν μελλοντικά χρειαστούν περισσότερες μετρήσεις για κάθε απόσταση, αυτό δε θα αλλάξει το μέγεθος του πίνακα, γιατί όπως αναφέρθηκε αποθηκεύεται ο μέσος όρος του πλήθους των μετρήσεων για κάθε απόσταση. Στις περιπτώσεις που πρέπει να αυξηθούν τα nodes ή και οι αποστάσεις, αυτό θα έχει ως συνέπεια και την αλλαγή του μεγέθους του πίνακα.

Τέλος δηλώνεται μια μονοδιάστατη λίστα/πίνακας, η dif mes, μεγέθους cols, όσες δηλαδή είναι και οι αποστάσεις. Σε αυτόν τον πίνακα αποθηκεύονται για συγκεκριμένο/τρέχων κόμβο, οι απόλυτες διαφορές των αποθηκευμένων μετρήσεων, με την τρέχουσα μέτρηση. Μέσω των τιμών αυτού του πίνακα προσδιορίζεται και η σχετική θέση του κόμβου. Δηλαδή σε ποια από τις προκαθορισμένες αποστάσεις βρίσκεται πιο κοντά, όπως εξηγήθηκε και στο παράδειγμα της ενότητας Mode B.

Σημείωση: Κατά την δήλωση των `nodes_averages` και `dif_mes`, οι αρχικές τιμές όλων των θέσεων, είναι μηδέν [0].

Μετά τις αρχικοποιήσεις των μεταβλητών/πινάκων ακολουθεί μία ατέρμονη `while True`, η οποία «περιμένει» να πατηθεί ένα από τα δύο πλήκτρα (A ή B) του cluster head ώστε να εισέλθει σε κατάσταση Mode A (`if button_a.is_pressed`) ή Mode B (`elif button_b.is_pressed`), αντίστοιχα. Όσο δεν επιλέγεται κάτι εμφανίζει συνεχώς στο led-display το '?', ένδειξη ότι ο cluster head είναι σε αναμονή κάποιας ενέργειας.

### 3.2.5. Επιλογή Mode A – Επαναλήψεις nodes/distances – REPL/led-display

```
if button_a.is_pressed():
    display.clear()
    # 1/2 second delay, because the next press button A is followed.
    sleep(500)
    # Create file on microbit file system.
    file = open('calibrates.txt', 'w')
    for i in range(0, rows):
        # Create a list of measurements for current node.
        nodes_mes = [[0] * cols for il in range(measurements)]
        for j in range(0, cols):
            # Real node number for displaying on screen.
            node = i + 1
            print ('Calibration measurements of the node', node, end='')
            print (' from a distance of', distances[j], 'm')
            print ('Press the button "A" on the microbit to get started ...')
            print ()
            display.set_pixel(i * 5, 0, 9)
            display.set_pixel(j * 5, 2, 9)
            display.set_pixel( 0, 4, 9)
            # Turn radio off
            radio.off()
            while not button_a.is_pressed():
                None
            # Turn radio on
            radio.on()
            display.clear()
            sleep(500)
```

Μόλις πατηθεί στιγμιαία το button A (if button\_a.is\_pressed), σβήνει το led-display και δημιουργείται μια καθυστέρηση ½ sec.

Πρέπει να σημειωθεί ότι, επειδή θα ακολουθήσουν και επόμενοι έλεγχοι για το αν έχει πατηθεί το button A, αυτή η καθυστέρηση μας εξασφαλίζει ότι ο έλεγχος για το αν έχει πατηθεί, θα ισχύσει μόνο για τη συγκεκριμένη συνθήκη. Στην περίπτωση που δεν υπήρχε αυτή η καθυστέρηση, με το αρχικό στιγμιαίο πάτημα του button A – λόγω της ταχύτητας εκτέλεσης του κώδικα – θα ίσχυε και η επόμενη συνθήκη (while not button\_a.is\_pressed) κάτι που δεν είναι επιθυμητό, γιατί αυτή η συνθήκη ελέγχει ουσιαστικά πότε θα πραγματοποιηθεί το επόμενο στιγμιαίο πάτημα του button A.

Οι μετρήσεις αποθηκεύονται στο αρχείο 'calibrates.txt' (file), σε μορφή κειμένου (text file) και όπως έχει αναφερθεί δημιουργείται στη εσωτερική flash memory του cluster head/micro:bit. Ουσιαστικά σε αυτό το αρχείο θα αποθηκεύονται τα περιεχόμενα του πίνακα nodes\_averages, διατηρώντας την ίδια διάταξη (γραμμές/στήλες) του αντίστοιχου πίνακα.

Στη συνέχεια ξεκινάει η διαδικασία των μετρήσεων με δυο εμφωλευμένες επαναλήψεις όπου για κάθε node (for i..) και για κάθε απόσταση/distance (for j..), θα γίνονται δεκαπέντε (15) μετρήσεις (measurements).

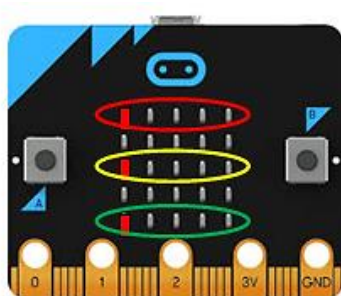
Προσοχή: Για λόγους πληρότητας της αποτύπωσης των μετρήσεων, με ενδεχόμενη μελλοντική μελέτη/επεξεργασία τους, κρίθηκε χρήσιμο να αποθηκεύονται για κάθε node,

όλες οι μετρήσεις του σε ξεχωριστό αρχείο κειμένου (text file). Έτσι για κάθε node, δημιουργείται/αρχικοποιείται – με περιεχόμενο μηδέν (0) – μία επιπλέον δισδιάστατη λίστα/πίνακας, η `nodes_mes`, στην οποία οι στήλες είναι όλες οι αποστάσεις (`cols`) και οι γραμμές είναι όλες το πλήθος των μετρήσεων (`measurements`). Όπως θα παρουσιαστεί στη συνέχεια, μόλις «γεμίσει» ο πίνακας, δημιουργείται και αντίστοιχο αρχείο (`measurements-X.txt`, όπου X ο αριθμός του node), στο οποίο αποθηκεύονται όλες οι μετρήσεις διατηρώντας την ίδια διάταξη (γραμμές/στήλες) του αντίστοιχου πίνακα.

Οι δεκαπέντε (15) μετρήσεις για το τρέχων node θα ξεκινήσουν με το επόμενο στιγμιαίο πάτημα του button A. Για το λόγο αυτό, εμφανίζονται οι κατάλληλες οδηγίες/ενδείξεις ώστε να τοποθετηθεί και το αντίστοιχο node στη σωστή/προκαθορισμένη απόσταση.

Αυτό πραγματοποιείται ταυτόχρονα με δύο τρόπους:

1. Εμφάνιση στην οθόνη του REPL (εντολές `print`), του αριθμού του node που αφορά η μέτρηση (`node`), όπως και το λεκτικό της τρέχουσας απόστασης (`distances[j]`) π.χ. «.. 0.5 m».
2. Εμφάνιση στο led-display ένδειξη, με την εξής κωδικοποίηση (ακολουθεί σχετική εικόνα):
  - Η 1<sup>η</sup> σειρά led (**κόκκινη έλλειψη**) δείχνει το ID του node για το οποίο γίνεται η μέτρηση. Στο σχήμα ανάβει το 1<sup>ο</sup> led, άρα θα ακολουθήσει μέτρηση για το 1<sup>ο</sup> node.
  - Η 3<sup>η</sup> σειρά led (**κίτρινη έλλειψη**) δείχνει τον α/α της τρέχουσας απόστασης. Στο σχήμα ανάβει το 1<sup>ο</sup> led, άρα θα ακολουθήσει μέτρηση για την 1<sup>η</sup> απόσταση (0.5 m).
  - Η 5<sup>η</sup> σειρά led (**πράσινη έλλειψη**) δείχνει τον α/α της μέτρησης. Στο σχήμα ανάβει το 1<sup>ο</sup> led, άρα θα ακολουθήσει η 1<sup>η</sup> μέτρηση από τις 15 που θα γίνουν συνολικά.



Εικόνα 31: Mode-A led display

Σημείωση: Επειδή για κάθε σειρά/έλλειψη υπάρχει ο περιορισμός των πέντε (5) led/θέσεων ακολουθείται η εξής λογική: Μόλις κάποια ένδειξη βρεθεί στην τελευταία 5<sup>η</sup> θέση, η επόμενη θα είναι πάλι η 1<sup>η</sup> θέση κ.ο.κ. Αυτό επιτυγχάνεται χρησιμοποιώντας τον τελεστή του υπολοίπου (%). Το υπόλοιπο της διαίρεσης του τρέχων α/α (node/απόστασης/μέτρησης) με το πλήθος των led της αντίστοιχης γραμμής (σταθερά 5 led), είναι πάντα μεταξύ 0 και 4 (modulo 5). Αυτό το αποτέλεσμα χρησιμοποιείται στην 1<sup>η</sup> παράμετρο της εντολής `display.set_pixel`, η οποία ενεργοποιεί το αντίστοιχο led: 0

για το 1<sup>ο</sup> led και έως 4 για το 5<sup>ο</sup> led. Η 2<sup>η</sup> παράμετρος της εντολής αφορά τη γραμμή led (0 για την 1<sup>η</sup> γραμμή και έως 4 για την 5<sup>η</sup> γραμμή). Τέλος η 3<sup>η</sup> παράμετρος αφορά τη φωτεινότητα του led (0 για σβηστό και έως 9 για μέγιστη φωτεινότητα). Η 1<sup>η</sup> παράμετρος της τελευταίας `display.set_pixel` είναι σταθερά 0 γιατί ουσιαστικά είναι η αρχική τιμή του μετρητή των μετρήσεων. Ουσιαστικά και οι τρεις (3) `display.set_pixel` λειτουργούν ως αρχικές ενδείξεις πριν ξεκινήσουν οι 15 μετρήσεις. Οι εντολές επαναλαμβάνονται στη συνέχεια του κώδικα.

Τα αντίστοιχα σχετικά μηνύματα που πρέπει να έχουν εμφανιστεί στο REPL φαίνονται στην ακόλουθη εικόνα.

```
BBC micro:bit REPL
MicroPython v1.9.2-34-gd64154c73 on 2017-09-01; micro:bit v1.0.1 with nRF51822
Type "help()" for more information.
>>>

R E S E T is pressed

Press the button "A" for calibration or "B" to calculate distance.

Calibration measurements of the node 1 from a distance of 0.5 m
Press the button "A" on the microbit to get started ...
```

Εικόνα 32: Repl mode-A μήνυμα αρχής μετρήσεων

Σημαντικό: Κάτι που παρατηρείται είναι ότι πριν ξαναπατηθεί το button A (while not `button_a.is_pressed`), απενεργοποιείται το radio (`radio.off`) και αφού πατηθεί ενεργοποιείται (`radio.on`). Αυτή η τακτική χρησιμοποιείται για να μειώσουμε στο ελάχιστο δυνατό την κατανάλωση ενέργειας. Σε πραγματικές συνθήκες, σε απομακρυσμένα και αυτόνομα ασύρματα δίκτυα, η εξασφάλιση ενέργειας για περισσότερη αυτονομία είναι ίσως η σημαντικότερη παράμετρος που πρέπει να προβλεφθεί.

Επειδή στο συγκεκριμένο σημείο του Mode A γίνεται η λήψη των μηνυμάτων από το τρέχων κόμβο, η ανάλυση του Mode A θα συνεχιστεί, αφού παρουσιαστεί ο κώδικας των nodes.

### 3.2.6. Επιλογή Mode A – Εκπομπή από κόμβο – Κώδικας κόμβων

Προϋπόθεση για να αρχίσει να λαμβάνει μηνύματα ο cluster head είναι το αντίστοιχο node, που ζητείται από τα σχετικά μηνύματα/ενδείξεις, να αρχίσει να εκπέμπει μηνύματα, από την καθορισμένη απόσταση που θα τοποθετηθεί. Η αναγνώριση/επιλογή του αντίστοιχου/σωστού node γίνεται με δύο (2) τρόπους:

1. Κάθε node, μόλις συνδεθεί με την τροφοδοσία του (μπαταρίες), εμφανίζει στο led-display του το ID του (1, 2 ή 3).
2. Κάθε node έχει αυτοκόλλητη ετικέτα πάνω δεξιά του, από την πλευρά που βρίσκονται τα button, το οποίο αναγράφει το ID του (1, 2 ή 3).

Σημαντικό: Στα ασύρματα δίκτυα, στην πράξη, ακόμα και στην περίπτωση που χρησιμοποιούνται ίδιοι κόμβοι (ίδια χαρακτηριστικά/hardware), υπάρχουν αποκλίσεις στη

συμπεριφορά τους. Δηλαδή τις περισσότερες φορές παρατηρείται, κάτω από τις ίδιες φαινομενικά συνθήκες (hardware/τροφοδοσία/περιβάλλον/απόσταση) να εκπέμπουν μηνύματα με διαφορετική ισχύ. Γι' αυτό σωστή τακτική – η οποία ακολουθείται και εδώ – είναι για κάθε κόμβο να γίνεται ξεχωριστή βαθμονόμηση. Σε όλη τη διάρκεια των πειραμάτων αυτή η αντιστοίχιση node/ID δεν αλλάζει.

Όπως έχει ήδη αναφερθεί το node εκπέμπει το ID του (1, 2 ή 3). Για να ξεκινήσει αυτή η εκπομπή, θα πρέπει – αφού τοποθετηθεί στη ζητούμενη απόσταση – να πατηθεί το button A.

Προσοχή: ΠΡΙΝ ξεκινήσει η εκπομπή μηνυμάτων (button A) από συγκεκριμένο κόμβο/node, θα πρέπει όλοι οι υπόλοιποι – κόμβοι – να είναι απενεργοποιημένοι ή να έχει πατηθεί το button RESET ώστε μόνο να εμφανίζονται το ID τους, χωρίς να εκπέμπουν.

Ο κώδικας που τρέχει στο node με ID 1, είναι ο ακόλουθος:

```
from microbit import *
import radio

# Node number
Node = 1

display.show(Node)
while not button_a.is_pressed():
    None
# Create the message.
msg = str(Node)

while True:
    # Turn radio on
    radio.on()
    # Send message.
    radio.send(msg)
    # Turn radio off
    radio.off()

    display.show(msg)
    sleep(500)
    display.clear()
    sleep(500)
```

Προσοχή: Το μόνο που αλλάζει στους κώδικες των υπόλοιπων nodes (2 και 3), είναι ο αντίστοιχος αριθμός της μεταβλητής Node (κόκκινο βέλος), σε 2 και 3 αντίστοιχα.

Λόγω της παρουσίασης προηγούμενων σεναρίων, η επεξήγηση του κώδικα των κόμβων είναι εύκολα κατανοητή: Κάθε node αρχικά δείχνει στο led-display του τον αριθμό του ID του (1, 2 ή 3) και περιμένει να πατηθεί το button A. Μόλις στιγμιαία πατηθεί το button A, μετατρέπει το ID του σε χαρακτήρες (str) και αρχίζει να το εκπέμπει (radio.send). Κάθε φορά που εκπέμπει ένα μήνυμα, αναβοσβήνει και στο led-display το αντίστοιχο ID. Η καθυστέρηση sleep(500) μετά τις εντολές display είναι απαραίτητη, γιατί λόγω ταχύτητας εκτέλεσης του κώδικα, δε θα γινόταν αντιληπτό ότι αναβοσβήνει το μήνυμα/ID.

Σημαντικό: Όπως και στον cluster head, έτσι και εδώ, πριν την αποστολή ενεργοποιείται το radio και απενεργοποιείται μετά την αποστολή. Η εξοικονόμηση κατανάλωσης ενέργειας είναι πολύ σημαντική και στους κόμβους.



### 3.2.7. Επιλογή Mode A – Λήψη μηνυμάτων

```
count = 0
sum_per = 0
while count < measurements:
    incoming = radio.receive_full()
    if incoming is not None:
        # Count the measurements.
        count += 1
        # Store the 3 values of tuple.
        msg, rssi, timestamp = incoming
        # Convert incoming bytes to string.
        msg = str(msg[3:], 'utf8')
        # RSSI (signal strength): a value between 0 (strongest) and
        # -255 (weakest) as measured in dBm.
        # Convert RSSI value to percent (%).
        rssi_per = (255+rssi)/255*100
        # Store in list the current measurement.
        nodes_mes[count-1][j] = rssi_per
        if count > rejected:
            sum_per += rssi_per
        print ('Measurement =', count, end='')
        print (' Message =', msg, end='')
        print (' from a distance of', distances[j], 'm', end='')
        print (' with signal strength =', '{:6.2f}'.format(rssi_per), '%')
        # Blinking received message.
        display.show(msg)
        sleep(500)
        display.clear()
        # Use first line (0) of pixels to show number of node.
        display.set_pixel(i % 5, 0, 9)
        # Use third line (2) of pixels to show number of distance.
        display.set_pixel(j % 5, 2, 9)
        # Use fifth line (4) of pixels to show number of measurement.
        display.set_pixel((count-1) % 5, 4, 9)
        sleep(500)
        display.clear()
```

Σε αυτό το τμήμα του κώδικα πραγματοποιούνται οι δεκαπέντε (15) μετρήσεις/επαναλήψεις (measurements) για συγκεκριμένο node (μεταβλητή *i*) και συγκεκριμένη απόσταση (μεταβλητή *j*).

Ως μετρητής – έτσι ώστε να γίνουν όσες μετρήσεις έχουν οριστεί – χρησιμοποιείται η μεταβλητή *count* η οποία αρχικοποιείται πριν τις επαναλήψεις (*while*). Επίσης πριν τις επαναλήψεις αρχικοποιείται σε μηδέν (0) και η μεταβλητή *sum\_per*, η οποία λειτουργεί ως αθροιστής των 15 μετρήσεων έτσι ώστε στο τέλος να υπολογιστεί ο μέσος όρος τους.

Με τη χρήση της *radio.receive\_full* – όπως έχει ήδη αναλυθεί σε προηγούμενη ενότητα – αποθηκεύεται στη μεταβλητή *incoming* το ληφθέν μήνυμα. Εδώ παρατηρείται ότι όσο δεν λαμβάνει κάτι ουσιαστικά επιστρέφει στη *while* και έτσι δημιουργείται ένας βρόγχος έως ότου το μήνυμα δεν είναι τίποτα (*if incoming is not None*).

Συνεπώς μόλις λάβει κάποιο μήνυμα ο cluster head (if incoming is not None), αυξάνει την count (τρέχουσα μέτρηση) και αποθηκεύει το περιεχόμενο της tuple μεταβλητής incoming, στις μεταβλητές msg, rssi, timestamp. Στο συγκεκριμένο σενάριο δε θα ασχοληθούμε με τη μεταβλητή timestamp. Η msg, που περιέχει το μήνυμα, επειδή είναι σε byte, μετατρέπεται σε string (<https://micro:bit-micropython.readthedocs.io/en/v1.0.1/radio.html#radio.receive>). Το περιεχόμενο της ουσιαστικά είναι το ID του αντίστοιχου κόμβου που κάνει εκπομπή. Το rssi, που περιέχει την ισχύ του σήματος, επειδή είναι σε dBm μετατρέπεται σε αντίστοιχο ποσοστό (rssi\_per) έτσι ώστε να είναι πιο εύκολα αντιληπτή. Αδύναμο σήμα (-255 dBm) θα αντιστοιχεί σε ποσοστό 0% και ισχυρό σήμα (0 dBm) σε ποσοστό 100%.

**Σημαντικό:** Πρέπει να γίνει ξεκάθαρο ότι οι μετρήσεις που αποθηκεύονται στους πίνακες και στα αρχεία του cluster head, είναι ουσιαστικά το περιεχόμενο της μεταβλητής rssi\_per η οποία εκφράζει την ισχύ του σήματος – σε ποσοστό – για κάθε μήνυμα που λαμβάνει.

Όπως ήδη έχει αναφερθεί, για κάθε node θα δημιουργηθεί αντίστοιχο αρχείο, με όλες του τις μετρήσεις (15 για κάθε απόσταση). Γι' αυτό το λόγο πρώτα θα αποθηκευτούν στο σχετικό πίνακα (nodes\_mes[count-1][j] = rssi\_per) και μετά το τέλος όλων των μετρήσεων – του κόμβου –, θα αποθηκευτούν στο αντίστοιχο αρχείο. Ο πίνακας όπως παρατηρείται γεμίζει ανά στήλη [j] η οποία εκφράζει την απόσταση. Ακολουθεί ενδεικτικό περιεχόμενό του:

		nodes_mes [15] [7]						
		j (cols)						
		0	1	2	3	4	5	6
count - 1 (measurements)	0	81.18	78.04	75.69	74.90	70.59	68.63	63.14
	1	81.18	77.65	77.25	74.90	71.37	69.80	64.31
	2	81.18	76.47	76.47	75.69	70.59	70.20	64.71
	3	81.18	79.22	76.08	74.90	70.59	70.20	63.14
	4	81.18	77.65	74.90	74.90	70.59	69.80	63.14
	5	81.18	78.04	75.29	74.90	70.98	70.20	64.31
	6	81.18	78.04	75.29	74.90	70.59	70.20	64.71
	7	81.18	78.43	75.29	74.90	70.59	70.20	63.14
	8	81.18	78.43	75.29	74.90	70.98	70.20	61.96
	9	81.18	78.04	75.29	74.90	70.59	70.20	63.14
	10	81.18	78.04	75.69	75.69	70.98	69.80	61.96
	11	81.57	78.04	75.69	76.08	71.37	70.20	63.92
	12	81.57	77.65	75.69	75.29	71.37	70.20	63.14
	13	81.57	78.04	75.69	76.47	70.98	70.20	62.35
	14	81.57	78.04	75.69	76.47	70.98	70.20	63.53

Πίνακας 7: Πίνακας μετρήσεων τρέχοντος κόμβου

**Σημείωση:** Η count – που εκφράζει την τρέχουσα μέτρηση από τις 15 –, επειδή χρησιμοποιείται για τα σχετικά μηνύματα στην οθόνη REPL ξεκινάει από την τιμή 1. Για να

«δείχνει» όμως στη σωστή γραμμή του πίνακα, μειώνεται κάθε φορά κατά ένα (-1). Υπενθυμίζεται ότι, οι δείκτες των πινάκων στην python, ξεκινάνε από το μηδέν (0).

Η σύγκριση που ακολουθεί (`if count > rejected`), για το συγκεκριμένο σενάριο, θα είναι πάντα αληθής, γιατί κατά την αρχικοποίηση των μεταβλητών έχει ορισθεί να μην απορρίπτονται μετρήσεις (`rejected = 0`). Συνεπώς στο άθροισμα (`sum_per += rssi_per`) θα υπολογιστούν και οι 15 μετρήσεις (`measurements`).

Προσοχή: Στον πίνακα `nodes_mes` ΑΠΟΘΗΚΕΥΟΝΤΑΙ ΟΛΕΣ ΟΙ ΜΕΤΡΗΣΕΙΣ. Δηλαδή και αυτές που λόγω της συνθήκης (`if count > rejected`) δε θα προστεθούν στο άθροισμα (`sum_per`) για τον υπολογισμό του μέσου όρου.

Σημαντικό: Σε δίκτυα WSN όταν λαμβάνουμε δεδομένα από κόμβους, πολλές φορές υπάρχουν μετρήσεις οι οποίες λόγω εξωγενών παραγόντων/παρεμβολών είναι αλλοιωμένες/λάθος. Παράδειγμα, όταν μετακινηθεί ένας κόμβος που μετρά θερμοκρασίες σε καινούργια θέση, λόγω της πιθανής αλλαγής τους περιβάλλοντος αλλά και της ταχύτητας απόκρισης του αισθητήρα του, αρχικά δε θα εκπέμψει σωστές τιμές. Πιθανή συνέπεια αυτού είναι, ο μέσος όρος των μετρήσεων, να έχει μεγάλη απόκλιση από την πραγματική τιμή της θερμοκρασίας. Αυτό όμως όπως γίνεται αντιληπτό, σε πραγματικές/κρίσιμες συνθήκες, θα οδηγήσει σε λάθος αποφάσεις. Σημαντικό ρόλο βέβαια στην απόκλιση, παίζει η αναλογία των λάθος μετρήσεων σε σχέση με το συνολικό πλήθος, δηλαδή κατά πόσο οι λάθος είναι ικανές να αλλοιώσουν το μέσο όρο. Στο συγκεκριμένο σενάριο έχει υπάρξει πρόβλεψη για την απόρριψη αρχικών μετρήσεων (`rejected`) οι οποίες δε θα ληφθούν υπ' όψιν στον υπολογισμό του μέσου όρου.

Στη συνέχεια, βάσει των περιεχομένων των μεταβλητών που ήδη έχουν εξηγηθεί (`count`, `msg`, `distances[j]`, `rssi_per`), ακολουθούν τα σχετικά μηνύματα (εντολές `print`) στην οθόνη REPL όπως φαίνονται στην ακόλουθη εικόνα:

```
R E S E T is pressed

Press the button "A" for calibration or "B" to calculate distance.

Calibration measurements of the node 1 from a distance of 0.5 m
Press the button "A" on the microbit to get started ...

Measurement = 1, Message = 1, from a distance of 0.5 m, with signal strength = 77.25 %
Measurement = 2, Message = 1, from a distance of 0.5 m, with signal strength = 77.25 %
Measurement = 3, Message = 1, from a distance of 0.5 m, with signal strength = 77.25 %
Measurement = 4, Message = 1, from a distance of 0.5 m, with signal strength = 77.25 %
Measurement = 5, Message = 1, from a distance of 0.5 m, with signal strength = 77.25 %
Measurement = 6, Message = 1, from a distance of 0.5 m, with signal strength = 77.25 %
Measurement = 7, Message = 1, from a distance of 0.5 m, with signal strength = 77.25 %
Measurement = 8, Message = 1, from a distance of 0.5 m, with signal strength = 77.25 %
Measurement = 9, Message = 1, from a distance of 0.5 m, with signal strength = 77.65 %
Measurement = 10, Message = 1, from a distance of 0.5 m, with signal strength = 77.65 %
Measurement = 11, Message = 1, from a distance of 0.5 m, with signal strength = 77.25 %
Measurement = 12, Message = 1, from a distance of 0.5 m, with signal strength = 77.65 %
Measurement = 13, Message = 1, from a distance of 0.5 m, with signal strength = 77.25 %
Measurement = 14, Message = 1, from a distance of 0.5 m, with signal strength = 77.25 %
Measurement = 15, Message = 1, from a distance of 0.5 m, with signal strength = 72.55 %
```

Εικόνα 33: `Repl mod-A` ενδεικτικές μετρήσεις

Επίσης στο `led-display` εμφανίζεται, για  $\frac{1}{2}$  sec το μήνυμα (μεταβλητή `msg`) που λαμβάνει ο `cluster head` και για άλλο  $\frac{1}{2}$  sec οι σχετικές ενδείξεις με την κωδικοποίηση που

ήδη έχει εξηγηθεί (modulo 5). Όπως παρατηρείται, επαναλαμβάνονται οι εντολές που αφορούν το led-display – έχουν ήδη αναλυθεί – με τη μόνη διαφορά στην τελευταία (display.set\_pixel), η οποία δείχνει την τρέχουσα τιμή της μέτρησης (count-1), πάντα σε modulo-5.

### 3.2.8. Επιλογή Mode A – Αποθήκευση μέσου όρου μετρήσεων

```
# Store in list the current average of measurements.
nodes_averages[i][j] = sum_per / (measurements - rejected)
# Write in the microbit file the current average of measurements.
file.write(' '+'{:6.2f}'.format(nodes_averages[i][j]))
print ('Summary ==> Node =', node, end='')
print (' , Distance =', distances[j], 'm', end='')
print (' , Average signal strength =', '{:6.2f}'.format(nodes_averages[i][j]), '%', end='')
print (' and', rejected, 'first measurements have been rejected.')
print ()
```

Μετά το τέλος των 15 μετρήσεων και PIN ακολουθήσουν οι μετρήσεις για την επόμενη επανάληψη/απόσταση (j) γίνονται οι εξής ενέργειες:

α) Υπολογίζεται ο μέσος όρος των 15 μετρήσεων για την τρέχουσα απόσταση και το αποτέλεσμα αποθηκεύεται στην αντίστοιχη θέση του πίνακα των μέσων όρων (nodes\_averages[i][j]).

Σημείωση: Επειδή στο άθροισμα (sum\_per) δεν έχουν προστεθεί οι απορριφθέντες μετρήσεις (rejected), αντίστοιχα το πλήθος τους αφαιρείται από τον υπολογισμό του μέσου όρου (measurements - rejected).

β) Αποθηκεύεται (file.write) ο τρέχων μέσος όρος (nodes\_averages[i][j]) στο αρχείο των μέσων όρων ('calibrates.txt'). Για να γίνεται διαχωρισμός των τιμών στο text αρχείο, χρησιμοποιείται το κενό (' '), το οποίο προστίθεται πριν από κάθε τιμή. Οι τιμές έχουν μήκος 6 θέσεις, τρία (3) ψηφία για το ακέραιο μέρος, μία (1) θέση για την υποδιαστολή και δύο (2) ψηφία για το δεκαδικό. (' '+'{:6.2f}').

Σημείωση: Στην επόμενη επανάληψη/απόσταση (j) η τιμή/μέσος όρος θα αποθηκευτεί δίπλα στην προηγούμενη (στην ίδια γραμμή) κ.ο.κ., έως ότου αλλάξει ο κόμβος (i), που τότε γίνεται και αλλαγή γραμμής. Ακολουθεί εικόνα με ενδεικτικό περιεχόμενο του σχετικού αρχείου (3 γραμμές/κόμβοι, 7 στήλες/αποστάσεις):

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια		
81.28	77.99	75.69	75.32	70.88	70.01	63.37
80.44	79.01	77.12	76.18	68.89	69.62	64.37
80.73	78.69	74.90	73.46	71.29	68.65	64.86

Εικόνα 34: Mod-A περιεχόμενα αρχείου μέσων όρωνcalibrates.txt

γ) Εμφανίζεται περιληπτικό (summary) μήνυμα σχετικά με τις 15 μετρήσεις που προηγήθηκαν (εντολές `print`), το οποίο περιλαμβάνει:

- Τον αριθμό του κόμβου για τον οποίο γίνονται οι μετρήσεις (Node)
- Την απόσταση των μετρήσεων (Distance)
- Το μέσο όρο των μετρήσεων (Average)
- Τις αρχικές μετρήσεις που ενδεχομένως έχουν απορριφτεί (rejected).

Ακολουθεί ενδεικτική εικόνα από την οθόνη REPL:

```
Summary ==> Node = 1, Distance = 0.5 m, Average signal strength = 81.23 % and 0 first measurements have been rejected.
```

Εικόνα 35: Repl mod-A περιληπτικό μήνυμα μετρήσεων

Σημείωση: Ο κώδικας που έχει αναλυθεί μέχρι αυτό το σημείο θα επαναληφθεί για όλες τις αποστάσεις (j).

### 3.2.9. Επιλογή Mode A - Δημιουργία αρχείου μετρήσεων για κάθε κόμβο

```
# New line in the microbit file.
file.write('\n')
# Create file with measurements of nodes.
file_mes = open('measurements-'+str(i+1)+'.txt', 'w')
# Write in the measurements file the current measurements.
for il in range(0, measurements):
    for jl in range (0, cols):
        file_mes.write(' '+{:6.2f}.format(nodes_mes[il][jl]))
    file_mes.write('\n')
file_mes.close()
```

Επειδή για κάθε κόμβο ( $i$ ) αρχικοποιείται – όπως έχει αναφερθεί – ο ίδιος πίνακας ( $nodes\_mes$ ) θα πρέπει, ΠΡΙΝ αλλάξει ο δείκτης ( $i$ ) του κόμβου, να προηγηθεί αποθήκευση των μετρήσεων ώστε να μην χαθούν. Συνεπώς, μόλις τελειώσουν οι μετρήσεις για όλες τις αποστάσεις ( $j$ ) και ΠΡΙΝ ξεκινήσουν για τον επόμενο κόμβο ( $i$ ), τα δεδομένα του «γεμάτου» πλέον πίνακα ( $nodes\_mes$ ) θα πρέπει να αποθηκευτούν στο αντίστοιχο text αρχείο ('measurements-'+str( $i+1$ )+'.txt').

**Σημείωση:** Ο πίνακας των μετρήσεων ( $nodes\_mes$ ), όπως έχει προηγουμένα αναφερθεί, «γεμίζει» ανά στήλη. Επειδή όμως στα text αρχεία, εξυπηρετεί τα δεδομένα να αποθηκεύονται ανά γραμμή, επιλέχθηκε να γίνει η αποθήκευση των στοιχείων/μετρήσεων του, αφού έχει συμπληρωθεί όλος ο πίνακας.

Η μορφή αποθήκευσης των μετρήσεων είναι η ίδια με αυτή που έχει ήδη εξηγηθεί για το αρχείο `calibrates.txt`. Επίσης, επειδή η αρίθμηση των κόμβων ξεκινάει από το μηδέν (0), στη διαμόρφωση του ονόματος του αρχείου προστίθεται το ένα ( $i+1$ ). Οι εντολές `file.write('\n')` και `file_mes.write('\n')` έχουν ως αποτέλεσμα, την αλλαγή γραμμής στα αντίστοιχα αρχεία, έτσι ώστε οι επόμενες τιμές να αποθηκεύονται σε νέα/επόμενη γραμμή.

Ακολουθεί εικόνα του αρχείου μετρήσεων του 1<sup>ου</sup> κόμβου:

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια		
81.18	78.04	75.69	74.90	70.59	68.63	63.14
81.18	77.65	77.25	74.90	71.37	69.80	64.31
81.18	76.47	76.47	75.69	70.59	70.20	64.71
81.18	79.22	76.08	74.90	70.59	70.20	63.14
81.18	77.65	74.90	74.90	70.59	69.80	63.14
81.18	78.04	75.29	74.90	70.98	70.20	64.31
81.18	78.04	75.29	74.90	70.59	70.20	64.71
81.18	78.43	75.29	74.90	70.59	70.20	63.14
81.18	78.43	75.29	74.90	70.98	70.20	61.96
81.18	78.04	75.29	74.90	70.59	70.20	63.14
81.18	78.04	75.69	75.69	70.98	69.80	61.96
81.57	78.04	75.69	76.08	71.37	70.20	63.92
81.57	77.65	75.69	75.29	71.37	70.20	63.14
81.57	78.04	75.69	76.47	70.98	70.20	62.35
81.57	78.04	75.69	76.47	70.98	70.20	63.53

Εικόνα 36: Περιεχόμενα αρχείου μετρήσεων 1ου κόμβου

### 3.2.10. Επιλογή Mode A – Τέλος κόμβων – Κλείσιμο αρχείου

```
# Close file on microbit.
file.close()
print ('Press the button "A" for calibration or "B" for calculate distance.')
print ()
```

Εφόσον τελειώσουν και όλες οι επαναλήψεις των κόμβων (i), ως τελευταίες ενέργειες απομένουν, να κλείσει το αρχείο `calibrates.txt` (`file.close`) και να εμφανιστεί σχετικό μήνυμα επιλογής mode στην οθόνη REPL (`'Press the button "A" for calibration or "B" for calculate distance.'`). Λόγω δομής/στοίχισης του κώδικα στην `python` (Παράρτημα A), η εκτέλεση θα επιστρέψει στην αρχική ατέρμονη επανάληψη (`while True:`), περιμένοντας να πατηθεί/επιλεγεί το `button A` ή `B`.

### 3.2.11. Επιλογή Mode B – Ενημέρωση πίνακα μέσων όρων από text αρχείο

Αφού έχει πραγματοποιηθεί τουλάχιστον μία φορά το mode A, στη συνέχεια μπορεί να επιλέγει το mode B.

Στη λειτουργία του mode B, όπως έχει ήδη αναφερθεί, πραγματοποιείται σύγκριση της τρέχουσας τιμής/ισχύς τυχαίου κόμβου (1, 2 ή 3), ο οποίος μπορεί να βρίσκεται σε τυχαία θέση και συγκρίνεται με την αντίστοιχη τιμή του πίνακα των μέσων όρων από τη διαδικασία του mode A. Ο σκοπός της σύγκρισης είναι, να προσδιοριστεί η απόσταση/θέση του κόμβου από τον σταθμό βάσης (`cluster head`).

Προϋπόθεση όμως για να γίνει αυτή η σύγκριση είναι ο αντίστοιχος πίνακας (`nodes_averages`) να έχει τις τιμές από το mode A. Για να εξασφαλιστεί αυτό – επειδή το `micro:bit` μπορεί να έχει αφαιρεθεί από την τροφοδοσία του – θα πρέπει από το αντίστοιχο `text` αρχείο (`'calibrates.txt'`) να ενημερωθεί ο πίνακας των μέσων όρων.



Ακολουθεί σχετική εικόνα:

```
# Press buttons B to load calibrates from microbit.
elif button_b.is_pressed():
    with open('calibrates.txt') as file:
        # Read the first line.
        f_line = file.readline()
        # Current row of list.
        i = 0
        # Read lines while to end of the file.
        while f_line:
            # Create list from values of line.
            s_line = f_line.split()
            # Covert values to float number and store them to current row of list.
            nodes_averages[i] = [float(j) for j in s_line]
            # Read the next line.
            f_line = file.readline()
            i += 1
        for i in range(0, rows):
            print (nodes_averages[i])
    # Turn radio on
    radio.on()
```

Το αρχείο των μέσων όρων ('calibrates.txt') «διαβάζεται» ανά γραμμή (file.readline). Κάθε γραμμή περιέχει τους μέσους όρους των αποστάσεων του αντίστοιχου κόμβου. Για κάθε γραμμή/κόμβο (while f\_line:) του αρχείου, γίνονται τα ακόλουθα βήματα:

- i. Δημιουργείται (f\_line.split) μια λίστα/πίνακας (s\_line) μήκους όσες και οι αποστάσεις. Λόγω του ότι το αρχείο είναι text file, κάθε θέση της λίστας περιέχει την αντίστοιχη τιμή της απόστασης σε μορφή χαρακτήρων (str).
- ii. Επειδή με αυτές τις τιμές θα γίνουν αριθμητικές πράξεις, μετατρέπονται σε float ([float(j) for j in s\_line]) και αποθηκεύονται στον πίνακα των μέσων όρων στην αντίστοιχη γραμμή/κόμβο (nodes\_averages[i]).
- iii. «Διαβάζεται» η επόμενη γραμμή του αρχείου (file.readline) και αυξάνεται ο αντίστοιχος δείκτης των κόμβων (i += 1).

Στη συνέχεια τυπώνετε στη REPL οθόνη το περιεχόμενο του πίνακα (nodes\_averages) για να υπάρχει οπτική επιβεβαίωση ότι ενημερώθηκε σωστά με τις αντίστοιχες τιμές του αρχείου. Ακολουθεί σχετική εικόνα:

```
Press the button "A" for calibration or "B" to calculate distance.
[81.28, 77.99, 75.69, 75.32, 70.88, 70.01, 63.37]
[80.44, 79.01, 77.12, 76.18, 68.89, 69.62, 64.37]
[80.73, 78.69, 74.9, 73.46, 71.29, 68.65, 64.86]
```

Εικόνα 37: Repl mode-B πίνακας μέσων όρων

Ως τελευταία ενέργεια – σε αυτό το τμήμα του κώδικα – είναι να εξασφαλιστεί ότι το module radio είναι ενεργό (radio.on).



### 3.2.12. Επιλογή Mode B – Λήψη και εκτύπωση μηνύματος

```
while True:
    incoming = radio.receive_full()
    if incoming is not None:
        # Store the 3 values of tuple.
        msg, rssi, timestamp = incoming
        # Convert incoming bytes to string.
        msg = str(msg[3:], 'utf8')
        # RSSI (signal strength): a value between 0 (strongest) and
        # -255 (weakest) as measured in dBm.
        # Convert RSSI value to percent (%).
        rssi_per = (255+rssi)/255*100
        # Convert node string to integer.
        num_node = int(msg)
        print ('Node =', num_node, ', Signal strength =', '{:6.2f}'.format(rssi_per), '%')
```

Ο cluster head εισέρχεται σε κατάσταση αναμονής/ατέρμονου βρόγχου (`while True:`) έως ότου λάβει μήνυμα (`incoming`) από κάποιον κόμβο. Μόλις λάβει κάποιο μήνυμα ο cluster head (`if incoming is not None`), εκτελεί τις ίδιες ενέργειες (μετατροπές/πράξεις) όπως και στο mode A (**ενότητα 3.2.7**). Στη συνέχεια, επειδή ο αριθμός του κόμβου από τον οποίο λαμβάνει μήνυμα θα χρησιμοποιηθεί σε αριθμητικές πράξεις, τον μετατρέπει σε ακέραια τιμή (`num_node`).

Ο κώδικας που «τρέχει» στους κόμβους είναι ακριβώς ίδιος, όπως είχε αναλυθεί στην **ενότητα 3.2.6**. Δηλαδή κάθε κόμβος, εκπέμπει ως μήνυμα το ID του (1, 2 ή 3).

Ουσιαστικά από το σήμα που λαμβάνεται, για να μπορέσει ο cluster head να προσδιορίσει τη θέση/απόσταση του κόμβου, δύο είναι οι απαραίτητες πληροφορίες:

1. Από ποιον κόμβο έλαβε μήνυμα (`num_node`)
2. Τη ισχύ του σήματος/μηνύματος (`rssi_per`)

Αυτές οι πληροφορίες τυπώνονται (`print`) στην οθόνη REPL (ακολουθεί εικόνα):

```
Node = 1 , Signal strength = 81.96 %
```

Εικόνα 38: Repl mode-B μήνυμα από κόμβο

Προσοχή: Στο mode B, τα μηνύματα μπορούν να στέλνονται από έναν ή ταυτόχρονα από περισσότερους κόμβους. Επίσης, η αντίστοιχη ενεργοποίηση/τοποθέτηση των κόμβων στο χώρο, δεν ακολουθεί κάποιον συγκεκριμένο κανόνα/σειρά. Άρα, μπορεί να έχει/ουν ενεργοποιηθεί ένας ή πολλοί κόμβοι, πριν ή αφού έχει επιλεγθεί το mode B. Αρχικά, για λόγους απλότητας, θεωρείται ότι έχει ενεργοποιηθεί μόνο ο κόμβος με ID=1.

### 3.2.13. Επιλογή Mode B – Προσδιορισμός θέσης – Εμφάνιση σε REPL

```
dif_mes[0] = abs(nodes_averages[num_node-1][0] - rssi_per)
min_mes = dif_mes[0]
pos_mes = 0
for j in range(1, cols):
    dif_mes[j] = abs(nodes_averages[num_node-1][j] - rssi_per)
    if dif_mes[j] < min_mes:
        min_mes = dif_mes[j]
        pos_mes = j
print ('          Calibrated measurements =', nodes_averages[num_node - 1])
print ('Difference (Calibrated - Measured) =', dif_mes)
print ('The node', num_node, end='')
print (' is close to', distances[pos_mes], 'meters.')
print ()
```

Υπενθυμίζεται ότι, για να προσδιοριστεί η θέση/απόσταση του κόμβου από το σταθμό βάσης, θα πρέπει να υπολογιστεί το εξής: Με ποια από τις αποθηκευμένες μέσες τιμές βαθμονόμησης, η τρέχουσα/μετρήσιμη τιμή, έχει την μικρότερη απόλυτη διαφορά (ενότητα 3.2.2).

Για αυτόν τον υπολογισμό, χρησιμοποιείται βοηθητικός πίνακας (ενότητα 3.2.4), του οποίου τις αντίστοιχες θέσεις (`dif_mes[j]`), αποθηκεύεται η απόλυτη (`abs`) διαφορά της μέσης τιμής βαθμονόμησης (`nodes_averages[num_node-1][j]`), με την τρέχουσα/μετρήσιμη τιμή (`rssi_per`).

Ο πίνακας διαφορών έχει μήκος όσο και το πλήθος των αποστάσεων (`cols`). Επίσης, η τιμή του τρέχων κόμβου μειώνεται κατά ένα (`num_node-1`) γιατί – όπως έχει ξανά αναφερθεί – στην `rython` οι δείκτες των πινάκων ξεκινάνε από το μηδέν (0), π.χ. ο πρώτος κόμβος (`ID=1`) αντιστοιχεί στη μηδενική (0) γραμμή.

Παράλληλα, στον πίνακα διαφορών (`dif_mes`), εφαρμόζεται ο κλασικός αλγόριθμος εύρεσης ελαχίστου στοιχείου σε μονοδιάστατο πίνακα. Συνεπώς, σε όποια θέση (`pos_mes`) βρεθεί η ελάχιστη διαφορά θα είναι και το ζητούμενο. Δηλαδή θα είναι η διαφορά με την πιο κοντινή τιμή στην αντίστοιχη θέση/απόσταση βαθμονόμησης. Με την εύρεση της θέσης της ελάχιστης τιμής, μπορεί να προσδιοριστεί και το αντίστοιχο λεκτικό π.χ. 1.5, από τον πίνακα των αποστάσεων (`distances[pos_mes]`).

Στην εμφάνιση των αποτελεσμάτων/πληροφοριών περιλαμβάνονται τα εξής:

- Οι μέσες τιμές της τρέχουσας γραμμής/κόμβου (`nodes_averages[num_node-1]`). Χρησιμοποιώντας μόνο το δείκτη των γραμμών, τυπώνονται όλες οι στήλες της αντίστοιχης γραμμής.
- Ο πίνακας/λίστα των διαφορών που έχουν υπολογιστεί (`dif_mes`). Επειδή δε χρησιμοποιείται δείκτης θα τυπωθεί όλο το περιεχόμενό του (μονοδιάστατος μιας γραμμής).
- Το ID του κόμβου από τον οποίο λήφθηκε το μήνυμα (`num_node`).

- Από την εύρεση της θέσης της ελάχιστης τιμής, τυπώνεται το αντίστοιχο λεκτικό της βαθμονομημένης απόστασης (distances [pos\_mes]).

Ακολουθεί παράδειγμα/ενδεικτική εικόνα της οθόνης REPL. Ο κόμβος με ID=1 έχει τοποθετηθεί μεταξύ 1.5 και 2 μέτρων:

```
Node = 1 , Signal strength = 73.73 %
  Calibrated measurements = [81.28, 77.99, 75.69, 75.32, 70.88, 70.01, 63.37]
Difference (Calibrated - Measured) = [7.55453, 4.26453, 1.96451, 1.59451, 2.84546, 3.71548, 10.3555]
The node 1 is close to 2 meters.
```

**Εικόνα 39: Repl mode -B υπολογισμός απόστασης κόμβου**

Στο συγκεκριμένο παράδειγμα δίνονται οι εξής πληροφορίες (ανά γραμμή):

- 1<sup>η</sup>: Το ID του Node από το οποίο λαμβάνεται το μήνυμα και η ισχύ του σήματος του (Node = 1 , Signal strength = 73.73 %).
- 2<sup>η</sup>: Η αντίστοιχη γραμμή του πίνακα μέσων όρων (Calibrated), που αντιστοιχεί στο Node 1. Συγκεκριμένα είναι η μηδενική (0) γραμμή.
- 3<sup>η</sup>: Οι απόλυτες διαφορές (Difference) των μέσων όρων (Calibrated) με την ισχύ της τρέχουσας μέτρησης (Measured) που έχει τυπωθεί στην 1<sup>η</sup> γραμμή (73.73 %).
- 4<sup>η</sup>: Ο προσδιορισμός της θέσης/απόστασης του Node 1 από το cluster head. Βρίσκεται κοντύτερα στην απόσταση των 2 μέτρων.

Όπως φαίνεται στην 3<sup>η</sup> γραμμή η μικρότερη απόλυτη διαφορά (75.32 - 73.73) είναι η 1.59451, η οποία αντιστοιχεί στην 3<sup>η</sup> θέση του πίνακα των αποστάσεων ([0.5, 1, 1.5, 2, 4, 8, 16]). Σημείωση: Η αρίθμηση του πίνακα ξεκινάει από το μηδέν (0).

Ακολουθούν παραδείγματα (screenshots) και με περισσότερους ενεργούς κόμβους σε διαφορετικές θέσεις/αποστάσεις.

Στην ακόλουθη εικόνα έχουν ενεργοποιηθεί δύο κόμβοι:

- με ID=1 τοποθετημένος περίπου στα 2 μέτρα
- με ID=2 τοποθετημένος περίπου στα 4 μέτρα

```
Node = 1 , Signal strength = 74.51 %
  Calibrated measurements = [81.28, 77.99, 75.69, 75.32, 70.88, 70.01, 63.37]
Difference (Calibrated - Measured) = [6.77023, 3.48022, 1.18021, 0.810211, 3.62976, 4.49979, 11.1398]
The node 1 is close to 2 meters.

Node = 2 , Signal strength = 69.02 %
  Calibrated measurements = [80.44, 79.01, 77.12, 76.18, 68.89, 69.62, 64.37]
Difference (Calibrated - Measured) = [11.4204, 9.99039, 8.1004, 7.1604, 0.129608, 0.600403, 4.6496]
The node 2 is close to 4 meters.
```

**Εικόνα 40: Repl mode-B Υπολογισμός απόστασης με δύο κόμβους ενεργούς**

Στην συνέχεια (επόμενη εικόνα) προστίθεται και τρίτος κόμβος (ID=3) τοποθετημένος περίπου στα 8 μέτρα:

```
Node = 3 , Signal strength = 67.45 %
    Calibrated measurements = [80.73, 78.69, 74.9, 73.46, 71.29, 68.65, 64.86]
Difference (Calibrated - Measured) = [13.279, 11.239, 7.44904, 6.00903, 3.83902, 1.19904, 2.59097]
The node 3 is close to 8 meters.

Node = 2 , Signal strength = 67.84 %
    Calibrated measurements = [80.44, 79.01, 77.12, 76.18, 68.89, 69.62, 64.37]
Difference (Calibrated - Measured) = [12.5969, 11.1669, 9.27689, 8.33688, 1.04688, 1.77689, 3.47311]
The node 2 is close to 4 meters.

Node = 1 , Signal strength = 75.29 %
    Calibrated measurements = [81.28, 77.99, 75.69, 75.32, 70.88, 70.01, 63.37]
Difference (Calibrated - Measured) = [5.9859, 2.69589, 0.395874, 0.0258789, 4.41409, 5.28412, 11.9241]
The node 1 is close to 2 meters.
```

Εικόνα 41: Repl mode-B Υπολογισμός απόστασης με τρεις κόμβους ενεργούς

Σημείωση: Τα μηνύματα στην οθόνη REPL εμφανίζονται με τη σειρά που λαμβάνονται από τον cluster head.

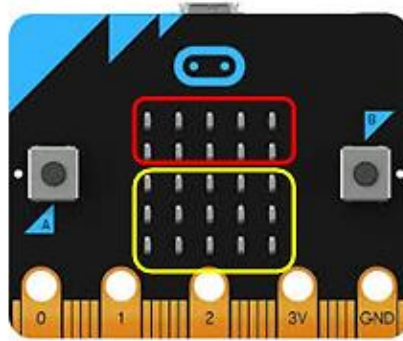
### 3.2.14. Επιλογή Mode B – Εμφάνιση σε led-display

```
display.clear()
sleep(500)
# Use first (0) and second line (1) of pixels to show number of node.
pixel_node = num_node - 1
display.set_pixel(pixel_node % 5, pixel_node // 5, 9)
# Use third (2), fourth (3) and fifth line (4) of pixels
# to show number of measurement.
display.set_pixel(pos_mes % 5, pos_mes // 5 + 2, 9)
sleep(500)
```

Όπως και στο mode A έτσι και εδώ εκτός της εμφάνισης των αποτελεσμάτων στην οθόνη REPL, υπάρχουν σχετικές ενδείξεις και στο led-display του cluster head.

Επίσης, χρησιμοποιείται παρόμοια κωδικοποίηση για την εμφάνιση των πληροφοριών. Συγκεκριμένα χωρίζουμε το led-display σε δύο τμήματα (ακολουθεί σχετική εικόνα):

- α) Η 1<sup>η</sup> και 2<sup>η</sup> σειρά led (**κόκκινο πλαίσιο**) χρησιμοποιούνται για να δείχνουν το ID του node από τον οποίο λαμβάνονται μηνύματα. Η 1<sup>η</sup> σειρά αφορά τα nodes 1-5 και η 2<sup>η</sup> σειρά τα nodes 6-10. Στο συγκεκριμένο σενάριο έχουν χρησιμοποιηθεί τρεις (3) κόμβοι.
- β) Η 3<sup>η</sup>, 4<sup>η</sup> και 5<sup>η</sup> σειρά led (**κίτρινο πλαίσιο**) χρησιμοποιούνται για να δείχνουν τον α/α της αντίστοιχης απόστασης που βρίσκεται το node. Η 1<sup>η</sup> σειρά αφορά τις αποστάσεις 1-5, η 2<sup>η</sup> σειρά τις αποστάσεις 6-10 και η 3<sup>η</sup> σειρά τις αποστάσεις 11-15. Στο συγκεκριμένο σενάριο, έχουν χρησιμοποιηθεί επτά (7) αποστάσεις ([0.5, 1, 1.5, 2, 4, 8, 16]).



Εικόνα 42: Mode-B led display

Προσοχή: Η συγκεκριμένη υλοποίηση/κωδικοποίηση, λόγω του περιορισμού του led-display, έχει τους εξής περιορισμούς:

- Μέγιστος αριθμός κόμβων: 2 γραμμές x 5 στήλες = 10
- Μέγιστος αριθμός αποστάσεων: 3 γραμμές x 5 στήλες = 15

Αρχικά σε αυτό το τμήμα του κώδικα γίνεται σβήσιμο του led-display από τυχόν προηγούμενο μήνυμα (`display.clear`). Στη συνέχεια για να γίνει ομαλά η εμφάνιση της επόμενης ένδειξης, δημιουργείται μια καθυστέρηση  $\frac{1}{2}$  sec (`sleep`) η οποία σε συνδυασμό με την καθυστέρηση της τελευταίας γραμμής δημιουργείται συνολική καθυστέρηση 1 sec. Συνεπώς η κάθε ένδειξη εμφανίζεται για 1 sec, μετά σβήνει και ακολουθεί η επόμενη.

Για τη απεικόνιση των ενδείξεων/led των κόμβων, θα πρέπει να προσδιοριστεί τόσο η στήλη όσο και η γραμμή του led-display:

- Ο δείκτης της στήλης υπολογίζεται, όπως είχε αναλυθεί και στην **ενότητα 3.2.5**, με modulo 5 (`pixel_node % 5`).
- Η σειρά/γραμμή όμως υπολογίζεται, με το ακέραιο μέρος της διαίρεσης του ID του node (`pixel_node // 5`) με το 5. Αν η τιμή του ID είναι από 1 έως και 5 το αποτέλεσμα θα είναι μηδέν (0), που αντιστοιχεί στην 1<sup>η</sup> σειρά στο led-display. Στη περίπτωση που η τιμή του ID είναι από 6 έως και 10, τότε το αποτέλεσμα είναι μονάδα (1), που αντιστοιχεί στην 2<sup>η</sup> σειρά στο led-display.

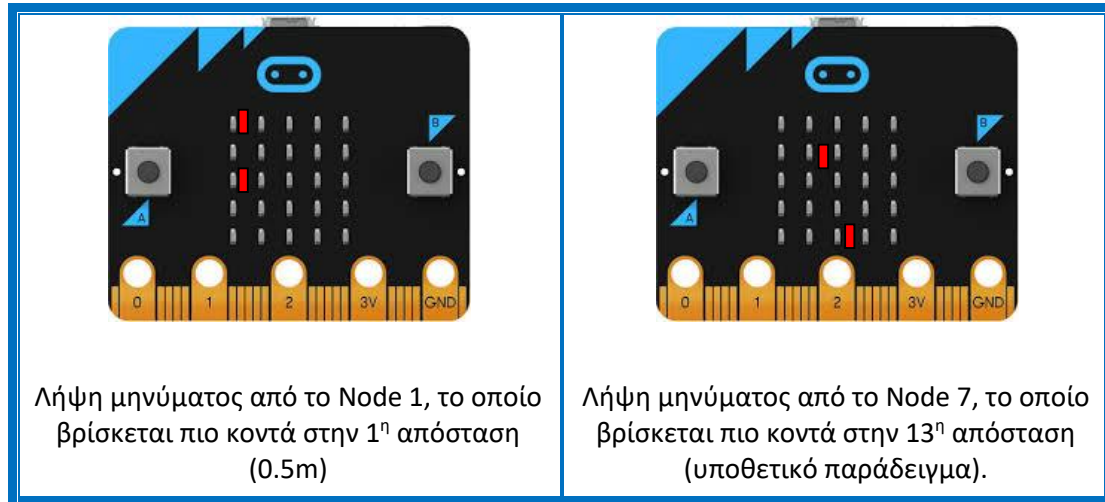
Για την απεικόνιση του α/α της αντίστοιχης απόστασης, εφαρμόζονται οι ίδιες πράξεις, με την αντίστοιχη όμως μεταβλητή (`pos_mes`):

- Ο δείκτης της στήλης υπολογίζεται, με modulo 5 (`pos_mes % 5`).
- Η σειρά/γραμμή όμως υπολογίζεται, με το ακέραιο μέρος της διαίρεσης του ID του node με το 5, αυξανόμενο κατά δύο μονάδες (`pixel_node // 5 + 2`). Επειδή οι ενδείξεις των αποστάσεων ξεκινάνε από την 3<sup>η</sup> γραμμή του led-display αυξάνουμε το αποτέλεσμα κατά δύο (+2). Π.χ. όταν το ακέραιο μέρος της πράξης είναι μηδέν (1<sup>η</sup> γραμμή led) να γίνεται δύο (3<sup>η</sup> γραμμή led) που είναι και 1<sup>η</sup> γραμμή – από τις τρεις – των ενδείξεων των αποστάσεων.

Σημείωση: Υπενθυμίζεται ότι η τελευταία παράμετρος στην `display.set_pixel` αφορά τη φωτεινότητα (`max=9`).

Προσοχή: Στη συνέχεια η λειτουργία του κώδικα επιστρέφει στην ατέρμονη (`while True:`) του mode B. Στη συγκεκριμένη υλοποίηση ΔΕΝ υπάρχει δυνατότητα εξόδου από το mode B, παρά μόνο πατώντας το button RESET.

Στις επόμενες εικόνες, παρουσιάζονται παραδείγματα λειτουργίας:



Εικόνα 43: Ενδεικτικές εικόνες led display mode - B

### 3.3. WSN – Γραφήματα

Όπως έχει αναφερθεί στην ενότητα 3.2.1 οι μετρήσεις πραγματοποιήθηκαν σε:

- α) κλειστό γήπεδο μπάσκετ
- β) κλειστή αίθουσα εκδηλώσεων

Οι χώροι επιλέχθηκαν με κριτήριο να επιτευχθεί το καλύτερο δυνατό περιβάλλον για μετρήσεις χωρίς παρεμβολές και χωρίς φυσικά ή τεχνητά εμπόδια. Εδώ καλό θα ήταν να αναφερθεί ότι στον πρώτο χώρο υπήρχαν διάσπαρτα στο χώρο, όχι στην περιοχή της λήψης των μετρήσεων, κάποια μεταλλικά αντικείμενα και υπήρχε εκείνη την ώρα και ανθρώπινη παρουσία από κάποιο συνεργείο που πραγματοποιούσε τεχνικές εργασίες σε διάφορα σημεία του. Ο δεύτερος χώρος ήταν εντελώς άδειος από αντικείμενα και ανθρώπινη παρουσία. Επιλέχθηκαν δύο χώροι με διαφορετικά χαρακτηριστικά για την δυνατότητα επαλήθευσης των αποτελεσμάτων σε δύο διαφορετικά περιβάλλοντα.

Επιλέχθηκε να παρατεθούν τα γραφήματα των μετρήσεων στον ένα και τον άλλο χώρο σε αντιπαραβολή το ένα κάτω από το άλλο για να είναι εύκολο να γίνουν συγκρίσεις.

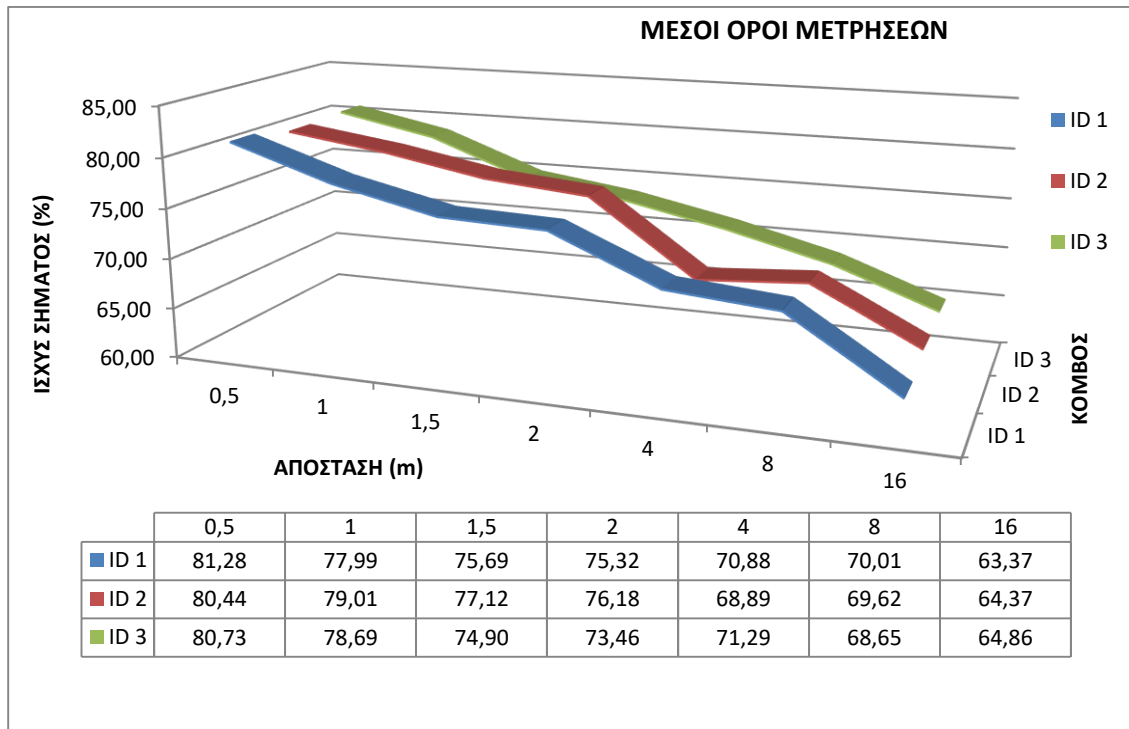
Ακολουθούν οι γραφικές παραστάσεις που μας αποτυπώνουν τη συμπεριφορά των κόμβων στο αντίστοιχο περιβάλλον.

#### 3.3.1. Μέσοι όροι Μετρήσεων

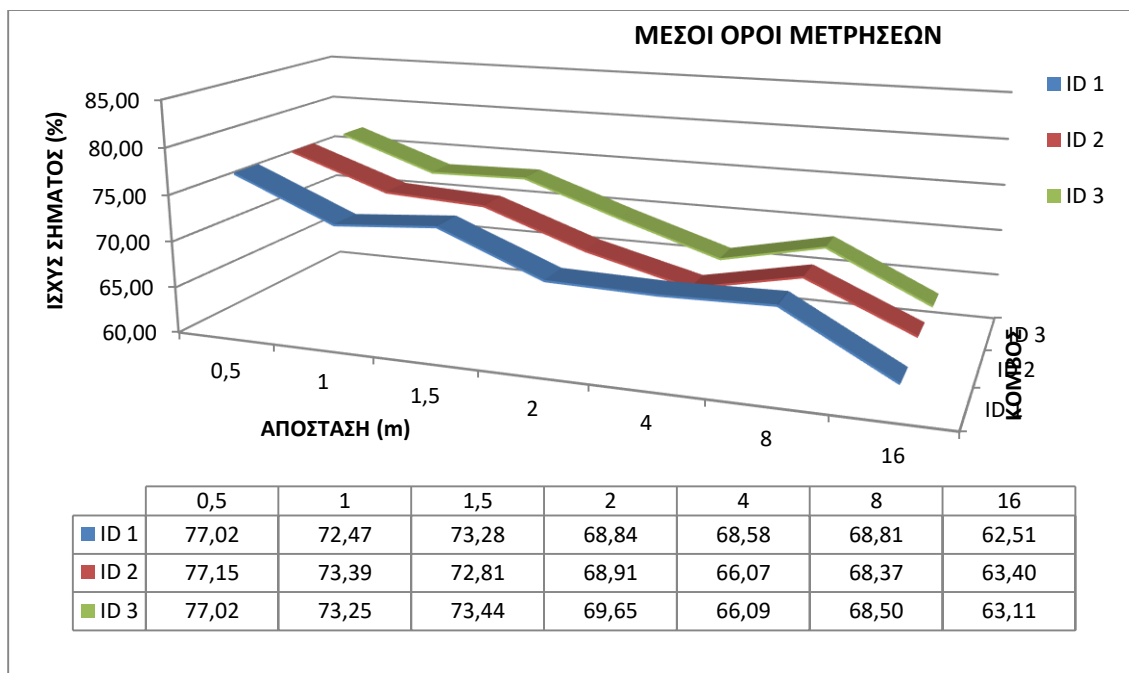
Στα δύο πρώτα διαγράμματα εμφανίζονται οι μέσοι όροι των μετρήσεων, και των τριών κόμβων. Ο μέσος όρος αναφέρεται στον μέσο όρο των 15 διαφορετικών μετρήσεων που λήφθηκαν σε κάθε διαφορετικό σημείο για κάθε έναν από τους κόμβους.

Μπορούμε να παρατηρήσουμε ότι οι μετρήσεις έχουν παρόμοια διακύμανση με μικρές αποκλίσεις και η συμπεριφορά των κόμβων έχει παρόμοια χαρακτηριστικά. Η μικρή

απόκλιση στις μετρήσεις στις ίδιες αποστάσεις θεωρείται φυσιολογική για την διαφορά του επιπέδου φορτίου των μπαταριών, τους διαφορετικούς χώρους, την διαφορά υλικών του περιβάλλοντος χώρου.



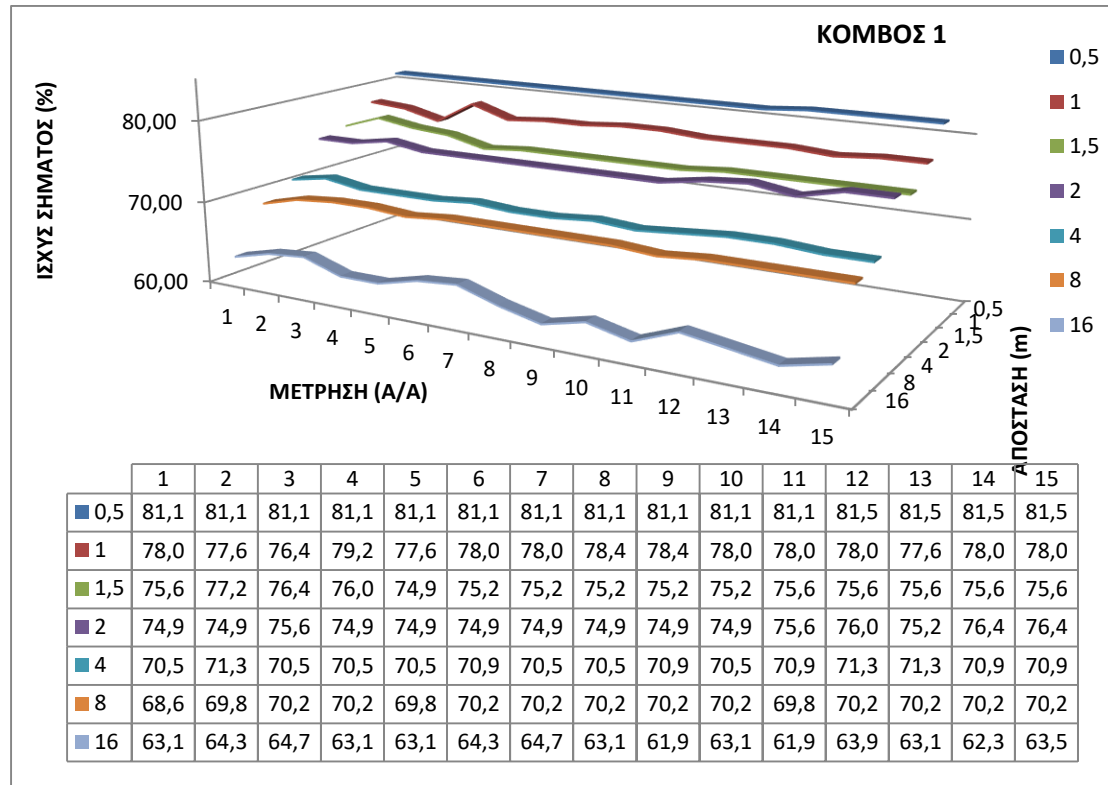
Εικόνα 44 Κλειστό Γήπεδο Μπάσκετ



Εικόνα 45: Κλειστή Αίθουσα Εκδηλώσεων

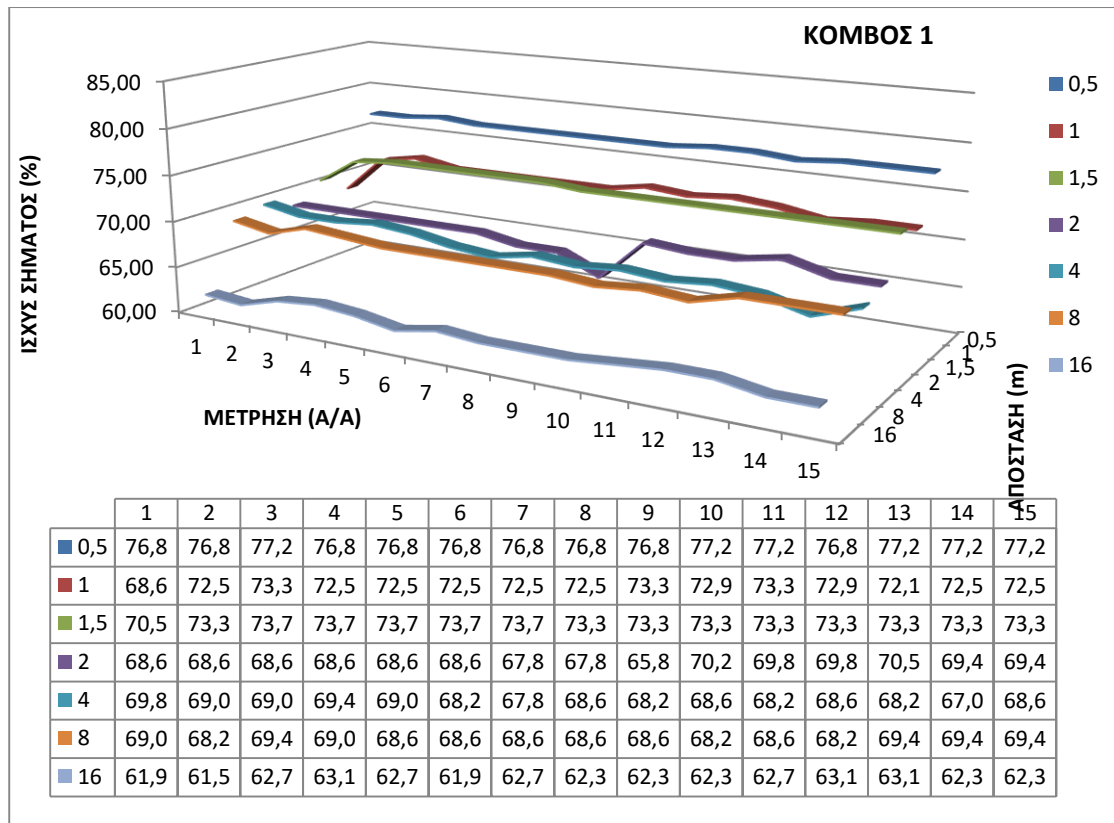


Στο δεύτερο διάγραμμα απεικονίζονται οι μετρήσεις που αφορούν τον 1<sup>ο</sup> κόμβο στις δύο διαφορετικές τοποθεσίες.

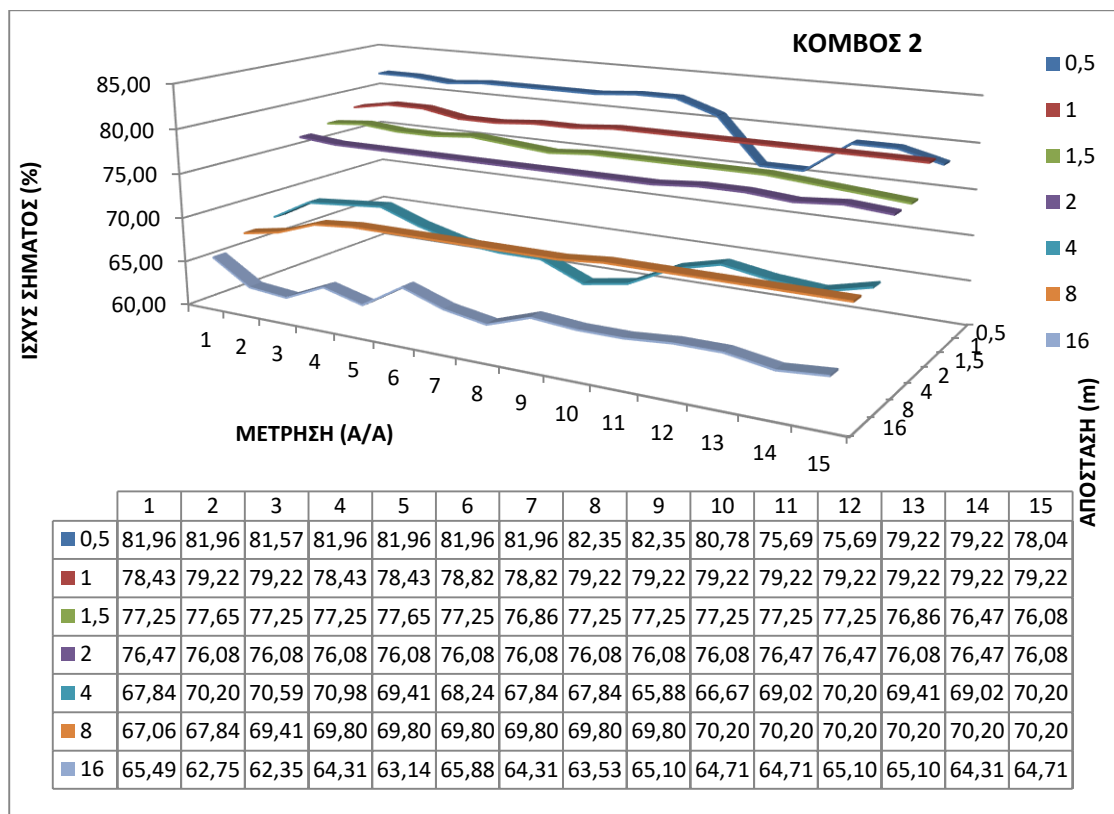


Εικόνα 46:Κόμβος 1 κλειστό γήπεδο μπάσκετ

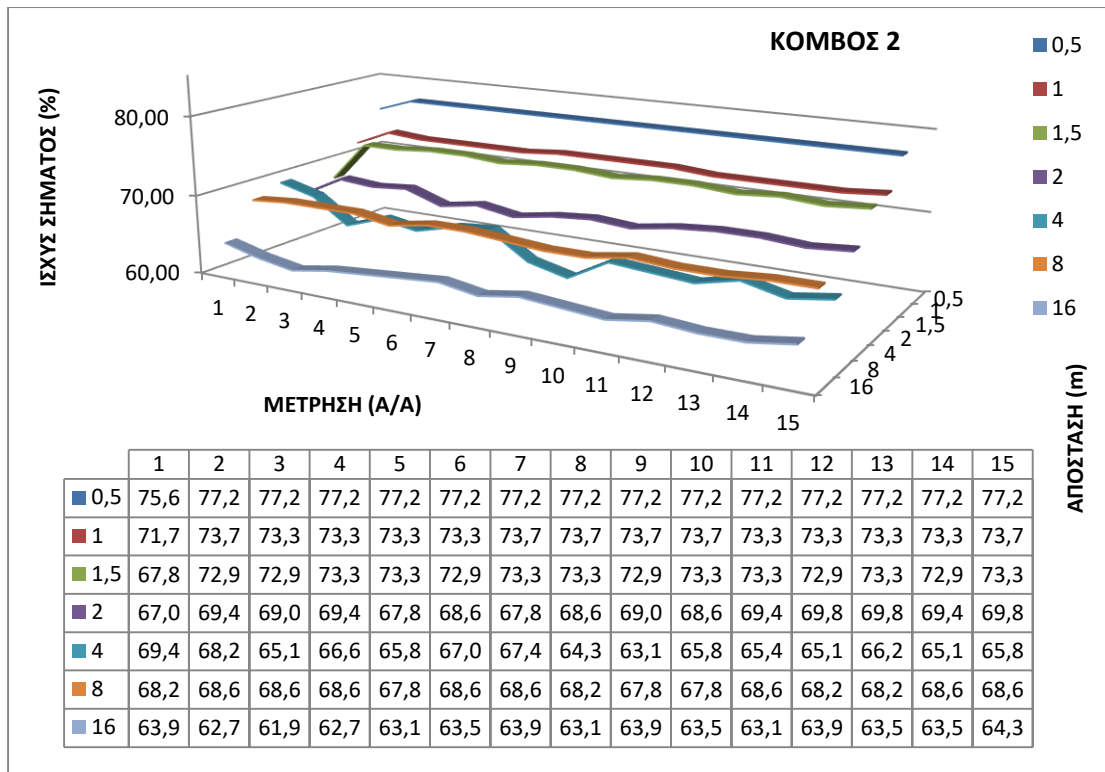




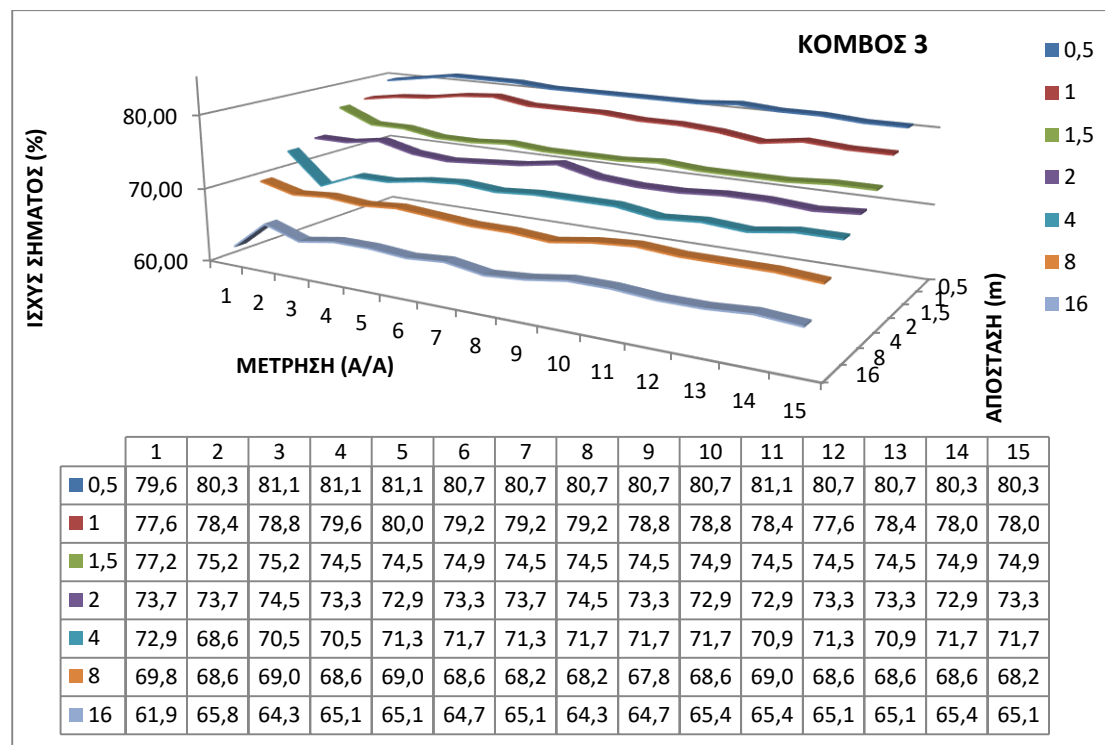
Εικόνα 47: Κόμβος 1 κλειστή αίθουσα εκδηλώσεων



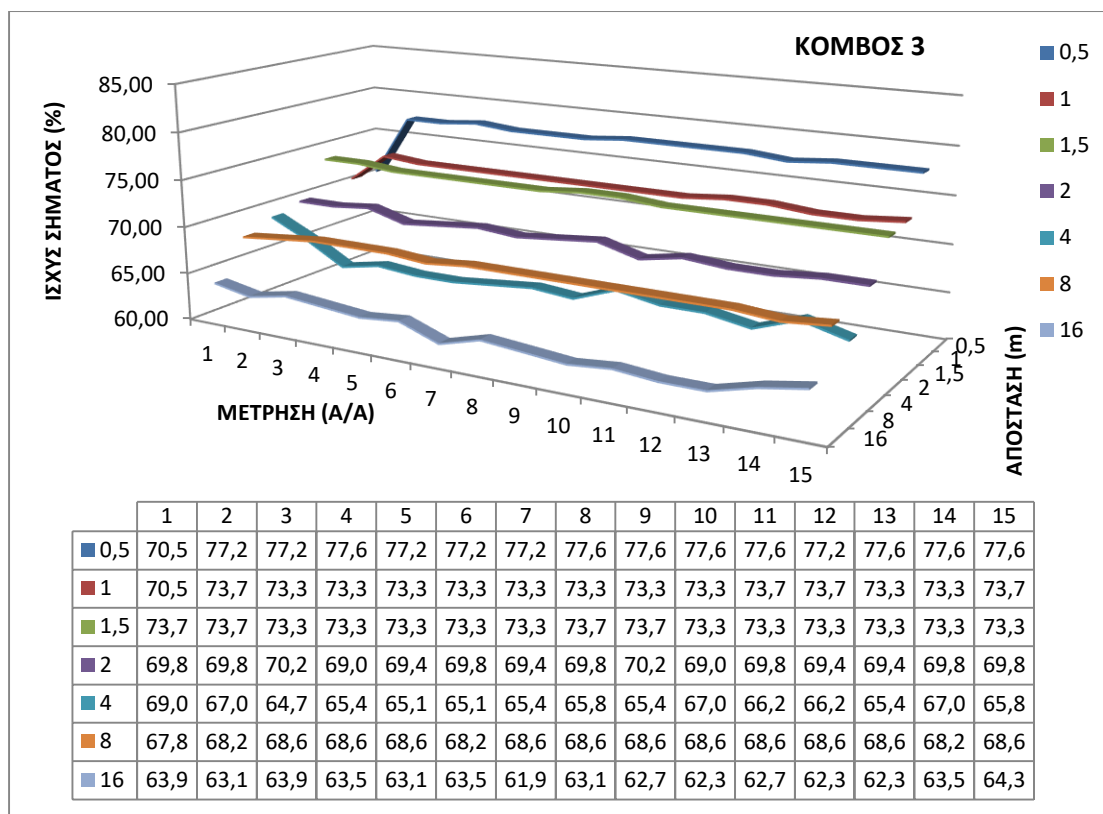
Εικόνα 48: Κόμβος 2 κλειστό γήπεδο του μπάσκετ



Εικόνα 49:Κόμβος 2 κλειστή αίθουσα εκδηλώσεων



Εικόνα 50: Κόμβος 3 κλειστό γήπεδο μπάσκετ



Εικόνα 51: Κόμβος 3 κλειστή αίθουσα εκδηλώσεων

## Κεφάλαιο 4: Επίλογος

### 4.1. Εισαγωγή

Στο κεφάλαιο αυτό γίνεται μια σύνοψη των θεμάτων με τα οποία ασχοληθήκαμε κατά την διάρκεια της παρούσης εργασίας, αναφέρονται τα προβλήματα που αντιμετωπίσαμε και οι τρόποι με τους οποίους προσπαθήσαμε να τα επιλύσουμε, καταγράφονται παρατηρήσεις και συμπεράσματα που προέκυψαν και επιχειρείται να γίνει μια αναφορά σε προτάσεις σχετικά με την εξέλιξη που θα μπορούσε να έχει το συγκεκριμένο πόνημα.

### 4.2. Σύνοψη εργασίας

Στην συγκεκριμένη εργασία μελετήσαμε την περίπτωση της δημιουργίας προγραμματιζόμενου δικτύου αισθητήρων με την χρήση μιας πλατφόρμας κόμβων που δεν έχει αναπτυχθεί αποκλειστικά για αυτή την χρήση αλλά έχει μία άλλη εκπαιδευτική φιλοσοφία. Μια πλατφόρμα που όπως είδαμε και από τα χαρακτηριστικά της πληροί όμως τις προϋποθέσεις εκείνες που πρέπει να έχει ένας κόμβος ενός WSN.

Η ιδέα δόθηκε όταν με την ιδιότητα του εκπαιδευτικού πληροφορικής δευτ/θμιας εκπαίδευσης πρωτοήρθαμε σε επαφή με την συγκεκριμένη πλατφόρμα και σκεφτήκαμε πως πέρα από τον εκπαιδευτικό της προσανατολισμό, θα μπορούσε ιδανικά να χρησιμοποιηθεί και ως πλατφόρμα για την ανάπτυξη WSN και IoT εφαρμογών.

Σχετικά με τον τρόπο με τον οποίο εξελίσσεται η πτυχιακή αυτή εργασία θα μπορούσε να αναφερθεί η σύνοψη που ακολουθεί:

Στο 1<sup>ο</sup> κεφάλαιο επιχειρείται μια όσο το δυνατόν συνοπτική αλλά και περιεκτική περιγραφή των τεχνολογιών, των εφαρμογών, των πρωτοκόλλων, καθώς και των πιο γνωστών πλατφορμών των ασύρματων δικτύων αισθητήρων.

Στο 2<sup>ο</sup> κεφάλαιο έγινε προσπάθεια να παρουσιαστεί η πλατφόρμα BBC micro:bit που επιλέχθηκε για την υλοποίηση WSN και να γίνει μια αναφορά στο υλικό, στο λογισμικό, στα πρωτόκολλα που υποστηρίζει καθώς και τις γλώσσες προγραμματισμού τις οποίες κάποιος μπορεί να χρησιμοποιήσει για να υλοποιήσει διάφορες εφαρμογές με αυτή.

Στο 3<sup>ο</sup> κεφάλαιο περιγράφονται οι ενέργειες σχετικά με την υλοποίηση των σεναρίων, πειραματικές μετρήσεις καθώς και θέματα που αντιμετωπίσαμε και στο τέλος παραθέτουμε γραφήματα που προέκυψαν από τις μετρήσεις που υλοποιήθηκαν.

Τέλος στο 4<sup>ο</sup> κεφάλαιο επιχειρείται να γίνει μια σύνοψη της εργασίας, να αναφερθούν τα προβλήματα και ο τρόπος με τον οποίο αντιμετωπίστηκαν, κάποιες παρατηρήσεις που προέκυψαν, καθώς και προτάσεις για μελλοντική εξέλιξη.

### 4.3. Προβλήματα και αντιμετώπιση

Σε κάθε πρωτότυπη εργασία που συμπεριλαμβάνει μετρήσεις/πειράματα τα προβλήματα είναι αναπόφευκτα. Έτσι και στην περίπτωση της συγκεκριμένης εργασίας τα προβλήματα ήταν αρκετά αλλά αντιμετωπίστηκαν.

Οι αρχικές δοκιμές στις βασικές επικοινωνίες των WSN (Broadcast/Multicast/Unicast) υλοποιήθηκαν σε οικιακό χώρο – σαλόνι – ο οποίος είχε αρκετά αντικείμενα και έπιπλα, με συνέπεια όσο αυξανόταν η απόσταση, λόγω κυρίως των πολλών αντανάκλασεων, να υπάρχει κακή ποιότητα στο σήμα, να παρατηρείται αστάθεια στην ισχύ του έως και μερικές φορές ολοκληρωτική απώλεια του. Η λύση ήταν να μεταφερθούν οι δοκιμές σε οικιακό χώρο με καλύτερα χαρακτηριστικά. Γι' αυτό το λόγο επιλέχθηκε στενόμακρος διάδρομος σπιτιού, χώρος με πολύ λιγότερα εμπόδια και λιγότερες αντανάκλασεις. Η ποιότητα του σήματος βελτιώθηκε αισθητά, τα προβλήματα μετάδοσης/λήψης μειώθηκαν, παρόλα αυτά επειδή υπάρχουν και παράγοντες που δε φαίνονται, όπως παρεμβολές από άλλες συχνότητες/επικοινωνίες, τα προβλήματα ήταν πολύ δύσκολο να εξαλειφθούν.

Στη διαδικασία της βαθμονόμησης των κόμβων διαπιστώθηκε ένα σοβαρό πρόβλημα. Όταν ολοκληρώνονταν οι 15 μετρήσεις π.χ. της αρχικής απόστασης του μισού μέτρου (0.5 m) και μετακινούνταν ο αντίστοιχος κόμβος στη επόμενη απόσταση π.χ. 1 m, ώστε να ληφθούν οι επόμενες μετρήσεις, ο cluster head λάμβανε μετρήσεις σαν ο κόμβος να βρίσκεται ακόμα στην προηγούμενη απόσταση (1 m). Δηλαδή, ενώ στο 0.5 m η ισχύς ήταν π.χ. 77%, στο 1 m η ισχύς μειωνόταν στο 72% – αναμενόμενο αποτέλεσμα εφόσον αυξήθηκε η απόσταση – αλλά με μεγάλη καθυστέρηση και σταδιακά, δηλαδή μετά από αρκετές μετρήσεις (10-15). Το πρόβλημα αρχικά αντιμετωπίστηκε με την τεχνική να απορρίπτονται αρχικές μετρήσεις (rejected). Παρόλα αυτά όμως, κατόπιν πολλών δοκιμών διαπιστώθηκε ότι, η ισχύς του σήματος είχε μεγάλη αστάθεια με μη προβλέψιμη συμπεριφορά, συνεπώς η τεχνική του rejected δεν έλυσε το πρόβλημα.

Τελικά μελετώντας τις δυνατότητες αλλά και τη συμπεριφορά του radio module διαπιστώθηκε μια πολύ σημαντική λεπτομέρεια, η οποία εξ' αρχής δεν είχε γίνει αντιληπτή. Στον αρχικό – μη αποτυπωμένο κώδικα στην παρούσα διπλωματική – όταν ο cluster head «έμπαινε» σε κατάσταση αναμονής, ώστε να μετακινηθεί το node στην επόμενη θέση βαθμονόμησης, το radio module – του cluster head – δεν έκλεινε (δεν υπήρχε η εντολή radio.off). Παρόλα αυτά, ενώ δεν υπήρχαν οπτικές ενδείξεις ότι γινόταν λήψη από τον cluster head, στην πραγματικότητα γινόταν. Αυτό είχε ως συνέπεια κατά τη μετακίνηση του node στην επόμενη θέση βαθμονόμησης, οι εκπομπές του – που δεν διακόπτονταν – να αποθηκεύονται στο buffer του radio module (του cluster head). Το αποτέλεσμα αυτού ήταν, όταν το node τοποθετούνταν στην καινούργια θέση βαθμονόμησης και δινόταν η εντολή στο cluster head να αρχίσει να λαμβάνει σήματα, ουσιαστικά εμφάνιζε πρώτα τις μετρήσεις που είχε αποθηκευμένες στο buffer του – κατά τη μετακίνηση του node – και μετά αυτές που λάμβανε από την καινούργια θέση του node. Αυτή ήταν και η λογική εξήγηση τις σταδιακής προσαρμογής του σήματος όταν το node μετακινιόταν σε νέα θέση.

Για να λυθεί το πρόβλημα έπρεπε να βρεθεί τρόπος να αδειάζει ο buffer του cluster head, πριν αρχίσει να λαμβάνει σήματα από τη νέα του θέση του node. Η λύση ήταν η απενεργοποίηση το radio module (radio.off) του cluster head πριν μπει σε κατάσταση

αναμονής. Διαπιστώθηκε ότι, απενεργοποιώντας το radio module γινόταν reset ο εσωτερικός buffer που έχει. Με αυτόν τον τρόπο επιτεύχθηκε και κάτι ακόμα σημαντικό, η εξοικονόμηση ενέργειας. Δηλαδή όσο ο cluster head ήταν σε αναμονή όχι μόνο δεν λάμβανε – όπως ήταν το ζητούμενο – σήματα, αλλά δεν του κατανάλωνε ενέργεια το radio module.

#### 4.4. Παρατηρήσεις – Συμπεράσματα

Λόγω της πειραματικής φύσης της διπλωματικής αλλά και του αχαρτογράφητου – όσο αφορά τη συμπεριφορά του σε WSN υλοποιήσεις micro:bit, προέκυψαν μέσω των προβλημάτων που αντιμετωπίστηκαν αρκετά χρήσιμα συμπεράσματα.

Το βασικότερο ίσως όλων, είναι οι αστάθμητοι παράγοντες του περιβάλλοντος χώρου οι οποίοι δημιουργούν στις περισσότερες των περιπτώσεων παρεμβολές έως και διακοπή της επικοινωνίας. Παρεμβολές μπορούν να δημιουργηθούν είτε από ακίνητα ή κινούμενα αντικείμενα στο χώρο όπως και από εκπομπές/λήψεις άλλων ασύρματων συσκευών επικοινωνίας. Πέρα από τον πειραματικό περιβάλλον, σε πραγματικές συνθήκες υπάρχουν επιπλέον οι καιρικές συνθήκες, η μορφολογία της περιοχής, η τυχαία διέλευση ανθρώπων, ζώων κ.α. Όλα αυτά επηρεάζουν την ποιότητα της επικοινωνίας και σε κάθε περίπτωση πρέπει να λαμβάνεται σχετική μέριμνα – όπου είναι δυνατόν – για την αποφυγή/αντιμετώπισή τους. Στους συγκεκριμένους χώρους που έγιναν οι πειραματικές μετρήσεις – πέρα από στεγασμένοι – έγινε προσπάθεια ώστε να υπάρχουν τα λιγότερα αντικείμενα στο χώρο, χωρίς την ύπαρξη άλλων ατόμων και στη λήψη κάθε μέτρησης να βρισκόμαστε τοποθετημένοι στην ίδια και σταθερή θέση στο χώρο. Παρόλα αυτά, όπως φάνηκε και από τα σχετικά γραφήματα, οι παρεμβολές ήταν αναπόφευκτες χωρίς όμως τελικά να επηρεάζουν τα αναμενόμενα αποτελέσματα.

Η εξοικονόμηση ενέργειας στα δίκτυα WSN – και όχι μόνο – είναι μια πολύ σημαντική παράμετρος έχοντας κατά νου ότι, σε πραγματικές συνθήκες μπορεί να αναφερόμαστε σε ασύρματο δίκτυο αισθητήρων τοποθετημένο σε απόμακρο και δύσβατο περιβάλλον, στο οποίο η πρόσβαση ενδεχομένως να έχει και υψηλό κόστος π.χ. τεράστια δασική έκταση για την πρόβλεψη πυρκαγιάς ή σε υψηλά υψόμετρα για τη μελέτη ακραίων καιρικών συνθηκών. Γι' αυτό το λόγο όταν δεν εκπέμπει/λαμβάνει οποιοδήποτε micro:bit του WSN πρέπει να απενεργοποιείται το radio module του, κάτι που εφαρμόστηκε και στα nodes και στο cluster head. Κατά τη διάρκεια των μετρήσεων η τροφοδοσία του κάθε micro:bit ήταν από δύο (2) αλκαλικές μπαταρίες (τύπου AAA), οι οποίες αν και χρησιμοποιήθηκαν αρκετές ώρες σε διάρκεια αρκετών μηνών, λόγω της πολύς χαμηλής κατανάλωσης, δε χρειάστηκαν αντικατάσταση. Για τις συγκεκριμένες συσκευές διαπιστώθηκε και πειραματικά, ότι έχουν πολύ χαμηλή κατανάλωση ενέργειας, συνεπώς και αυτονομία, οπότε ενδείκνυνται για υλοποιήσεις WSN.

Όσο αφορά το ίδιο το hardware των micro:bit παρατηρήθηκε μία «λεπτομέρεια» σημαντική αλλά και αναπόφευκτη! Ενώ πριν ξεκινήσουν οι μετρήσεις σε όλα τα micro:bit είχαν τοποθετηθεί καινούργιες μπαταρίες και λαμβάνοντας υπ' όψιν ότι κάθε ένα απ' αυτά τοποθετούνταν σε ίδια σημεία/αποστάσεις και στις ίδιες περιβαλλοντικές συνθήκες, παρόλα αυτά η ισχύς/ποιότητα των σημάτων είχε πολλές φορές απόκλιση. Αυτό είναι βέβαιο είναι κάτι αναμενόμενο και αναπόφευκτο γιατί είναι ουσιαστικά κατασκευαστικό «πρόβλημα». Η λέξη πρόβλημα εκφράζεται μέσα σε εισαγωγικά γιατί στη ουσία δεν είναι πρόβλημα, αλλά

οι απειροελάχιστες κατασκευαστικές αποκλίσεις που μπορεί να έχουν δυο φαινομενικά ίδια προϊόντα, κατασκευασμένα από το ίδιο εργοστάσιο, τα ίδια υλικά και το ίδιο ρομποτικό σύστημα. Παρόλα αυτά λόγω της υψηλής κλίμακας ολοκλήρωσης μπορεί να έχουν διαφορές οι οποίες γίνονται πλέον αισθητές στο πείραμα, στην πράξη.

Μια επιπλέον σημαντική παρατήρηση όσο αφορά το hardware είναι σχετικά με τη συμπεριφορά των αισθητήρων κυρίως σε σχέση με τις περιβαλλοντικές συνθήκες στις οποίες λειτουργούν. Η τιμές που λαμβάνονται από κάποιον αισθητήρα π.χ. θερμοκρασίας δεν μεταβάλλονται απότομα, συνεπώς όταν για κάποιο λόγο αλλάξουν οι συνθήκες ή αλλάξει θέση ο ίδιος ο αισθητήρας, μέχρι να προσαρμοστεί στο καινούργιο περιβάλλον θα ληφθούν κάποιες μετρήσεις με μεγάλη απόκλιση της πραγματικής. Σε αυτήν την περίπτωση πρέπει να προβλεφθεί η αγνόηση μετρήσεων (rejected) οι οποίες θα έχουν ως συνέπεια να αλλοιώσουν αισθητά την τελική πληροφορία.

Όσο αφορά την αξιοπιστία των μετρήσεων, πρέπει να αναφερθεί ότι εκτός από την τακτική να αγνοείται πλήθος αρχικών μετρήσεων, μπορεί να εφαρμοστεί και η τεχνική αξιολόγηση/εκτίμησης των μετρήσεων και επιλεκτικής απόρριψή τους. Πολλές φορές το σφάλμα/παρεμβολή στα WSN δεν παρατηρείται στις αρχικές μετρήσεις, αλλά σε ενδιάμεσες ή τυχαίες. Δηλαδή συγκεκριμένες μετρήσεις που για κάποιο λόγο έχουν μεγάλη απόκλιση με τις υπόλοιπες. Σε αυτή την περίπτωση με κατάλληλο «έξυπνο» λογισμικό θα πρέπει να φιλτράρονται ώστε να απορρίπτονται και στον υπολογισμό του μέσου όρου να χρησιμοποιούνται αυτές με την μεταξύ τους μικρότερη απόκλιση.

## 4.5. Προτάσεις – Βελτιώσεις

Έγινε μια προσπάθεια να υλοποιήσουμε ένα δίκτυο WSN και να αναπτύξουμε κάποια σενάρια δικτυακής επικοινωνίας μεταξύ κόμβων. Οι δυνατότητες που είχαμε περιόρισαν τα αποτελέσματα σε μελέτη της συμπεριφοράς 4 κόμβων στους οποίους εναλλάσσαμε ρόλους Cluster Head και απλών κόμβων. Για την καλύτερη υλοποίηση των σεναρίων επιθυμητό θα ήταν να είχαμε στην διάθεσή μας ένα μεγαλύτερο αριθμό κόμβων, ίσως μερικών δεκάδων ή εκατοντάδων. Σε αυτή την περίπτωση θα μπορούσαν να υλοποιηθούν και σενάρια για την μελέτη ιεραρχικών πρωτοκόλλων δρομολόγησης και τα συμπεράσματα που θα προέκυπταν σίγουρα θα ήταν πιο ολοκληρωμένα.

Μια άλλη περίπτωση που θα μπορούσε να μελετηθεί θα ήταν εάν υπήρχε δυνατότητα χρήσης ρομποτικών οχημάτων που θα μπορούσαν να χρησιμοποιηθούν ως βάσεις έδρασης κινητών κόμβων. Σίγουρα θα είχαμε μια καλύτερη εικόνα της συμπεριφοράς του δικτύου και της επικοινωνίας μεταξύ των κόμβων καθώς και της διαχείρισης των ενεργειακών τους αποθεμάτων σε αυτή την περίπτωση.

Ένα άλλο θέμα που παρατηρήσαμε κατά την διάρκεια των μετρήσεων μας είναι ότι ουσιαστικά λήφθηκαν στο επίπεδο του πατώματος άρα ουσιαστικά ήταν σαν να λήφθηκαν σε δύο διαστάσεις. Μια βελτίωση που θα μπορούσε να μελετηθεί θα ήταν να υπήρχε η δυνατότητα χρήσης drone και να αναρτούσαμε κάποιον ή κάποιους από τους κόμβους σε αυτό ώστε να είχαμε την ευκαιρία να μελετήσουμε την συμπεριφορά του δικτύου σε περιβάλλον 3 διαστάσεων.







# ΠΑΡΑΡΤΗΜΑ Α

## Πηγαίος κώδικας cluster head

```
1  from microbit import *
2  import radio
3  print ()
4  print ()
5  print ('R E S E T is pressed')
6  print ()
7  print ('Press the button "A" for calibration or "B" to calculate distance.')
8  print ()
9  # Number of nodes.
10 rows = 3
11 # Number of distances.
12 cols = 7
13 # Create a list of measurement distance names.
14 distances = [0.5, 1, 1.5, 2, 4, 8, 16]
15 # Number of measurements per distance.
16 measurements = 15
17 # Initial rejected measurements.
18 rejected = 0
19 # Create a list of size rows x cols, filled with zeros.
20 nodes_averages = [[0] * cols for i in range(rows)]
21 # List with differences between current and calibrated measurements.
22 dif_mes = [0] * cols
23
24 while True:
25     # Show ? on display.
26     display.show('?')
27     # Press buttons A to calibrate.
28     if button_a.is_pressed():
29         display.clear()
30         # 1/2 second delay, because the next press button A is followed.
31         sleep(500)
32         # Create file on microbit file system.
33         file = open('calibrates.txt', 'w')
34
35         for i in range(0, rows):
36             # Create a list of measurements for current node.
37             nodes_mes = [[0] * cols for il in range(measurements)]
38             for j in range(0, cols):
39                 # Real node number for displaying on screen.
40                 node = i + 1
41                 print ('Calibration measurements of the node', node, end='')
42                 print (' from a distance of', distances[j], 'm')
43                 print ('Press the button "A" on the microbit to get started ...')
44                 print ()
45                 display.set_pixel(i % 5, 0, 9)
46                 display.set_pixel(j % 5, 2, 9)
47                 display.set_pixel( 0, 4, 9)
48                 # Turn radio off
49                 radio.off()
50                 while not button_a.is_pressed():
51                     # None
52                     # Turn radio on
53                     radio.on()
54                     display.clear()
55                     sleep(500)
56                     count = 0
57                     sum_per = 0
58                     while count < measurements:
59                         incoming = radio.receive_full()
60                         if incoming is not None:
61                             # Count the measurements.
62                             count += 1
63                             # Store the 3 values of tuple.
64                             msg, rssi, timestamp = incoming
65                             # Convert incoming bytes to string.
66                             msg = str(msg[3:], 'utf8')
67                             # RSSI (signal strength): a value between 0 (strongest) and
```

```

67 # -255 (weakest) as measured in dBm.
68 # Convert RSSI value to percent (%).
69 rssi_per = (255+rssi)/255*100
70 # Store in list the current measurement.
71 nodes_mes[count-1][j] = rssi_per
72 if count > rejected:
73     sum_per += rssi_per
74     print ('Measurement =', count, end='')
75     print (' Message =', msg, end='')
76     print (' from a distance of', distances[j], 'm', end='')
77     print (' with signal strength =', '{:6.2f}'.format(rssi_per), '%')
78 # Display on Mu Plotter and create data_capture file in
79 # username\mu_code\data_capture folder.
80 # print ( int(msg), rssi_per )
81 # -----
82 # Blinking received message.
83 display.show(msg)
84 sleep(500)
85
86 display.clear()
87 # Use first line (0) of pixels to show number of node.
88 display.set_pixel( i % 5, 0, 9)
89 # Use third line (2) of pixels to show number of distance.
90 display.set_pixel( j % 5, 2, 9)
91 # Use fifth line (4) of pixels to show number of measurement.
92 display.set_pixel((count-1) % 5, 4, 9)
93 sleep(500)
94 display.clear()
95 # Store in list the current average of measurements.
96 nodes_averages[i][j] = sum_per / (measurements - rejected)
97 # Write in the microbit file the current average of measurements.
98 file.write(' '+{:6.2f}'.format(nodes_averages[i][j]))
99 print ('Summary ==> Node =', node, end='')
100 print (' Distance =', distances[j], 'm', end='')
101 print (' Average signal strength =', '{:6.2f}'.format(nodes_averages[i][j]), '%', end='')
102 print (' and', rejected, 'first measurements have been rejected.')
103 print ()
104 # New line in the microbit file.
105 file.write('\n')
106 # Create file with measurements of nodes.
107 file_mes = open('measurements-'+str(i+1)+'.txt', 'w')
108 # Write in the measurements file the current measurements.
109 for il in range(0, measurements):
110     for jl in range (0, cols):
111         file_mes.write(' '+{:6.2f}'.format(nodes_mes[il][jl]))
112     file_mes.write('\n')
113 file_mes.close()
114 # Close file on microbit.
115 file.close()
116 print ('Press the button "A" for calibration or "B" for calculate distance.')
117 print ()
118 # Press buttons B to load calibrates from microbit.
119 elif button_b.is_pressed():
120     with open('calibrates.txt') as file:
121         # Read the first line.
122         f_line = file.readline()
123         # Current row of list.
124         i = 0
125         # Read lines while to end of the file.
126         while f_line:
127             # Create list from values of line.
128             s_line = f_line.split()
129             # Covert values to float number and store them to current row of list.
130             nodes_averages[i] = [float(j) for j in s_line]
131             # Read the next line.
132             f_line = file.readline()
133             i += 1

```

```

133 for i in range(0, rows):
134     print (nodes_averages[i])
135 # Turn radio on
136 radio.on()
137 while True:
138     incoming = radio.receive_full()
139     if incoming is not None:
140         # Store the 3 values of tuple.
141         msg, rssi, timestamp = incoming
142         # Convert incoming bytes to string.
143         msg = str(msg[3:], 'utf8')
144         # RSSI (signal strength): a value between 0 (strongest) and
145         # -255 (weakest) as measured in dBm.
146         # Convert RSSI value to percent (%).
147         rssi_per = (255+rssi)/255*100
148         # Convert node string to integer.
149         num_node = int(msg)
150         print ('Node =', num_node, ', Signal strength =', '{:6.2f}'.format(rssi_per), '%')
151         dif_mes[0] = abs(nodes_averages[num_node-1][0] - rssi_per)
152         min_mes = dif_mes[0]
153         pos_mes = 0
154         for j in range(1, cols):
155             dif_mes[j] = abs(nodes_averages[num_node-1][j] - rssi_per)
156             if dif_mes[j] < min_mes:
157                 min_mes = dif_mes[j]
158                 pos_mes = j
159         print ('          Calibrated measurements =', nodes_averages[num_node - 1])
160         print ('Difference (Calibrated - Measured) =', dif_mes)
161         print ('The node', num_node, end='')
162         print (' is close to', distances[pos_mes], 'meters.')
163         print ()
164         display.clear()
165         sleep(500)
166
167 # Use first (0) and second line (1) of pixels to show number of node.
168 pixel_node = num_node - 1
169 display.set_pixel(pixel_node % 5, pixel_node // 5 , 9)
170 # Use third (2), fourth (3) and fifth line (4) of pixels
171 # to show number of measurement.
172 display.set_pixel( pos_mes % 5, pos_mes // 5 + 2, 9)
173 sleep(500)

```

## **Βιβλιογραφία - Αναφορές**

- [1] Judy J. (2001). Microelectromechanical systems (MEMS): Fabrication, design and applications. *Smart Materials and Structures*. 10. 1115-1134. 10.1088/0964-1726/10/6/301.
- [2] R. T. Howe, R. S. Muller, K. J. Gabriel and W. S. N. Trimmer, "Silicon micromechanics: sensors and actuators on a chip," in *IEEE Spectrum*, vol. 27, no. 7, pp. 29-31, July 1990, doi: 10.1109/6.58424.
- [3] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks," in *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, Aug. 2002, doi: 10.1109/MCOM.2002.1024422.
- [4] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: mobile networking for "smart dust". In *Proceedings of ACM MobiCom '99*, pp. 271–278, Seattle, WA, USA, August 1999
- [5] K.A. Delin. The Sensor Web: A Macro-Instrument for Coordinated Sensing. *Sensors*. 2002; 2(7):270-285.
- [6] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh. Monitoring volcanic eruptions with a wireless sensor network. In *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN05)*, 2005
- [7] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, *Computer Networks*, Volume 52, Issue 12, 2008, Pages 2292-2330, ISSN 1389-1286, <https://doi.org/10.1016/j.comnet.2008.04.002>.
- [8] K. Sohraby, D. Minoli and T. Znati. "Introduction and Overview of Wireless Sensor Networks 1.1 Introduction and Overview of Wireless Sensor Networks Introduction and Overview of Wireless Sensor Networks."
- [9] Karray F, Jmal MW, Abid M, BenSaleh MS, Obeid AM (2014) A review on wireless sensor node architectures. In: *The Ninth International Symposium on ReCoSoC*. IEEE, pp 1–8
- [10] Karray, Fatma & Jmal, Wassim & Abid, Mohamed & Houssaini, D.E. & Obeid, Abdulfattah & Qasim, Syed Manzoor & Bensaleh, Mohammed. (2014). Architecture of Wireless Sensor Nodes for Water Monitoring Applications: From Microcontroller-based System to SoC Solutions. 5th IMEKO TC19 Symposium on Environmental Instrumentation and Measurements 2014.
- [11] E. M. Sozer, M. Stojanovic and J. G. Proakis, "Underwater acoustic networks," in *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 72-83, Jan. 2000, doi: 10.1109/48.820738.
- [12] Chien, Thang & Nguyen Chan, Hung & Thanh, Nguyen. (2011). A Comparative Study on Hardware Platforms for Wireless Sensor Networks. *International Journal on Advanced Science, Engineering and Information Technology*. 2. 10.18517/ijaseit.2.1.157.
- [13] Arduino UNO technical specifications:  
<https://www.arduino.cc/en/pmwiki.php?n=Main/arduinoBoardUno#techspecs>

- [14] BBC micro:bit technical Specifications:  
<https://www.bbc.co.uk/mediacentre/mediapacks/micro:bit/specs/>
- [15] Adi Mallikarjuna Vanteddu, Reddy Avu, Phani Kumar, D Janakiram, G Kumar. (2007). Operating Systems for Wireless Sensor Networks: A Survey Technical Report.
- [16] Farooq, M.O.; Kunz, T. Operating Systems for Wireless Sensor Networks: A Survey. *Sensors* 2011, *11*, 5900-5930.
- [17] Challouf, Sabri & Kriaa, Lobna & Saidane, Leila. (2017). Comparison of IoT Constrained Devices Operating Systems: A Survey. 369-375. 10.1109/AICCSA.2017.187.
- [18] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM*, 20(1):46 - 61, 1973.
- [19] Philip Levis, Nelson Lee, Matt Welsh, and David Culler. 2003. TOSSIM: accurate and scalable simulation of entire TinyOS applications. In Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys '03). Association for Computing Machinery, New York, NY, USA, 126–137. DOI: <https://doi.org/10.1145/958491.958506>
- [20] Dunkels, A. et al. "Contiki - a lightweight and flexible operating system for tiny networked sensors." 29th Annual IEEE International Conference on Local Computer Networks (2004): 455-462.
- [21] TinyOS <http://www.tinyos.net/>
- [22] ContikiOS [www.contiki-os.org](http://www.contiki-os.org)
- [23] Mbed Docs <https://os.mbed.com/docs/mbed-os/v6.6/introduction/index.html>
- [24] RIOT OS Documentation <https://doc.riot-os.org/>
- [25] Zephyr OS Docs <https://docs.zephyrproject.org/latest/>
- [26] J. B. Carruthers, "Wireless infrared communications", Chapter in Encyclopedia of Telecommunications, John Wiley & Sons, Inc., 2002.
- [27] M. Stojanovic, "Acoustic (underwater) communications", In John G. Proakis, editor, Encyclopedia of Telecommunications. John Wiley & Sons, Inc., 2003.
- [28] I. F. Akyildiz, M. C. Vuran, and Z. Sun, "Signal propagation techniques for wireless underground communication networks", *Physical Communication Journal*, 2(3):167–183, September 2009.
- [29] H. Fahmy, "Wireless Sensor Networks Essentials", In *Wireless Sensor Networks, Signals and Communication Technology*, Springer, Cham. [https://doi.org/10.1007/978-3-030-29700-8\\_1](https://doi.org/10.1007/978-3-030-29700-8_1), 2020
- [30] C. Nakas, D. Kandris, and G. Visvardis, "Energy Efficient Routing in Wireless Sensor Networks: A Comprehensive Survey", *Algorithms*, vol. 13, no. 3, p. 72, Mar. 2020 [Online]. Available: <http://dx.doi.org/10.3390/a13030072>
- [31] Γεώργιος Ν. Μπράβος «Μελέτη Προσαρμοστικών Τεχνικών Μετάδοσης Πληροφορίας Σε Ασύρματα Δίκτυα Αισθητήρων», Διδακτορική διατριβή, 2008

- [32] E. Shih, S. H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks", In Proceedings of ACM MobiCom'01, pp. 272–287, Rome, Italy, 2001.
- [33] A. Sinha, A. Chandrakasan, "Dynamic power management in wireless sensor networks", IEEE Design and Test of Computers, March/April 2001.
- [34] S. Mishra and A. Nasipuri, "An adaptive low power reservation based MAC protocol for wireless sensor networks", In Proceedings of the IEEE International Conference on Performance, Computing, and Communications, pp. 731–736, Phoenix, AZ, USA, April 2004.
- [35] I. Demirkol, C. Ersoy, and F. Alagoz. MAC protocols for wireless sensor networks: a survey. IEEE Communications Magazine, 44(4):115–121, April 2006
- [36] T. Herman and S. Tixeuil. A distributed TDMA slot assignment algorithm for wireless sensor networks. In Proceedings of the Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS'04), pp. 45–58, Turku, Finland, July 2004.
- [37] C. Y. Chong, S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," Proceedings of the IEEE, Aug. 2003.
- [38] J. Kulik, W. R. Heinzelman, H. Balakrishnan, "Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks," Wireless Networks, Vol. 8, 2002, pp. 169–185.
- [39] D. Braginsky, D. Estrin, "Rumor Routing Algorithm for Sensor Networks," Proceedings of the 1st Workshop on Sensor Networks and Applications (WSNA'02), Atlanta, GA, Oct. 2002.
- [40] J. N. Al-Karaki, A. E. Kamal, "Routing techniques in wireless sensor networks: a survey", IEEE Wireless Communications, 11(6):6–28, December 2004.
- [41] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks", Ad Hoc Networks, 3(3):325–349, May 2005.
- [42] G. Khanna, S. Bagchi, and Y. Wu, "Fault tolerant energy aware data dissemination protocol in sensor networks", In Proceedings of the International Conference on Dependable Systems and Networks, pp. 795–804, Florence, Italy, June 2004.
- [43] Stann F, Heidemann J (2003) RMST: reliable data transport in sensor networks. In: First IEEE International Workshop on Sensor Network Protocols and Applications, Anchorage, AK, USA, pp. 102–112
- [44] C. Wang, K. Sohraby, V. Lawrence, B. Li, Y. Hu "Priority-based congestion control in wireless sensor networks", 2006, In: Proc. IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, pp. 22–31,
- [45] A. Rahman, A. El Saddik, W. Gueaieb, "Wireless Sensor Network Transport Layer: State of the Art", 10.1007/978-3-540-69033-7\_11, 2008.
- [46] Rathnayaka A.J.D, Potdar V. M., "Wireless Sensor Network transport protocol: A critical review", Journal of Network and Computer Applications, Volume 36, Issue 1, 2013, Pages 134-146, ISSN 1084-8045, <https://doi.org/10.1016/j.jnca.2011.10.001>.

- [47] "IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPAN)," in IEEE Std 802.15.4-2003 , vol., no., pp.1-680, 1 Oct. 2003, doi: 10.1109/IEEESTD.2003.94389.
- [48] "IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)," in IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003) , vol., no., pp.1-320, 7 Sept. 2006, doi: 10.1109/IEEESTD.2006.232110.
- [49] "IEEE Standard for Low-Rate Wireless Networks," in IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015), vol., no., pp.1-800, 23 July 2020, doi: 10.1109/IEEESTD.2020.9144691.
- [50] J. Zheng, J. L. Myung, "A comprehensive performance study of IEEE 802.15. 4", Sensor network operations 4, 218-237, 2016.
- [51] N. Sastry, D. Wagner, "Security considerations for IEEE 802.15.4 networks." Proceedings of the 2004 ACM Workshop on Wireless Security, WiSe. 32-42. 10.1145/1023646.1023654.
- [52] ZigBee Alliance. <http://www.zigbee.org/>.
- [53] ZigBee Specification : <https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf>
- [54] J. S. Lee, Y. W. Su, C. C. Shen (2007). "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi.", IECON Proceedings (Industrial Electronics Conference). 46 - 51. 10.1109/IECON.2007.4460126.
- [55] H. Labiod, H. Afifi, C. de Santis, WiFi TM, "Bluetooth TM, ZigBee TM, and Wimax TM", Springer, 2007 Available : <https://doi.org/10.1007/978-1-4020-5397-9>
- [56] S. Farahani, Chapter 1 - ZigBee Basics, "ZigBee Wireless Networks and Transceivers", Newnes, 2008, Pages 1-24, ISBN 9780750683937, <https://doi.org/10.1016/B978-0-7506-8393-7.00001-7>.
- [57] <https://en.wikipedia.org/wiki/Bluetooth>
- [58] [https://en.wikipedia.org/wiki/Bluetooth\\_Low\\_Energy](https://en.wikipedia.org/wiki/Bluetooth_Low_Energy)
- [59] M. Woolley, "Bluetooth Core Specification Version 5.2 Feature Overview, Version: 1.0.1", Revision Date: 9 December 2020, Available at: [https://www.bluetooth.com/wp-content/uploads/2020/01/Bluetooth\\_5.2\\_Feature\\_Overview.pdf](https://www.bluetooth.com/wp-content/uploads/2020/01/Bluetooth_5.2_Feature_Overview.pdf)
- [60] C. Gomez, J. Oller, and J. Paradells, "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology," Sensors, vol. 12, no. 9, pp. 11734–11753, Aug. 2012 [Online]. Available: <http://dx.doi.org/10.3390/s120911734>
- [61] M. Siekkinen, M. Hiienkari, J. Nurminen, J. Nieminen, "How low energy is Bluetooth low energy? Comparative measurements with ZigBee/802.15.4.", IEEE Wireless



Communications and Networking Conference Workshops (WCNCW), 2012, Paris, France. 10.1109/WCNCW.2012.6215496.

- [62] N. Todtenberg, R. Kraemer, "A survey on Bluetooth multi-hop networks, Ad Hoc Networks", Volume 93, 2019, 101922, ISSN 1570-8705, <https://doi.org/10.1016/j.adhoc.2019.101922>.
- [63] S.M. Darroudi, C. Gomez, "Bluetooth Low Energy Mesh Networks: A Survey", Sensors, vol. 17, no. 7, p. 1467, Jun. 2017. <https://doi.org/10.3390/s17071467>
- [64] T. Lee, M. Lee, H. Kim and S. Bahk, "A Synergistic Architecture for RPL over BLE," *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, London, UK, 2016, pp. 1-9, doi: 10.1109/SAHCN.2016.7732968.
- [65] WirelessHART communications protocol: <http://www.hartcomm2.org/index.html>.
- [66] D. Kandris, C. Nakas, D. Vomvas, G. Koulouras (2020). "Applications of Wireless Sensor Networks: An Up-to-Date Survey" *Applied System Innovation*. 3. 14. 10.3390/asi3010014.
- [67] B. Rashid, M. H. Rehmani, "Applications of wireless sensor networks for urban areas: A survey, *Journal of Network and Computer Applications*", Volume 60, 2016, Pages 192-219, ISSN 1084-8045, <https://doi.org/10.1016/j.jnca.2015.09.008>.
- [68] Boomerang shooter detection system. <http://bbn.com/boomerang>.
- [69] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, G. Zhou, J. Hui, and B. Krogh. "VigilNet: an integrated sensor network system for energy-efficient surveillance", *ACM Transactions on Sensor Networks*, 2(1):1–38, February 2006
- [70] Bokareva, T.; Hu, W.; Kanhere, S.; Ristic, B.; Gordon, N.; Bessell, T.; Jha, S. "Wireless sensor networks for battlefield surveillance", In *Proceedings of the Land Warfare Conference, Brisbane, Australia, 24-27 October 2006*; pp. 1–8.,
- [71] Lim, H.B.; Ma, D.; Wang, B.; Kalbarczyk, Z.; Iyer, R.K.; Watkin, K.L. "A soldier health monitoring system for military applications", In *Proceedings of the 2010 International Conference on Body Sensor Networks, Singapore, 7–9 June 2010*; pp. 246–249.
- [72] micro:bit Technical Specifications : <https://tech.micro:bit.org/>
- [73] <https://lancaster-university.github.io/micro:bit-docs/concepts/>
- [74] Nordic Semiconductor nRF51822 Product Brief : <https://tinyurl.com/nRF51822SoC>