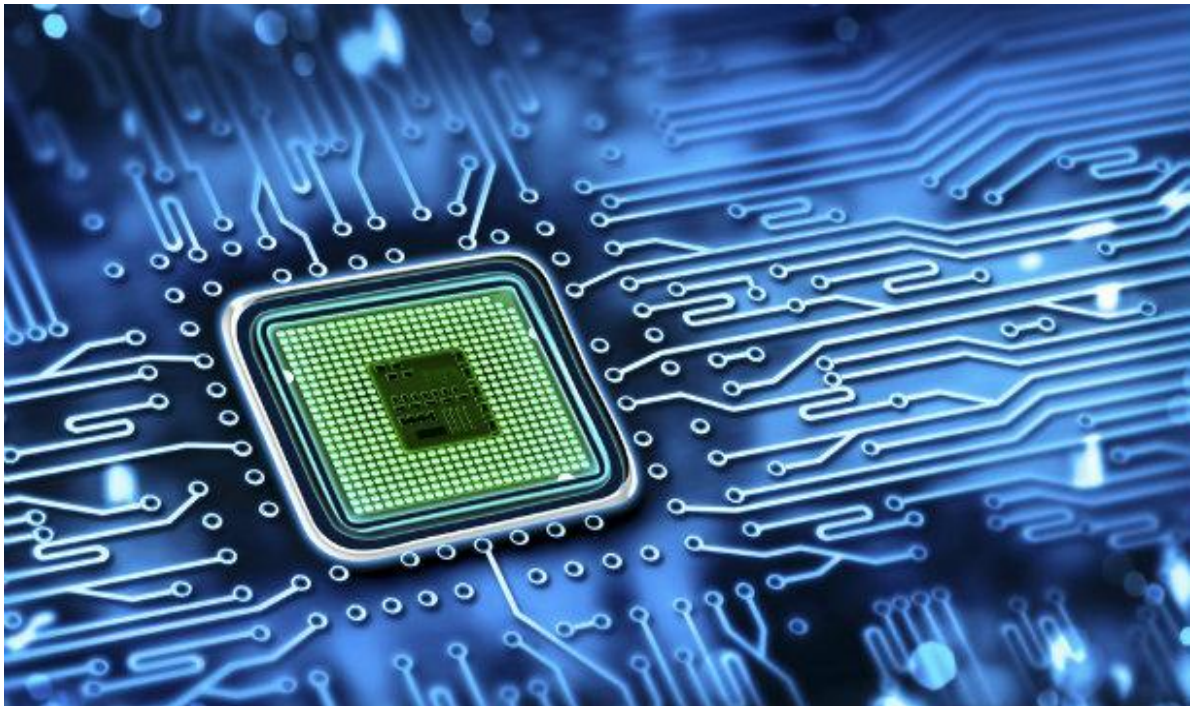


ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΣΥΡΜΑΤΟΙ ΑΙΣΘΗΤΗΡΕΣ ΚΑΙ ΜΙΚΡΟΕΛΕΓΚΤΕΣ



ΣΠΟΥΔΑΣΤΗΣ: Κυρίστης Σωτήρης

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Καρέλης Δημήτρης

ΠΑΤΡΑ 2016

©2016 - All rights reserved

©2016, Copyright από Κυρίτση Σωτήρη

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω ιδιαίτερος τον καθηγητή Καρέλη Δ. κυρίως για την εμπιστοσύνη που μου έδειξε, και την υπομονή που έκανε κατά τη διάρκεια υλοποίησης της παρούσας πτυχιακής εργασίας. Όπως επίσης και για την πολύτιμη βοήθεια και καθοδήγηση του, για την επίλυση διάφορων θεμάτων.

Επίσης, θα ήθελα να ευχαριστήσω θερμά και να εκφράσω την ειλικρινή μου ευγνωμοσύνη, σε όσους στάθηκαν δίπλα μου και με στήριξαν με κάθε τρόπο και με βοήθησαν στην ολοκλήρωση της πτυχιακής μου εργασίας.

Πάτρα, Μάιος 2016

Σωτήρης Κυρίτσης

ΠΡΟΛΟΓΟΣ

Η λέξη τεχνολογία αποτελείται από δύο μέρη τα οποία είναι η τέχνη και ο λόγος. Για αρκετούς αιώνες η τεχνολογία θεωρείται αποτέλεσμα μιας εφαρμογής, η οποία είναι θεωρητικής επιστημονικής γνώσης και έχει ως στόχο τη δημιουργία ενός αντικειμένου με πρακτικό όφελος. Η τεχνολογία ορίζεται ως η εφαρμογή της επιστημονικής γνώσης για πρακτικούς σκοπούς, ιδιαίτερα στη βιομηχανία. Ωστόσο, υπάρχει και η έννοια της λέξης που αφορά στην ανάπτυξη συσκευών και μηχανισμών για επιστημονικούς σκοπούς και εκείνον τον κλάδο της γνώσης που σχετίζεται είτε με τις εφαρμοσμένες επιστήμες είτε με τη μηχανολογία. Υπάρχουν όμως και περιπτώσεις που ο όρος τεχνολογία αναφέρεται στη μεθοδολογία που χαρακτηρίζει μια τέτοια διαδικασία.

Τα τελευταία χρόνια υπάρχει η τάση η έννοια της τεχνολογία να αναφέρεται αποκλειστικά στην υψηλή τεχνολογία και στην τεχνολογία υπολογιστών. Ο όρος έχει συνδεθεί στενά με την επιστήμη των υπολογιστών και την επεξεργασία της πληροφορίας. Ιδιαίτερα από τον καιρό όπου η πληροφορία έχει πάρει ψηφιακή μορφή δίνει την δυνατότητα στην εύκολη επεξεργασία της, με συνέπεια να δημιουργούνται απαιτήσεις υψηλότερου επιπέδου στην διαδικασία μεταφοράς πληροφοριών.

Οι απαιτήσεις αυτές βρίσκουν απαντήσεις στην δημιουργία νέων δικτύων υπολογιστών μέσω - συνδυασμού ή όχι - ενσύρματων, ασύρματων ή οπτικών καναλιών μετάδοσης. Η εξέλιξη των τελευταίων χρόνων ειδικότερα στον χώρο των ασύρματων τηλεπικοινωνιών και ηλεκτρονικών, έχει ανοίξει τον δρόμο για την δημιουργία ασύρματων δικτύων αισθητήρων – WSN (Wireless Sensor Networks) χαμηλού κόστους. Ένα τέτοιο δίκτυο αποτελείται από ένα σύνολο κόμβων πυκνά τοποθετημένων μεταξύ τους. Κάθε ένας από αυτούς αποτελείται από μια διάταξη αισθητήρων και έναν μικροελεγκτή.

Ο τρόπος αυτός δίνει την δυνατότητα να υλοποιεί απλούς υπολογισμούς και να στέλνει τα απαραίτητα μηνύματα στο δίκτυο. Η μεταφορά των μηνυμάτων μπορεί να ακολουθεί είτε τυχαία είτε προκαθορισμένη διαδρομή με απόσταση έναν κόμβο τη φορά. Μέσα από μια σειρά τέτοιων μεταφορών, τα δεδομένα που συλλέγονται από τους κόμβους φθάνουν στον κεντρικό υπολογιστή για την τελική επεξεργασία και απεικόνιση. Θα πρέπει να σημειωθεί ότι τα ασύρματα δίκτυα αισθητήρων χρησιμοποιούνται σε εφαρμογές που έχουν σχέση με την παρατήρηση και καταγραφή φυσικών φαινομένων, ή ανίχνευση συμβάντων.

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας πτυχιακής εργασίας είναι η ανάλυση του τρόπου λήψης των μηνυμάτων που στέλνονται από κόμβους ενός ασύρματου δικτύου αισθητήρων WSN. Θα πραγματοποιηθεί εκτενέστερη περιγραφή των συστημάτων που χρησιμοποιούνται σε τέτοιες περιπτώσεις με στόχο την κάλυψη του θεωρητικού υποβάθρου τους.

Στο 1^ο Κεφάλαιο περιγράφονται τα ασύρματα δίκτυα αισθητήρων. Αρχικά πραγματοποιείται μια εισαγωγή με στόχο να την κατανόηση αυτών των δικτύων καθώς και τα χαρακτηριστικά που παρουσιάζουν. Επιπλέον, δίνεται έμφαση στα πλεονεκτήματα που έχουν και αναλύονται οι εφαρμογές που μπορούν να υλοποιηθούν με την βοήθεια αυτών.

Στο 2^ο Κεφάλαιο πραγματοποιείται ανάλυση του πρωτοκόλλου IEEE 802.15.4/ZigBee. Στην αρχή περιγράφονται οι τοπολογίες δικτύου και τα χαρακτηριστικά τους. Ακολουθεί η αρχιτεκτονική WPAN και η ανάλυση των επιπέδων PHY Layer και MAC Layer. Επιπλέον, δίνεται έμφαση στο ZigBee και προγράφεται το επίπεδο δικτύου, το επίπεδο εφαρμογής και η πραγματοποίηση της δρομολόγηση μηνυμάτων στο δίκτυο.

Στο 3^ο Κεφάλαιο γίνεται αναφορά στο λειτουργικό TinyOS και στη γλώσσα nesC. Το TinyOS είναι το λειτουργικό σύστημα ανοικτού κώδικα που δίνει την δυνατότητα στον χρήστη να επέμβει και να το τροποποιήσει σύμφωνα με τις απαιτήσεις του. Η σπουδαιότητα του μας δίνει το ερέθισμα να παρουσιαστή η αρχιτεκτονική του. Επιπλέον, αναφέρονται και τα δομικά χαρακτηριστικά της γλώσσα προγραμματισμού που χρησιμοποιείται και στην περίπτωση αυτή είναι η γλώσσα nesC.

Στο 4^ο Κεφάλαιο παρουσιάζεται το hardware της εταιρίας Crossbow, το οποίο χρησιμοποιείται για την υλοποίηση μεγάλου φάσματος εφαρμογών. Δίνονται τα τεχνικά χαρακτηριστικά των στοιχείων που αποτελούν ένα κόμβο όπως ο μικροελεγκτής ATmega128L της ATMEL, ο RF transceiver CC2420 της Chipcon και η διάταξη των αντίστοιχων αισθητήρων.

Στο 5^ο Κεφάλαιο αναλύεται η software πλατφόρμα Moteworks της Crossbow. Αρχικά παρουσιάζονται τα χαρακτηριστικά και ο τρόπος λειτουργίας του λογισμικού XMesh που βασίζεται στα πρότυπα ZigBee και 802.15.4.

Τέλος στο 6^ο Κεφάλαιο παρατίθενται τα συμπεράσματα της παρούσας πτυχιακής εργασίας καθώς και η βιβλιογραφία που χρησιμοποιήθηκε για την υλοποίηση της.

ΠΕΡΙΕΧΟΜΕΝΑ

Ευχαριστίες

Πρόλογος

Περίληψη

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1: *Ασύρματα δίκτυα αισθητήρων*

1.1	Εισαγωγή	1
1.2	Πλεονεκτήματα των Ασυρμάτων Δικτύων Αισθητήρων	2
1.3	Χαρακτηριστικά των Ασυρμάτων Δικτύων Αισθητήρων	3
1.4	Εφαρμογές	5

Κεφάλαιο 2: *Πρωτόκολλο IEEE 802.15.4/ZigBee*

2.1	Εισαγωγή	10
2.2	Τοπολογίες δικτύου	11
2.3	Αρχιτεκτονική WPAN	14
2.4	PHY Layer	17
2.5	MAC Layer	22
2.6	Ασφάλεια	28
2.6.1	Υπηρεσίες ασφάλειας	29
2.6.2	Security modes	29
2.7	ZigBee	30
2.7.1	Επίπεδο δικτύου	33
2.7.2	Επίπεδο εφαρμογής	33
2.7.3	Δρομολόγηση μηνυμάτων στο δίκτυο	34

Κεφάλαιο 3: *Το λειτουργικό TinyOS και η γλώσσα nesC*

3.1	Εισαγωγή	36
-----	----------	----

3.2	Αρχιτεκτονική TinyOS	37
3.3	Η γλώσσα προγραμματισμού nesC	39
3.3.1	Δομικά χαρακτηριστικά nesC	42
3.3.2	Τρόπος λειτουργίας	46
Κεφάλαιο 4:	Hadware της εταιρίας Crossbow	
4.1	Εισαγωγή	48
4.2	Module MICAZMPR2600	49
4.2.1	Atmel ATmega 128L microprocessor	34
4.2.2	RF transceiver CC2420	41
4.3	Module MTS400 sensorboard	70
4.3.1	Αισθητήρας θερμοκρασίας και υγρασίας	71
4.3.2	Αισθητήρας πίεσης	73
4.3.3	Αισθητήρας μέτρησης δονήσεων στους X και Y άξονες	76
4.3.4	Αισθητήρας φωτός	79
4.4	Module MIB520	83
Κεφάλαιο 5:	Πλατφόρμα λογισμικού Moteworks	
5.1	Εισαγωγή	84
5.2	Βαθμίδες δικτύου	85
5.3	Mote tier	86
5.3.1	Χαρακτηριστικά του XMesh	88
5.3.2	Σχηματισμός του δικτύου	90
5.3.3	Προγραμματισμός κόμβων και σταθμού βάσης	92
5.4	Server tier	96
5.4.1	Εντολές του Xserve	97
5.5	Client Tier	98
Κεφάλαιο 6:	Συμπεράσματα	100
ΒΙΒΛΙΟΓΡΑΦΙΑ		101

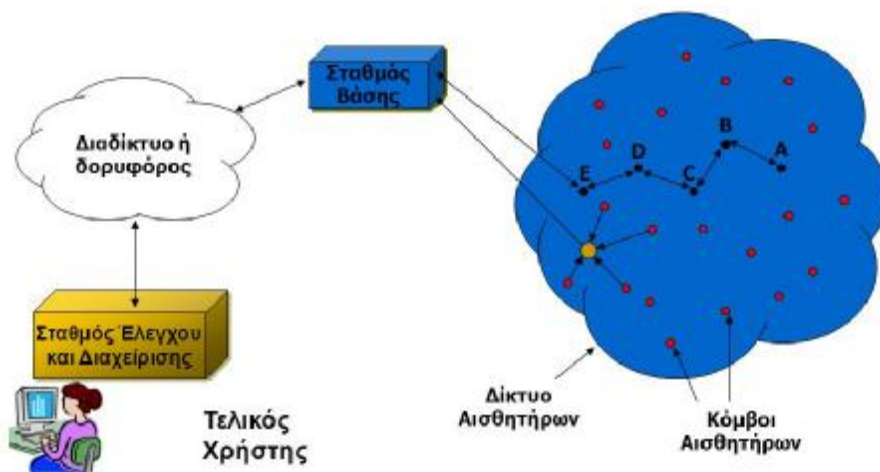
ΚΕΦΑΛΑΙΟ 1: ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ

1.1. ΕΙΣΑΓΩΓΗ

Τα ασύρματα δίκτυα αισθητήρων (WSN - Wireless Sensor Networks) θεωρούνται ως τα καταλληλότερα συστήματα για τον έλεγχο και την παρατήρηση φυσικών φαινομένων, ιδιαίτερα σε περιπτώσεις που η τοποθεσία θεωρείται απομακρυσμένη ή δύσκολα προσβάσιμη. Τα ασύρματα δίκτυα αποτελούνται από έναν κεντρικό αποδέκτη (base station) και κάποιους περιφερικούς κόμβους. Κάθε κόμβος έχει δυνατότητα συλλογής δεδομένων που έχουν να κάνουν με τις περιβαλλοντικές συνθήκες γύρωθεν αυτού. Τα δεδομένα αυτά μπορεί να είναι:

- Θερμοκρασία
- Υγρασία
- Πίεσης
- Κίνηση αντικειμένων
- Φωτεινότητα

Όλα αυτά δρομολογούνται μέσω μιας σειράς μεταφορών (multi – hop)επικοινωνίας προς τη βάση, η οποία με την σειρά της επικοινωνεί με τον τελικό χρήστη μέσω του διαδικτύου ή άλλου δικτύου (Μανωλόπουλος Χ., 2010)



Σχήμα 1.1. Ασύρματο δίκτυο αισθητήρων
(Παπαβασιλείου Σ., 2010)

1.2. ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΩΝ ΑΣΥΡΜΑΤΩΝ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ

Τα ασύρματα δίκτυα αισθητήρων προσφέρουν πλεονεκτήματα τα οποία δεν μπορούν να δώσουν τα αντίστοιχα συμβατικά συστήματα αισθητήρων, αφού για την λειτουργία γίνεται χρήση καλωδίων με σκοπό την μεταφορά δεδομένων και οι διατάξεις των αισθητήρων έχουν μεγάλο κόστος (Μανωλόπουλος Χ., 2010).

Τα πλεονεκτήματα που παρουσιάζουν τα ασύρματα δίκτυα αισθητήρων είναι τα εξής:

✓ *Εύκολη εγκατάσταση και συντήρηση:*

Η τοποθέτηση των αισθητήρων δεν παρουσιάζει τους περιορισμούς των εγκαταστάσεων με καλώδια. Καθίσταται δυνατή η ανάπτυξη και επέκταση του δικτύου σε τοποθεσίες όπου η εγκατάσταση, η συντήρηση και ο έλεγχος των συσκευών ενός ενσύρματου δικτύου είναι από πολύ δύσκολη έως ανέφικτη.

✓ *Χαμηλό κόστος*

Συγκριτικά με αντίστοιχες ενσύρματες εγκαταστάσεις έχουν χαμηλό κόστος, αφού περιορίζεται αισθητά η χρήση καλωδιώσεων. Εδώ θα πρέπει να σημειωθεί ότι σε βιομηχανικές εφαρμογές μεγάλης κλίμακας, το κόστος των καλωδιώσεων και της εγκατάστασης ενός ενσύρματου συστήματος μπορεί να φτάσει μέχρι και το 80 % του συνολικού κόστους του συστήματος.

✓ *Αυξημένη κάλυψη του δικτύου*

Υπάρχει μεγάλη δυνατότητα κάλυψης των δικτύων ως αποτέλεσμα του μεγάλου αριθμού αυτόνομων κόμβων που το αποτελούν. Κάθε σημείο μέτρησης γίνεται προσβάσιμο μέσω αυτού. Επιπλέον, είναι πολύ εύκολη η επέκταση του δικτύου με την προσθήκη νέων κόμβων, χωρίς να δημιουργείται πρόβλημα στη λειτουργία και τη διαχείρισή του από τον χρήστη.

✓ *Απλοποιημένη σχεδίαση του δικτύου.*

Έχουν αναπτυχθεί πολλά ευέλικτα πρωτόκολλα που υποστηρίζουν πλήθος δυναμικών τοπολογιών. Ανάλογα με την εφαρμογή, μπορεί να επιλεγεί η κατάλληλη τοπολογία, η οποία δύναται να αλλάζει παράλληλα με τη μεταβολή των επικρατουσών συνθηκών, για παράδειγμα σε περιπτώσεις βλάβης ενός κόμβου ή αλλαγής τοποθεσίας της βάσης συλλογής των δεδομένων.

✓ Υψηλότερος ρυθμός μετάδοσης δεδομένων

Ο υψηλότερος ρυθμός μετάδοσης δεδομένων μπορεί να γίνει κατανοητός με ένα παράδειγμα από το χώρο των βιομηχανικών δικτύων. Το δημοφιλές πρωτόκολλο HART έχει ρυθμό μετάδοσης 1.2 Kbps και το Foundation Fieldbus 31.25 Kbps, ενώ το ασύρματο Wireless HART, που βασίζεται στο πρότυπο επικοινωνίας IEEE 802.15.4, έχει ρυθμό μετάδοσης 250 Kbps. Η διαφορά μεταξύ των ενσύρματων και του ασύρματου δικτύου είναι εμφανής.

✓ Μεγάλη ανοχή σε σφάλματα

Τα ασύρματα δίκτυα αισθητήρων παρουσιάζουν μεγάλη αντοχή σε σφάλματα, που αυτό οφείλεται στη θέση όπου είναι παρατεταγμένοι οι κόμβοι στο χώρο. Η μεγάλη πυκνότητα και η σωστή διασπορά τους στο χώρο μπορεί να δώσει ένα πλεόνασμα πληροφορίας που σε περίπτωση αστοχίας ενός κόμβου θα χρησιμοποιηθεί για την ανάκτηση της χαμένης πληροφορίας. Σε αντίστοιχη περίπτωση αστοχίας κόμβου σε ενσύρματο δίκτυο, είναι πιθανότερη η απώλεια δεδομένων, κατά συνέπεια η λανθασμένη λειτουργία του δικτύου (Κατσαούνης Γ., 2013).

1.3.ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ ΑΣΥΡΜΑΤΩΝ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ

Τα ασύρματα δίκτυα αισθητήρων παρουσιάζονται σε πληθώρα εφαρμογών. Τα χαρακτηριστικά που διαθέτουν προκύπτουν από τις προδιαγραφές που ορίζεται από την χρήση τους. Τα κύρια χαρακτηριστικά των ασύρματων δικτύων αισθητήρων είναι τα ακόλουθα:

✓ Περιορισμένοι Πόροι:

Οι κόμβοι των WSN έχουν ως μόνες πηγές ενέργειας τις μπαταρίες. Η παροχή ενέργειας δεν είναι μεγάλης ισχύος και δεν είναι διαθέσιμη επ' αόριστον. Οι κόμβοι θα πρέπει να είναι σχεδιασμένοι με σκοπό να λειτουργούν για μεγάλα χρονικά διαστήματα, αυτό έχει ως συνέπεια να παρουσιάζουν τα εξής:

- Περιορισμένη υπολογιστική ισχύ
- Περιορισμένη μνήμη

- Περιορισμένο εύρος ζώνης συχνοτήτων

Ωστόσο, θα πρέπει να σημειωθεί ότι υπάρχει εναλλακτική περίπτωση και είναι τα φωτοβολταϊκά panels τα οποία μπορούν να χρησιμοποιηθούν μόνο για εφαρμογές εξωτερικών χώρων.

✓ *Multi – hot επικοινωνία :*

Για να πραγματοποιηθεί η μεταφορά των μηνυμάτων από ένα κόμβο ο οποίος βρίσκεται στα όρια περιοχής του δικτύου προς τον κεντρικό αποδέκτη, γίνεται μέσω ενδιάμεσων κόμβων. Οι κόμβοι αυτοί έχουν την δυνατότητα συλλογής και αποστολής των δικών τους μετρήσεων, καθώς επίσης και προώθηση μηνυμάτων άλλων κόμβων όταν αυτοί βρίσκονται στο ενδιάμεσο τμήμα της διαδρομής. Στην ουσία γίνεται μια επαναχρησιμοποίηση του εύρους ζώνης. Θα πρέπει να σημειωθεί ότι χρησιμοποιούνται μικρότερες αποστάσεις για την μετάδοση των μηνυμάτων που έχει σαν συνέπεια να απαιτείται μικρότερη ισχύ εκπομπής άρα και μείωση της απαιτούμενης ενέργειας (εξοικονόμηση ενέργειας).

✓ *Ad- hoc αρχιτεκτονική:*

Με αυτή την αρχιτεκτονική οι κόμβοι των WSN έχουν την δυνατότητα να παραταχθούν τυχαία σε μια περιοχή και να οργανωθούν μόνοι τους με στόχο τον σχηματισμό πλεγμάτων. Τονίζεται ότι παρουσιάζουν μεγάλη σταθερότητα και αντοχή σφάλματος, αφού δίνεται η δυνατότητα όταν κάποιος κόμβος τεθεί εκτός λειτουργίας αναδιοργανώνονται μεταξύ τους και σχηματίζουν νέες διαδρομές προώθησης.

✓ *Απόκρυψη δεδομένων*

Το σημαντικότερο σε αυτά τα συστήματα είναι η ασφάλεια, ιδιαίτερα σε συστήματα που εφαρμόζονται κάτω από συγκεκριμένες συνθήκες (στρατιωτικές εφαρμογές) και απαιτούν την μέγιστη ασφάλεια των δεδομένων που μεταφέρονται, την προστασία της λειτουργίας καθώς και την απόκρυψη κάποιων δεδομένων. Στην τελευταία περίπτωση απαιτείται περισσότερη υπολογιστή ισχύ με συνέπεια περισσότερη ενέργεια και πόρους (Μανωλόπουλος Χ., 2010).

1.4. ΕΦΑΡΜΟΓΕΣ

Τα ασύρματα δίκτυα αισθητήρων λόγω των πλεονεκτημάτων που προσφέρουν, της δυνατότητας τους να συλλέγουν δεδομένα από ένα τεράστιο φάσμα φυσικών μεγεθών ακόμη και σε αντίξοες συνθήκες, καθώς και της αξιοπιστίας, ευελιξίας και εύκολης επέκτασής τους, μπορούν να χρησιμοποιηθούν για τον έλεγχο ή τον αυτοματισμό πολλών διαδικασιών. Έτσι, εφαρμόζονται σε πολλούς κλάδους της βιομηχανίας ή της επιστήμης.



Σχήμα 1.2: Τα ασύρματα δίκτυα αισθητήρων μπορούν να χρησιμοποιηθούν σε πληθώρα εφαρμογών.

[Πηγή: <http://microcontrollerslab.com>]

Επιπλέον, τα ασύρματα δίκτυα αισθητήρων μπορούν να διαχωριστούν σε δυο βασικές κατηγορίες, οι οποίες είναι οι εξής:

- Της επίβλεψης (monitoring)
- Της ανίχνευσης (tracking)

Αυτές με την σειρά τους μπορούν να χωριστούν σε:

- Παρακολούθηση χώρου
- Παρακολούθηση αντικειμένων
- Παρατήρηση της αλληλεπίδρασης των αντικειμένων και περιβάλλοντος χώρου

Κάποιες εφαρμογές είναι οι κάτωθι:

- Περιβαλλοντικές εφαρμογές,
- Γεωργικές εφαρμογές
- Εφαρμογές πρόληψης καταστροφών και παροχής βοήθειας
- Οικιακές εφαρμογές
- Επιτήρηση μηχανών και βιομηχανικές εφαρμογές
- Επιτήρηση αντικειμένων
- Εφαρμογές ασφαλείας
- Στρατιωτικές εφαρμογές
- Τηλεματική - έλεγχος μεταφορών και συγκοινωνιών
- Ιατρικές εφαρμογές και Υγιεινή
- Άλλες εμπορικές εφαρμογές (Σπένδας Α., 2012).

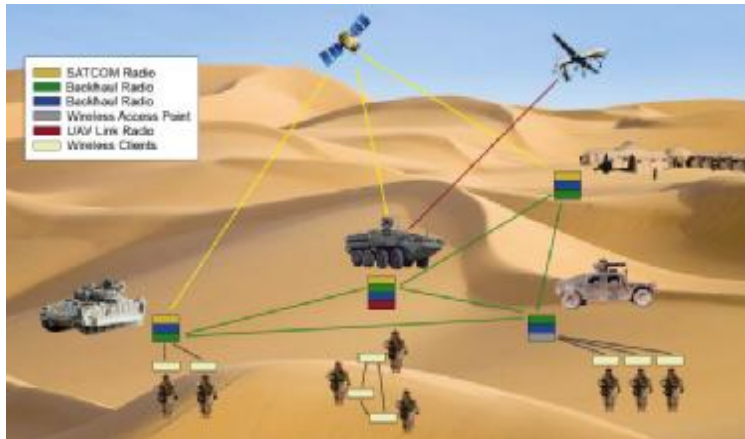
Στην συνέχεια ακολουθεί η περιγραφή κάποιων από τις προαναφερόμενες εφαρμογές οι οποίες χρησιμοποιούνται ευρύτατα στις μέρες μας (Κατσαούνης Γ., 2013).

▼ Περιβαλλοντικές Εφαρμογές

Το πρώτο μέρος περιβαλλοντικών εφαρμογών αφορά τη διενέργεια μετεωρολογικών και γεωλογικών ερευνών με χρήση αισθητήρων νερού, βροχής, ξηρασίας, θερμοκρασίας, ταχύτητας αέρα και άλλων. Το δεύτερο μέρος αφορά τη χαρτογράφηση της βιοποικιλότητας του φυσικού περιβάλλοντος. Περιλαμβάνει την τοποθέτηση αισθητήρων σε διάφορα ζώα για τη μελέτη των συνηθειών και των κινήσεών τους, με σκοπό την καταγραφή και τη μελέτη τους, καθώς και τη μέτρηση διαφόρων μεγεθών του αέρα και της θάλασσας. Τέλος, σημαντική είναι η χρήση WSN για την παρακολούθηση της παραγωγής σε γεωργικές γαίες.

▼ Στρατιωτικές Εφαρμογές

Εφαρμογές WSN για στρατιωτικούς σκοπούς χρησιμοποιούνται για την παρακολούθηση των στρατιωτικών δυνάμεων, του εξοπλισμού και των πυρομαχικών, την επίβλεψη του πεδίου μάχης, την αναγνώριση εχθρικών δυνάμεων, την κατασκευή σύγχρονων συστημάτων ανίχνευσης και στόχευσης καθώς και τον έλεγχο βιολογικών ή χημικών απειλών.

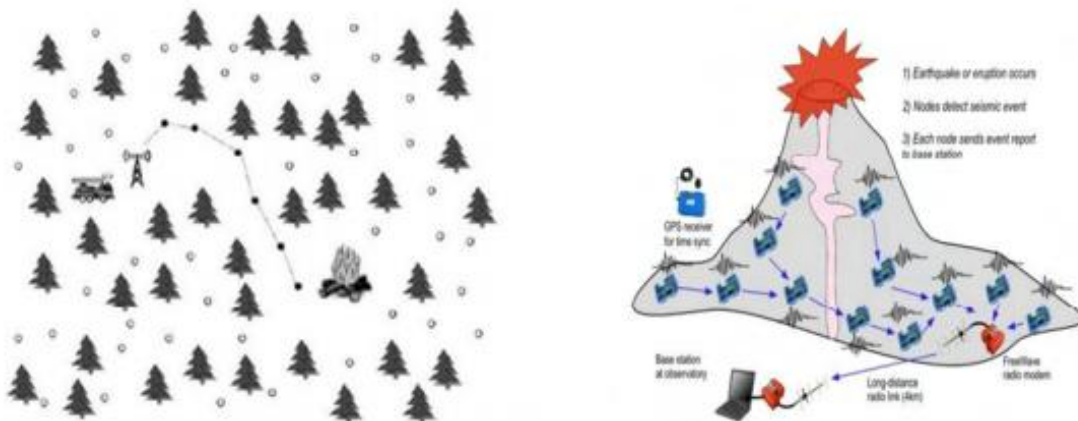


Σχήμα 1.3: Εφαρμογές WSN για στρατιωτικούς σκοπούς.

[Πηγή: <http://microcontrollerslab.com>]

✓ Πρόληψη Φυσικών Καταστροφών

Παρότι η κατηγορία θα μπορούσε να ενταχθεί στο ευρύτερο φάσμα των περιβαλλοντικών εφαρμογών, αναφέρεται ξεχωριστά εξαιτίας της ιδιαίτερης σημασίας της. Έχουν αναπτυχθεί WSN για την παρακολούθηση δασικών εκτάσεων για ανίχνευση πυρκαγιών, όπως και για εξέταση τυχόν διαρροών σε περιβάλλοντα με χημικά προϊόντα. Μια άλλη σημαντική εφαρμογή είναι η παρακολούθηση σεισμογενούς ή ηφαιστιογενούς δραστηριότητας.



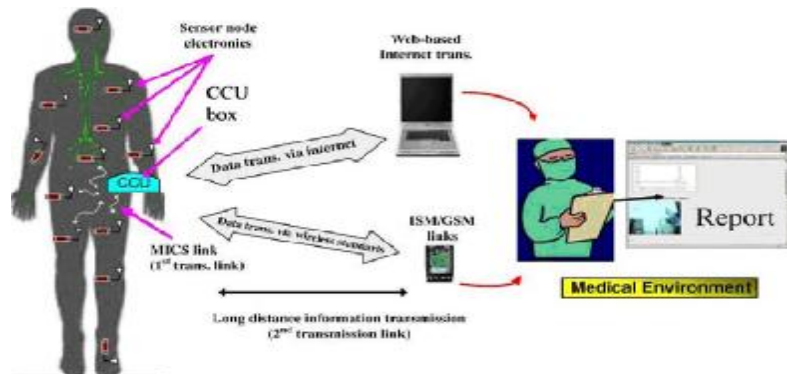
Σχήμα 1.4: Η πρόληψη φυσικών καταστροφών μπορεί να πραγματοποιηθεί με την χρήση

WSN, α) Παρακολούθηση δασικών εκτάσεων, β) παρακολούθηση ηφαιστειών

[Πηγή: <http://microcontrollerslab.com>]

▼ Ιατρική

Σε πολλές περιπτώσεις ενσωματώθηκαν ειδικοί αισθητήρες στο σώμα ασθενών ή ηλικιωμένων για την παρακολούθηση ορισμένων φυσιολογικών δεδομένων, είτε κατά τη διάρκεια κάποιας επέμβασης, είτε κατά τη μετεγχειρητική περίοδο. Επίσης, ακόμα και μέσα στο νοσοκομείο, γιατροί κουβαλούν μαζί τους έναν ειδικό κόμβο που δείχνει τη θέση τους, ώστε να επιτευχθεί η έγκαιρη ειδοποίησή τους σε περίπτωση ανάγκης.

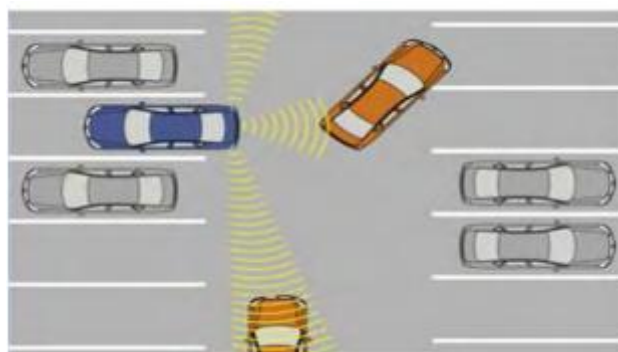


Σχήμα 1.4: Χρήση WSN στον τομέα της ιατρικής.

[Πηγή: <http://www.cse.wustl.edu/~jain/cse574-08/ftp/medical/>]

▼ Εφαρμογές Ανίχνευσης

Είναι πολύ διαδεδομένη η χρήση αισθητήρων τοποθετημένων σε κατάλληλες θέσεις για τον έλεγχο και τον ακριβή προσδιορισμό της θέσης κάποιου αντικειμένου ή της διαδρομής του στο χώρο. Σπουδαία παραδείγματα είναι τα σύγχρονα αυτοκίνητα, που διαθέτουν συστήματα εντοπισμού αντικειμένων για χρήση, συνήθως, κατά το παρκάρισμα, όπως και τα σύγχρονα συστήματα ασφαλείας.

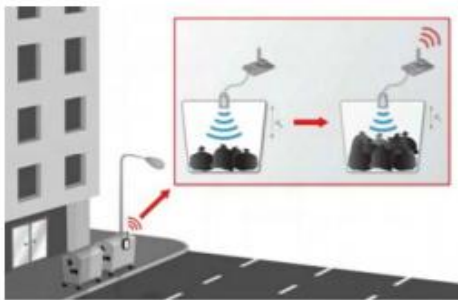


Σχήμα 1.5: Χρήση WSN για τον εντοπισμό εμποδίων

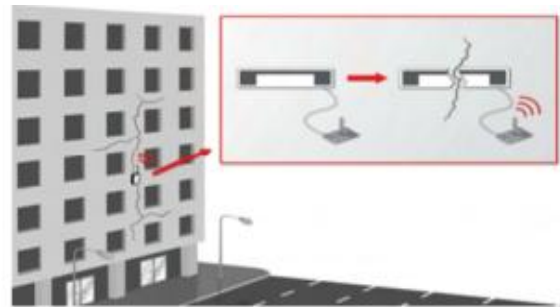
[Πηγή: <http://microcontrollerslab.com>]

▼ Αυτοματισμοί Έξυπνων Κτιρίων και Εγκαταστάσεων

Κατά το σχεδιασμό αυτοματισμών διαχείρισης κτιριακών εγκαταστάσεων είναι απαραίτητη η χρήση WSN. Επιτυγχάνεται εξοικονόμηση ενέργειας και ο έλεγχος των συνθηκών κλιματισμού μέσω μετρήσεων μεγεθών, όπως για παράδειγμα θερμοκρασίας ή υγρασίας, αυτοματοποίηση διαδικασιών, όπως το άνοιγμα-κλείσιμο θυρών ή το πότισμα κήπων, έλεγχος ασφαλείας, καθώς και έλεγχος δομικών στοιχείων για την εξακρίβωση της καταλληλότητάς τους. Ενδεικτικές είναι οι εικόνες που ακολουθούν σχετικά με μία αντίστοιχη εφαρμογή στους δρόμους μιας πόλης.



Σχήμα 1.6. α: Έλεγχος πληρότητας κάδων



Σχήμα 1.6. β: Έλεγχος δομικών στοιχείων κτιρίων

Σχήμα 1.6: Έξυπνη πόλη

Εφαρμογή WSN για αυτοματισμούς έξυπνων εγκαταστάσεων

[Πηγή: <http://microcontrollerslab.com>]

ΚΕΦΑΛΑΙΟ 2: ΠΡΩΤΟΚΟΛΛΟ ΙΕΕΕ 802.15.4/ZIGBEE

2.1. ΕΙΣΑΓΩΓΗ

Το Personal Area Network WPAN είναι ένα ασύρματο δίκτυο, χαμηλού κόστους το οποίο έχει την δυνατότητα να επιτρέπει ασύρματη επικοινωνία σε εφαρμογές μικρών απαιτήσεων και περιορισμένης ενεργειακής αυτονομίας. Τα κύρια χαρακτηριστικά ενός τέτοιου συστήματος είναι τα εξής:

- Εύκολη εγκατάσταση
- Αξιοπιστία στην μεταφορά δεδομένων
- Short – range λειτουργία
- Χαμηλή κατανάλωση ενέργειας

Επιπλέον, πρέπει να σημειωθεί ότι το πρωτόκολλο επικοινωνίας είναι το IEE802.15.4. Το WPAN το αποτελούν δύο διαφορετικοί τύποι και είναι οι εξής:

✓ Συσκευές πλήρους λειτουργίας (Full Function Device FFD)

Η συσκευή FFD έχει την δυνατότητα να επικοινωνήσει και με τους δύο τύπους συσκευών, δηλαδή και FFD και RFD. Ακόμα, χρησιμοποιείται ως συντονιστής του δικτύου PAN coordinator.

✓ Συσκευές περιορισμένης λειτουργίας (Reduced Function Device FFD)

Η συσκευή RFD δεν έχει τόσες δυνατότητες όπως η FFD, μπορούν να επικοινωνήσουν μόνο με των τύπο FFD και χρησιμοποιούνται για απλές εφαρμογές και συνήθως για μεταφορά δεδομένων από αισθητήρες. Αυτό είναι αποτέλεσμα ότι οι πόροι και η μνήμη που χρησιμοποιούν είναι περιορισμένα (Μανωλόπουλος X., 2010).

2.2. ΤΟΠΟΛΟΓΙΕΣ ΔΙΚΤΥΟΥ

Ένα WPAN δίκτυο μπορεί να λειτουργήσει με δύο διαφορετικούς τύπους τοπολογιών και είναι οι εξής:

- Star topology
- Peer to Peer topology

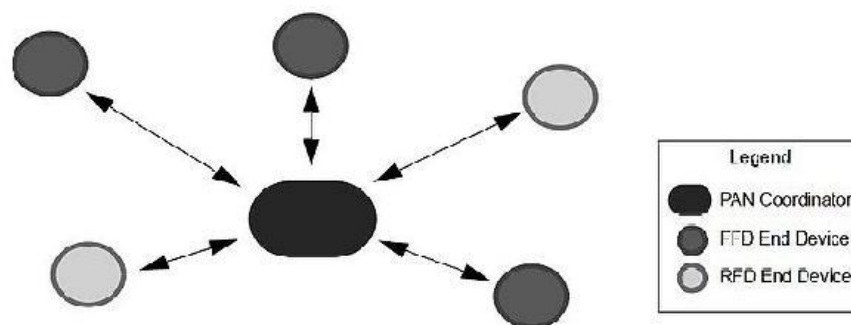
Στην συνέχεια του εδαφίου θα πραγματοποιηθεί ανάλυση των προαναφερόμενων.

▼ Star topology

Στην Star topology η επικοινωνία υλοποιείται από ένα πλήθος περιφερειακών συσκευών RFD/FFD και έναν PAN coordinator τύπου FFD. Ο PAN coordinator λειτουργεί ως σημείο αναφοράς του δικτύου και σε αυτόν καταλήγουν τα δεδομένα που έρχονται από τις περιφερειακές συσκευές. Μπορεί να χρησιμοποιηθεί για την αρχικοποίηση, την λήξη και την δρομολόγηση της επικοινωνίας στο δίκτυο και αποτελεί τον σταθμό βάσης του δικτύου PAN.

Όταν ένα FFD ενεργοποιηθεί για πρώτη φορά, έχει την δυνατότητα να δημιουργήσει το δικό του δίκτυο και να λειτουργεί ως PAN coordinator. Τα δίκτυα τύπου Star λειτουργούν ανεξάρτητα από τα υπόλοιπα δίκτυα του ίδιου τύπου που βρίσκονται στην ίδια περιοχή. Αυτό επιτυγχάνεται επιλέγοντας ένα PAN identifier, ο οποίος δεν χρησιμοποιείται από άλλο δίκτυο που έχει διαφορετική ασύρματη επικοινωνία.

Σημειώνεται ότι όταν επιλεγεί ο PAN identifier, ο PAN coordinator επιτρέπει σε άλλες συσκευές να προσαρτηθούν στο δικό του δίκτυο, είτε οι συσκευές RFD είτε FFD. Επιπλέον η Star topology χρησιμοποιείται για εφαρμογές αυτοματισμού και για τον έλεγχο περιφερικών υπολογιστών (PC).

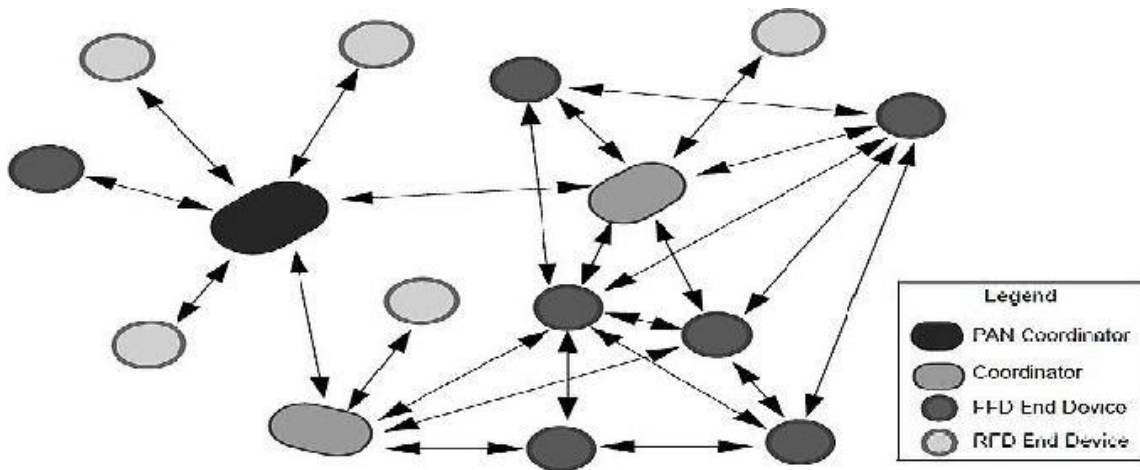


Σχήμα 2.1: Απεικόνιση Star Topology

(Μακρή Χ., 2011)

▼ Peer - to - Peer topology

Στην Peer - to - Peer topology υπάρχει ένας PAN coordinator, έχει την δυνατότητα να επικοινωνεί με οποιαδήποτε συσκευή βρίσκεται εντός εμβέλειας του. Επιπλέον, έχει την δυνατότητα να υλοποιήσει πολύπλοκες δομές δικτύου όπως μια mesh topology. Χρησιμοποιείται συνήθως σε εφαρμογές όπως είναι οι έξυπνες καλλιέργειες, ο βιομηχανίες έλεγχος ακόμα και σε ελέγχους ασφαλείας. Ένα δίκτυο Peer - to - Peer μπορεί να είναι ad-hoc, να οργανώνεται και να διορθώνεται μόνο του. Όλες οι συσκευές μπορούν να επικοινωνήσουν μεταξύ τους ακόμα κι αν βρίσκονται εκτός εμβέλειας, χρησιμοποιώντας ενδιάμεσες συσκευές για την δρομολόγηση των μηνυμάτων (multi hop λειτουργία).



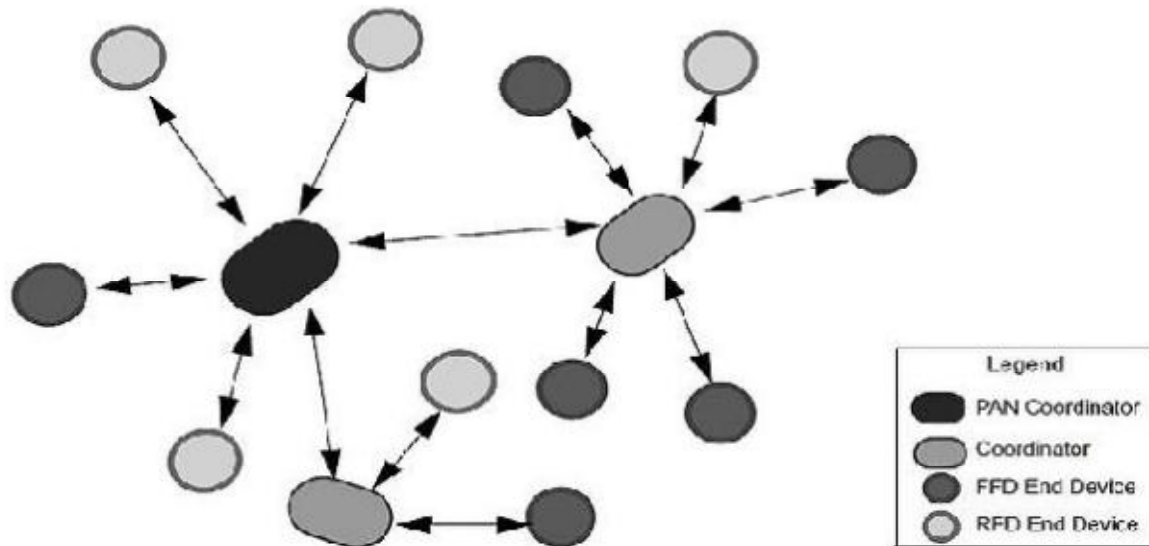
Σχήμα 2.2: Απεικόνιση Peer to Peer Topology

(Μακρή Χ., 2011)

Θα πρέπει να σημειωθεί ότι υπάρχει μια ειδική περίπτωση Peer - to - Peer δικτύου και είναι η τοπολογία cluster-tree όπως απεικονίζεται στο Σχήμα 2.3. Στο cluster-tree οι περισσότερες συσκευές είναι FFD. Μια RFD συσκευή μπορεί να συνδεθεί μόνο όταν βρίσκεται στο τέλος της διακλάδωσης του cluster γιατί έχει την δυνατότητα επικοινωνίας μόνο με μία συσκευή FFD την φορά.

Οποιαδήποτε FFD συσκευή μπορεί να λειτουργήσει σαν coordinator και να παρέχει συγχρονισμό σε άλλες συσκευές και σε άλλους coordinators. Μόνο ένας coordinator μπορεί να λειτουργήσει ως ολικός PAN coordinator του δικτύου, ο οποίος καταναλώνει τους περισσότερους υπολογιστικούς πόρους από κάθε άλλη συσκευή. Ο PAN coordinator

σχηματίζει την πρώτη ομάδα cluster, ο οποίος αποτελεί το cluster head CLH της ομάδας με το cluster identifier CID να έχει τιμή μηδέν (0). Επιλέγει έναν αχρησιμοποίητο PAN identifier και μεταδίδει ευρέως ακολουθίες δεδομένων, σε γειτονικές συσκευές.



Σχήμα 2.3: Τοπολογία cluster-tree

(Μακρή X., 2011)

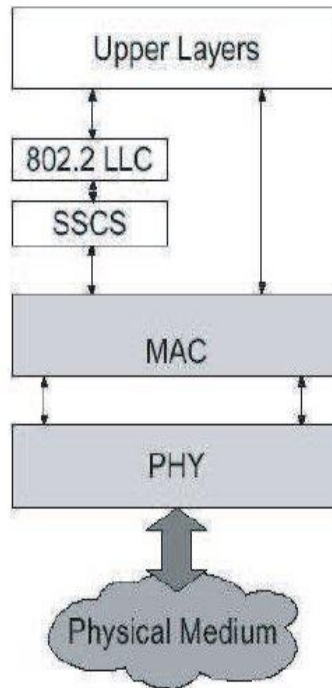
Όταν μία υποψία συσκευή προς σύνδεση λάβει μια ακολουθία δεδομένων, μπορεί να απαιτήσει να συνδεθεί στο δίκτυο μέσω του cluster head CLH. Αν ο PAN coordinator επιτρέψει την σύνδεση, θα προσθέσει την συσκευή ως child στη λίστα των γειτονικών του. Στην συνέχεια η συσκευή αυτή θα προσθέσει το CLH ως parent στην λίστα γειτονικών του και θα ξεκινήσει να μεταδίδει περιοδικές ακολουθίες.

Σε περίπτωση που η υποψία προς σύνδεση συσκευή δεν μπορεί να συνδεθεί σε ένα cluster δίκτυο μέσω του αντίστοιχου CLH, θα ψάξει για όλο CLH- parent που ανήκει σε άλλο cluster. Ο PAN coordination έχει την δυνατότητα να οδηγήσει μια συσκευή ώστε να αποτελεί ένα CLH ενός νέου cluster με συνέπεια την βαθμιαία εξάπλωση του δικτύου, αυτό απεικονίζεται και στο Σχήμα 2.3.

Ωστόσο, αυτό που θα πρέπει να τονιστεί για αυτό τον τύπο του δικτύου είναι το πλεονέκτημα του, το οποίο είναι η ευρεία κάλυψη μιας περιοχής, και το μειονέκτημα του, το οποίο είναι η μειωμένη ταχύτητα μετάδοσης των μηνυμάτων (Μανωλόπουλος X., 2010 ; Μακρή X., 2011).

2.3.ΑΡΧΙΤΕΚΤΟΝΙΚΗ WPAN

Η αρχιτεκτονική κάθε WPAN (Wireless Personal Network), στην κατηγορία των οποίων ανήκουν και τα WSNs, κατηγοριοποιείται σε μία σειρά από επίπεδα (layers), τα οποία διευκολύνουν τη μελέτη και το σχεδιασμό του δικτύου και προτυποποιούνται από μία σειρά πρωτοκόλλων. Η αρχιτεκτονική ενός WPAN δικτύου απεικονίζεται στο Σχήμα 2.4.



Σχήμα 2.4: Αρχιτεκτονική WPAN

[Πηγή: <http://mobliezigbee.blogspot.gr>]

Όπως προαναφέρθηκε ο WPAN αποτελείται από επίπεδα τα οποία κάθε ένα από αυτά είναι υπεύθυνο για ένα τμήμα λειτουργιών του προτύπου και παρέχει υπηρεσίες στα υψηλότερα επίπεδα. Τα επίπεδα είναι το κάτωθι:

- Φυσικό επίπεδο Physical Layer- PHY: περιέχει τον RF πομποδέκτη μαζί με κάποιους μηχανισμούς χαμηλού επιπέδου.
- Επίπεδο Media Access Control – MAC: παρέχει πρόσβαση στο φυσικό κανάλι για όλους τους τύπους μετάδοσης.

Η πρόσβαση του MAC επιπέδου από υψηλότερα επίπεδα πραγματοποιείται μέσω του interface Logical Link Control (LLC). Το LLC έχει την δυνατότητα να παρέχει μηχανισμούς πολυπλεξίας καθώς και ελεγχόμενη ροή δεδομένων. Επιπλέον, επιτρέπει την συνύπαρξη και συνεργασία ανάμεσα σε ευρέως διαδεδομένα πρωτόκολλα δικτύων όπως είναι:

- IP
- IPX
- Multipoint (δίκτυα πολλαπλών σημείων)

Τα υψηλότερα επίπεδα διαχωρίζονται με την σειρά τους σε δύο κατηγορίες, οι οποίες είναι οι ακόλουθες:

- Επίπεδο εφαρμογής: παρέχει την προτεινόμενη λειτουργία της συσκευής
- Επίπεδο δικτύου: παρέχει το configuration, τον χειρισμό του δικτύου και την προώθηση των μηνυμάτων (Μακρή Χ., 2011).

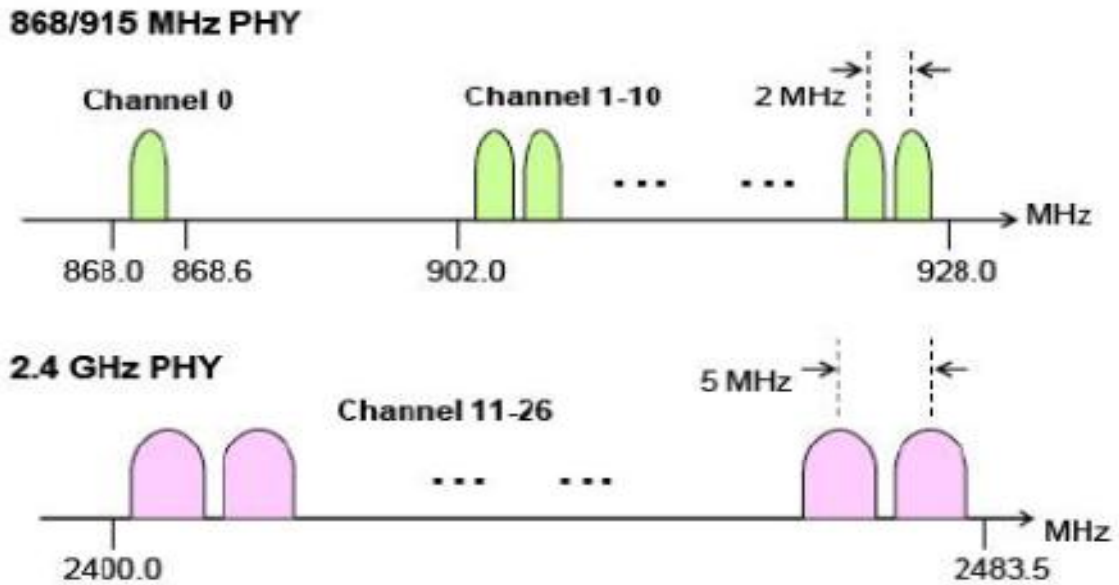
▼ *Physical Layer- PHY:*

Το PHY layer έχει πρόσβαση από το MAC layer μέσω του Interfaces Physical Layer Data Entity Service Access Point (PLDE – SAP) και Physical Layer Management Entity (PLME – SAP). Το Interfaces Physical Layer Data Entity – PLDE ενεργοποιεί την διαδικασία αποστολής και λήψης δεδομένων του συγκεκριμένου επιπέδου, μέσω φυσικού καναλιού. Ωστόσο, το Physical Layer Management Entity – PLME έχει να κάνει με την διαχείριση συχνοτήτων του PHY επιπέδου, για την διάδοση των μηνυμάτων.

Οι συχνότητες που χρησιμοποιούνται για την μετάδοση δεδομένων χωρίζονται σε τρία bands παρουσιάζονται κάτωθι:

- 868-868,6 MHz → Ευρώπη
- 902-928 MHz → Β. Αμερική
- 2400-2483,5 MHz → Worldwide

Στο Σχήμα 2.5. απεικονίζονται τα προαναφερόμενα κανάλια συχνοτήτων για το πρωτόκολλο 802.15.4.



Σχήμα 2.5: Τα κανάλια συχνοτήτων για το πρωτόκολλο 802.15.4
(Μακρή Χ., 2011)

▼ *Media Access Control Layer – MAC*

Για την πρόσβαση του MAC Layer από υψηλότερα επίπεδα χρησιμοποιούνται τα interfaces MAC Layer Data Entity – MLDE, όπου ενεργοποιεί την αποστολή και λήψη μηνυμάτων μέσω του RHY Layer, και MAC Layer Management Entity – MLME, όπου αλληλεπιδρά με την οντότητα διαχείρισης του MAC.

Ορισμένα από τα χαρακτηριστικά του MAC sublayer είναι τα εξής:

- Beacon management: για την διαχείριση του beacon frame για τον συγχρονισμό PAN
- Channel Access: αφορά την πρόσβαση καναλιού
- GTS Management: για την διαχείριση τμημάτων GTS
- Frame Validation: για την επιβεβαίωση των frames
- Acknowledged Frame Delivery: για την διανομή frames αναγνώρισης.

Με βάση τα προαναφερόμενα χαρακτηριστικά είναι προφανές ότι το MAC sublayer αποτελεί βασικό παράγοντα για την ασφάλεια των δεδομένων. Αφού μέσω αυτού μπορούν να υλοποιηθούν κάποιο μηχανικοί ασφαλείας (Μανωλόπουλος Χ., 2010).

2.4. PHY LAYER

Σε αυτό το εδάφιο θα πραγματοποιηθεί περιγραφή του τρόπου μεταφοράς δεδομένων από τον PAN coordinator και αντίστροφα. Θα γίνει χρήση του Ανιχνευτή Φέροντος Πολλαπλής Πρόσβασης με Αποφυγή Συγκρούσεων ή CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance). Επιπλέον, η χρήση ενός superframe για τον συγχρονισμό των συσκευών μπορεί να είναι προαιρετική, τονίζοντας ότι αν γίνει χρήση θα επιλεγεί slotted CSMA-CA εκδοχή ή αλλιώς θα επιλεγεί unslotted CSMA-CA.

Οι αρμοδιότητες του είναι οι ακόλουθες:

- Εκτέλεση ED
- Πραγματοποίηση του LQI
- Εκτέλεση του CCA
- Επιλογή ακριβούς συχνότητας καναλιού
- Αποστολή και λήψη δεδομένων

Στην συνέχεια θα πραγματοποιηθεί ανάλυση και περιγραφή των εξής σημείων του PHY layer:

- Πρωτόκολλο CSMA-CA
- Δομή Superframe
- Μοντέλο μεταφοράς δεδομένων

✓ *Πρωτόκολλο CSMA-CA*

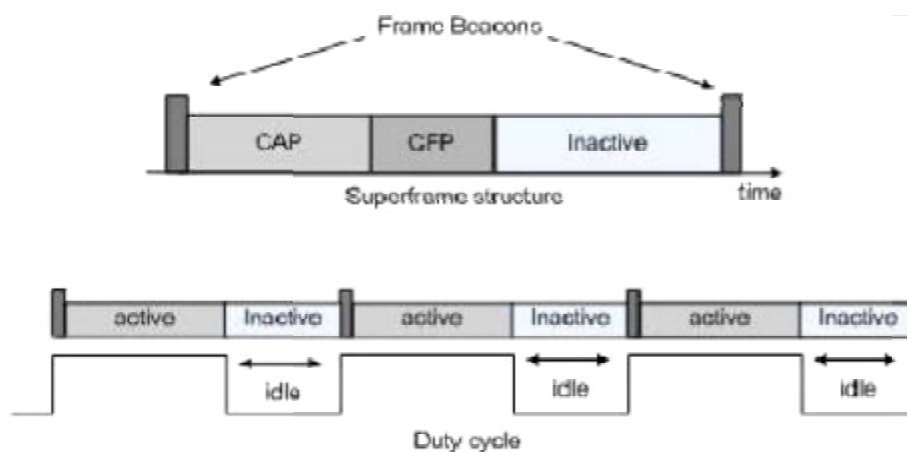
Στην slotted CSMA-CA εκδοχή, για την πραγματοποίηση του απαραίτητου συγχρονισμού γίνεται χρήση beacon frames καθώς και χρονικών υποδοχών slots. Κάτω από αυτές τις συνθήκες ο coordinator έχει την δυνατότητα να επικοινωνεί με τις άλλες συσκευές δίνοντας και το πλεονέκτημα της μειωμένης κατανάλωσης ενέργειας. Θα πρέπει να σημειωθεί ότι με τον προαναφερόμενο τρόπο αποφεύγεται σε μεγάλο ποσοστό οι συγκρούσεις. Επιπλέον, το slotted CSMA-CA θεωρείται κατάλληλο για τοπολογίες star όπου υπάρχει επικοινωνία ενός coordinator με τις περιφερειακές συσκευές.

Η unslotted CSMA-CA εκδοχή, είναι υπεύθυνη για την ανίχνευση του καναλιού πριν πραγματοποιηθεί με μετάδοση. Όταν μια συσκευή θέλει να μεταδώσει δεδομένα περιμένει

για ένα διάστημα back off περιόδων. Στην περίπτωση που είναι αδρανές τότε ξεκινά η μετάδοση. Κατά την διάρκεια της μετάδοσης δεν έχει την δυνατότητα να ανιχνεύσει το κανάλι με συνέπεια να στέλνει ολόκληρη την ακολουθία. Ωστόσο, λόγω των παρεμβολών μπορεί η ακολουθία να χαθεί. Στη περίπτωση που το κανάλι είναι απασχολημένο ή υπάρχει σύγκρουση πακέτων, ο αποστολέας αναβάλλει την μετάδοση και αναμένει ξανά για ένα διάστημα back off περιόδων, μέχρι το κανάλι να γίνει αδρανές (Μανωλόπουλος X., 2010 ; Σπένδας Α., 2012).

▼ Δομή Superframe

Η δομή ενός Superframe προσδιορίζεται από τον PAN coordinator και το εύρος του μπορεί να καθοριστεί από δύο Frame Beacons που είναι τοποθετημένα στην αρχή και στο τέλος του πλαισίου (Σχήμα 2.6).



Σχήμα 2.6: Απεικόνιση Δομής Superframe

(Μανωλόπουλος X., 2010).

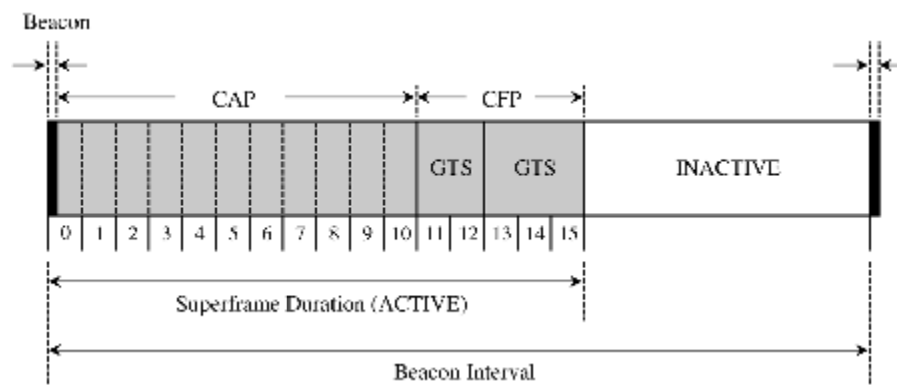
Το Frame Beacon μεταδίδεται στο πρώτο slot του κάθε πλαισίου. Χρησιμοποιείται για τον συγχρονισμό μεταξύ των συσκευών, για την ανάγνωση του PAN και την περιγραφή της δομής των Superframes.

Το superframe αποτελείται από ένα ενεργό τμήμα, το οποίο με την σειρά του αποτελείται από 16 slots του ίδιου μεγέθους και ένα τμήμα ανενεργό. Στην περιοχή του ανενεργού τμήματος ο coordinator μπορεί να μην αλληλεπιδρά με στο Pan δίκτυο με

συνέπεια να βρίσκεται σε αναμονή. Όσο αφορά το ενεργό τμήμα διαχωρίζεται με την σειρά του στα εξής τμήματα:

- Contention Access Period – CAP
- Contention Free Period – CFP

Στις εφαρμογές που απαιτείται είτε συγκεκριμένο εύρος ζώνης δεδομένων (data bandwidth) είτε μικρή καθυστέρηση κατά την μεταφορά των δεδομένων, ο PAN coordinator έχει την δυνατότητα να χρησιμοποιεί τμήματα (slots) από το CFP για την υλοποίηση των εφαρμογών αυτών. Τα τμήματα αυτά ονομάζονται guaranteed time slots - GTS. Ένα τμήμα GTS μπορεί να χρησιμοποιεί slots, περισσότερα του ενός. Επιπλέον, μέχρι και επτά (7) τμήματα GTS μπορούν να προσδιοριστούν από τον PAN coordinator. Στο Σχήμα 2.7 απεικονίζεται ένα superframe που το CFP έχει χωριστεί σε τρία GTS διαφορετικού μεγέθους το καθένα (Μανωλόπουλος X., 2010).



Σχήμα 2.7: Τμήματα GTS στο Contention Free Period – CFP του Superframe

[Πηγή: <http://usn-pioneer.tistory.com>]

✓ Μοντέλο μεταφοράς δεδομένων

Για την μεταφορά δεδομένων ανάμεσα σε δύο συσκευές υπάρχουν τρεις περιπτώσεις και είναι οι κάτωθι:

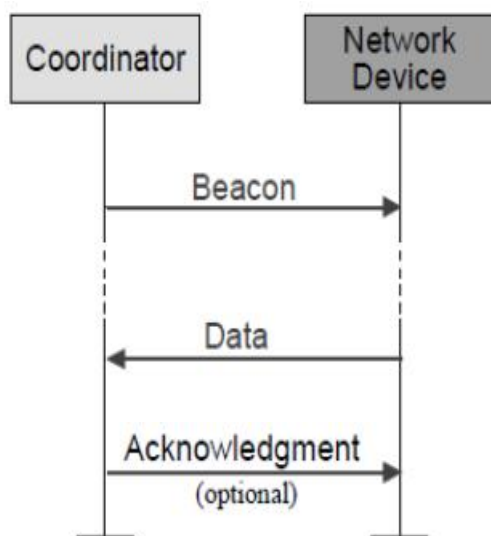
- Μεταφορά δεδομένων από συσκευή σε coordinator
- Μεταφορά δεδομένων από coordinator σε συσκευή
- Μεταφορά δεδομένων από δύο συσκευές peer- to - peer

Σε κάθε μία χρησιμοποιείται είτε η slotted είτε η unslotted CSMA-CA εκδοχή, ανάλογα με την χρήση ή όχι του superframe. Στην συνέχεια ακολουθεί η ανάλυση των προαναφερόμενων περιπτώσεων.

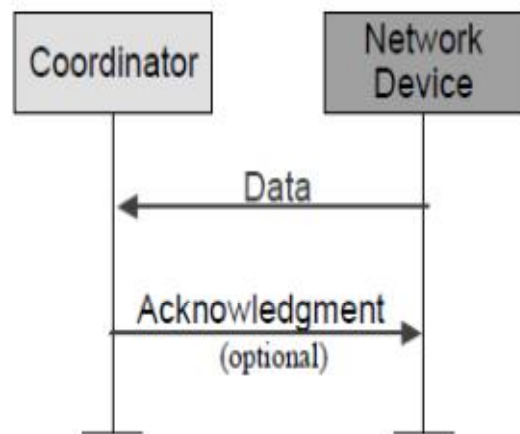
∅ *Περίπτωση 1^η : Μεταφορά δεδομένων από συσκευής σε coordinator*

Στην περίπτωση που μία συσκευή επιθυμεί να μεταφέρει δεδομένα στον coordinator κάνοντας χρήση superframe, θα πρέπει να το εντοπίσει και έπειτα να στο συγχρονίσει με αυτό. Θα πρέπει να επιλέγει η «σωστή» στιγμή όπου η συσκευή μεταδίδει την ακολουθία χρησιμοποιώντας τον slotted CSMA-CA μηχανισμό - στον coordinator, ο οποίος με την σειρά του στέλνει ένα μήνυμα αναγνώρισης (τύπου acknowledgement) ως ένδειξη επιτυχούς μεταφοράς (Σχήμα 2.8).

Σε περίπτωση που γίνεται χρήση του superframe, μπορεί να πραγματοποιηθεί απευθείας μετάδοσης μεταφοράς δεδομένων μέσω του unslotted CSMA-CA μηχανισμού, από την συσκευή προς τον coordinator, έπειτα ακολουθεί ένα μήνυμα αναγνώρισης (Σχήμα 2.9) (Σπένδας Α., 2012)



Σχήμα 2.8. Χρήση Slotted-CSMA μηχανισμού
[Πηγή: <http://www.slideshare.net>]



Σχήμα 2.9. Χρήση Unslotted-CSMA μηχανισμού
[Πηγή: <http://www.slideshare.net>]

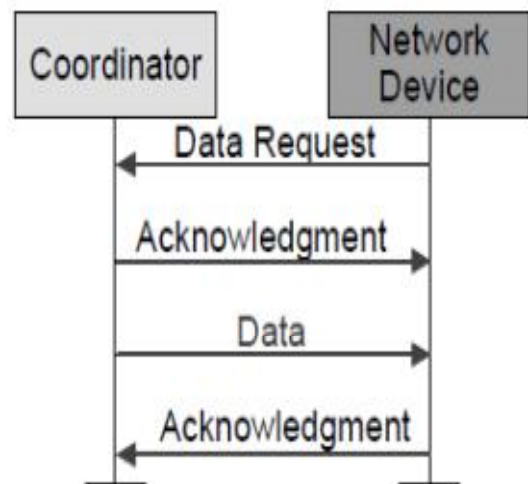
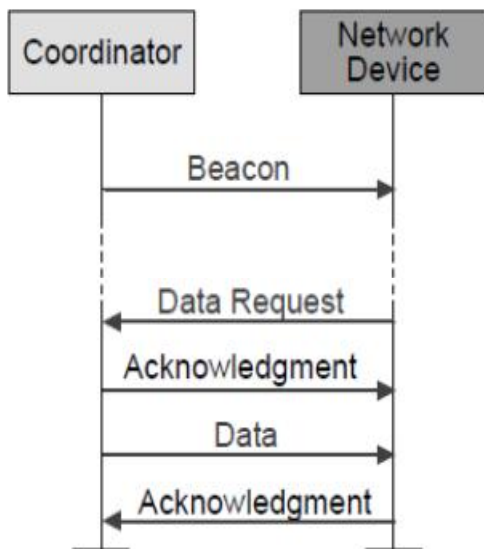
∅ *Περίπτωση 2^η : Μεταφορά δεδομένων από coordinator σε συσκευή*

Όταν ο coordinator επιθυμεί να στείλει δεδομένα προς μια συσκευή δηλώνει μέσω του superframe, ότι υπάρχει ένα μήνυμα σε αναμονή. Στην συνέχεια η συσκευή εντοπίζει και

διαβάζει το superframe, στέλνει request (MAC command) για λήψη δεδομένων από coordinator. Ο coordinator στέλνει ένα frame αναγνώρισης ως ανασκόπηση μαζί με την ακολουθία δεδομένων προς την συσκευή. Όταν πραγματοποιηθεί η διαδικασία μεταφοράς, η συσκευή στέλνει μήνυμα αναγνώρισης προς τον coordinator για την επιτυχή διεξαγωγή της διαδικασίας, ο οποίος στην συνέχεια διαγράφει από την λίστα αναμονής μηνυμάτων την ακολουθία (Σχήμα 2.10)

Στην περίπτωση που ο coordinator επιθυμεί να στείλει δεδομένα χωρίς να γίνει χρήση του superframes, τότε τα δεδομένα αποθηκεύονται με στόχο να γίνει αμέσως μετά η μετάδοση. Η συσκευή έχει την δυνατότητα να στέλνει από μόνη της data request (MAC command) μέσω του unslotted CSMA-CA μηχανισμού στον coordinator, ο οποίος έπειτα στέλνει ένα frame αναγνώρισης (acknowledgment) ως ανταπόκριση μαζί με την ακολουθία δεδομένων.

Στην περίπτωση που υπάρχει κάποιο μήνυμα σε αναμονή τότε μεταδίδεται προς την συσκευή μέσω του ίδιου μηχανισμού. Αν υπάρχει διαφορετική περίπτωση η ακολουθία των δεδομένων είναι μηδέν (0), δείχνοντας με αυτόν τον τρόπο ότι δεν υπάρχουν δεδομένα. Αφού ολοκληρωθεί η διαδικασία μεταφοράς η συσκευή με την σειρά της στέλνει μήνυμα αναγνώρισης προ του coordinator για την επιτυχή διεξαγωγή της διαδικασίας. Ο τρόπος ο οποίος πραγματοποιείται η διαδικασία απεικονίζεται στο Σχήμα 2.11.



Σχήμα 2.10: Χρήση Slotted-CSMA μηχανισμού

[Πηγή: <http://www.slideshare.net>]

Σχήμα 2.11: Χρήση Unslotted-CSMA μηχανισμού

[Πηγή: <http://www.slideshare.net>]

Ø Περίπτωση 3^η: Μεταφορά δεδομένων από δύο συσκευές peer- to - peer

Σε δίκτυο PAN peer-to-peer, κάθε συσκευή έχει την δυνατότητα να επικοινωνήσει με οποιαδήποτε άλλη, με την προϋπόθεση όμως να βρίσκονται στο ίδιο δίκτυο. Επιπλέον, κάθε μία από αυτές τις συσκευές λειτουργεί είτε ως παραλήπτης είτε ως αποστολέας χρησιμοποιώντας την unslotted CSMA-CA εκδοχή για την μεταφορά δεδομένων (Μανωλόπουλος Χ., 2010).

2.5. MAC LAYER

Το επίπεδο έλεγχου πρόσβασης μέσω εξασφαλίζει την διασύνδεση των ανώτερων επιπέδων με το φυσικό. Είναι το επίπεδο που ελέγχει άμεσα το PHY. Οι αρμοδιότητές του MAC Layer είναι:

- Παραγωγή των beacons
- Συγχρονισμός των συσκευών στο εισερχόμενο beacon
- Να επιτρέπει την σύνδεση και την αποσύνδεση μεταξύ των συσκευών στα ZigBee δίκτυα
- Να υποστηρίζει τις παραμέτρους ασφαλείας του πρωτοκόλλου
- Να χρησιμοποιεί τη CSMA-CA για να επιτρέψει την πρόσβαση στο κανάλι
- Παραχώρηση των GTS (Σπένδας Α., 2012)

Για την κατανόηση της λειτουργίας θα αναλυθούν οι κατασκευές πλαισίου που συνήθως αναφέρονται ως frame structures. Τα Frame structures έχουν σχεδιαστεί με σκοπό την ασύρματη μετάδοση δεδομένων, όπου επικρατούν συνθήκες με μεγάλο ποσοστό θορύβου. Η μετάδοση αυτή πραγματοποιείται με όσο τον δυνατόν λιγότερη πολυπλοκότητα. Το δίκτυο WPAN ορίζεται από τα εξής είδη Frame structures:

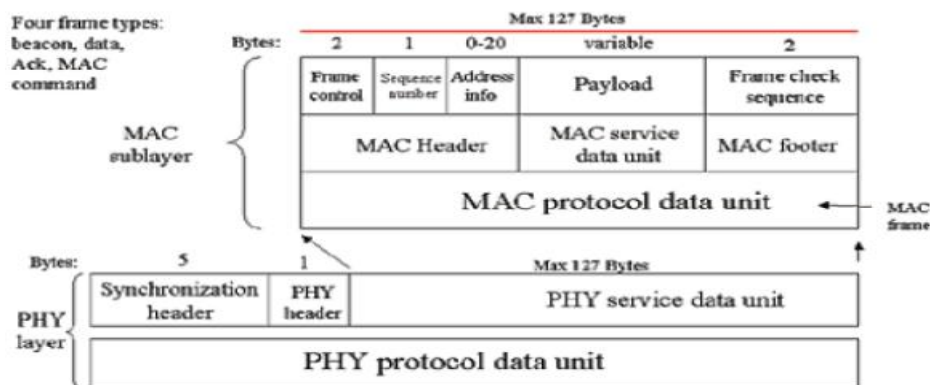
- Beacon Frame: για τον συγχρονισμό PAN μέσω του coordinator
- Data Frame: για την μεταφορά δεδομένων
- Acknowledgement Frame: για την επιβεβαίωση επιτυχούς λήψης
- MAC Command Frame: για την λήψη δεδομένων από τον coordinator (Dvorak J., 2005)



Εικόνα 2.1: Τα είδη Frame structures

[Πηγή: <http://www.slideshare.net>]

Η δομή των προαναφερόμενων Frame Structures αναλυτικότερα δίνεται στην Εικόνα 2.2. Το Synchronization Header –SHR χρησιμοποιείται με σκοπό να πραγματοποιείται ο συγχρονισμός μεταξύ του πομπού και του δέκτη. Το PHY Header – PHR ορίζει το μήκος του MAC protocol data unit – MPDU. Το μέγεθος MPDU είναι διαφορετικό ανάλογα με τον τύπο του Frame structures που χρησιμοποιείται.



Εικόνα 2.1: Δομή ενός frame structure

(Μανωλόπουλος X., 2010)

Στην συνέχεια του εδαφίου θα πραγματοποιηθεί ανάλυση σύμφωνα με τον Μανωλόπουλο X. (2012) των έξι σημείων, με σκοπό την καταγραφή των χαρακτηριστικών και την λειτουργία καθενός ξεχωριστά:

- Synchronization Header –SHR
- PHY Header – PHR
- MAC protocol data unit – MPDU

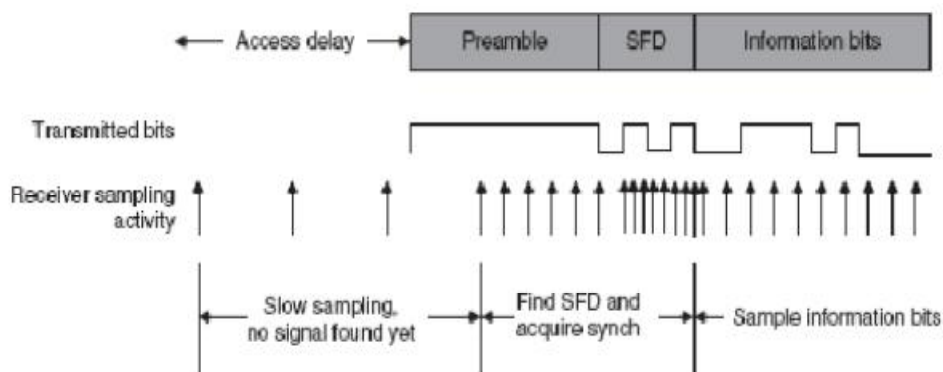
Ø Synchronization Header –SHR

Το Synchronization Header –SHR αποτελείται από τα εξής:

- Preamble sequence → 4 Bytes
- SFD – Start of frame delimiter → 1 Bytes

Θα πρέπει να σημειωθεί ότι επειδή η δημιουργία της ακολουθίας των bits που έχει την δυνατότητα να μεταδοση ένας πομποδέκτης στηρίζεται στην χρήση ενός εσωτερικού ταλαντωτή τις περισσότερες φορές λόγω αποκλίσεων της ονομαστικής συχνότητας δημιουργείται σφάλμα στον συγχρονισμό πομπού και δέκτη με συνέπεια η αποκωδικοποίηση αποτυγχάνει.

Για τον προαναφερόμενο λόγο σε κάθε frame structure προστίθεται μια δοκιμαστική ακολουθία, που θα πρέπει να επιτρέπει στο δέκτη να αναγνωρίσει τις παραμέτρους μετάδοσης του πομπού και να συγχρονιστεί (Σχήμα 2.12). Η ακολουθία αυτή βρίσκεται στην αρχή του frame και ακολουθεί ο χαρακτήρας έναρξης του frame SDF.



Σχήμα 2.12: Δομή Synchronization Header –SHR

[Πηγή: <http://www.slideshare.net>]

Ø PHY Header – PHR

Το PHY Header – PHR αποτελείται από το Length Field. Ορίζει τον αριθμό των bytes που αποτελούν το MPDU (MAC protocol data unit) χωρίς όμως να προσμετρά τον εαυτό του. Επιπλέον, στην τιμή αυτή περιλαμβάνει και το FCS (frame check sequence) παρόλο που παράγεται από το hardware. Έχει μήκος 8 Bits, το LSB είναι προκαθορισμένο να έχει την τιμή μηδέν (0), ενώ η μέγιστη τιμή που παίρνει είναι 127. Το Length Field χρησιμοποιείται

αμφότερα στην εκπομπή και λήψη δεδομένων, οπότε υπάρχει πάντοτε σε μία ακολουθία (Μανωλόπουλος Χ., 2010).

Ø *MAC Protocol Data Unit – MPDU*

Το MAC Protocol Data Unit - MPDU αποτελείται από τα εξής:

1. MAC Header (MHR)
2. MAC Service Data Unit (MSDU)
3. MAC Footer (MFR)

Ανάλογα τον τύπο του frame structure, το MSDU έχει διαφορετικό μήκος. Στην συνέχεια θα πραγματοποιηθεί ανάλυση των προαναφερόμενων.

1. *MAC Header (MHR)*

Το MAC Header (MHR) αποτελείται από τα εξής:

- Το Frame Control έχει μέγεθος 2 bytes, το οποίο είναι υπεύθυνο για τον καθορισμό του frame, το format των διευθύνσεων και ελέγχει το acknowledgement.
- Το Sequence Number θα πρέπει να είναι κατάλληλο για το acknowledgement frame σύμφωνα με την προηγούμενη μετάδοση.
- Το Address Field έχει την δυνατότητα να παρουσιάζει μεταβλητό μέγεθος. Όταν το frame structure είναι beacon frame, στο address field περιέχεται μόνο η διεύθυνση της πηγής. Όταν το frame structure είναι data frame, περιέχεται η διεύθυνση της πηγής μαζί με την διεύθυνση προορισμού, ενώ όταν είναι τύπου acknowledgement το address field είναι κενό.

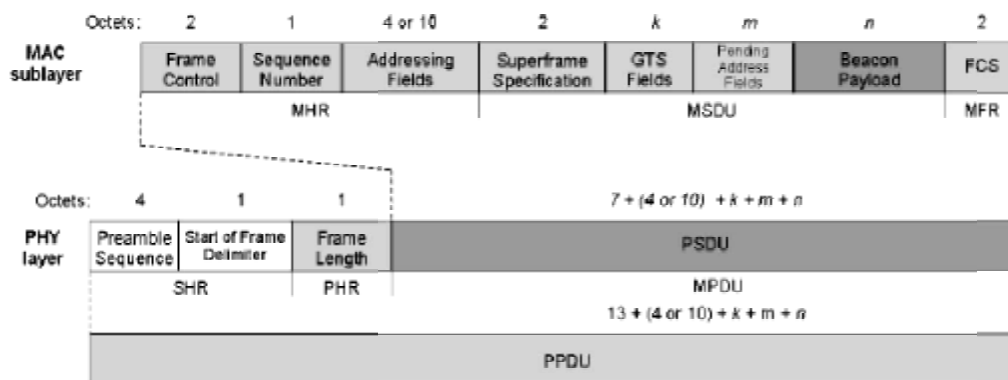
2. *MAC Service Data Unit (MSDU)*

Το μήκος MAC Service Data Unit – MSDU είναι διαφορετικό για κάθε ένα από τα τέσσερα διαφορετικά είδη frame structures που χρησιμοποιούνται. Στην συνέχεια παρουσιάζεται κάθε ένα από τα αυτά με σκοπό να κατανοηθούν οι διαφορές.

a) Beacon Frame

Το MSDU του Beacon Frame αποτελείται από τα ακόλουθα:

- Το Superframe specification έχει μέγεθος 2 Bytes και ορίζει τις προδιαγραφές του superframe
- Το GTS Fields έχει μέγεθος k Bytes και ορίζει το πλήθος GTS τμημάτων για την υλοποίηση εφαρμογών με συγκεκριμένο bandwidth.
- Το Pending Address Field έχει μέγεθος m Bytes και ορίζει το πλήθος των διευθύνσεων που βρίσκονται στο address field του MHR
- Το Beacon Payload έχει μέγεθος n Bytes και περιέχει δεδομένα σχετικά με το beacon frame.

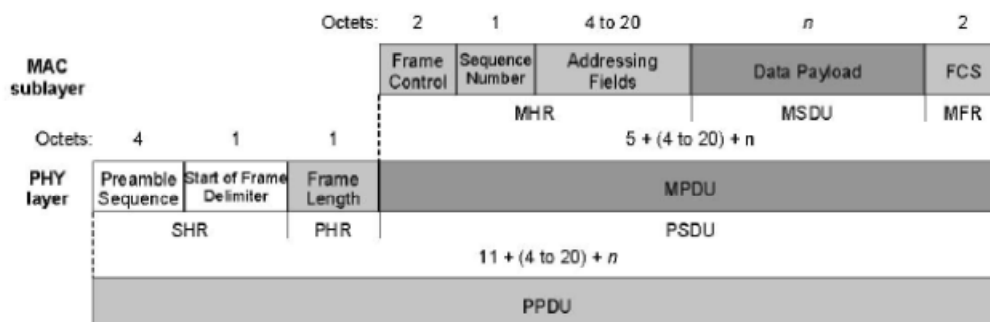


Σχήμα 2.13: Δομή του Beacon frame

(Σπένδας Α., 2012)

b) Data Frame

Το data frame αποτελείται μόνο από το data payload το οποίο είναι μεγέθους n Bytes. Όταν προέρχονται από μία συσκευή, το data payload περιέχει δεδομένα συλλογής μετρήσεων από ένα σύνολο αισθητήρων, ενώ αν προέρχονται από coordinator το data payload περιέχει δεδομένα που αφορούν τον προγραμματισμό της συσκευής (OTAP – On the air programming).

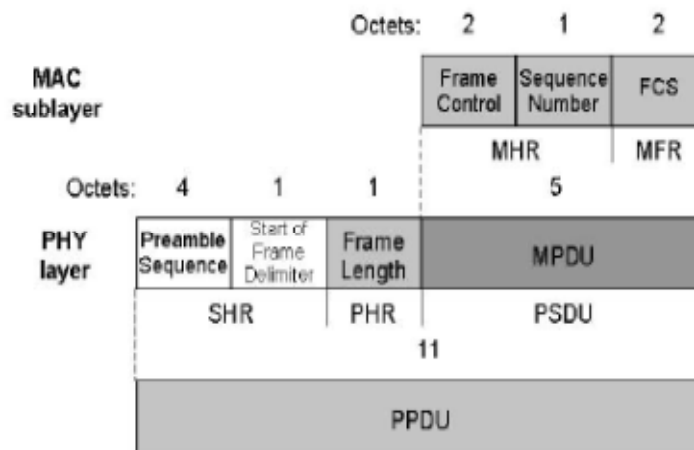


Σχήμα 2.14: Δομή του Data frame

(Σπένδας Α., 2012)

c) Acknowledgement Frame

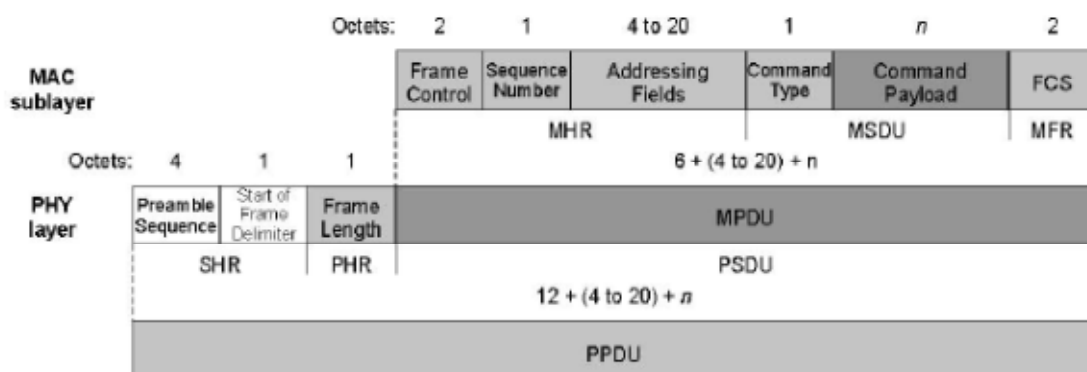
Όταν είναι τύπου acknowledgement, στο τμήμα MHR δεν περιέχεται το πεδίο addressing fields. Επιπλέον, το MSDU είναι και αυτό απόν διότι το συγκεκριμένο frame structure δεν χρησιμοποιείται για την αποστολή κάποιων δεδομένων, αλλά ως σήμα ανταπόκρισης ανάμεσα σε μια συσκευή και έναν coordinator



Σχήμα 2.14: Δομή του Acknowledgement frame
(Σπένδας Α., 2012)

d) Command Frame

Στο command frame το MHR παραμένει ίδιο. Το MSDU αποτελείται από το Command Type μήκους 1 Byte όπου δίνεται η δυνατότητα να προσδιοριστεί η MAC εντολή και το Command Payload το οποίο έχει μέγεθος n Bytes, βρίσκεται η MAC εντολή.



Σχήμα 2.15: Δομή του Command frame
(Σπένδας Α., 2012)

Command frame identifier	Command name
0x01	Association request
0x02	Association response
0x03	Disassociation Notification
0x04	Data request
0x05	PAN ID Conflict Notification
0x06	Orphan Notification
0x07	Beacon request
0x08	Coordinator Realignment
0x09	GTS request
0x0 _a -0xff	Reserved

Πίνακας 2.1. Εντολές του Command frame
(Μανωλόπουλος X., 2010)

3. MAC Footer (MFR)

Το MAC Footer (MFR) έχει μήκος 2 Bytes όπου υπάρχει το Frame Check Sequence FCS. Στέλνεται από τον πομπό μαζί με την υπόλοιπη ακολουθία και χρησιμοποιείται με σκοπό να επαληθευση το μήνυμα από τον δέκτη. Το FCS υπολογίζεται από τον δέκτη με τη μέθοδο του κυκλικού πολυωνυμικού κώδικα, που χρησιμοποιεί ως είσοδο τα δεδομένα που υπάρχουν στο MPDU χωρίς όμως να προσμετρά τον εαυτό του. Ο πομπός και ο δέκτης πέφτει να συμφωνήσουν πάνω σε ένα πολυώνυμο γεννήτορα εκ των προτέρων.

2.6.ΑΣΦΑΛΕΙΑ

Το IEEE 802.15,4 παρουσιάζει μεγάλο εύρος εφαρμογών, αυτό παρουσιάζει την ανάγκη να υπάρχουν μηχανισμοί για να παρέχονται υπηρεσίες ασφάλειας ανάμεσα στις συσκευές. Οι μηχανισμοί αυτοί που έχουν δημιουργηθεί έχουν την δυνατότητα να υποστηρίζουν την πρόσβαση σε μία λίστα ελέγχου και να χρησιμοποιούν συμμετρική κρυπτογράφηση για την προστασία των μεταδιδόμενων frames. Ωστόσο, θα πρέπει να σημειωθεί ότι οι υπηρεσίες ασφαλείας δεν έχουν την δυνατότητα να χρησιμοποιούνται σε οποιαδήποτε χρονική στιγμή με οποιαδήποτε συσκευή.

Τα υψηλότερα επίπεδα του πρωτοκόλλου καθορίζουν πότε οι υπηρεσίες ασφαλείας θα χρησιμοποιηθούν στο MAC sublayer και παρέχουν όλους τους απαραίτητους κώδικες για την ενεργοποίηση των μηχανισμών ασφαλείας.

2.6.1. Υπηρεσίες ασφαλείας

Οι υπηρεσίες ασφαλείας που χρησιμοποιούνται σε ένα τέτοιο σύστημα είναι οι ακόλουθες:

- **Access Control:** Υπηρεσία όπου παρέχει την δυνατότητα μια συσκευή να επιλέξει άλλες συσκευές που πρόκειται να συνδεθεί. Στην περίπτωση ενεργοποίησης του η συσκευή αποθηκεύεται στην λίστα ACL τις αντίστοιχες συσκευές προέλευσης των εισερχόμενων frames.
- **Data Encryption:** η υπηρεσίας χρησιμοποιείται έναν συμμετρικό κρυπτογράφημα για την προστασία δεδομένων, με στόχο να μην δίνεται η δυνατότητα ανάγνωσης από άλλες συσκευές. Με την χρήση ενός κώδικα – κλειδί κρυπτογραφούνται τα δεδομένα. Η κρυπτογράφηση δεδομένων χρησιμοποιείται στα αντίστοιχα payloads των beacon, command και data frame structure.
- **Frame Integrity:** στην υπηρεσία αυτή χρησιμοποιείται ένα μήνυα – κώδικας ακεραιότητας για την προστασία των δεδομένων από διαφορετικές ομάδες συσκευών που δεν έχουν κώδικα – κλειδί. Επιπλέον, διαβεβαιώνει ότι τα δεδομένα προέρχονται από μια ομάδα με τον ίδιο κώδικα – κλειδί. Η υπηρεσία ακεραιότητας των frames χρησιμοποιείται σε data frames, beacon frames, command frames. Ο κώδικας χρησιμοποιείται για την ακεραιότητα των frames είτε αφορά ομάδα συσκευών είτε δύο Peer συσκευές.
- **Sequential Freshness:** υπηρεσία όπου χρησιμοποιείται μια συγκεκριμένη φόρμα ακολουθίας ως είσοδος με σκοπό την απόρριψη επαναλαμβανόμενων frames. Κάθε frame έχει έναν αύξοντα αριθμό, όπου συγκρίνεται κάθε φορά με τον προηγούμενο αριθμό που έχει παραλάβει μία συσκευή. Το frame ανανεώνεται με συνέπεια εξασφαλίζει ότι τα δεδομένα είναι τα πιο πρόσφατα, όμως δεν παρέχεται ο ακριβής χρόνος ανανέωσης.

2.6.2. Security modes

Ανάλογα με τον τρόπο που έχει ρυθμιστεί να λειτουργεί MAC sublayer μπορεί να παρέχει διαφορετικά είδη ασφάλειας και είναι τα ακόλουθα:

- Secured mode: μπορούν να εκτελεστούν όλες οι υπηρεσίες που περιγράφηκαν πιο πάνω, ανάλογα με το επίπεδο ασφαλείας που χρειάζεται μια εφαρμογή
- Unsecured mode: υπάρχει κίνδυνος κλοπής ή παραποίησης των δεδομένων, αφού όλες οι υπηρεσίες παραμένουν ανενεργές.
- ACL mode: παρέχει συγκεκριμένες υπηρεσίες για επικοινωνία με άλλες συσκευές. Τα υψηλότερα επίπεδα μπορούν να απορρίψουν κάποια frames, όταν το MAC sublayer εντοπίσει ότι κάποιο frame διαδίδεται από μια συγκεκριμένη συσκευή. Η μόνη υπηρεσία ασφαλείας που υποστηρίζεται στο ACL mode είναι η access control (Μανωλόπουλος Χ., 2010).

2.7.ZIGBEE

Το πρωτόκολλο ZigBee έχει επικρατήσει για τα χαμηλού ρυθμού μετάδοσης τοπικά ασύρματα δίκτυα (LR-WPAN). Δημιουργήθηκε από συμμαχία 200 εταιριών (ZigBee Alliance) με στόχο την παγκόσμια προτυποποίηση σε τεχνολογίες ασύρματης δικτύωσης. Στα χαμηλότερα επίπεδα του ανοιχτού μοντέλου επικοινωνιών OSI (φυσικό επίπεδο – επίπεδο γραμμής) το πρωτόκολλο ZigBee υλοποιεί το πρότυπο 802.15.4, ο σχεδιασμός του οποίου εστιάζει κυρίως στην ελαχιστοποίηση της κατανάλωσης ενέργειας και στην αξιόπιστη μετάδοση πληροφορίας.



Εικόνα 2.3: Λογότυπο ZigBee Alliance.

[Πηγή: <http://www.regent-elec.com>]

Το πρότυπο 802.15.4 υλοποιείται κυρίως σε 3 ζώνες συχνοτήτων με ένα ή περισσότερα κανάλια μετάδοσης. Στη φασματική περιοχή των 868 – 869.8 MHz με ένα κανάλι μετάδοσης και ρυθμούς μέχρι 20 Kbps, στη ζώνη 905 – 928 MHz χωριζόμενη σε 10 κανάλια και ρυθμούς μετάδοσης μέχρι 40 Kbps και στη ζώνη 2.4 – 2.4855 GHz, όπου επιτυγχάνονται ρυθμοί μετάδοσης έως και 250 Kbps. Συνήθως, η περιοχή κάλυψης φτάνει τα 75 m και μπορεί να αυξηθεί εις βάρος του ρυθμού μετάδοσης δεδομένων.

Επιπλέον, στα πλαίσια του επιπέδου γραμμής, το πρότυπο καθορίζει το επίπεδο ελέγχου πρόσβασης στο μέσο (MAC), που χρησιμοποιεί πρόσβαση καναλιού CSMA – CA (Carrier Sense Multiple Access with Collision Avoidance – Πολλαπλή πρόσβαση στο μέσο με αποφυγή συγκρούσεων). Σύμφωνα με αυτή, κάθε κόμβος του δικτύου αρχικά ελέγχει το μέσο για να εντοπίσει αν κάποιος άλλος κόμβος στέλνει δεδομένα. Αν αυτό συμβαίνει, τότε ο κόμβος περιμένει για λίγο χρόνο πριν προχωρήσει σε νέο έλεγχο του μέσου.

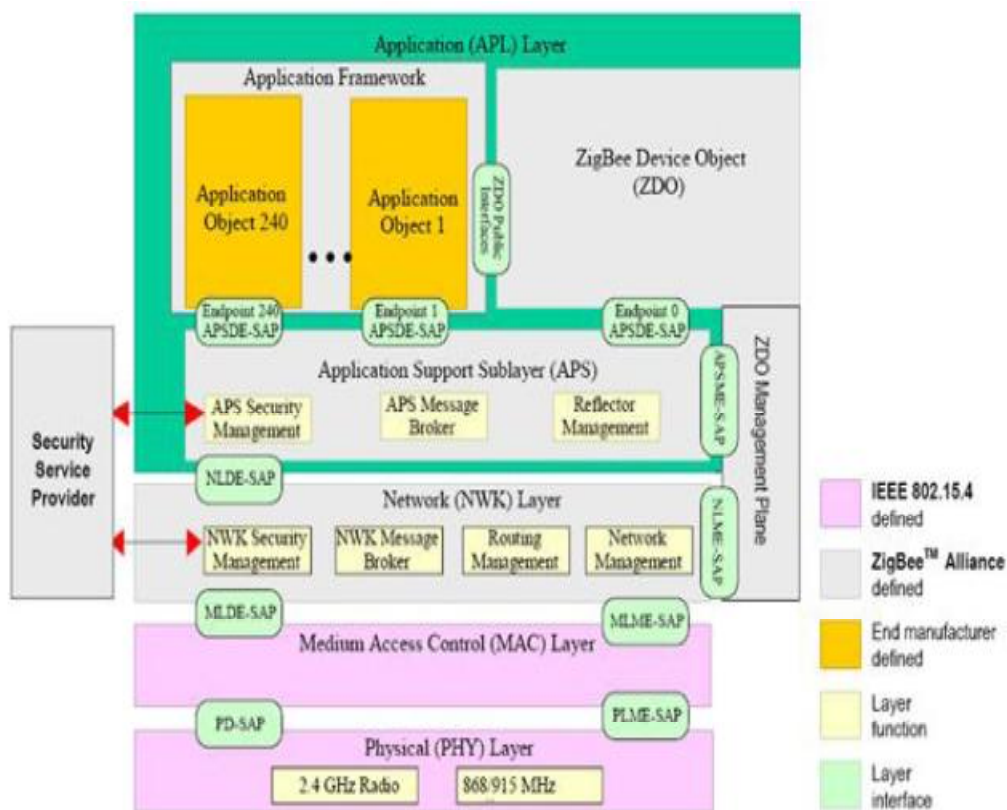
Όταν διαπιστωθεί ότι κανένας άλλος κόμβος δε στέλνει δεδομένα, τότε ο κόμβος έχει το ελεύθερο να ξεκινήσει την αποστολή των δεδομένων του. Μετά την ολοκλήρωση της αποστολής, ο κόμβος περιμένει μήνυμα επιβεβαίωσης (acknowledgement packet) από τη βάση συλλογής δεδομένων για την επιτυχή λήψη του μηνύματος δεδομένων. Σε περίπτωση που δεν πραγματοποιηθεί η επιβεβαίωση, ο κόμβος υποθέτει ότι το μήνυμα δεδομένων που έστειλε συγκρούστηκε με μηνύματα δεδομένων απεσταλμένα από άλλους κόμβους του δικτύου και προχωρά σε διαδικασία επανάληψης της μετάδοσης του πακέτου.

Εκτός από το φυσικό επίπεδο και το επίπεδο γραμμής, το πρωτόκολλο ZigBee προσθέτει δύο επιπλέον επίπεδα, το επίπεδο δικτύου (ZigBee Network Layer) και το επίπεδο εφαρμογής (ZigBee Application Layer). Χρησιμοποιεί τοπολογία πλέγματος, τεχνολογία δυναμικής δρομολόγησης και επιβεβαίωσης λήψης πακέτου σε διάφορα επίπεδα, με αποτέλεσμα να μπορεί να σχηματίσει αξιόπιστα, ευέλικτα δίκτυα με καλή ανοχή σε σφάλματα και εύκολα επεκτάσιμα (Κατσαούνης Γ., 2013).

Ωστόσο, θα πρέπει να τονιστούν τα πλεονεκτήματα που παρουσιάζει ZigBee, τα οποία είναι και τα βασικά χαρακτηριστικά του και είναι τα ακόλουθα:

- Χαμηλός κύκλος καθήκοντων - παρέχει μακριά ζωή μπαταριών
- Χαμηλή λανθάνουσα κατάσταση
- Υποστηρίζει πολλές τοπολογίες δικτύων: Στατικός, δυναμικός, αστέρι και πλέγμα (Static, dynamic, star and mesh)
- Άμεσο απλωμένο φάσμα ακολουθίας (DSSS) Direct Sequence Spread Spectrum

- Μέχρι 65.000 κόμβοι σε ένα δίκτυο
- 128-bit AES encryption κρυπτογράφηση - παρέχει ασφαλείς συνδέσεις μεταξύ των συσκευών
- Αποφυγή σύγκρουσεων
- Ποιοτική ένδειξη συνδέσεων
- Σαφής αξιολόγηση των καναλιών (Asvest T., 2014)



Εικόνα 2.4: Αρχιτεκτονική πρωτοκόλλου ZigBee
(Κατσαούνης Γ., 2013)

Οι συσκευές του πρωτοκόλλου ZigBee χωρίζονται στις επόμενες κατηγορίες.

▼ Συντονιστής (ZigBee Coordinator)

Ο συντονιστής είναι η συσκευή που είναι υπεύθυνη για το σχηματισμό και τη διαχείριση του δικτύου. Υπάρχει πάντα μόνο ένας συντονιστής σε κάθε δίκτυο. Απαιτεί συνεχή τροφοδοσία.

▼ *Δρομολογητής (ZigBee Router ή Full Function Device)*

Οι δρομολογητές μπορούν να προωθούν μηνύματα από τις υπόλοιπες συσκευές στους αντίστοιχους προορισμούς τους. Παράλληλα μπορούν να εκτελούν την εκάστοτε εφαρμογή που τρέχει στο δίκτυο. Απαιτούν συνεχή τροφοδοσία.

▼ *Τελική Συσκευή (ZigBee End Device ή Reduced Function Device)*

Οι τελικές συσκευές συνδέονται απαραίτητα σε ένα γονέα, είτε δρομολογητή είτε τον συντονιστή. Δεν προωθούν μηνύματα, αλλά μπορούν να βρίσκονται για αρκετό χρόνο στην αναμονή, εξοικονομώντας ενέργεια (Κατσαούνης Γ., 2013).

2.7.1. Επίπεδο δικτύου

Το επίπεδο δικτύου αφορά την διαχείριση της multi-hop επικοινωνίας και γεφυρώνει τα δύο πρωτόκολλα μέσω των interfaces NLDE-SAP και NLME-SAP. Το interface NLDE-SAP (Network Layer Data Entity – Service Access Point) δημιουργεί τα κατάλληλα πακέτα βάσει των δεδομένων που λαμβάνει από το ανώτερο επίπεδο και αποφασίζει για την σωστή δρομολόγηση τους στο δίκτυο. Το interface NLME - SAP (Network Layer Management Entity – Service Access Point) αφορά λειτουργίες όπως εγκατάσταση νέου δικτύου, αναγνώριση γειτονικών συσκευών, εύρεση και καταγραφή διαδρομών δρομολόγηση μηνυμάτων διευθυνσιοδότηση νέων συσκευών και επιλογή μηχανισμού δρομολόγησης.

2.7.2. Επίπεδο εφαρμογής

Το επίπεδο εφαρμογών αποτελείται από τα εξής:

- Υποεπίπεδο υποστήριξης εφαρμογών (Application support sublayer, APS)
- Πλαίσιο εφαρμογών (Application framework, AF)
- Αντικείμενα συσκευής ZigBee (ZigBee Device Objects, ZDO)
- Καθορισμένες από τον κατασκευαστή εφαρμογές.

Το υποεπίπεδο APS είναι υπεύθυνο για τη μεταφορά των δεδομένων των εφαρμογών σε άλλες συσκευές του δικτύου. Επίσης υποστηρίζει την ανακάλυψη συσκευών και την εγκατάσταση συνδέσεων με αυτές. Τα ZDO είναι αυτά που καθορίζουν το ρόλο της κάθε συσκευής στο δίκτυο, τον τρόπο λειτουργίας της και παρέχουν τη δυνατότητα για ανακάλυψη υπηρεσιών και συσκευών στις εφαρμογές. Επίσης διαχειρίζονται όλους τους μηχανισμούς που έχουν σχέση με την ασφάλεια. Το πλαίσιο εφαρμογών είναι το περιβάλλον στο οποίο φιλοξενούνται οι εφαρμογές μέσα σε μία συσκευή ZigBee. Σε αυτό μπορούν να υπάρξουν μέχρι και 240 εφαρμογές. Για το διαχωρισμό τους, καθεμία έχει το δικό της σημείο τερματισμού (Asvest T., 2014)

2.7.3. Δρομολόγηση μηνυμάτων στο δίκτυο

Ο πιο απλό τρόπο μεταφοράς μηνυμάτων θεωρείται το ασύρματο δίκτυο αισθητήρων και ονομάζεται Flooding. Κάθε κόμβος έχει την δυνατότητα να μεταδίδει τις μετρήσεις τους προς τους γειτονικούς κόμβους, οι οποίοι με την σειρά τους ακολουθούν την ίδια διαδικασία με σκοπό να καταλήξει το μήνυμα στον προορισμό όπου είναι και το τελευταίο σημείο μετάδοσης. Θα πρέπει να σημειωθεί ότι αυτό ο τρόπος μεταδόσεις παρουσιάζει σημαντικά μειονεκτήματα. Ένα από τα βασικά είναι ότι σε περίπτωση υπερφόρτωσης του δικτύου τα μηνύματα δεν φτάνουν στον τελικό προορισμό με συνέπεια την μεγάλη κατανάλωση ενέργειας και πόρων.

Ο πλέον διαδεδομένος τρόπος δρομολόγησης των μηνυμάτων είναι με την χρήση του αλγορίθμου AD-Hoc On Demand Distance Vector – AODV. Σε αυτή την περίπτωση ο κάθε κόμβος επεξεργάζεται πληροφορίες σχετικά με τις διάφορες δυνατές διαδρομές προορισμού και τις αποθηκεύει σε ένα πίνακα δρομολόγησης (Routing Table). Τα δεδομένα που περιέχει ο πίνακας δρομολόγησης του κάθε κόμβου αφορούν τα εξής:

- Την ID διεύθυνση του τελικού προορισμού
- Την απόσταση (αριθμός Hops)
- Το κόστος μετάδοσης
- Την ID διεύθυνση του επόμενου κόμβου που βρίσκεται στον δρόμο για το τελικό προορισμό.

Ο AODV αλγόριθμός πέρα από την εύρεση των κατάλληλων διαδρομών θα πρέπει ταυτόχρονα να ελέγχει χρήση των πόρων του κάθε κόμβου.

Η διαδρομές δρομολόγησης ανανεώνονται όταν ο ένας κόμβος προσπαθεί να εισέλθει στο δίκτυο. Στην περίπτωση αυτή εκπέμπει μια αίτηση σύνδεσης προς τους γειτονικούς κόμβους, οι οποίοι με την σειρά τους καταγράφουν τον κόμβο από τον οποίο προήλθε το μήνυμα και στέλνουν προς αυτόν ένα σύνολο εναλλακτικών διαδρομών. Όταν ο γειτονικός κόμβος λάβει την αίτηση σύνδεσης, στέλνει στον υποψήφιο κόμβο τα δεδομένα της τρέχουσας διαδρομής που είναι αποθηκευμένα στον προαναφερόμενο πίνακα δρομολόγησης. Ο υποψήφιος κόμβος θα εισέλθει στο δίκτυο μέσω του κόμβου που έχει την διαδρομή τον μικρότερο αριθμό Hops προς τον τελικό προορισμό. Ωστόσο, θα πρέπει να επισημανθεί ότι στην περίπτωση που ο ένας κόμβος βγει εκτός λειτουργίας η διαδικασία διακόπτεται και επαναλαμβάνεται από την αρχή.

Για την διατήρηση της χωρητικότητας του δικτύου, κάθε σύνδεση έχει ένα αύξων αριθμό ο οποίος χρησιμοποιείται από τους γειτονικούς κόμβους με σκοπό να αποφεύγονται οι επαναλήψεις. Το μεγάλο πλεονέκτημα του αλγορίθμου είναι ότι δεν δημιουργείται περισσότερο φορτίο στο ήδη υπάρχον δίκτυο σε περίπτωση εισόδων νέων κόμβων, με συνέπεια την εξοικονόμηση πόρων στο δίκτυο. Τονίζεται ότι ο προαναφερόμενος αλγόριθμός χρειάζεται περισσότερο χρόνο για την δημιουργία μια σύνδεσης αφού θα πρέπει να υπάρξει ανταπόκριση και από τους υπόλοιπους κόμβους του δικτύου (Μανωλόπουλος Χ., 2010).

ΚΕΦΑΛΑΙΟ 3: ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ TinyOS ΚΑΙ Η ΓΛΩΣΣΑ NesC

3.1. ΕΙΣΑΓΩΓΗ

Το TinyOS αναπτύχθηκε και εξελίχθηκε από το πανεπιστήμιο του Berkley και είναι λογισμικό ανοιχτού κώδικα. Πρόκειται για ένα λειτουργικό σύστημα μικρό σε μέγεθος και οικονομικό ενεργειακά στη διαχείριση των αισθητήρων. Παρέχει ένα σύνολο από δομικές μονάδες λογισμικού από τις οποίες ο προγραμματιστής μπορεί να διαλέξει τα κατάλληλα συστατικά (components). Το μέγεθος τέτοιων αρχείων είναι τις τάξης των 200 bytes έτσι το μέγεθος του συνολικού προγράμματος παραμένει το ελάχιστο δυνατό. Το λειτουργικό σύστημα αυτό διαχειρίζεται τόσο το υλικό όσο και το ασύρματο δίκτυο εκτελώντας τις μετρήσεις των αισθητήρων, παίρνοντας αποφάσεις δρομολόγησης και ελέγχοντας την κατανάλωση ενέργειας.

Λόγω των περιορισμών που επέβαλλε η φύση των αισθητήρων, μια νέα γλώσσα προγραμματισμού αναπτύχθηκε, η nesC, που υλοποιεί τις δομικές σχεδιαστικές ανάγκες και την επαναχρησιμοποίηση κώδικα του TinyOS για μικροσκοπικούς αισθητήρες. Για την υλοποίηση της επαναχρησιμοποίησης το TinyOS εφαρμόζει μια αρχιτεκτονική κατανομής σε επιμέρους συστατικά (component – based). Επιπρόσθετα για την βελτιστοποίηση της διαχείρισης ενέργειας χρησιμοποιεί ένα μοντέλο εκτέλεσης βασισμένο σε γεγονότα (event – based) όπου τα γεγονότα οδηγούν τα προγράμματα και οι σχετικοί πόροι αποδεσμεύονται με το πέρας της χρήσης τους.



Εικόνα 3.1: Λογότυπο TinyOS

[Πηγή: <http://neobenblog.blogspot.gr>]

Επίσης, το TinyOS έχει βελτιστοποιηθεί ως προς την χρήση της μνήμης και προσφέρει υψηλή αποδοτικότητα ως προς την κατανάλωση ενέργειας. Επιπλέον, παρέχει διεπαφές (interfaces) μεταξύ των επιμέρους τμημάτων που το συνθέτουν και που ανήκουν σε γειτονικά στρώματα της αρχιτεκτονικής του.

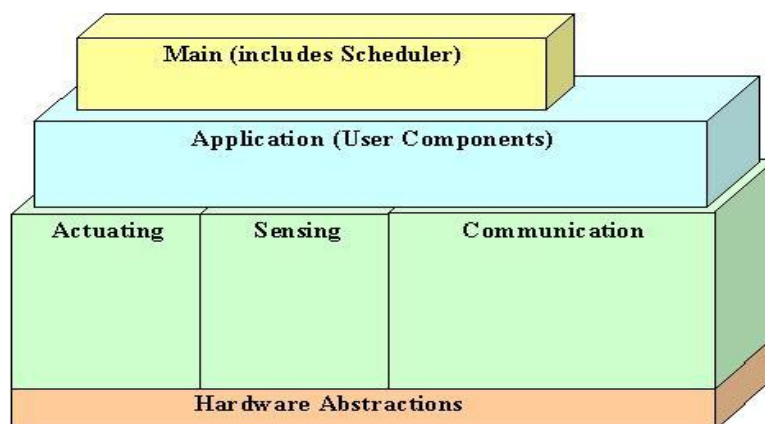
Η προσαρμογή του TinyOS σε διάφορα είδη πλατφόρμων είναι εύκολη εξαιτίας της εύχρηστης και αφαιρούμενης διαστρωμάτωσης σε επίπεδο υλικού στον πυρήνα. Η διαστρωμάτωση αυτή επίσης διευκολύνει πολύ την ανάπτυξη των ασύρματων δικτύων. Το TinyOS ακόμα προσφέρει μια σειρά από εφαρμογές και εργαλεία ανάπτυξης όπως:

- Το Tossim (προσομοιωτής δικτύων του TinyOS)
- Το deluge και το TinyDB που βοηθούν στην ανάπτυξη και έρευνα των WSN.

Λόγω της αποδοτικής σχεδίασης, της μεγάλης κοινότητας υποστήριξης και του ανοικτού κώδικα το TinyOS έγινε το πλέον διαδεδομένο λειτουργικό σύστημα για τα WSN (Πλευράκη Α., 2011).

3.2.ΑΡΧΙΤΕΚΤΟΝΙΚΗ TinyOS

Κάθε λειτουργικό σύστημα χωρίζει τα δομικά του στοιχεία (components) σε επίπεδα, με αυτή την αρχιτεκτονική έχει δημιουργηθεί και το TinyOS (Σχήμα 3.1).



Σχήμα 3.1: Αρχιτεκτονική TinyOS

[Πηγή: <http://old.ceid.upatras.gr>]

Όπως παρατηρείται στο Σχήμα 3.1 στο χαμηλότερο επίπεδο (Hardware Abstraction) βρίσκονται τα components που αφορούν το hardware του συστήματος τα οποία είναι:

- I/O συσκευές
- Αισθητήρες
- Πομποδέκτης

Στο επόμενο επίπεδο υπάρχουν τα εξής:

- Components: είναι υπεύθυνα για την ενεργοποίηση (actuating)
- Sensing: την συλλογή δεδομένων
- Communication επικοινωνία ενός mote

Ωστόσο, τα υψηλότερα επίπεδα (Main και Application) αφορούν components που έχουν δημιουργηθεί από την χρήση, τα οποία οργανώνουν τον τρόπο λειτουργίας των components χαμηλότερων επιπέδων. Στην συνέχεια του εδαφίου θα πραγματοποιηθεί ανάλυση στα component τα οποία αποτελούνται από τρία είδη συναρτήσεων και είναι τα ακόλουθα:

- Commands
- Events
- Tasks

▼ *Commands*

Οι συναρτήσεις Commands είναι αιτήσεις (request) προς τα components χαμηλότερου επιπέδου για την έναρξη και την εκτέλεση μιας ενέργειας. Η εκτέλεση τους είναι άμεση χωρίς την δυνατότητα αποκλεισμού και η ολοκλήρωση μπορεί να πραγματοποιηθεί σε πολύ μικρό χρονικό διάστημα. Επιπλέον, έχουν την δυνατότητα να σηματοδοτήσουν τα Events. Σημειώνεται ότι μαζί με τα Events αποτελούν τον μηχανισμό επικοινωνίας μεταξύ των components.

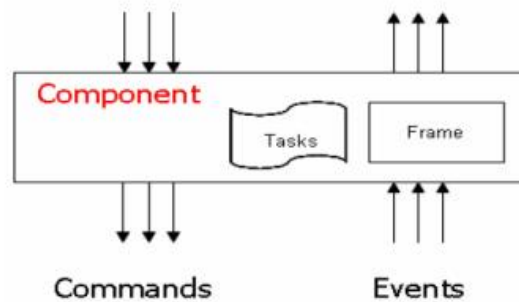
▼ *Events*

Οι συναρτήσεις Events σηματοδοτούν το τέλος μιας εργασίας, οι οποία βέβαια έχει ξεκινήσει από το command. Η command έχει την δυνατότητα να ζητήσει από έναν αισθητήρα

την έναρξη συλλογής δεδομένων, ενώ το event προκαλείται όταν ο αισθητήρας ολοκληρώσει την συλλογή μετρήσεων. Ωστόσο, ένα event μπορεί να εμφανιστεί ασύγχρονα λόγω μιας διακοπής του hardware και η εκτέλεση μιας λειτουργίας λογισμικού να πραγματοποιηθεί σε δύο φάσεις(split phase operation). Μετά από την αίτηση εκτέλεσης μιας λειτουργίας, η command επιστρέφει και το αντίστοιχο event μετά τη μεσολάβηση κάποιου χρονικού διαστήματος, σηματοδοτώντας με αυτό τον τρόπο το τέλος της λειτουργίας. Τέλος το event έχουν την δυνατότητα να καλέσουν τα commands και tasks χαμηλότερου επίπεδο και να σηματοδοτήσουν τα events υψηλότερου επιπέδου.

▼ Tasks

Τα tasks καλούνται για να εκτελέσουν λειτουργία χαμηλότερης προτεραιότητας μετά την ολοκλήρωση των τρεχόντων commands και events. Τα tasks έχουν την δυνατότητα να καλέσουν μία μια command χαμηλότερου επιπέδου, να σηματοδοτήσει ένα event ή ακόμα και ένα καινούργιο task. Η εκτέλεση του δεν μπορεί να διακοπεί από κάποιο σύγχρονο event ή άλλο task, παρά μόνο από κάποιο σύγχρονο event. Επιπλέον, έχουν πρόσβαση σε δεδομένα που αφορούν σε ένα συγκεκριμένο component, στο οποίο ανήκουν. Χρησιμοποιούν τους περισσότερους πόρους του συστήματος γιατί εκτελούνται συνεχώς και δεν διακόπτονται, παρά μόνον σε ειδικές περιπτώσεις (ασύγχρονο event) (Μανωλόπουλος X., 2010).



Σχήμα 3.2: Δομή ενός component
(Πλευράκη Α., 2011)

3.3. Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ nesC

Η γλώσσα NesC (network embedded systems C), η οποία αναπτύχθηκε από ερευνητές του Πανεπιστημίου UC Berkeley, αντιπροσωπεύει ένα πολλά υποσχόμενο πεδίο για τους σχεδιαστές εφαρμογών. Είναι κατάλληλα σχεδιασμένη για ενσωματωμένα συστήματα

δικτύων και υποστηρίζει ένα προγραμματιστικό μοντέλο που ενσωματώνει αντιδραστικότητα με το περιβάλλον, ταυτοχρονισμό και δυνατότητα επικοινωνίας. Ένα βασικός άξονας επικέντρωσης της NesC είναι η ολιστικός σχεδιασμός συστημάτων.

Οι εφαρμογές των μονάδων-αισθητήρων (motes) είναι βαθιά συνδεδεμένες στο υλικό και κάθε μονάδα τρέχει μια εφαρμογή κάθε φορά. Έτσι υπάρχει ένας αριθμός μοναδικών προκλήσεων που η γλώσσα NesC πρέπει να επιληφθεί:

∅ *Οδήγηση από την αλληλεπίδραση με το περιβάλλον:*

Σε αντίθεση με τα παραδοσιακά συστήματα υπολογιστών, τα motes χρησιμοποιούνται για τη συλλογή δεδομένων και τον έλεγχο του τοπικού περιβάλλοντος, παρά για γενικής φύσεως υπολογισμούς. Αυτή η ιδιαιτερότητα οδηγεί σε δυο παρατηρήσεις:

- 1^η Παρατήρηση: τα motes είναι στοιχειωδώς οδηγούμενα από συμβάντα (event driven), αντιδρώντας σε αλλαγές του περιβάλλοντος (άφιξη ενός μηνύματος, ανάκτηση δεδομένων από αισθητήρες) παρά οδηγούμενα από διαδραστική (interactive) ή κατά δεσμίδες (batch) επεξεργασία.
- 2^η Παρατήρηση: Η «άφιξη» ενός συμβάντος ή η επεξεργασία δεδομένων είναι συντρέχουσες δραστηριότητες, απαιτώντας έτσι μια μεθόδευση για διαχείριση του ταυτοχρονισμού αυτού που επιλαμβάνεται ενδεχόμενων σφαλμάτων (bugs) όπως οι συνθήκες συναγωνισμού (race conditions).

∅ *Περιορισμένοι πόροι:*

Οι μονάδες αυτές (motes) έχουν πολύ περιορισμένους φυσικούς πόρους, λόγω των ιδιαίτερων αναγκών για μικρό μέγεθος, χαμηλό κόστος και μικρή κατανάλωση ενέργειας. Οι περιορισμοί αυτοί δεν αναμένεται να εκλείψουν, καθώς τα οφέλη από την προσδοκία του νόμου του Moore θα οδηγεί συνεχώς σε μείωση του μεγέθους και του κόστους παρά σε αύξηση δυνατοτήτων-ικανοτήτων στο ίδιο μέγεθος.

✓ *Αξιοπιστία:*

Αν και είναι αναμενόμενο οι μονάδες αυτές να παθαίνουν βλάβη λόγω σφαλμάτων υλικού, είναι έντονη η ανάγκη για εφαρμογές οι οποίες μπορεί να τρέχουν για μεγάλο χρονικό διάστημα. Για παράδειγμα, οι εφαρμογές παρακολούθησης περιβάλλοντος πρέπει να είναι ικανές να συλλέγουν δεδομένα χωρίς την ανθρώπινη παρέμβαση για μήνες κάθε φορά. Ένας σημαντικός στόχος είναι η μείωση των σφαλμάτων κατά τη διάρκεια της εκτέλεσης

(run-time errors), καθώς δεν υπάρχει ουσιαστικός μηχανισμός ανάκαμψης σφαλμάτων εκτός από την αυτόματη επανεκκίνηση του συστήματος.

▼ *Μικρές απαιτήσεις για λειτουργίες πραγματικού χρόνου:*

Παρόλο που υπάρχουν κάποιες εργασίες που είναι χρονικά κρίσιμες, όπως η διαχείριση της ασύρματης επικοινωνίας ή η σταθμοσκόπηση των αισθητήρων, σε γενικές γραμμές δεν υπάρχουν μεγάλες απαιτήσεις για πραγματικού χρόνου λειτουργίες. Μάλιστα η εμπειρία έχει δείξει ότι οι όποιοι χρονικοί περιορισμοί μπορούν να ικανοποιηθούν έχοντας απόλυτο έλεγχο της εφαρμογής και του λειτουργικού συστήματος και παράλληλα μειώνοντας την χρήση (utilization). Μια από τις λίγες κρίσιμες από πλευράς χρόνου λειτουργίες στα δίκτυα αισθητήρων είναι η ασύρματη επικοινωνία. Δεδομένου όμως της βασικής αναξιοπιστίας της ραδιοζεύξης γενικότερα δε θα χρειαστεί απαραίτητα να ικανοποιήσουμε δύσκολες απαιτήσεις στον τομέα αυτό.

Η γλώσσα NesC είναι προέκταση της C και αναμενόμενα, έχει όμοια σύνταξη, εντούτοις παρέχει τρία σημαντικά στοιχεία που τη διαφοροποιούν σημαντικά:

- Η γλώσσα NesC ορίζει ένα μοντέλο συστατικών που υποστηρίζει συστήματα οδηγούμενα από συμβάντα. Το μοντέλο αυτό παρέχει αμφίδρομες διεπαφές (interfaces) προς απλοποίηση της ροής των συμβάντων και επιτρέπει αποδοτική και ελαφριά υλοποίηση χωρίς τη δημιουργία εικονικών συναρτήσεων και δυναμικών στοιχείων.
- Παράλληλα ορίζει ένα απλό αλλά συγκεκριμένο μοντέλο ταυτοχρονισμού σε συνδυασμό με εκτεταμένη ανάλυση κατά τη μεταγλώττιση: ο μεταγλωττιστής (compiler) της NesC εντοπίζει τις πλειονότητα των περιπτώσεων ανταγωνισμού δεδομένων (data race) κατά τη διάρκεια της μεταγλώττισης. Αυτός ο συνδυασμός επιτρέπει τη δημιουργία σύγχρονων εφαρμογών που απαιτούν περιορισμένους πόρους.
- Η γλώσσα NesC παρέχει μια μοναδική ισορροπία μεταξύ της ανάλυσης προγράμματος, για τη βελτίωση της αξιοπιστίας και τη μείωση του κώδικα, και της δυνατότητας για δημιουργία ολοκληρωμένων εφαρμογών.

Επειδή η γλώσσα NesC έχει αποδειχθεί ότι είναι αποτελεσματική στην περίπτωση ανάπτυξης εφαρμογών για ασύρματα δίκτυα αισθητήρων, χρησιμοποιείται σαν την προγραμματιστική γλώσσα για το λειτουργικό σύστημα TinyOS, ένα μικρό λειτουργικό σύστημα για ασύρματα δίκτυα αισθητήρων που έχει υιοθετηθεί από ένα μεγάλο πλήθος ερευνητικών ομάδων σε όλο τον κόσμο. Σε εξέλιξη βρίσκονται έρευνες προς ανάπτυξη στο πρότυπο του TinyOS και άλλων γλωσσών προγραμματισμού αλλά μέχρι στιγμής η NesC είναι η μόνη γλώσσα που μπορεί να χρησιμοποιηθεί για την ανάπτυξη προγραμμάτων στο TinyOS (Gay D. et al, 2003 ; Πλευράκη Α., 2011).

3.3.1. Δομικά χαρακτηριστικά nesC

Η nesC (network embedded systems C) είναι μια γλώσσα προγραμματισμού ανοικτού κώδικα (open source). Επιπλέον η nesC βασίζεται στην γλώσσα προγραμματισμού C για δύο λόγους, οι οποίοι είναι οι εξής:

- Οι μεταγλωττιστές τις C παράγουν αποδοτικότερο κώδικα για όλους τους Microcontrollers
- Μεγάλο μέρος προγραμματιστών σε embedded systems την χρησιμοποιούν ως κύρια γλώσσα προγραμματισμού.

Για να πραγματοποιηθεί ο έλεγχος των λαθών και για να υπάρχει μεγαλύτερη απόδοση, πραγματοποιείται ολική ανάλυση του κώδικα κατά την διάρκεια του compiling. Σημειώνεται ότι ο κώδικας αυτός είναι στατιστικός, αφού η μνήμη κατανέμεται στατιστικά σε μία εφαρμογή TinyOS κατά τη διάρκεια της μεταγλώττισης (compiling). Το γράφημα διασυνδέσεων (wiring) μεταξύ των διάφορων components είναι σταθερό και γνωστό, εξαλείφοντας με αυτόν τον τρόπο την ανάγκη για δυναμική δέσμευση μνήμης.

Τα δομικά χαρακτηριστικά που αφορούν την nesC είναι τα ακόλουθα:

- Interfaces
- Modules
- Configuration

Στην συνέχεια του εδαφίου θα πραγματοποιηθεί ανάλυση των προαναφερόμενων δομικών χαρακτηριστικών της nesC με σκοπό την περιγραφή τους και την ανάλυση της χρησιμότητάς τους.

▼ Interfaces

Τα interfaces χρησιμεύουν για την αλληλεπίδραση μεταξύ των components και αποτελούν αναπόσπαστο κομμάτι ενός προγράμματος. Δίνεται η δυνατότητα είτε να παρέχονται είτε να χρησιμοποιούνται από ένα component. Τα παρεχόμενα interfaces αντιπροσωπεύουν τις λειτουργικές που παρέχει ένα component – provider. Αυτά που χρησιμοποιούνται αντιπροσωπεύουν τις λειτουργικές που απαιτεί ένα component – user. Δεν περιέχουν κώδικα για την εκτέλεση υπολογισμών, το μόνο που περιέχουν είναι commands που υλοποιούνται από το component-provider του interface, καθώς και events που υλοποιούνται από το component –user του interface.

Με αυτόν τον τρόπο, δηλαδή της χρήσης των interfaces δίνει την δυνατότητα εκτέλεσης μιας λειτουργίας σε slip-phase operation. Δεν μπορεί να εκτελέσει μια command αν δεν παρέχει μια υλοποίηση για τα events που θα προκαλέσουν το τέλος της. στην συνέχεια ακολουθεί η Εικόνα 3.2 όπου απεικονίζεται ένα interface με το όνομα Timer, το οποίο χρησιμοποιείται για τον συγχρονισμό ανάμεσα σε δύο ή περισσότερα components.

```
includes Timer; // make TIMER_x constants available
interface Timer {
    command result_t start(char type, uint32_t interval);
    command result_t stop();
    event result_t fired();
}
```

Εικόνα 3.2: Ορισμός του Timer Interface

(Μανωλόπουλος X., 2010).

Όπως παρατηρείται και παραπάνω στον Timer Interface δηλώνονται δυο commands (start και stop) και ένα event (fired). Το component που έχει την δυνατότητα να παρέχει το interface θα πρέπει να έχει τις υλοποιήσεις των commands (start και stop). Το component που θα κάνει χωρίς του Timer Interface θα πρέπει να υλοποιεί το event fired.

▼ Modules

Το module είναι στην ουσία ένα component που χρησιμοποιεί ένα ή περισσότερα interfaces. Μέσα στον κώδικα του module βρίσκεται ο πυρήνας του TinyOS και καθορίζεται η λειτουργία του προγράμματος. Στην συνέχεια ακολουθεί ένα παράδειγμα με το όνομα BlinkM, με το οποίο καθορίζεται η λειτουργία ενός LED να αναβοσβήνει 1000ms.

```
module BlinkM {
  //interfaces που παρέχει το module BlinkM
  provides {
    interface StdControl;
  }
  //interfaces που χρησιμοποιεί το module BlinkM
  uses {
    interface Timer;
    interface Leds;
  }
}
implementation {
  //υλοποίηση commands του interface που παρέχει
  command result_t StdControl.init() {
    call Leds.init();
    return SUCCESS;
  }
  //με την init() command του StdControl interface
  //καλούμε το Leds interface να εκτελέσει την init()
  //command που έχει.
  command result_t StdControl.start() {
    return call Timer.start(TIMER_REPEAT, 1000);
  }
  //με την start() command του StdControl interface
  //καλούμε το Timer interface να εκτελέσει την start()
  //command που έχει.
  command result_t StdControl.stop() {
    return call Timer.stop();
  }
  //με την stop() command του StdControl interface
  //καλούμε το Timer interface να εκτελέσει την stop()
  //command που έχει.
  //υλοποίηση event του interface που χρησιμοποιεί
  event result_t Timer.fired()
  {
    call Leds.redToggle();
    return SUCCESS;
  }
  //με το fired() event του Timer interface καλούμε
  //το Leds interface να εκτελέσει την redToggle()
  //command που έχει.
}
```

Εικόνα 3.3: Το Module αρχείο BlinkM.nc της εφαρμογής Blink

(Μανωλόπουλος X., 2010)

Το module BlinkM με την χρήση του StdControl interface έχει την δυνατότητα μέσω των commands `init()` `start()` και `stop()` έχει να καλέσει τα interfaces Timer μέσω του `call`. Αυτά στην συνέχεια εκτελούν την εντολή command που περιέχουν.

Στο Timer interface στην εντολή `start` έχουν ορισμένα `TIMER_REPEAT` και `1000`. Το πρώτο δηλώνει την συνεχή επανάληψη του Timer και το δεύτερο το διάστημα που μεσολαβεί σε ms. Σε περίπτωση που χρειάζεται μια μόνο εκκίνηση του Timer, χρησιμοποιείται η τιμή `TIMER_ONE_SHOT`. Όταν περάσει αυτό το χρονικό διάστημα τότε καλείται το event `fired()` που ανάβει το κόκκινο LED.

▼ Configuration

Όπως προαναφέρθηκε παραπάνω το module BlinkM μέσω του interface StdControl, μπορεί να καλέσει τα Leds και Timer interfaces και να εκτελέσει τα commands αυτών. Ωστόσο, οι λειτουργίες σε αυτά τα commands γίνονται σε άλλα modules που παρέχουν τα interface. Για να γνωρίζει το module BlinkM που γίνεται η υλοποίηση των λειτουργιών χρειάζεται ένα ξεχωριστό component που ονομάζεται Configuration.

Η διαδικασία σύνδεσης των components με αυτόν τον τρόπο μεταξύ τους αποκαλείται wiring. Διαθέτουν τους τελεστές \rightarrow , \leftarrow και $=$. Τα βέλη δηλώνουν την κατεύθυνση, συγκεκριμένα το interface που βρίσκεται στο component που χρησιμοποιείται, συνδέεται με το interface του component που παρέχει. Στην Εικόνα 3.4 ακολουθεί το configuration με το όνομα Blink. Στην αρχή δηλώνονται τα modules-components τα οποία θα συνδεθούν μεταξύ τους και έπειτα ακολουθεί ο τρόπος με τον οποίο συνδέονται μεταξύ τους, με βάση τα interfaces που διαθέτουν.

```
configuration Blink {
}
implementation {
  components Main, BlinkM, SingleTimer, LedsC;
  Main.StdControl -> SingleTimer.StdControl; //σύνδεση StdControl interface του component-χρήστη "Main"
                                                    //με το StdControl interface του component-παροχέα "SingleTimer"
  Main.StdControl -> BlinkM.StdControl;       //σύνδεση StdControl interface του component-χρήστη "Main"
                                                    //με το StdControl interface του component-παροχέα "BlinkM"
  BlinkM.Timer -> SingleTimer.Timer;         //σύνδεση Timer interface του component-χρήστη "BlinkM"
                                                    //με το Timer interface του component-παροχέα "SingleTimer"
  BlinkM.Leds -> LedsC;                       //σύνδεση Leds interface του component-χρήστη "BlinkM"
  //BlinkM.Leds -> LedsC.Leds;                //με το Leds interface του component-παροχέα "LedsC"
}
```

Εικόνα 3.4. Το Configuration αρχείο Blink.nc της εφαρμογής Blink

(Μανωλόπουλος X., 2010)

Το interface StdControl χρησιμοποιείται από το Main component για την αρχικοποίηση (reset) των components που παρέχουν, δηλαδή του BlinkM και Single Timer. Έπειτα το BlinkM component χρησιμοποιεί τα Single Timer και Leds components, μέσω των interface Timer και Leds. Όπως είναι κατανοητό από όσα προαναφέρθηκαν το module Single Timer χρησιμοποιείται και από το Main αλλά και από το BlinkM μέσω των StdControl και του Timer αντίστοιχα. Ωστόσο, έχει την δυνατότητα να κάνει και τις διασυνδέσεις των interfaces. Αν παρατηρήσουμε τον κώδικα του Single Timer στην αρχή παρέχει τα interfaces Timer και StdControl, όπου γίνεται χρήση από τα Main και BlinkM modules.

```

configuration SingleTimer (
  provides interface Timer;           //παρέχει τα interfaces Timer και StdControl
  provides interface StdControl;
)

implementation (
  components TimerC;                 //δήλωση του component TimerC

  Timer = TimerC.Timer[unique("Timer")]; //ταυτοποίηση interface Timer του component SingleTimer
  //με το interface Timer του component TimerC
  StdControl = TimerC;              //ταυτοποίηση interface StdControl του component SingleTimer
  //με το interface StdControl του component TimerC
)

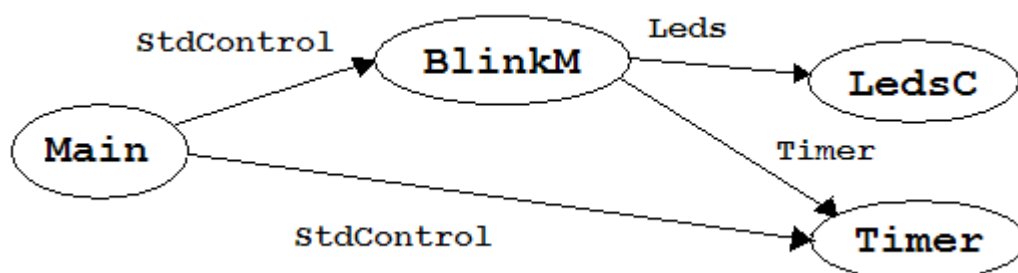
```

Εικόνα 3.5: Παράδειγμα διασύνδεσης μεταξύ των interfaces Timer και StdControl
(Μανωλόπουλος X., 2010)

Στην Εικόνα 3.5 χρησιμοποιείται ο τελεστής =, όπου χρησιμοποιείται για την ταυτοποίηση των interfaces ενός user – component με αυτά του provider – component. Με αυτόν τον τρόπο τα παρεχόμενα interfaces Timer και StdControl του SingleTimer είναι τα ίδια με αυτά του component TimerC, αφού δεν παρεμβάλλεται ανάμεσα τους κάποιο άλλο component. (Μανωλόπουλος X., 2010)

3.3.2. Τρόπος λειτουργίας

Για την κατανόηση των όσων προαναφέρθηκαν δίνεται το Σχέδιο 3.3. με σκοπό την απεικόνιση του τρόπου διασύνδεσης των modules που ανάβουν ή σβήνουν το κόκκινο Led. Θα πρέπει να σημειωθεί ότι τα βέλη του σχήματος αντιστοιχούν στα interfaces ενώ τα πλαίσια στα modules.



Σχήμα 3.3: Διασύνδεση των modules της εφαρμογής Blink μέσω των αντίστοιχων interfaces (Μανωλόπουλος X., 2010)

Άρα σύμφωνα με το Σχήμα 3.3 για την αρχικοποίηση όλων των components καλείται η συνάρτηση `Main.StdControl.init()`, μέσω της οποίας μπορεί και καλείται η `BlinkM.StdControl.init()`, που με την σειρά της καλεί την συνάρτηση αρχικοποίησης του component `LedC`, η `LedsC.Leds.init()`. Αφού ολοκληρωθεί η διαδικασία αρχικοποίηση πρέπει να πραγματοποιηθεί στην συνέχεια η έναρξη της λειτουργίας.

Με αυτό τον τρόπο καλείται η συνάρτηση έναρξης `Main.StdControl.start()`, μέσω της οποίας καλείται η `BlinkM.StdControl.start()` που με την σειρά της καλεί την `Timer.start(TIMER_REPEAT,1000)`. Ο `Timer` ξεκινά και κάθε 1000ms, μετά από κάθε διακοπή καλείται το event `BlinkM.Timer.fired()` που καλεί τον `BlinkM.LedToggle()` για να ανάψει η να σβήσει το κόκκινο LED (Μανωλόπουλος X., 2010).

ΚΕΦΑΛΑΙΟ 4: HADWARE ΤΗΣ ΕΤΑΙΡΙΑΣ CROSSBOW

4.1.ΕΙΣΑΓΩΓΗ

Τα ασύρματα δίκτυα αισθητήρων (WNS) χρησιμοποιούνται με σκοπό την καταγραφή και την παρατήρηση συμβάντων σε περιοχές που είναι δύσκολα προσβάσιμες ή ασύμφορες να τοποθετηθεί ένα ενσύρματο δίκτυο. Όπως προαναφέρθηκε σε προηγούμενο Κεφάλαιο ένα WPAN αποτελείται από δύο τύπους συσκευών οι οποίες είναι:

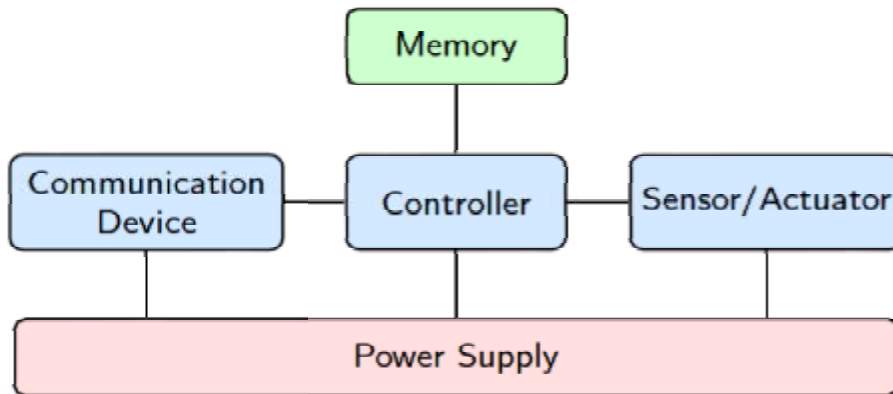
- Συσκευή πλήρους λειτουργίας: Full Function Device FFD
- Συσκευή περιορισμένης λειτουργίας: Reduced Function Device RFD

Στην περίπτωση του WSN που ανήκει στα WPAN δίκτυα, η συσκευή FFD θεωρείται ο κεντρικό αποδέκτης και ονομάζεται Base Station, ενώ οι συσκευές RFD θεωρούνται περιφερειακοί κόμβοι και ονομάζονται motes. Ο τρόπος λειτουργίας τους είναι προκαθορισμένος στα motes γίνεται η λήψη μετρήσεων και η αποστολή πληροφοριών είτε προς γειτονικά motes είτε προς το Base Station.

Όπως είναι λογικό στον κεντρικό αποδέκτη καταλήγουν όλα τα δεδομένα των μετρήσεων που προέρχονται από τους περιφερικούς κόμβους. Έτσι μέσω ηλεκτρονικού υπολογιστή, όπου έχει πρόσβαση ο χρήστης, δίνεται η δυνατότητα να πραγματοποιηθεί η επεξεργασία σε πραγματικό χρόνο και στην συνέχεια να γίνει αποθήκευση, ωστόσο, μπορεί να γίνει αποθήκευση αρχικά και μετά επεξεργασία κάποιου αρχείου με συνέπεια την ευκολία του χρήστη.

Η δομή των mote είναι προκαθορισμένη και συνήθως αποτελείται από 2 module και τις μπαταρίες. Στο ένα module βρίσκεται ο microcontroller, η Flash memory RF transceiver, ενώ στο άλλο η διάταξη αισθητήρων. Στο Σχήμα 4.1 απεικονίζεται η βασική δομή ενός mote.

Η ανάλυση του Hardware της εταιρίας Crossbow πραγματοποιείται με σκοπό την κατανόηση λειτουργίας τέτοιων συστημάτων, διότι παρατηρείται ότι χρησιμοποιείται αρκετά συχνά σε εφαρμογές και ειδικές επιστημονικές εργασίες.



Σχήμα 4.1: Αρχιτεκτονική WSN κόμβου.

[Πηγή: <http://docplayer.gr>]

Το Hardware της εταιρίας Crossbow αποτελείται από τα εξής:

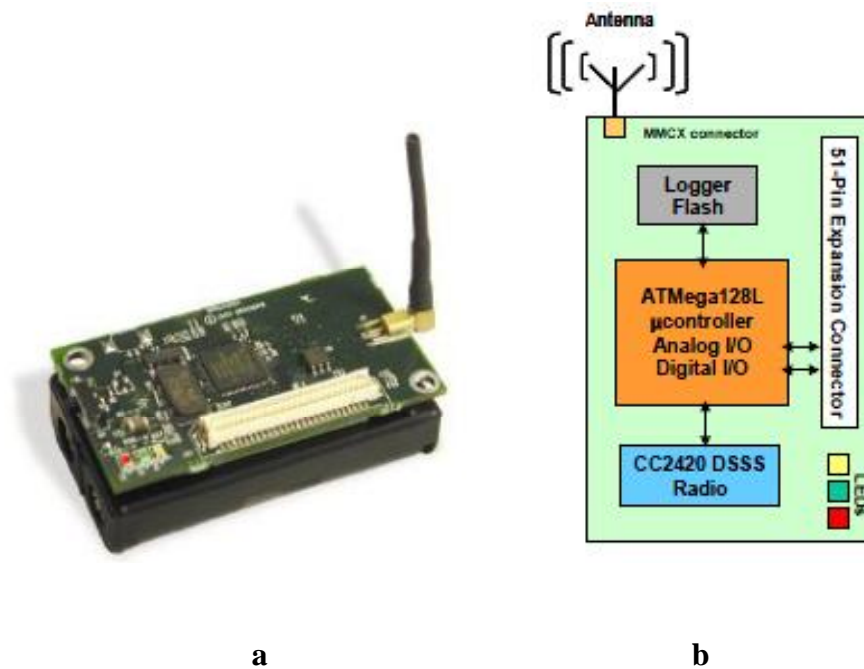
- Τέσσερα modules MICAZ MPR2600 που έχουν τον microcontroller και τον RF tranceiver
- Τέσσερα modules MTS400 αποτελούμενα από ένα σύνολο αισθητήρων
- Ένα module MIB520 για τον προγραμματισμό των motes και την λήψη δεδομένων από το δίκτυο.

Σημειώνεται ότι ένα module MICAZ MPR600 συνδεδεμένο με ένα module MTS400 συνθετούν ένα ασύρματο κόμβο – mote, ενώ αν συνδεθεί ένα MICAZ MPR2600 με το module MIB520 τότε έχουμε τον κεντρικό αποδέκτη – base station για την λήψη των δεδομένων από τα περιφερειακά motes.

4.2.MODULE MICAZ MPR2600

Το module MICAZ MPR2600 χρησιμοποιεί έναν μικροελεγκτή τύπου ATmega128L της εταιρίας Atmel και έναν RF transceiver CC420 της εταιρίας Chipcon. Η Εικόνα 4.1. απεικονίζει ένα MICAZ MPR2600 καθώς και το block διαγράμματός του. Σημειώνεται ότι η υποδοχή των 51 pins χρησιμοποιείται για την σύνδεση του συγκεκριμένου module, είτε με το

module των αισθητήρων (MTS400) για την υλοποίηση ενός mote, είτε με το module MIB520 για την υλοποίηση του για την υλοποίηση του base station (Crossbow, 2007).



Εικόνα 4.1. a) Όψη ενός WSN κόμβου **b)** Block διάγραμμα του WSN κόμβου

[Πηγή: <http://www.xbow.com>]

4.2.1. Atmel ATmega 128L microprocessor

Για την διεκπεραίωση κάποιων εφαρμογών σε ασύρματα δίκτυα, είναι υποχρεωτικός ένας μικροελεγκτής. Ο μικροελεγκτής έχει την δυνατότητα να επεξεργάζεται τα δεδομένα που προέρχονται από ένα σύνολο αισθητήρων και με την βοήθεια – συμβουλή ενός πομποδέκτη στέλνει πληροφορίες σε μία βάση με σκοπό είτε την συλλογή μετρήσεων είτε την ανίχνευση κάποιων συμβάντων.

Για να γίνει σωστή η επιλογή ενός τέτοιου μικροελεγκτή δίνεται έμφαση στους εξής παραμέτρους:

- Μικρή κατανάλωση ενέργειας
- Σχετικά απλή γκάμα εντολών
- Δυνατότητα απενεργοποίησης (όταν δεν εκτελεί κάποια λειτουργία)

Ο ATmega 128L είναι ένα 8 – bit Mc υλοποιημένος βάσει της αρχιτεκτονικής RISC (Reduced Instruction Set Computing), όπου οι εντολές οι οποίες είναι απλοποιημένες έχουν την δυνατότητα να πετύχουν υψηλότερη απόδοση υπό τν όρο ότι κάθε απλοποίηση παρέχει μικρότερο χρόνο εκτέλεσης της εντολής.

Μερικά από τα βασικά χαρακτηριστικά καταγράφονται και αναλύονται στην συνέχεια.

▼ *RISC architecture*

- Δυνατότητα εκτέλεσης 133 εντολών
- 32x8 registers γενικής χρήσης
- 16MHz clock δυνατότητα εκτέλεσης μέχρι και 16 MIPS

▼ *Peripheral features*

- SPI interfaces, για την διασύνδεση microcontroller –RF transceiver
- Διπλή USART σύγχρονης και ασύγχρονης λειτουργίας. Η USART0 χρησιμοποιείται για την επικοινωνία του microcontroller με τον H/Y, ενώ η USART1 είναι διαθέσιμη για τον χρήστη.
- ADC 10 – bit των 8 καναλιών, για την μετατροπή μιας τάσης εισόδου σε μια τιμή μήκους 10 bit με ανάλυση 1/1024 (0.1%). Μέσω ενός πολυπλέκτη 8 προς 1 υπάρχει δυνατότητα χρήσης διαφορετικών εισόδων (Port F)
- Two Wire Serial Interface – TWI, για την διασύνδεση μέχρι και 128 συσκευών χρησιμοποιώντας δύο αμφίδρομης κατεύθυνσης γραμμές, ένα για τον χρονισμό (SCL) και ένα για μεταφορά δεδομένων (SDA)
- Τέσσερις Timers/counters εκ των οποίων οι δύο είναι των 8 bit που χρησιμοποιούνται από τον TinyOs και 2 των 16 bit οι οποίοι είναι διαθέσιμοι στον χρήστη.
- Real Time Counter - RCT με στόχο την ρύθμιση της ημερομηνίας και της ώρας.

▼ *ATmega128L memories*

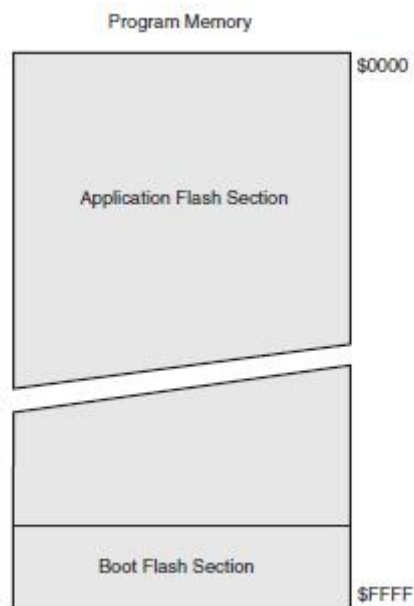
- Flash memory

Έχει μέγεθος 128 kBytes, έχει δυνατότητα υποστηρίξεις μέχρι και 10.000 κύκλους ανάγνωσης και εγγραφής. Θεωρείται κατάλληλη για την αποθήκευση του κώδικα της εκάστοτε εφαρμογής. Επιπλέον, οι εντολές του μικροελεγκτή είναι μήκους είτε 16 είτε 32

bits, η μνήμη είναι οργανωμένη ως 64K x 16 με συνέπεια η γραμμή κάθε μήκους να είναι 2 Bytes.

Για την καλύτερη ασφάλεια διαχωρίζεται σε δύο τμήματα τα οποία είναι τα ακόλουθα:

- Application program: αποθηκεύεται ο κύριος κώδικας
- Boot program: αποθηκεύονται κάποιοι παράμετροι για την εκκίνηση της λειτουργίας του μικροελεκτή.



Σχήμα 4.1: Εύρος διευθύνσεων 128Kbytes μνήμης Flash (64K x 16)

[Πηγή: <http://www.xbow.com>]

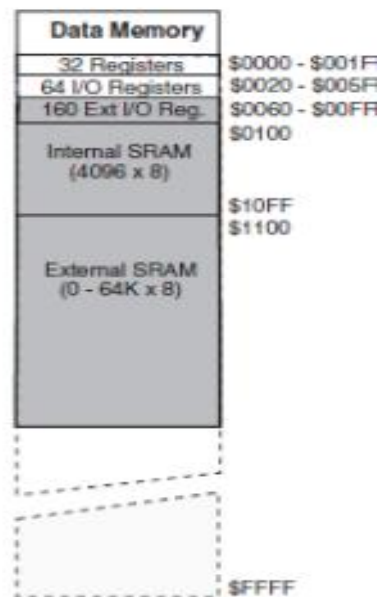
Ü Data Memory / external SRAM

Η μνήμη αυτή χρησιμοποιείται για την αποθήκευση κάποιων παραμέτρων της εφαρμογής, όπως μεταβλητές του TinyOS. Επιπλέον, είναι κατάλληλη για την αποθήκευση προσωρινών δεδομένων με στόχο την εκτέλεση εντολών, καθώς και δεδομένων που προέρχονται από τις περιφερειακές συσκευές οι οποίες είναι συνδεδεμένες στον μικροελεκτή. Κάθε γραμμή μήκους είναι του 1 Byte. Οι πρώτες 32 διευθύνσεις ανήκουν στους registers που χρησιμοποιούνται από την ALU. Οι επόμενες 224 αντιστοιχούν στην I/O μνήμη, η οποία χωρίζεται στην κύρια I/O μνήμη με πλήθος 64 διευθύνσεων και σε μια

προέκταση αυτής, πλήθους 160 διευθύνσεων. Ακόμα υπάρχει και η SRAM μεγέθους 4 kBytes, τονίζεται ότι δίνεται η δυνατότητα προέκτασης αυτής.

Η προαιρετική SRAM καταλαμβάνει μια περιοχή που αντιστοιχεί σε 64K διευθύνσεις. Η περιοχή ξεκινά αμέσως μετά το τέλος της τελευταίας διεύθυνσης της κύριας SRAM. Σε αυτήν μπορεί να συνδεθούν διάφορες περιφερειακές συσκευές όπως LCD οθόνη, A/D και D/A converters.

Memory Configuration A



Σχήμα 4.2: Οργάνωση της εξωτερικής μνήμης SRAM

[Πηγή: <http://www.xbow.com>]

ü EEPROM memory

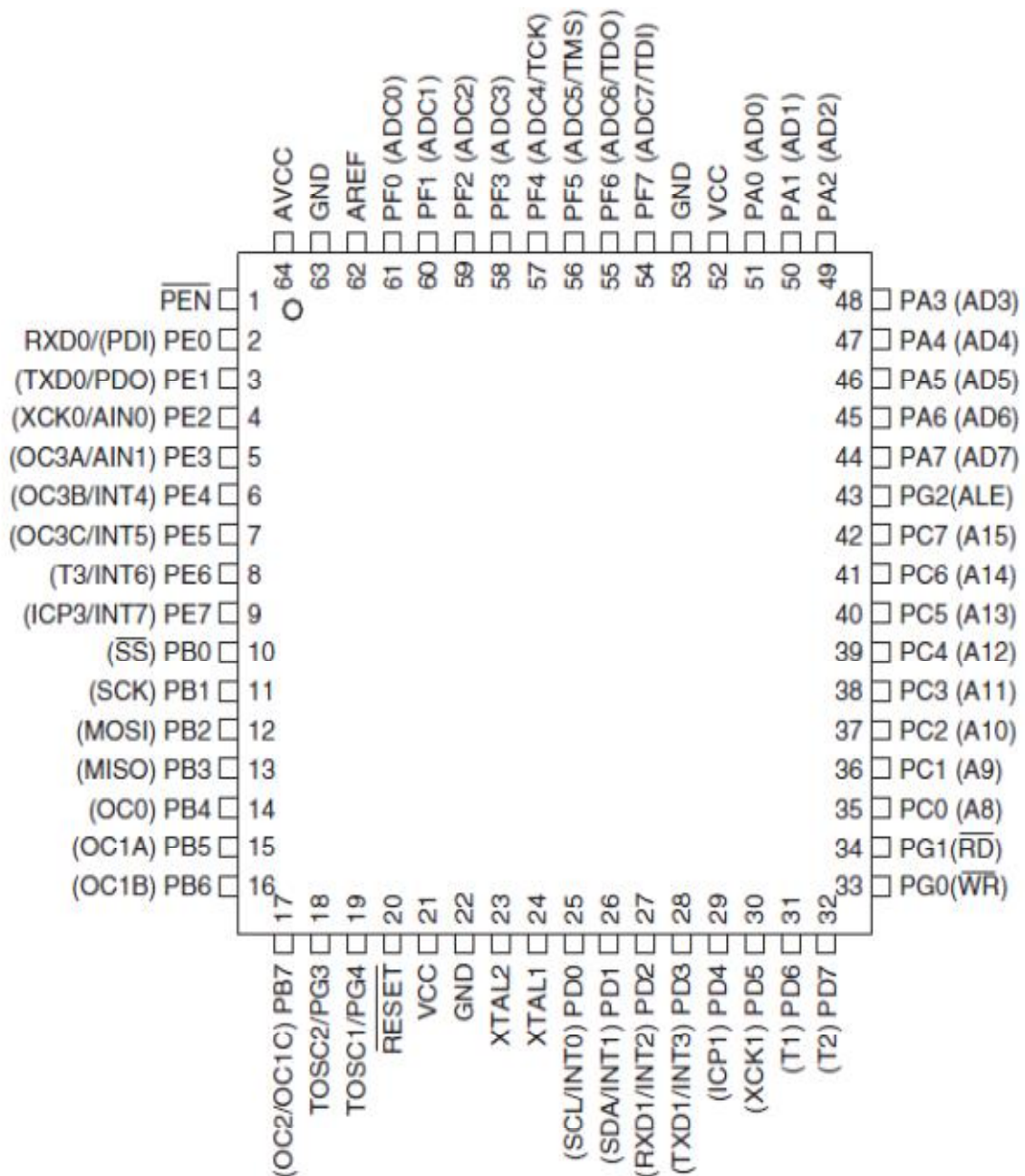
Ο ATmega 128L περιέχει μια EEPROM μνήμη δεδομένων, έχει μέγεθος 4 kBytes, έχει δυνατότητα υποστηρίξεις μέχρι και 100.000 κύκλους ανάγνωσης και εγγραφής. Στην μνήμη αυτή αποθηκεύονται κάποιες σταθερές όπως η διεύθυνση και η ομάδα που ανήκει σε κάθε κόμβο. Η πρόσβαση ανάμεσα στην EEPROM και στον CPU γίνεται μέσω των καταχωριστών οι οποίοι είναι οι εξής:

- EEPROM Address Register
- EEPROM Data Register

Σημειώνεται ότι ο προγραμματισμός της EEPROM δεν μπορεί να πραγματοποιηθεί όταν την ίδια στιγμή η CPU κάνει εγγραφή στη FLASH.

▼ ATmega 128L Pinout / Block Diagram

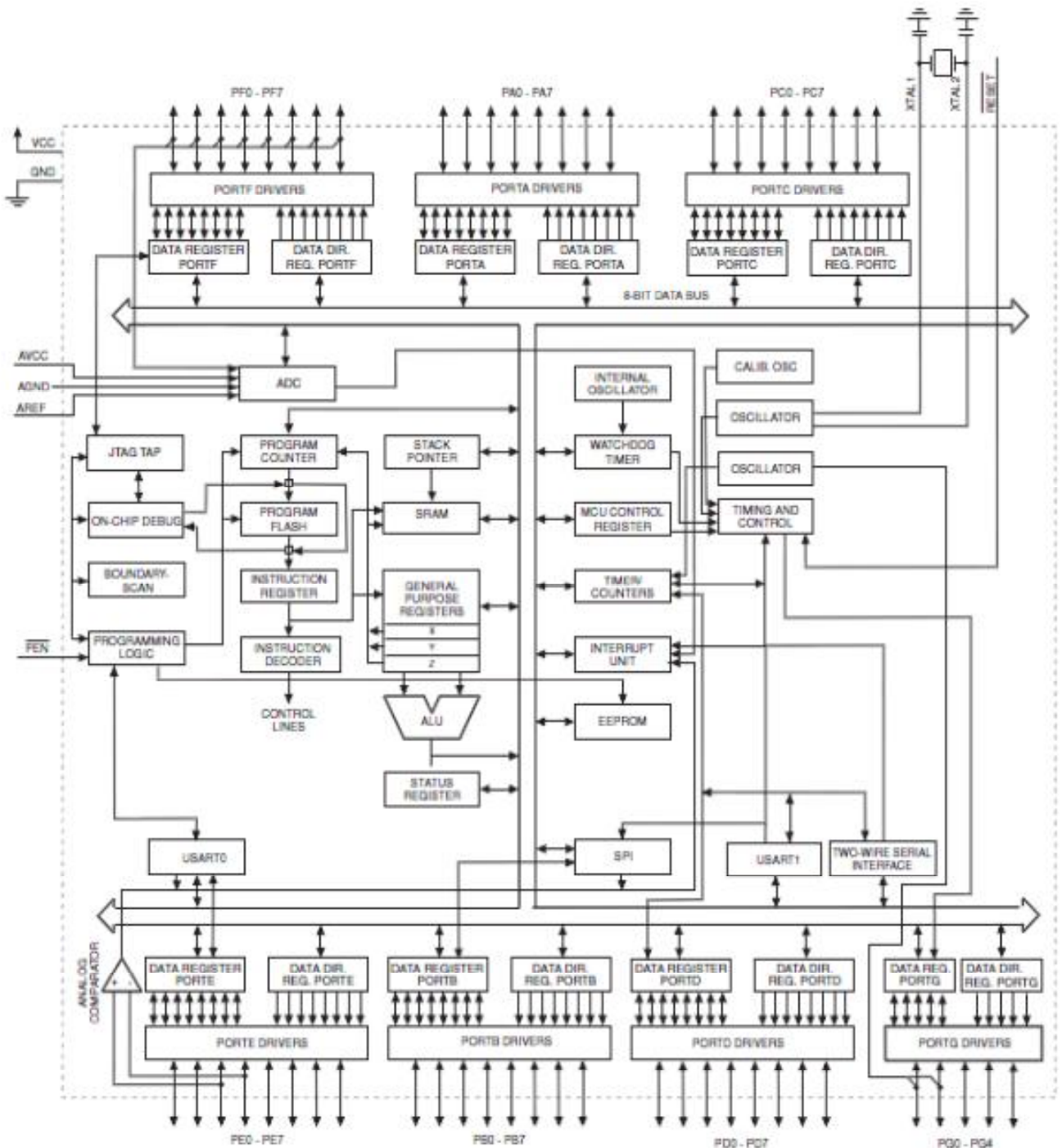
Στο Σχήμα 4.3. απεικονίζονται τα pins από τα οποία αποτελείται ο ATmega 128L.



Σχήμα 4.3: Pins του ATmega 128L microcontroller

[Πηγή: <http://www.xbow.com>]

Στο Σχήμα 4.4 απεικονίζεται το block διάγραμμα του ATmega 128L.



Σχήμα 4.4: Block διάγραμμα του ATmega 128L microcontroller

[Πηγή: <http://www.xbow.com>]

Για την κατανόηση των όσων απεικονίζονται στα σχήματα του εδαφίου δίνεται η επεξήγηση των συντομογραφιών με στόχο την πλήρη κατανόηση τους. Οι βασικότερες συντομογραφίες είναι οι κάτωθι:

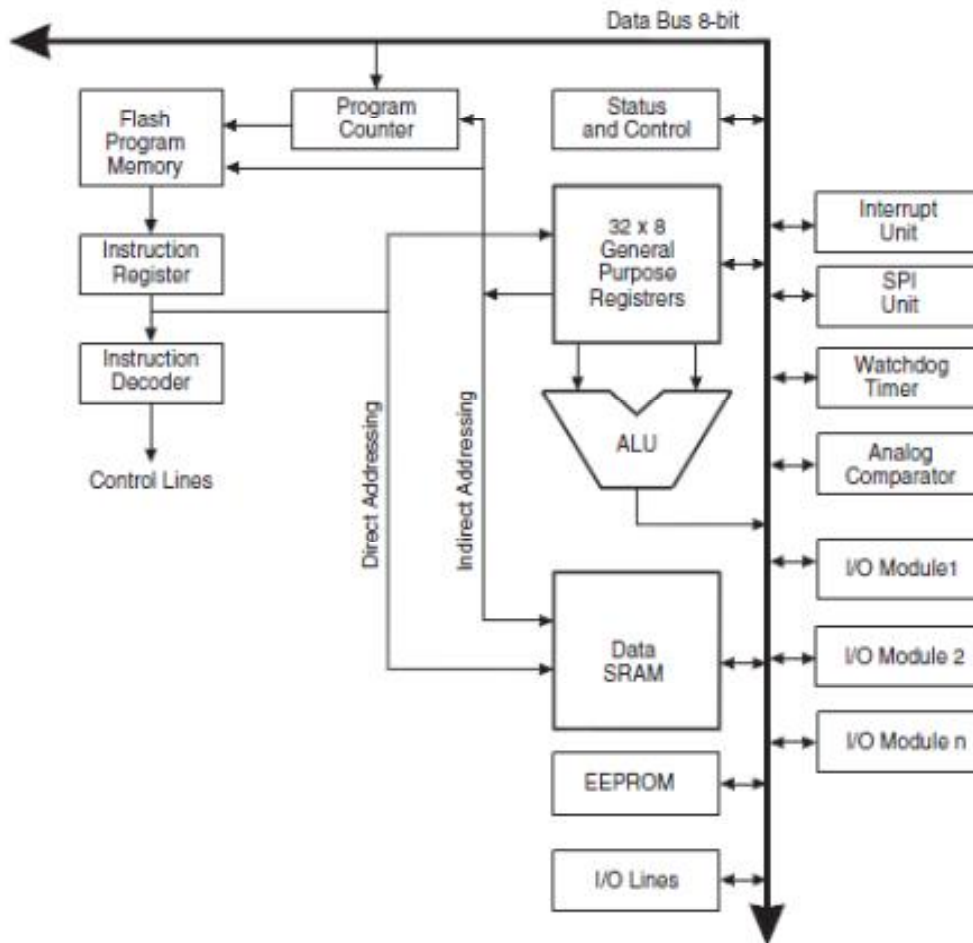
- VCC: Τροφοδοσία (2.7- 5.5 volts)
- GND: Γείωση
- RESET: Αρχικοποίηση του μC
- XTAL1: Είσοδος για την παροχή συχνότητας στη γεννήτρια συχνοτήτων του μC
- XTAL2: Έξοδος στη γεννήτρια συχνοτήτων του μC
- AVCC: Τροφοδοσία του F port και του A/D Converter
- ARFF: Pin αναφοράς του A/D Converter
- PEN: Pin ενεργοποίησης του SPI serial programming mode. Σε κανονική λειτουργία έχει την λογική 1. Για την ενεργοποίηση του serial programming mode, θα πρέπει κατά την διάρκεια reset του μC να πάρει την τιμή 1.
- I/O Ports: ATmega microprocessor αποτελείται από 7 ports διπλής κατεύθυνσης των 8 bits: port A(PA7...PA0), port B(PB7..PB0), port C(PC7..PC0), port D(PD7..PD0), port E(PE7..PE0) port F(PF7..PF0) και port G(PG7..PG0).

▼ ATmega AVR CPU Core

Οι κύριες λειτουργίες του CPU core είναι οι εξής:

- Η δυνατότητα πρόσβασης σε διαφορές μνήμες του μC (SRAM, EEPROM, FLASH)
- Η εκτέλεση αριθμητικών και λογικών πράξεων
- Ο έλεγχος των περιφερειακών συσκευών
- Ο χειρισμός των interrupts.

Στην συνέχεια ακολουθεί το block διάγραμμα του CPU core, το οποίο είναι βασισμένο στην τεχνολογία του AVR.



Σχήμα 4.5: Ο πυρήνας AVR του ATmega 129L

[Πηγή: <http://www.xbow.com>]

Με σκοπό την υψηλότερη απόδοση αλλά και την δυνατότητα εκτέλεσης εντολών ταυτόχρονα η AVR χρησιμοποιεί διαφορετικές μνήμες και διαύλους δεδομένων και προγραμματισμού. Οι εντολές που βρίσκονται στην Flash memory εκτελούνται κατά σειρά δηλαδή όταν η μια εντολή εκτελείται η άλλη είναι σε σειρά και είναι έτοιμη για εκτέλεση. Αυτή η μέθοδος ονομάζεται single level pipeline, και οι εντολές εκτελούνται σε κάθε κύκλο ρολογιού.

Η διάταξη των 32x8 bit registers χρησιμοποιείται για την λειτουργία της ALU. Έχει σαν είσοδο δύο γραμμές δεδομένων, οι οποίες προέρχονται από την διάταξη των registers. Το αποτέλεσμα το οποίο παράγεται στην ALU επιστρέφεται ξανά πίσω στους registers. Οι έξι από τους 32 registers έχουν την δυνατότητα να χρησιμοποιηθούν ως τρεις 16-bit έμμεσους address register pointers (X Y Z registers) για τον διαχωρισμό των διευθύνσεων των

δεδομένων. Επιπλέον, ένας από τους τρεις register pointer μπορεί να χρησιμοποιηθεί και ως address pointer για την FLASH memory.

Η ALU έχει την δυνατότητα να εκτελέσει αριθμητικές και λογικές πράξεις. Οι πράξεις αυτές μπορούν να πραγματοποιηθούν ανάμεσα σε δύο registers ή σε έναν register και μια μεταβλητή. Ο status register είναι αυτός που παίρνει την τιμή του αποτελέσματος, η οποία έχει προκύψει από μια πράξη συνήθως αριθμητική της ALU. Οι εντολές που μπορεί να υποστηρίξει η AVR έχει μήκος 16 bit. Η εκτέλεση ενός προγράμματος μπορεί να περιέχει εντολές jump ή call, με τις οποίες μπορεί να διαπεράσει όλο το μέγεθος της μνήμης FLASH.

Κάθε διεύθυνση μνήμης ενός προγράμματος μπορεί να περιέχει εντολή μήκους 16 ή 32 bit. Κατά την διάρκεια κάποιων interrupts ι subroutine calls, η διεύθυνση που βρίσκεται στον program counter (PC) αποθηκεύεται στον Stack Pointer (SP) ο οποίος βρίσκεται στην SRAM. το μέγεθος του σωρού περιορίζεται από το μέγεθος που έχει μνήμη SRAM. Σημειώνεται ότι όλα τα προγράμματα αρχικοποιούν τον Stack Pointer κατά την χρονική στιγμή του reset πριν εκτελεστούν τα πιθανά interrupts ή subroutines. Ο Stack Pointer είναι κατάλληλος για εγγραφή και ανάγνωση και χρησιμοποιείται από την μνήμη I/O μνήμη.

Η μνήμη FLASH χωρίζεται σε δύο τομείς, όπου κάθε ένας από αυτούς είναι κατάλληλος για εγγραφή και ανάγνωση. Οι τομείς αυτοί είναι οι κάτωθι:

- Boot Program
- Application Program

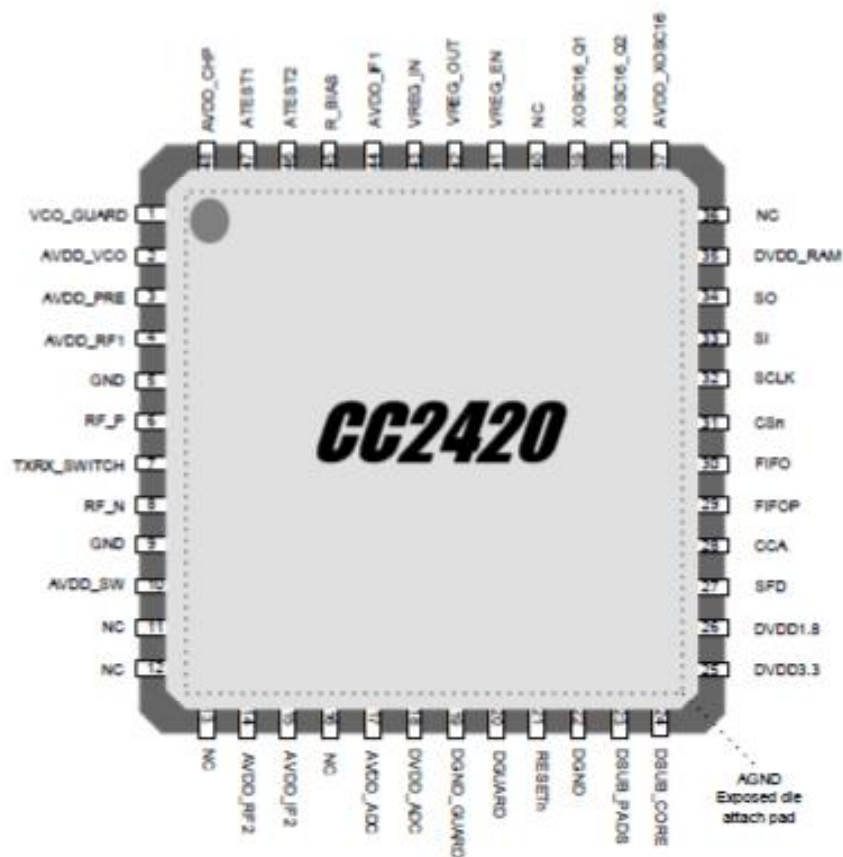
Το interrupt SPM έχει την δυνατότητα να κάνει εγγραφή στο τομέα του Application Program αφού πρώτα καθορίζεται πάλι το μέγεθος του Boot Program. Το interrupt unit ελέγχεται από ένα σύνολο control registers που βρίσκονται στην I/O μνήμη και ένα interrupt enable bit που βρίσκεται στον Status Register. Κάθε interrupt έχει την δική του θέση μέσα στο interrupt unit, και έχει προτεραιότητα ανάλογα σε ποία θέση βρίσκεται. Όσο χαμηλότερη διεύθυνση μνήμης έχει, τόσο μεγαλύτερη προτεραιότητα έχει (Crossbow, 2007).

4.2.2. RF transceiver CC2420

Ο RF transceiver είναι κατάλληλος για την ασύρματη επικοινωνία μεταξύ των motes και του σταθμού βάσης σε ένα Wireless Sensor Network. Το πρωτόκολλο επικοινωνία είναι IEEE 802.15.4 στα 2.4 GHz (ZigBee). Η υποστήριξη που παρέχει ήτοι:

- Packet Handling → διαχείριση data packets
- Data Buffering → προσωρινή αποθήκευση δεδομένων
- Burst Transmissions → διαρκή μεταφορά δεδομένων
- Data Encryption → απόκρυψη δεδομένων
- Data Authentication → πιστοποίηση δεδομένων
- Link Quality Indication → ποιότητα σύνδεσης
- Packet Timing Information → συγχρονισμός data packets

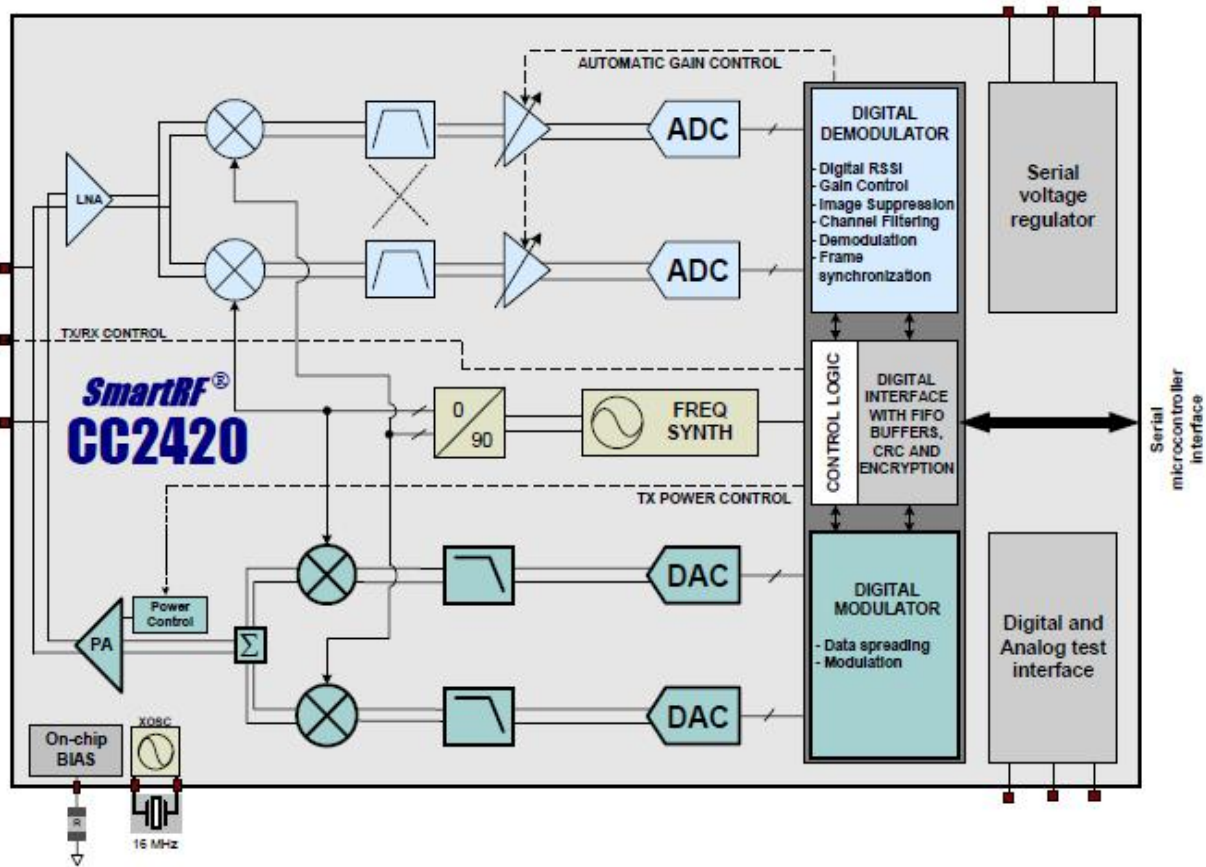
Τα προαναφερόμενα χαρακτηριστικά του CC2420 έχουν την δυνατότητα να μειώσουν τον φόρτο εκτέλεσης του μικροελεγκτή επιτρέποντας την χρήση του πομποδέκτη με μικροελεγκτές μικρότερης ισχύος. Η αλληλεπίδραση ανάμεσα στον RF transceiver και στον microcontroller γίνεται μέσω του Serial Peripheral Interface bus (SPI). Σημειώνεται ότι οι πληροφορίες του εδαφίου αυτό έχουν προέλθει από <http://www.ti.com/product/CC2420>.



Σχήμα 4.6: Ο RF transceiver CC2420

[Πηγή: <http://www.chipcon.com>]

Ακολουθεί το Σχήμα 4.7. όπου είναι η απεικόνιση ενός απλοποιημένου Block διάγραμμα του CC2420



Σχήμα 4.7: Block διάγραμμα του CC2420 RF transceiver

[Πηγή: <http://www.chipcon.com>]

▼ CC2420 Receiver

Το CC2420 διαθέτει ένα δέκτη low -IF receive. Το λαμβανόμενο σήμα RF ενισχύεται από τον ενισχυτή χαμηλού θορύβου (low – noise amplifier) ο οποίος συμβολίζεται LNA. Μέσω της O- QPSK αναλύεται σε άρτιες και περιττές δυαδικές ακολουθίες (I και Q) με την ενδιάμεση συχνότητα (IF). Στο IF (2 MHz), το συγκρότημα σήμα I / Q φιλτράρεται και ενισχύεται, και τότε ψηφιοποιηθεί από τα κέντρα διοικητικής κράτησης. Ο αυτόματος έλεγχος κέρδους, ο συγχρονισμός Bytes και ο συσχετισμός συμβόλων εκτελείται ψηφιακά.

Όταν το SFD γίνεται High, αυτό δείχνει ότι η αρχή της οριοθέτησης του πλαισίου έχει ανιχνευθεί. Το CC2420 αποθηκεύει προσωρινά το ληφθέντα δεδομένα σε μια FIFO ακολουθία 128 bytes λαμβάνετε. Ο χρήστης μπορεί να διαβάσει την FIFO μέσω μιας διεπαφή

SPI. Το CRC (cyclic redundancy check) επαληθεύεται στο hardware. Το RSSI (Receive Signal Strength Indicator) και συσχετισμός των τιμών επισυνάπτεται στο πλαίσιο.

▼ CC2420 Transmitter

Ο CC2420Transmitter βασίζεται στην άμεση και προσωρινή αποθήκευση των δεδομένων. Τα δεδομένα είναι ρυθμισμένα σε μία ακολουθία 128 bytes, η οποία είναι ξεχωριστή από τον δέκτη. Οι χαρακτήρες αρχής του πλαισίου που είναι για μετάδοση παράγεται από το hardware. Η ακολουθία δεδομένων μαζί με τους χαρακτήρες αρχής και τέλους περνά από τους μετατροπείς, οι οποίοι ονομάζονται DACs και πραγματοποιούν την ψηφιοποίηση. Έπειτα ένα αναλογικό LP φίλτρο περνά το σήμα στον O-QPSK διαμορφωτή (I και Q) και το παραγόμενο RF σήμα οδηγείται στον τελικό ενισχυτή (PA power amplifier). Έτσι μέσω του τελικού ενισχυτή το σήμα οδηγείται στην κεραία και μεταδίδεται μέσω του πρωτόκολλου IEEE 802.15.4.

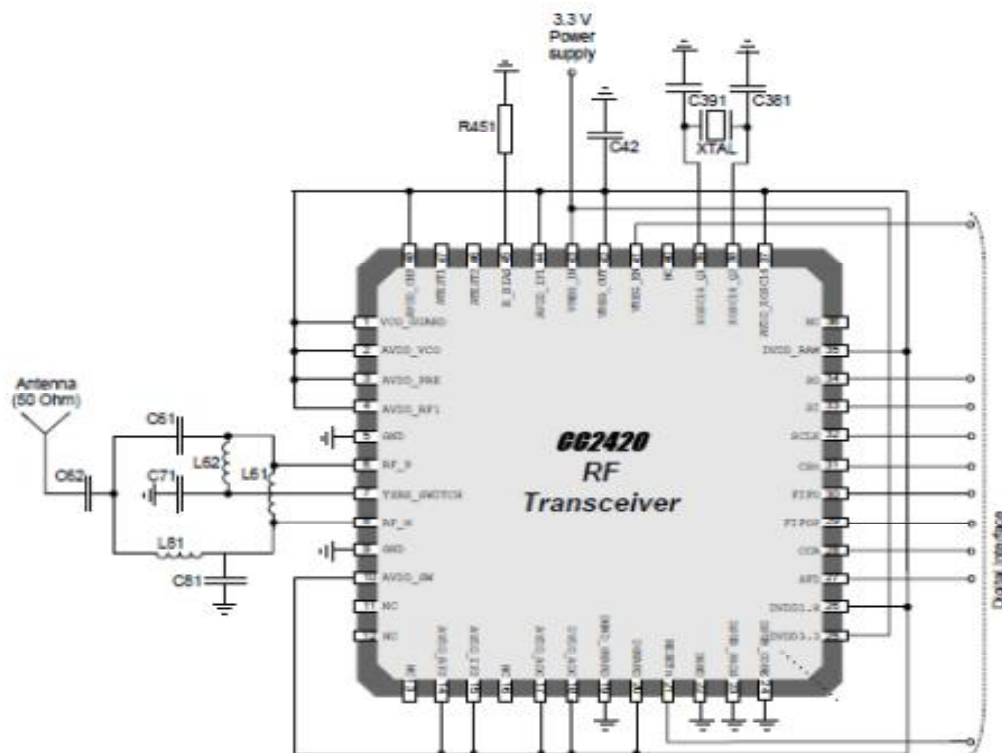
Η σύνδεση ανάμεσα στον PA ενισχυτή του πομπού και του LNA ενισχυτή του δέκτη, πραγματοποιείται συνδέοντας το TXRX_SWITCH και των RF_P και RF_N μέσω μιας εξωτερικής διαδρομής DC. Η γεννήτρια συχνοτήτων αποτελείται από ένα LC VCO (voltage controlled oscillator) και έναν Phase Splitter 90° για την παραγωγή των I και Q διαμόρφωσης ή αποδιαμόρφωσης. Το VCO παράγει συχνότητες στην περιοχή φάσματος 4800Hz-4966MHz, οι οποίες διαιρούνται στο μισό όταν χωριστούν σε I και Q.

Ο ταλαντωτής συνδέεται στο XOSC16_Q1) και στο XOSC16_Q2, ο οποίος δίνει συγκεκριμένη συχνότητα στην γεννήτρια συχνοτήτων. Ένα PLL χρησιμοποιείται με σκοπό να συνδυάσει την φάση και την συχνότητα του εισερχόμενου σήματος με αυτά του επιθυμητού. Το block control logic (Σχήμα 4.8) χρησιμοποιείται για τα εξής:

- Χειρισμό των frames
- Προσωρινή αποθήκευση δεδομένων
- Αναγνώριση διευθύνσεων
- MAC security

Σημειώνεται ότι control logic επικοινωνεί με τον microcontroller μέσω του SPI interface τεσσάρων αγωγών.

Η προαναφερόμενη διάταξη χρησιμοποιείται για τον βέλτιστο τερματισμό του RF σήματος μέσω μιας κεραίας single-ended των 50 Ωhm. Η αντίσταση R451 στο R_BIAS χρησιμοποιείται για την ρύθμιση της τάσης πόλωσης του chip. Ο τρόπος σύνδεσης του CC2420 με τα εξωτερικά στοιχεία καταγράφεται στον Πίνακα 4.1.



Σχήμα 4.8: Κύκλωμα λειτουργίας του CC2420 RF transceiver

[Πηγή: <http://www.chipcon.com>]

Item	Single ended output, transmission line balun	Single ended output, discrete balun	Differential antenna
C42	10 μ F, 0.5 Ω < ESR < 5 Ω	10 μ F, 0.5 Ω < ESR < 5 Ω	10 μ F, 0.5 Ω < ESR < 5 Ω
C61	Not used	0.5 pF, +/- 0.25pF, NP0, 0402	Not used
C62	Not used	5.6 pF, +/- 0.25pF, NP0, 0402	Not used
C71	Not used	5.6 pF, 10%, X5R, 0402	Not used
C81	5.6 pF, +/- 0.25pF, NP0, 0402	0.5 pF, +/- 0.25pF, NP0, 0402	Not used
C381	27 pF, 5%, NP0, 0402	27 pF, 5%, NP0, 0402	27 pF, 5%, NP0, 0402
C391	27 pF, 5%, NP0, 0402	27 pF, 5%, NP0, 0402	27 pF, 5%, NP0, 0402
L61	8.2 nH, 5%, Monolithic/multilayer, 0402	7.5 nH, 5%, Monolithic/multilayer, 0402	27 nH, 5%, Monolithic/multilayer, 0402
L62	Not used	5.6 nH, 5%, Monolithic/multilayer, 0402	Not used
L71	22 nH, 5%, Monolithic/multilayer, 0402	Not used	12 nH, 5%, Monolithic/multilayer, 0402
L81	1.8 nH, +/- 0.3nH, Monolithic/multilayer, 0402	7.5 nH, 5%, Monolithic/multilayer, 0402	Not used
R451	43 k Ω , 1%, 0402	43 k Ω , 1%, 0402	43 k Ω , 1%, 0402
XTAL	16 MHz crystal, 16 pF load (C _L), ESR < 60 Ω	16 MHz crystal, 16 pF load (C _L), ESR < 60 Ω	16 MHz crystal, 16 pF load (C _L), ESR < 60 Ω

Πίνακας 4.1. Οι τιμές των στοιχείων του CC2420 RF transceiver

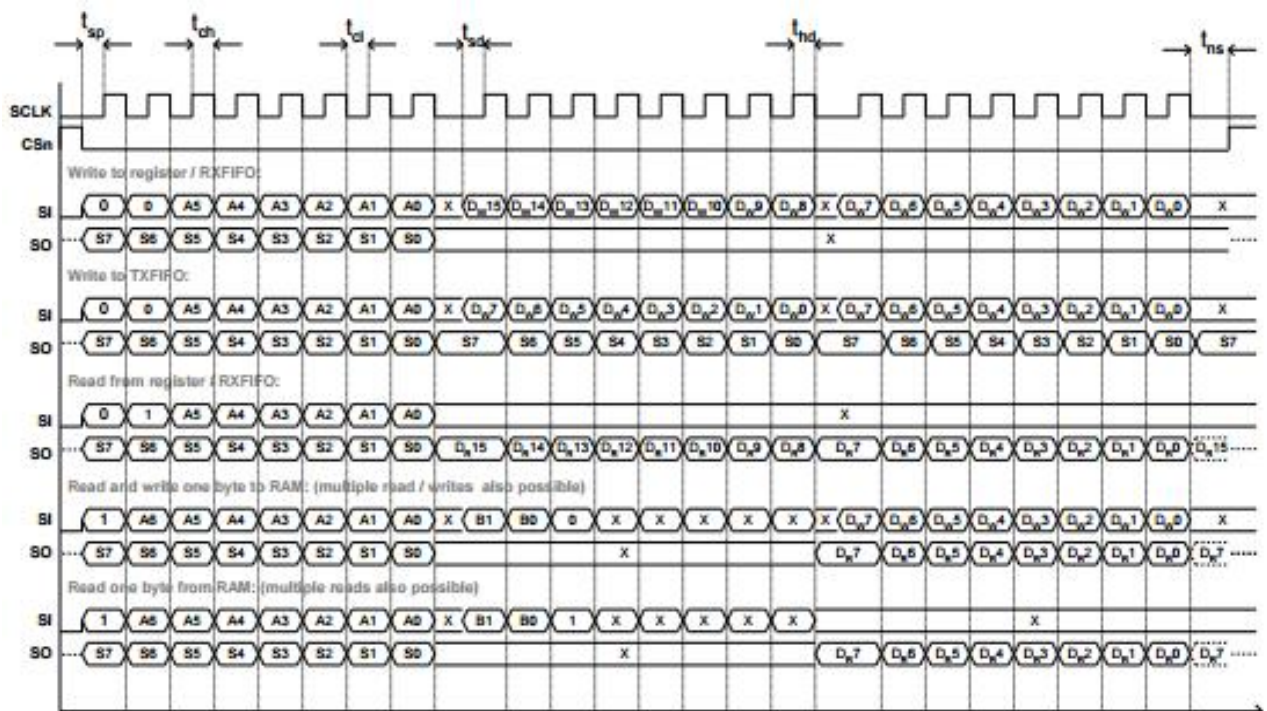
[Πηγή: <http://www.ti.com>]

▼ Serial Configuration και Data Interface - S.P.I.

• Πρόσβαση καταχωρήσεων

Το configuration του CC2420 έχει την δυνατότητα να πραγματοποιηθεί μέσω των pins SI, SO, SCLK και CSn που βρίσκεται στην δεξιά πλευρά. Αυτό μπορεί να παρατηρηθεί και από το Σχήμα 4.8. Ωστόσο, για το configuration χρειάζονται 33 των 16- bits registers που αφορούν την διαχείριση (conf) και την κατάσταση του (status) και 15 registers για αποθήκευση και εκτέλεση εντολών. Επιπλέον, από τα ίδια pins μπορεί να πραγματοποιηθεί ανάγνωση και εγγραφή προσωρινών δεδομένων με το MSB να είναι το αρχικό στην μετάδοση. Τα δεδομένα αυτά μπορούν να μεταφερθούν μέσω των 2 registers των 8 bit, εκ των οποίων ο ένας είναι για λαμβάνει την FIFO ακολουθία και ο άλλος για την εκπέμπει την FIFO ακολουθία.

Κάθε ένας από τα 50 μητρώα – καταχωρητές ελέγχεται από μια 6 -bit διεύθυνση. Για την ληφθείσα RXFIFO ακολουθίας που πρόκειται για την πρόσβαση του κάθε καταχωρητή στέλνονται 24 bits μέσω του SI pin. Το RAM/Register bit πρέπει να μηδενίζεται για την πρόσβαση του κάθε καταχωρητή. Μέσω του Read/Write bit επιλέγεται αν θα γίνει εγγραφή ή ανάγνωση του καταχωρητή. Σημειώνεται ότι το 0 είναι για εγγραφή και το 1 για ανάγνωση.



Σχήμα 4.9: Τρόπος πρόσβασης RXFIFO, TXFIFO και μνήμης RAM

[Πηγή: <http://www.ti.com>]

Το bit που στέλνεται πρώτο είναι το RAM/Register bit ακολουθούμενο από το Read/Write bit. Τα απομείναντα 6 bits είναι αυτά που αποτελούν την διεύθυνση. Επιπλέον, τα bits δεδομένων μεταφέρονται στέλνοντας το MSB αρχικά. Όμως με την εγγραφή την εκπεμπόμενη TXFIFO ακολουθίας στο SO pin βρίσκεται το status byte για κάθε data byte του SI pin.

Η ανάγνωση της ληφθείσας RXFIFO ακολουθίας έχει την δυνατότητα να διαβαστεί από τον μC, όπου είναι στη σειρά τρία, μέσω του SO pin αφού είναι η έξοδος του CC2420. Το Read/Write bit γίνεται 1, δηλαδή λογικό με σκοπό να ξεκινήσει η ανάγνωση. Το CC2420 λειτουργεί με συγκεκριμένη διαδικασία η οποία είναι η εξής, αρχικά στέλνει την διεύθυνση του καταχωρητή, ο οποίος είναι συγκεκριμένος κάθε φορά και έπειτα τα 16 bit δεδομένων. Αφού πραγματοποιηθεί και η μεταφορά του D0, ακολουθεί η αποθήκευση της ακολουθίας στον configuration register.

Θα πρέπει να σημειωθεί ότι ο τρόπος πρόσβασης των καταχωρητών στην υλοποίηση των εντολών παραμένει ίδιο με την διαφορά όμως ότι δεν πραγματοποιείται μεταφορά δεδομένων. Το RAM/Register bit και το Read/Write bit είναι 0 ενώ τα υπόλοιπα αφορούν την διεύθυνση τους με το εύρος τιμών 0x00-0x0E.

- *Πρόσβαση μνήμης RAM*

Η εσωτερική μνήμη RAM του CC420 μεγέθους 368 bytes(0x00-0x16F) μπορεί να προσπελαστεί μέσω του SPI interface. Η εσωτερική μνήμη RAM αποτελείται από τρία (3) banks, τα οποία είναι τα ακόλουθα:

- TXFIFO – bank 0
- RXFIFO – bank1
- Security – bank 2

Το μέγεθος που έχουν τα FIFO banks είναι 128 bytes το κάθε ένα από αυτά, όμως το security έχει μέγεθος 112 bytes. Ο αναλυτικός καταμερισμός την εσωτερικής μνήμης RAM καταγράφεται στο Πίνακα 4.2.

Επιπλέον, δεδομένα μπορούν να είτε να εγγραφούν είτε να διαβαστούν στέλνοντας την διεύθυνση μόνο μια φορά. Στην συνέχεια αυξάνονται κατά 1 από το hardware. Επιπλέον, η εγγραφή και ανάγνωση την μνήμης RAM πραγματοποιείται κατά 1 Byte κάθε φορά σε αντίθεση με την πρόσβαση των καταχωρητών που χρειάζεται 2 Bytes κάθε φορά, αυτό διαφαίνεται στην τέταρτη σειρά του Σχήματος 4.9.

Address	Byte Ordering	Name	Description
0x16F – 0x16C	-	-	Not used
0x16B – 0x16A	MSB LSB	SHORTADR	16-bit Short address, used for address recognition.
0x169 – 0x168	MSB LSB	PANID	16-bit PAN identifier, used for address recognition.
0x167 – 0x160	MSB LSB	IEEEADR	64-bit IEEE address of current node, used for address recognition.
0x15F – 0x150	MSB LSB	CBCSTATE	Temporary storage for CBC-MAC calculations
0x14F – 0x140	MSB (Flags) LSB	TXNONCE / TXCTR	Transmitter nonce for in-line authentication and transmitter counter for in-line encryption.
0x13F – 0x130	MSB LSB	KEY1	Encryption key 1
0x12F – 0x120	MSB LSB	SABUF	Stand-alone encryption buffer, for plaintext input and ciphertext output
0x11F – 0x110	MSB (Flags) LSB	RXNONCE / RXCTR	Receiver nonce for in-line authentication or receiver counter for in-line decryption.
0x10F – 0x100	MSB LSB	KEY0	Encryption key 0
0x0FF – 0x080	MSB LSB	RXFIFO	128 bytes receive FIFO
0x07F – 0x000	MSB LSB	TXFIFO	128 bytes transmit FIFO

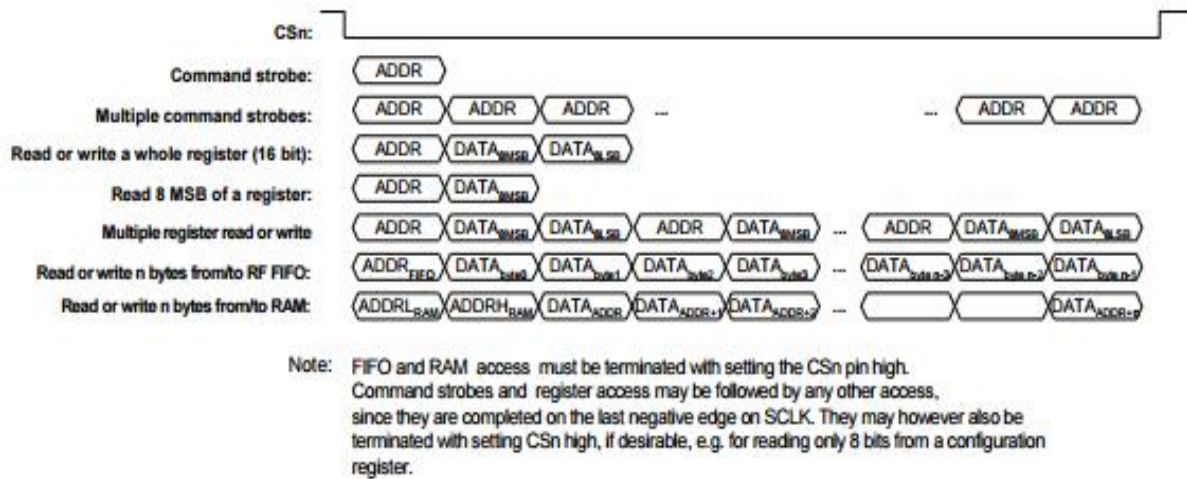
Πίνακας 4.2: Καταμερισμός της μνήμης RAM του CC2420.

[Πηγή: <http://www.ti.com>]

Το RAM/Register bit κα πρέπει να είναι 1 για να υπάρξει η δυνατότητα να ενεργοποιηθεί η πρόσβαση της μνήμης RAM. Το μέγεθος των διευθύνσεων είναι 9 bits. Τα B0:1 επιλέγουν κάποιο από τα banks της RAM, ενώ τα A6:0 bits αντιστοιχούν στην διεύθυνση του ήδη επιλεγμένου bank. Το B3 bit αντιστοιχεί στο Read/Write bit όπου το 0 read και το 1 write, ενώ τα B4:7 δεν λαμβάνονται υπόψη.

Θα πρέπει να σημειωθεί ότι κατά την εγγραφή της μνήμης RAM τα δεδομένα στέλνονται από τον μC μέσω του SI pin αμέσως μετά από δεύτερο byte της διεύθυνσης. Όμως παρόμοιος τρόπος ακολουθείται κατά την αποστολή των δεδομένων όταν πραγματοποιείται η ανάγνωση στον μC μέσω του SO pin.

Στην συνέχεια παρατίθεται το Σχήμα 4.10. όπου απεικονίζονται όλες οι περιπτώσεις μεταφοράς δεδομένων μέσω του SPI interface σε μια εφαρμογή, μπορεί να αφορούν είτε καταχωρητές είτε μνήμη RAM.



Σχήμα 4.10: Περιπτώσεις μεταφοράς δεδομένων για καταχωρήσεις και μνήμη RAM

[Πηγή: <http://www.ti.com>]

· *Πρόσβαση FIFO*

Η πρόσβαση στην RAM δεν χρησιμοποιείται για εγγραφή FIFO ακολουθιών. Διότι ο FIFO counter δεν μπορεί να μηδενιστεί. Η TXFIFO και οι RXFIFO ακολουθίες μπορούν να έχουν πρόσβαση μέσω του TXFIFO (0x3E) και RXFIFO (0x3F) καταχωρητών αντίστοιχα.

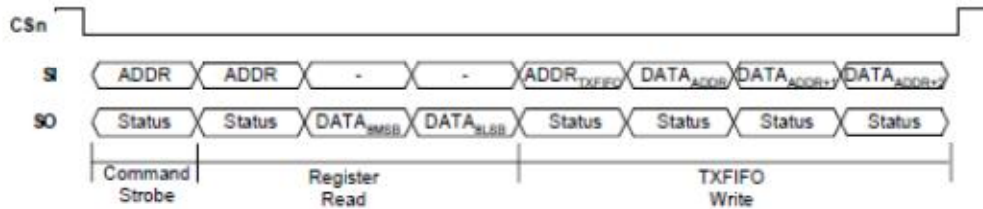
Οι TXFIFO είναι μόνο για εγγραφή ωστόσο δίνεται η δυνατότητα να διαβαστούν κατά 1 byte την φορά χρησιμοποιώντας την μνήμη RAM. Οι RXFIFO είναι εγγραφής και ανάγνωσης, όπου η εγγραφή πραγματοποιείται μόνο σε περιπτώσεις που η RXFIFO χρησιμοποιείται για τους εξής λόγους :

- Απόκρυψη δεδομένων (decryption)
- Πιστοποίηση δεδομένων (authentication)

Όταν γράφετε το TXFIFO (pin SI) το status byte έχει έξοδο το SO για κάθε data byte. Αυτό θα μπορούσε να χρησιμοποιηθεί για την ανίχνευση στον TXFIFO καταχωρητή αν έχει πραγματοποιηθεί εγγραφή με μειωμένη ταχύτητα από αυτήν της ανάγνωσης. Η πρόσβαση των FIFO μπορεί να ολοκληρωθεί και να τερματιστεί όταν το CSn γίνει λογικό 1.

Το FIFO και FIFOP pins ανήκουν στο SPI και παρέχουν επίσης πρόσθετες πληροφορίες σχετικά με τα δεδομένα που λαμβάνουν και χρησιμοποιούνται από τον μC. Η πρόσβαση των καταχωρητών, της μνήμης RAM και των FIFO μπορεί να πραγματοποιηθεί με

συνεχή τρόπο χωρίς να γίνει το CSn λογικό 1. Σημειώστε ότι τα FIFO και FIFOP pins ισχύουν μόνο για το RXFIFO.



Σχήμα 4.11: Πρόσβαση των καταχωρητών της μνήμης RAM και των FIFO με συνεχή τρόπο χωρίς να γίνει το CSn λογικό 1.

[Πηγή: <http://www.ti.com>]

✓ Διασύνδεση Microcontroller με το CC240

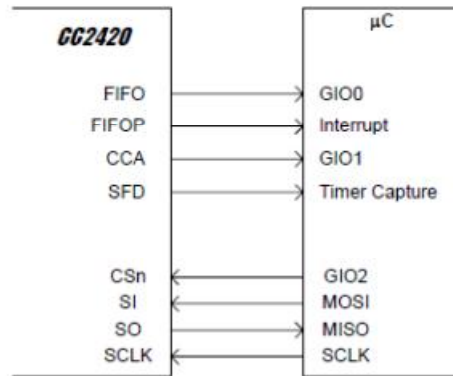
Όταν χρησιμοποιείται ένα τυπικό σύστημα, το CC2420 θα πρέπει να συνδεθεί με ένα μικροελεγκτή (μC). Ωστόσο, με την πραγματοποίηση αυτής της σύνδεσης θα πρέπει να ισχύουν οι απαραίτητες λειτουργίες, οι οποίες είναι οι κάτωθι:

- § Εγγραφή και ανάγνωση προσωρινών δεδομένων έσω του SPI interface
- § Αρχικοποίηση του καναλιού μέσω του CCA pin
- § Αλληλεπίδραση μέσω των FIFO και FIFOP status pins του CC2420 για τις εκπεμπόμενες και λαμβανόμενες FIFO ακολουθίες
- § Παροχή πληροφοριών χρονισμού μέσω του SFD pin στον μC.

Στην συνέχεια στο Σχήμα 4.12 παρουσιάζεται ο τρόπος σύνδεσης του CC2420 με έναν μC. Παρατηρείται λοιπόν ότι ο μC χρησιμοποιεί 4 I/O pins για το SPI interface (SI, SO, CSn, SCLK). Το SI αποτελεί την είσοδο του μC, τα υπόλοιπα είναι οι έξοδοι αυτού. Τα SI, SO, και SCLK pins έχουν την δυνατότητα να χρησιμοποιηθούν ταυτόχρονα από όλες τις συσκευές που υποστηρίζουν το SPI interface.

Το FIFOP pin χρησιμοποιείται για τα interrupts του μC όταν είτε έχει πραγματοποιηθεί η ολοκλήρωση της λήψης μιας ακολουθίας εισερχόμενης είτε όταν ο αριθμός των bytes της ακολουθίας αυτής είναι μεγαλύτερος από ένα το ανώτερο όριο που

έχει οριστεί. Το FIFO pin χρησιμοποιείται για τα δεδομένα της εισερχόμενης ακολουθίας. Το SFD pin συνδέεται με το αντίστοιχο timer capture pin του μC για το χρονισμό των εισερχόμενων/εξερχόμενων ακολουθιών κατά την λήψη/αποστολή δεδομένων αντίστοιχα.

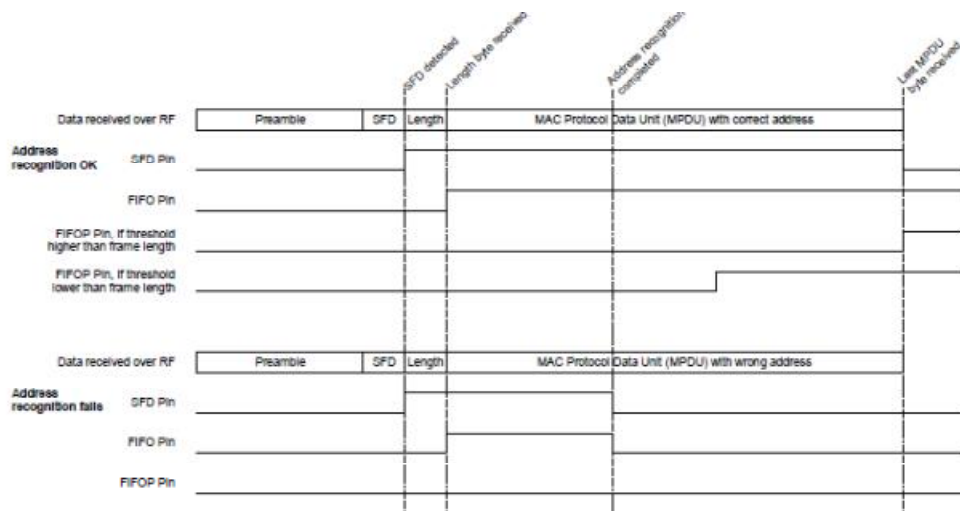


Σχήμα 4.12: Τρόπος διασύνδεσης του CC2420 με έναν microcontroller

[Πηγή: <http://www.ti.com>]

- *Λήψη δεδομένων – Receive mode*

Για την λήψη δεδομένων είναι απαραίτητη η αναγνώριση διεύθυνσης. Με την ολοκλήρωση της λήψης του Frame Delimiter από την λαμβανόμενη ακολουθία, το SFD pin γίνεται λογικό και ξεκινά η διαδικασία αναγνώρισης διεύθυνσης. Στην περίπτωση που είναι επιτυχής το SFD pin παραμένει λογικό 1 μέχρι την ολοκλήρωση της ακολουθίας δεδομένων. Στην αντίθετη περίπτωση επιτυχής το SFD pin γίνεται λογικό 0.



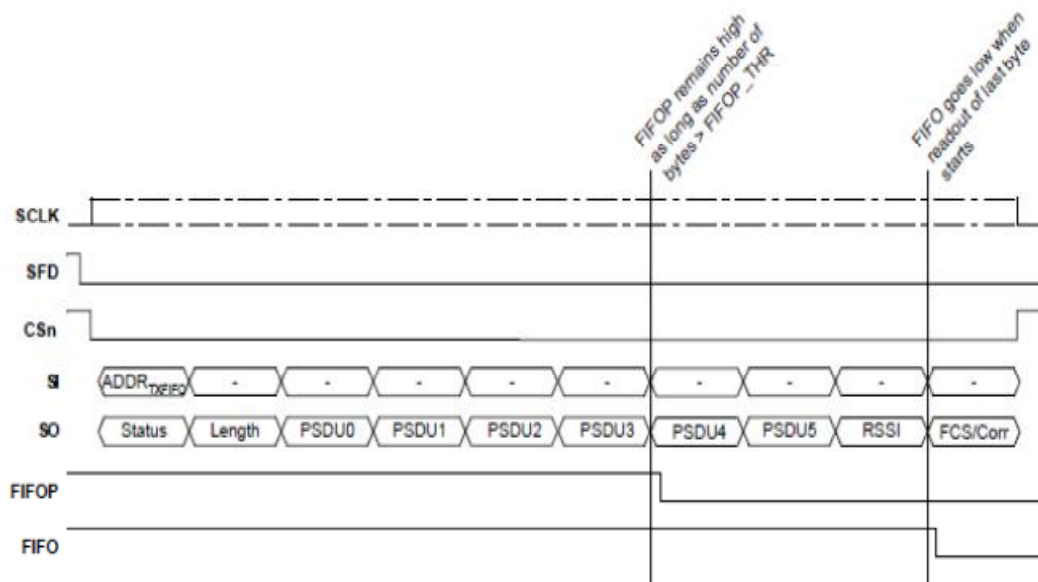
Σχήμα 4.13: Διαδικασία αναγνώρισης διεύθυνσης.

[Πηγή: <http://www.ti.com>]

Σύμφωνα με το Σχήμα 4.13 παρατηρείται ότι το FIFO pin γίνεται λογικό 1 όταν το RXFIFO μνήμη θα αποθηκεύσει την εισερχόμενη RXFIFO ακολουθία. Τονίζεται ότι παραμένει σε αυτή την κατάσταση μέχρι την στιγμή που η μνήμη θα μείνει κενή από την παρούσα ή από προηγούμενη ακολουθία. Το πρώτο byte που αποθηκεύεται στην RXFIFO είναι αυτό που αποτελεί το πεδίο της διεύθυνσης της αντίστοιχης λαμβανόμενης ακολουθίας.

Το FIFO pin γίνεται λογικό 1 σε δύο περιπτώσεις. Η πρώτη περίπτωση είναι όταν γίνεται η λήψη του τελευταίου byte μιας εισερχόμενης RXFIFO ακολουθίας και η δεύτερη όταν ο αριθμός των bytes που θα διαβαστούν είναι μικρότερος από το προκαθορισμένο όριο που έχει οριστεί στην IOCFG0.FIFOP_THR μεταβλητή. Θα πρέπει να σημειωθεί ότι παραμένει 1 μέχρι την στιγμή που ο αριθμός των bytes που θα είναι μεγαλύτερος από το προκαθορισμένο όριο που έχει οριστεί. Κατά την διαδικασία αναγνώρισης των διευθύνσεων η λαμβανόμενη ακολουθία δε διαβάζεται από την RXFIFO μέχρι που η διαδικασία φτάσει στο τέλος της.

Σε περίπτωση που υπάρξει που υπάρχει υπερφόρτωση στην RXFIFO μνήμη των 128 bytes γίνεται ενημέρωση του μC μέσω του FIFO pin παίρνοντας την τιμή 0, ενώ ταυτόχρονα το FIFOP pin είναι 1. Αποτέλεσμα αυτού είναι ότι σε αυτό το σημείο δεν μπορεί να υπάρξει υπερφόρτωση της μνήμης αφού τα δεδομένα που υπάρχουν σε αυτή προωθούνται για περαιτέρω επεξεργασία.

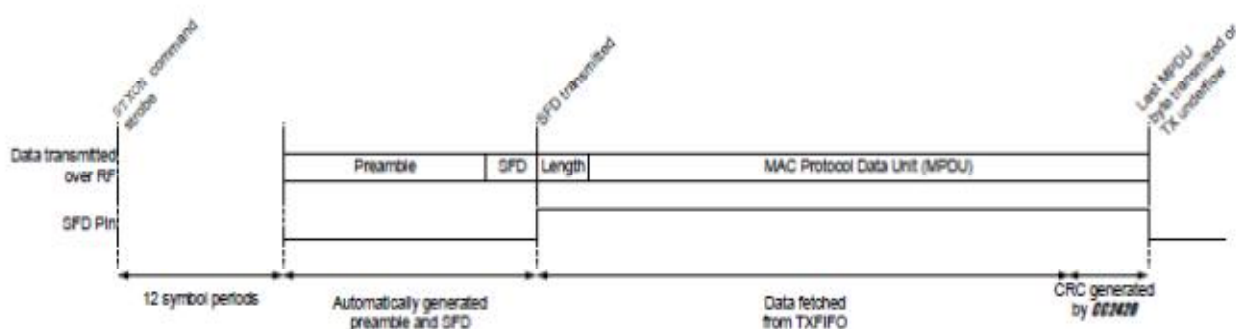


Σχήμα 4.14: Διαδικασία λήψης δεδομένων

[Πηγή: <http://www.ti.com>]

- *Αποστολή δεδομένων - Transmit mode*

Κατά την διάρκεια της αποστολής των δεδομένων , τα FIFO και FIFOP pins έχουν σχέση μόνο με την RXFIFO μνήμη. Το SFD pin γίνεται λογικό 1 κατά την διάρκεια της αποστολής της ακολουθίας (Σχήμα 4.15) και στην συνέχεια μετά την λήψη της ακολουθίας γίνεται παίρνει ξανά την τιμή 0. Σύμφωνα με όσα έχουν προαναφερθεί είναι προφανές ότι το SFD pin έχει την ίδια συμπεριφορά είτε πρόκειται για μετάδοση είτε για αποστολή δεδομένων με μία καθυστέρηση της τάξεως των 2μs λόγω των περιορισμών που παρουσιάζονται στο εύρος φάσματος του πομπού και του δέκτη.



Σχήμα 4.15: Διαδικασία αποστολής δεδομένων

[Πηγή: <http://www.ti.com>]

Θα πρέπει να σημειωθεί ότι οι πληροφορίες του εδαφίου αυτού έχουν προέλθει από <http://www.ti.com/product/CC2420>.

4.3. MODULE MTS400 SENSOR BOARD

Για την επιλογή του sensor board πρέπει να λαμβάνεται υπόψη η συμβατότητα του με τις προδιαγραφές του συστήματος που θα χρησιμοποιηθεί. Το module MTS400 συνδέεται με το module MICAZ MPR2600 μέσω υποδοχής των 51 pins. Οι δυνατότητες του συγκεκριμένου μοντέλου είναι ιδανικές για την καταγραφή ισχυρών περιβαλλοντικών φαινομένων. Έχουν την δυνατότητα καταγραφής των εξής:

- Θερμοκρασία
- Υγρασία
- Βαρομετρική πίεση
- Φωτισμό
- Επιτάχυνση στους άξονες x,y.

Οι δυνατότητες που προσφέρονται με την MTS400 επιτρέπει μια μεγάλη ποικιλία από εφαρμογές να αναπτυχθούν, από ένα απλό ασύρματο μετεωρολογικό σταθμό μέχρι ένα δίκτυο κόμβων περιβαλλοντικής παρακολούθησης. Η πλακέτα MTS400 χρησιμοποιεί την τελευταία γενιά αισθητήρων που είναι ενεργειακά αποδοτική. Αυτή η δυνατότητα παρέχει εκτεταμένη διάρκεια ζωής της μπαταρίας, οπότε προσδίδει ένα χαμηλό κόστος συντήρησης.



Εικόνα 4.2: MTS400 sensor board

[Πηγή: <http://www.memsic.com>]

Θα πρέπει να σημειωθεί ότι η πλακέτα MTS400 χρησιμοποιεί διαφορετικούς μικροεπεξεργαστές που είναι σχεδιασμένοι για να μετρούν συγκεκριμένα περιβαλλοντικά φαινόμενα (Κωνσταντίνου Γ., 2012).


4.3.1. Αισθητήρας θερμοκρασίας και υγρασίας

Ο Sensirion® SHT11 είναι ένας single-chip αισθητήρας υγρασίας και θερμοκρασίας με βαθμονομημένη ψηφιακή έξοδο. Διαθέτει εσωτερικό analog-to-digital μετατροπέα 14-bit και σειριακή θύρα. Η τροφοδοσία του αισθητήρα ενεργοποιείται μέσω ενός

προγραμματιζόμενου διακόπτη, όπως επίσης και τα σήματα της θύρας ελέγχου. Τα αναλογικά σήματα που λαμβάνονται για την θερμοκρασία και υγρασία, μετατρέπονται σε ψηφιακές μονάδες με τον analog-to-digital μετατροπέα (Κωνσταντίνου Γ., 2012).

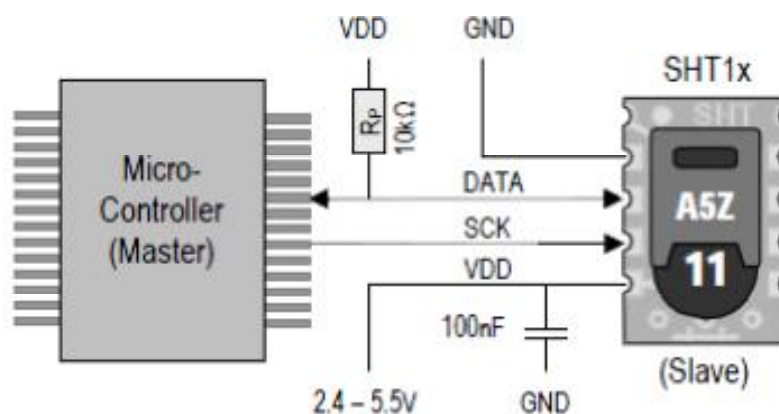
Ο Sensirion® SHT11 αποτελείται από 8 ακροδέκτες εκ των οποίων χρησιμοποιούνται οι 4, όπως απεικονίζεται στο Σχήμα 4.16.

Pin	Name	Comment
1	GND	Ground
2	DATA	Serial Data, bidirectional
3	SCK	Serial Clock, input only
4	VDD	Source Voltage
NC	NC	Must be left unconnected



Σχήμα 4.16: Pins του αισθητήρα θερμοκρασίας και υγρασίας SHT11
(Μανωλόπουλος Χ., 2010)

Όπως προαναφέρθηκε το ολοκληρωμένο SHT11 διαθέτει ένα A/D converter 14 bits, οποίος χρησιμεύει για την απόδοση της θερμοκρασίας (14 bits) και της υγρασίας (12 bits). Ο τρόπος διασύνδεσης με τον μC γίνεται μέσω του Two Wire Interface – TWI (Σχήμα 4.17) (Μανωλόπουλος Χ., 2010).



Σχήμα 4.17: Διασύνδεση αισθητήρα με τον microcontroller
(Μανωλόπουλος Χ., 2010)

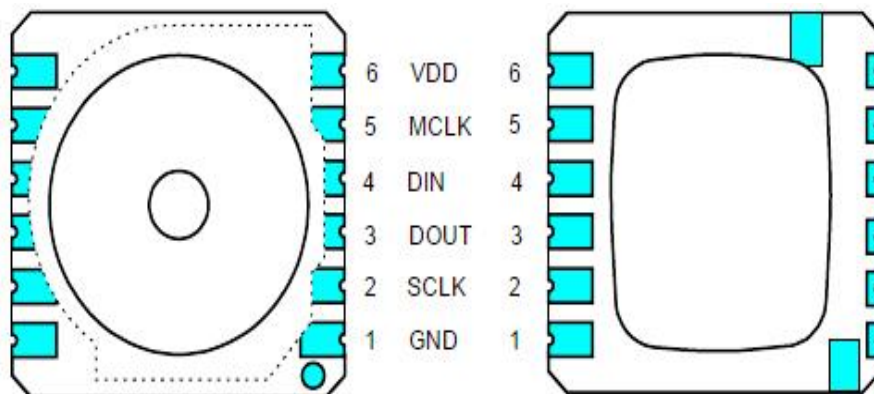
Στην συνέχεια ακολουθεί ο Πίνακας 4.3 όπου παρουσιάζει συνοπτικά τα τεχνικά χαρακτηριστικά του συγκεκριμένου αισθητήρα.

Τύπος Αισθητήρα	Sensirion SHT11	
Κανάλια	Υγρασία	Θερμοκρασία
Διακύμανση	20- 80 %	-40° C έως 100° C
Ακρίβεια	3.5% RH	± 0.5 - 2.5 ° C
Χρόνος απόκρισης	5 sec	8 sec
Εμβέλεια λειτουργίας	3.6 μέχρι 2.4 volts	
Θύρα	Ψηφιακή θύρα	

Πίνακας 4.3: Τεχνικά χαρακτηριστικά του Sensirion SHT11
(Μανωλόπουλος Χ., 2010)

4.3.2. Αισθητήρας πίεσης

Για την μέτρηση πίεσης χρησιμοποιείται αισθητήρας πίεσης MS5534A της εταιρίας Intersema, το οποίο έχει την δυνατότητα να μετρήσει και θερμοκρασία. Αποτελείται από 6 ακροδέκτες όπως αυτοί απεικονίζονται στο Σχήμα 4.18.



Σχήμα 4.18: Αισθητήρας πίεσης MS5534A της εταιρίας Intersema

[Πηγή: <http://www.datasheetlib.com>]

Για την διασύνδεση με τον μC χρησιμοποιούνται οι ακροδέκτες DIN, DOUT για την λήψη και την αποστολή δεδομένων. Ο χρονισμός αυτών πραγματοποιείται στα 500KHz χρησιμοποιώντας τον ακροδέκτη SCLK. Ο ακροδέκτης MCLK χρησιμοποιείται για τον συγχρονισμό ολόκληρου του συστήματος στα 32.768 KHz.

Pin Name	Pin	Type	Function
VDD	6	P	Positive Supply Voltage
MCLK	5	I	Master Clock (32.768kHz)
DIN	4	I	Data Input
DOUT	3	O	Data Output
SCLK	2	I	Serial Data Clock
GND	1	G	Ground

Πίνακας 4.4: Τα pins του αισθητήρα πίεσης και θερμοκρασίας Intersema MS5534A

[Πηγή: <http://www.datasheetlib.com>]

Στην συνέχεια ακολουθεί ο Πίνακας 4.5 όπου παρουσιάζει συνοπτικά τα τεχνικά χαρακτηριστικά του συγκεκριμένου αισθητήρα.

Τύπος Αισθητήρα	MS5534A – Intersema	
Κανάλια	Πίεση	Θερμοκρασία
Διακύμανση	300mbar έως 1100mbar.	-40° C έως 125° C
Ακρίβεια	± 0.5 mbar	$\pm 0.5 - 2.5$ ° C
Εμβέλεια λειτουργίας	2.2 έως 3.6 volts	
Θύρα	Ψηφιακή θύρα	

Πίνακας 4.5: Τεχνικά χαρακτηριστικά του MS5534A - Intersema

(Μανωλόπουλος X., 2010)

Ωστόσο, η πίεση με την θερμοκρασία παρουσιάζουν κάποιες διακυμάνσεις μεταξύ τους, για παράδειγμα η θερμοκρασία είναι στους 25° C τότε η πραγματική τιμή της πίεσης

κυμαίνεται μεταξύ των 750mbar έως 1100mbar. Στην συνέχεια ακολουθεί ο Πίνακας 4.6 με αποκλίσεις των τιμών πίεσης και ο Πίνακας 4.7 με τις αποκλίσεις των τιμών της θερμοκρασίας.

Parameter	Conditions	Min	Typ	Max	Unit	Note
Resolution			0.1		mbar	1
Absolute Pressure Accuracy	$p = 750 \dots 1100\text{mbar}$ $T_a = 25^\circ\text{C}$	-1.5		+1.5	mbar	2
Relative Pressure Accuracy	$p = 750 \dots 1100\text{mbar}$ $T_a = 25^\circ\text{C}$	-0.5		+0.5	mbar	3
Maximum Error over Temperature	$T_a = -10 \dots +60^\circ\text{C}$ $p = \text{const.}$	-1		+1	mbar	4
		-1		+6	mbar	5
Long-term Stability	12 month		-1		mbar	6
Maximum Error over Supply Voltage	VDD = 2.2...3.6V	-1.5	0	+1.5	mbar/V	

Πίνακας 4.6: Αποκλίσεις τιμών στην μέτρηση πίεσης του MS5534A - Intersema

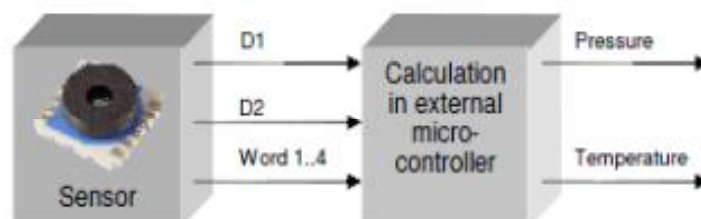
[Πηγή: <http://www.datasheetlib.com>]

Parameter	Conditions	Min	Typ	Max	Unit
Resolution		0.005		0.015	$^\circ\text{C}$
Accuracy	at reference temperature	-0.8		0.8	$^\circ\text{C}$
Maximum Error over Supply Voltage	VDD = 2.2...3.6V	-0.08		+0.08	$^\circ\text{C/V}$

Πίνακας 4.7: Αποκλίσεις τιμών στην μέτρηση θερμοκρασίας του MS5534A - Intersema

[Πηγή: <http://www.datasheetlib.com>]

Για την μετατροπή των τιμών πίεσης και της θερμοκρασίας από binary format σε πραγματικές τιμές στέλνονται μέσω του DOUT ακροδέκτη οι παράμετροι D1 και D2. Έχουν μήκος 16 bit και αφορούν την μέτρηση της πίεσης και της θερμοκρασίας. Οι 4 λέξεις μήκους 16 bit χρησιμοποιούνται για μεγαλύτερη ακρίβεια. Ο υπολογισμός των τιμών πίεσης και της θερμοκρασίας γίνεται από το μC και απεικονίζεται στο Σχήμα 4.19.



Σχήμα 4.19: Αποστολή δεδομένων στον μC για επεξεργασία

(Μανωλόπουλος X., 2010)

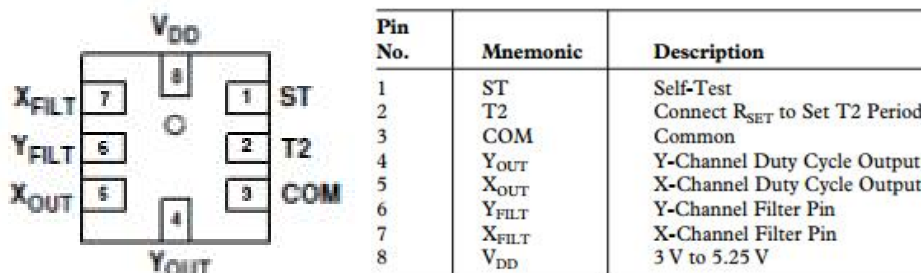
Στις 4 λέξεις μήκους 16 bit κρύβονται 6 συντελεστές οι οποίοι βοηθούν για την μεγαλύτερη ακρίβεια στις τιμές πίεσης και θερμοκρασίας. Η ακρίβεια που προσφέρεται μπορεί να διακριθεί από το Σχήμα 4.20. ωστόσο, είναι απαραίτητο να σημειωθεί ότι ο συντελεστής C5 προκύπτει από τα δεδομένα των δύο πρώτων λέξεων με συγκεκριμένη σειρά, ενώ με παρόμοιο τρόπο προκύπτει και ο συντελεστής C2 μέσω των δύο τελευταίων λέξεων (Μανωλόπουλος X., 2010 ; Κωνσταντίνου Γ., 2012).



Σχήμα 4.20: Προσδιορισμός της θέσης των συντελεστών C1-C6
(Μανωλόπουλος X., 2010)

4.3.3. Αισθητήρας μέτρησης δονήσεων στους X και Y άξονες

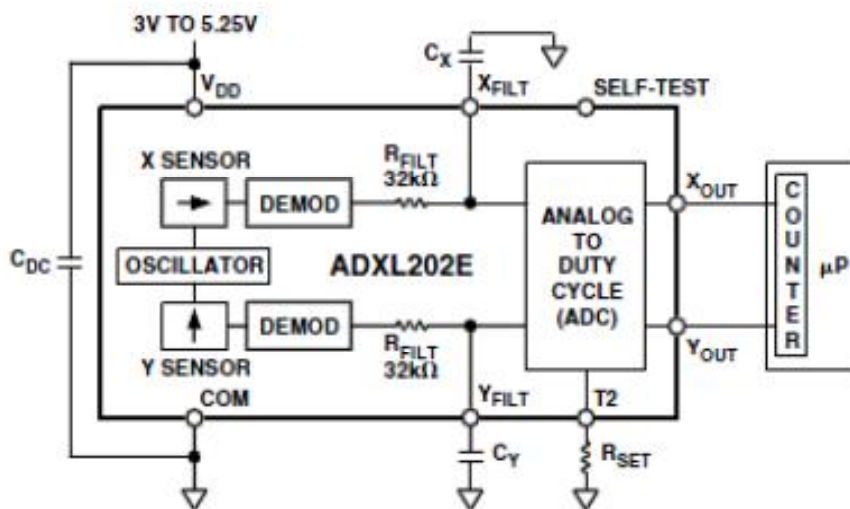
Για την υλοποίηση της μέτρησης της επιτάχυνσης (g) X και Y αξόνων χρησιμοποιείται το ADXL202JE της εταιρίας Analog Divices. Ο αισθητήρας αυτός αποτελείται από 8 ακροδέκτες (Σχήμα 4.21).



Σχήμα 4.21: Περιγραφή των pins τους αισθητήρα ADXL202JE

[Πηγή: <http://www.analog.com>]

Το ST pin είναι υπεύθυνο για τον σωστό έλεγχο της ορθής λειτουργίας του ολοκληρωμένου. Όταν πραγματοποιηθεί η σύνδεση με την τάση τροφοδοσίας VDD, μια ηλεκτροστατική δύναμη ασκείται εσωτερικά του ολοκληρωμένου κυκλώματος και μέσω της κίνησης που προκύπτει, φαίνεται αν λειτουργεί κανονικά. Σε κανονική λειτουργία ο ακροδέκτης είτε γειώνεται είτε δεν είναι συνδεδεμένος. Για κάθε άξονα ένα κύκλωμα εξόδου 14 bit μετατρέπει το αναλογικό σήμα σε ένα DCM (Duty Cycle Modulated) ψηφιακό σήμα, το οποίο στέλνεται μέσω των pins Xout και Yout σε ένα counter/port του μC για αποκωδικοποίηση.



Σχήμα 4.22: Block διάγραμμα του αισθητήρα ADXL202JE

[Πηγή: <http://www.analog.com>]

Το εύρος μέτρησης της επιτάχυνσης του αισθητήρα είναι $\pm 2g$. Στα X_{FILT} και Y_{FILT} pins συνδέονται πυκνωτές με στόχο την υλοποίηση LP φίλτρων για την μείωση του θορύβου, ο οποίος φέρνει απόκλιση στις πραγματικές τιμές από 0,8mg 5,7 και mg, ανάλογα με το bandwidth του φίλτρου. Το εύρος bandwidth μπορεί να είναι από 10Hz μέχρι 500Hz σύμφωνα με την χωρητικότητα του πυκνωτή που είναι συνδεδεμένος στους ακροδέκτες (Πίνακας 4.8).

Bandwidth	Capacitor Value
10 Hz	0.47 μF
50 Hz	0.10 μF
100 Hz	0.05 μF
200 Hz	0.027 μF
500 Hz	0.01 μF
5 kHz	0.001 μF

Πίνακας 4.8: Καθορισμός του εύρους bandwidth μέσω των πυκνωτών C_x , C_y

[Πηγή: <http://www.analog.com>]

Η περίοδος T_2 του ψηφιακού σήματος DCM που προέρχεται από A/D converter μπορεί να ρυθμιστεί από 1 msec μέχρι και 10 msec και για τους δύο άξονες, η μέτρηση της επιτάχυνσης γίνεται μέσω ενός counter που μετρά την διάρκεια του T_2 και T_1 . Θα πρέπει να σημειωθεί ότι το T_1 είναι ενεργό τμήμα της περιόδου T_2 . Και υπολογίζει τον λόγο T_1/T_2 . Επιπλέον, η μηδενική επιτάχυνση (0g) ισούται με το 50% του λόγου T_1/T_2 (Μανωλόπουλος X., 2010).

T_2 (ms)	R_{SET} (k Ω)	ADXL202E Sample Rate	Counter-Clock Rate (MHz)	Counts per T_2 Cycle	Counts per g	Resolution (mg)
1.0	124	1000	2.0	2000	250	4.0
1.0	124	1000	1.0	1000	125	8.0
1.0	124	1000	0.5	500	62.5	16.0
5.0	625	200	2.0	10000	1250	0.8
5.0	625	200	1.0	5000	625	1.6
5.0	625	200	0.5	2500	312.5	3.2
10.0	1250	100	2.0	20000	2500	0.4
10.0	1250	100	1.0	10000	1250	0.8
10.0	1250	100	0.5	5000	625	1.6

Πίνακας 4.9: Καθορισμός ακρίβειας μετρήσεων

[Πηγή: <http://www.analog.com>]

Στην συνέχεια ακολουθεί ο Πίνακας 4.10 με τα χαρακτηριστικά του ADXL202JE της εταιρίας Analog Divices.

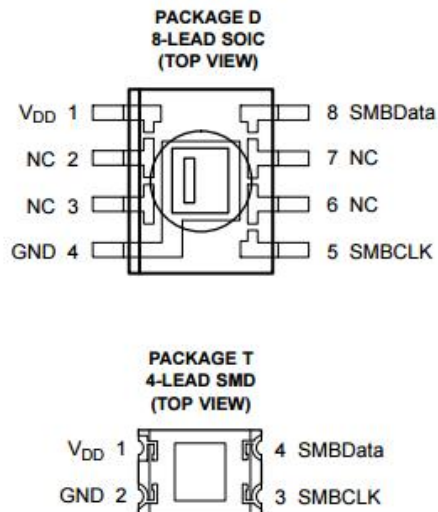
Τύπος Αισθητήρα	Αναλογική συσκευή ADXL202JE
Κανάλια	X (ADC1), Y(ADC2)
Διακύμανση	$\pm 2g$ ($1g=9,81 \text{ m/sec}^2$)
Ευαισθησία	167 mV/g, $\pm 17\%$
Ανάλυση	2mg(0.002 g) RMS
Offset	VBATTERY /2 $\pm 0,4V$
Εμβέλεια λειτουργίας	3.6 έως 3. volts
Θύρα	Αναλογική θύρα

Πίνακας 4.10: Τεχνικά Χαρακτηριστικά του ADXL202JE της εταιρίας Analog Divices.

(Κωνσταντίνου Γ., 2012)

4.3.4. Αισθητήρας φωτός

Για την μέτρηση της φωτεινότητας σε Lux χρησιμοποιείται ο αισθητήρας TSL2550D της εταιρίας TAOS (Texas Advanced Optoelectronic Solutions). Η τάση τροφοδοσίας είναι 2.7 – 5.5 Volts και η επικοινωνία με τον μC πραγματοποιείται έσω του pins SMBCLK και SMBData (TWI interface). Η θερμοκρασία λειτουργίας είναι από -40°C μέχρι τους 85°C .



Σχήμα 4.22: Αισθητήρας TSL2550D της εταιρίας TAOS

[Πηγή: <http://www.farnell.com>]

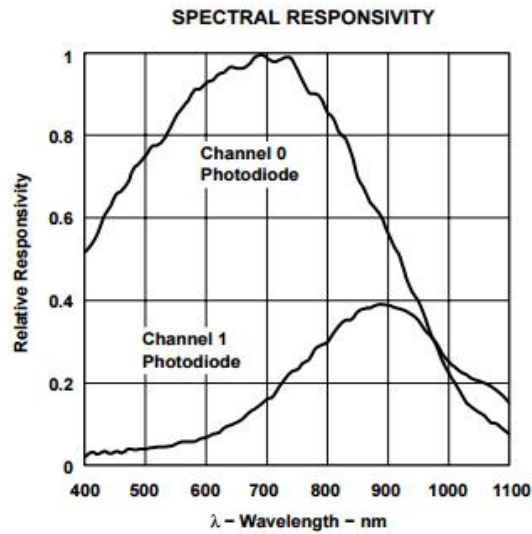
TERMINAL NAME	TERMINAL		TYPE	DESCRIPTION
	D PKG NO.	T PKG NO.		
GND	4	2		Power supply ground. All voltages are referenced to GND.
SMBCLK	5	3	I	SMBus serial clock input terminal — clock signal for SMBus serial data.
SMBData	8	4	I/O	SMBus serial data I/O terminal — serial data I/O for SMBus.
V _{DD}	1	1		Supply voltage.

Πίνακας 4.11: Περιγραφή των pins του αισθητήρα φωτός TSL2550D

[Πηγή: <http://www.farnell.com>]

Αποτελείται από δύο φωτοдиодους, οι οποίες έχουν εύρος ευαισθησίας ακτινοβολίας από 380 nm μέχρι 1100 nm. Η μία φωτοδιόδος (channel 0) είναι ευαίσθητη στο ορατό φως (380nm – 750nm) και στις υπέρυθρες ακτίνες (750nm – 1100nm), ενώ η δεύτερη (channel 1)

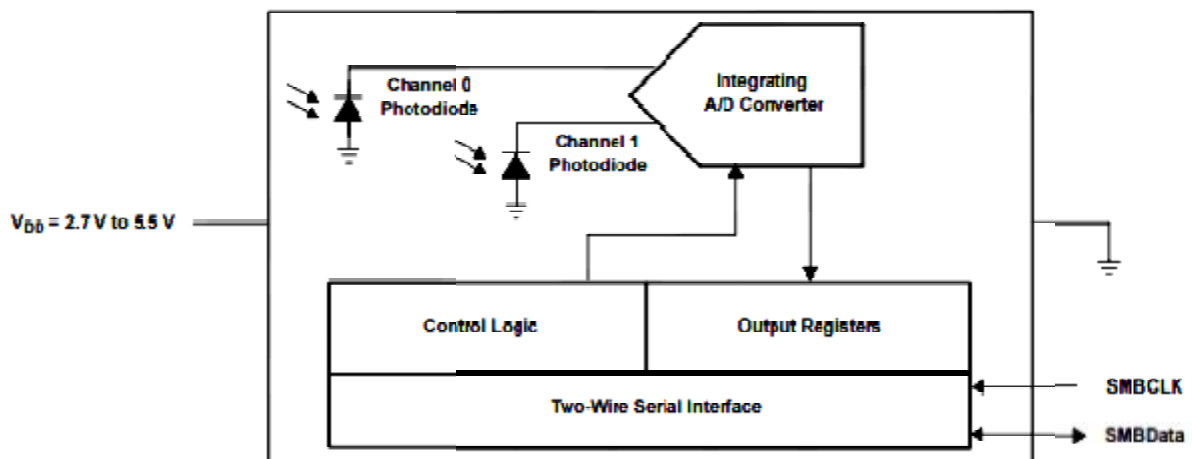
έχει ευαισθησία αποκλειστικά στις υπέρυθρες ακτίνες όπως διακρίνεται στο Διάγραμμα 4.1. Το channel 1 χρησιμοποιείται κυρίως για την αντιστάθμιση της επίδρασης συνιστώσας που προέρχεται από το channel 0.



Διάγραμμα 4.1: Απεικόνιση channel 0 και channel 1.

[Πηγή: <http://www.farnell.com>]

Ο αισθητήρας μέτρησης φωτεινότητας περιέχει έναν A/D converter 12 bit. Για την απόδοση της τιμής της φωτεινότητας, αρχικά μετατρέπονται οι τάσεις των φωτοδιόδων σε ψηφιακές τιμές και ύστερα αποθηκεύονται στο αντίστοιχο ADC register του κάθε καναλιού.



Σχήμα 4.23: Block διάγραμμα του αισθητήρα φωτός TSL2550D

[Πηγή: <http://www.farnell.com>]

Σε control Logic Block σχήματος υπάρχει ένας command register μέσω του οποίου επιλέγονται λειτουργίες όπως:

- Ανάγνωση των ADC register
- Ταχύτητα μετατροπής A/D converter (400ms ή 800ms)
- Κατάσταση λειτουργίας του αισθητήρα (power up/down state)

Στην συνέχεια ακολουθεί ο Πίνακας 4.12 όπου καταγράφονται οι εντολές Command Register.

COMMAND	FUNCTION
0x00h	Power-down state
0x03h	Power-up state/Read command register
0x1Dh	Write command to assert extended range mode
0x18h	Write command to reset or return to standard range mode
0x43h	Read ADC channel 0
0x83h	Read ADC channel 1

Πίνακας 4.12: Οι εντολές του Command Register

[Πηγή: <http://www.farnell.com>]

Κάθε ADC register μεγέθους 8 bit περιέχει δύο πεδία τα οποία είναι CHORD και STEP και χρησιμοποιούνται για τον υπολογισμό της φωτεινότητας. Το MSB του ADC register (B7 Valid Bit) χρησιμοποιείται για την ένδειξη αποθηκεύσεις δεδομένων στον καταχωρητή. Το πεδίο CHORD (B6..B4) αντιστοιχεί στο σημαντικό τμήμα της τιμής του ADC, ενώ το STEP (B3..B0) αντιστοιχεί στο πιο ασήμαντη τμήμα της τιμής της.

VALID	CHORD BITS			STEP BITS			
B7	B6	B5	B4	B3	B2	B1	B0
VALID	C2	C1	C0	S3	S2	S1	S0

FIELD	BITS	DESCRIPTION
VALID	7	ADC channel data is valid. One indicates that the ADC has written data into the channel data register, since ADCEN was asserted in the COMMAND register.
CHORD	6 to 4	CHORD number.
STEP	3 to 0	STEP number.

Πίνακας 4.13: Τα πεδία CHORD και STEP του ADC register

[Πηγή: <http://www.farnell.com>]

Ωστόσο, θα πρέπει να σημειωθεί ότι υπάρχει σχέση ανάμεσα στις CHORD και VALUE τιμές και καταγράφεται στον Πίνακα 4.14.

CHORD BITS B6, B5, B4	C, CHORD NUMBER	CHORD VALUE (Note A)	STEP VALUE (Note B)
000	0	0	1
001	1	16	2
010	2	49	4
011	3	115	8
100	4	247	16
101	5	511	32
110	6	1039	64
111	7	2095	128

Πίνακας 4.14: Σχέση ανάμεσα στις CHORD και VALUE τιμές

[Πηγή: <http://www.farnell.com>]

Στην συνέχεια ακολουθεί ο πίνακας με συνοπτικά τα χαρακτηριστικά του αισθητήρα TSL2550D – TAOS.

Τύπος Αισθητήρα	TSL2550D - TAOS
Κανάλια	Φως
Διακύμανση	380 nm μέχρι 1100 nm
Εμβέλεια λειτουργίας	2.7 – 5.5 Volts
Θύρα	Ψηφιακή θύρα

Πίνακας 4.15: Τεχνικά χαρακτηριστικά του TSL2550D - TAOS

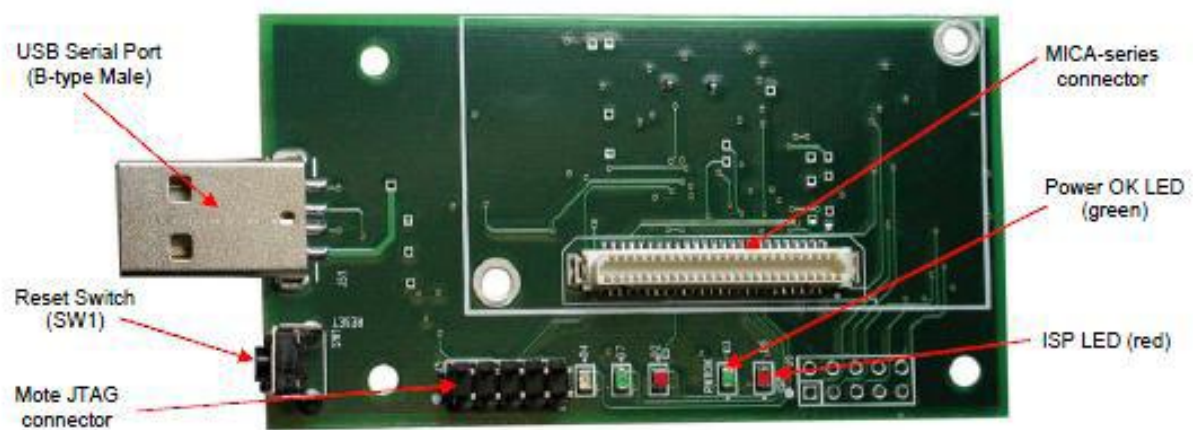
(Κωνσταντίνου Γ., 2012)

Θα πρέπει να σημειωθεί ότι για ο ακριβή υπολογισμός των προαναφερόμενων χαρακτηριστικών πραγματοποιείται με την βοήθεια μαθηματικών σχέσεων. Ωστόσο, η παρούσα πτυχιακή εργασία δεν θα επεκταθεί περαιτέρω στην ανάλυση των αισθητήρων.

4.4.MODULE MIB520

Η MIB520 (Mote Interface Board) έχει την δυνατότητα σύνδεσης με Ηλεκτρονικό Υπολογιστή (Εικόνα 4.3) μέσω μια θύρας USB. Αφού πραγματοποιηθεί μια εγκατάσταση από συγκεκριμένους driver, η θύρα αυτή έχει την δυνατότητα να λειτουργεί σαν δύο εικονικές σειριακές θύρες COM με διαφορετική λειτουργία κάθε μια. Η πρώτη COM, δίνει την δυνατότητα στο MIB520 να λειτουργεί ως πλατφόρμα προγραμματισμού του MICAZ MPR2600 και μέσω της δεύτερης, λειτουργεί ως σταθμός βάσης του WSN.

Σημειώνεται ότι σε κάθε περίπτωση λειτουργίας του MIB520, θα πρέπει να είναι τοποθετημένο σε ένα MICAZ MPR2600, είτε για τον προγραμματισμό του ATmega 128L μέσω ενός επεξεργαστή ATmega16L (δυνατότητα ISP - In System Programming), είτε για την αποστολή / λήψη δεδομένων από το δίκτυο μέσω του CC2420 RF σε πραγματικό χρόνο.



Εικόνα 4.3: Όψη του MIB520

[Πηγή: <http://vivliothmmy.ee.auth.gr>]

Θα πρέπει να σημειωθεί ότι το κάθε mote αποτελείται από τα MICAZ MPR2600 και MTS400 modules, ενώ ο σταθμός βάσης αποτελείται από τα MIB520 και MICAZ MPR2600 modules. Ο σταθμός βάσης είναι συνδεδεμένος σε ένα ηλεκτρονικό υπολογιστή και μέσω της θύρας USB και έχει σταθερή τροφοδοσία. Τα motes μπορούν να στέλνουν τις μετρήσεις τους, αρκεί να βρίσκονται σε εύρος 75 – 100m από το base station ή κάποιον γειτονικό κόμβο, ανάλογα με τις περιβαλλοντικές συνθήκες (Μανωλόπουλος Χ., 2010).

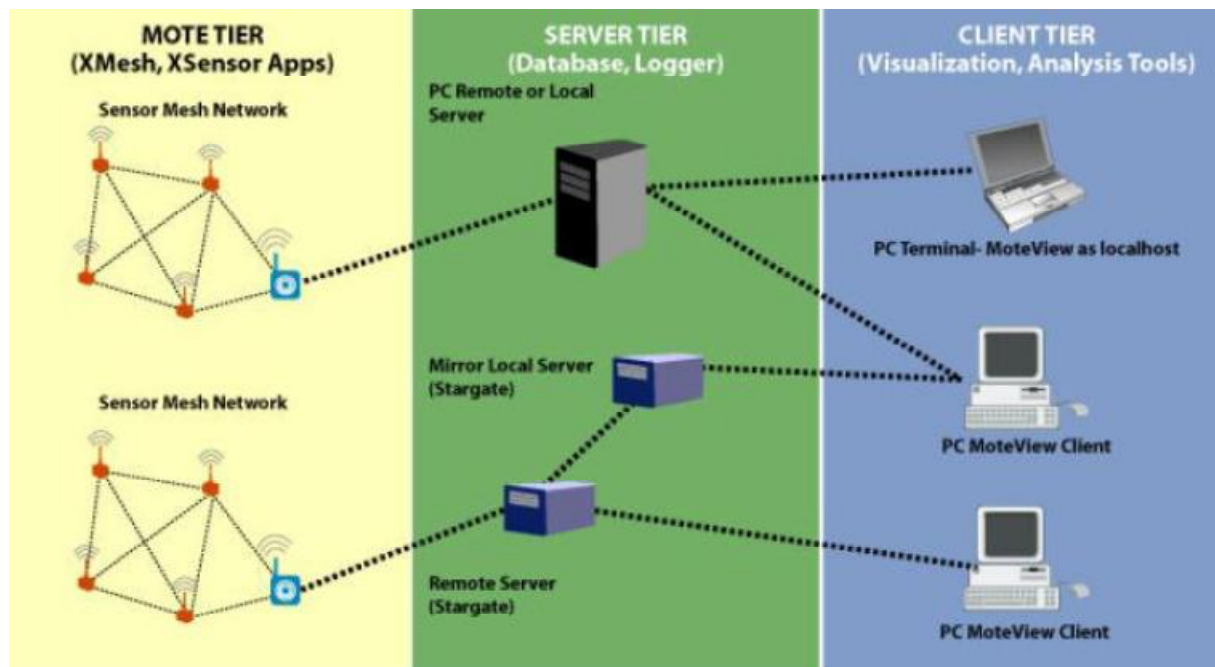
ΚΕΦΑΛΑΙΟ 5: ΠΛΑΤΦΟΡΜΑ ΛΟΓΙΣΜΙΚΟΥ MOTEWORKS

5.1.ΕΙΣΑΓΩΓΗ

Σε μία εφαρμογή εκτός από το Hardware που πολλές φορές είναι παραπλήσιο της εταιρίας Crossbow όπως προαναφέρθηκε και στο προηγούμενο κεφάλαιο, πρέπει να υπάρχει και η αντίστοιχη software πλατφόρμα λογισμικού MoteWorks για την υλοποίηση διαχείρισης ενός WSN δικτύου. Η πλατφόρμα αυτή αποτελείται από τα εξής προγράμματα:

- Programmer's Notepad – PN
- Cygwin
- MoteView

Κάθε ένα από τα προαναφερόμενα προγράμματα αντιστοιχεί σε μια από τις τρεις βαθμίδες (tiers) του δικτύου (Εικόνα 5.1)



Εικόνα 5.1: Οι βαθμίδες του WSN δικτύου της Crossbow
(Μανωλόπουλος X., 2010)

5.2.ΒΑΘΜΙΔΕΣ ΔΙΚΤΥΟΥ

Οι βαθμίδες δικτύου είναι τρεις και είναι οι εξής:

- Mote tier
- Server tier
- Client Tier

Στην συνέχεια ακολουθεί σύντομη ανάλυση των προαναφερόμενων σύμφωνα με τον Μανωλόπουλο X., (2010).

▼ *Mote tier*

Σε αυτή τη βαθμίδα δικτύου βρίσκεται το XMesh. Το Mote tier είναι ένα πρωτόκολλο δικτύου το οποίο είναι σχεδιασμένο για την δυναμική προώθηση μηνυμάτων από κόμβο σε κόμβο (multi –hop) προς τον σταθμό βάσης. Όταν υπάρξει φυσικό εμπόδιο ή πρέπει να υπάρξει κάλυψη μεγάλης απόστασης τότε δημιουργεί νέες διαδρομές προώθησης μηνυμάτων. Οι διαδρομές αυτές επιλέγονται βάση των δυνατοτήτων που παρουσιάζει το δίκτυο (ad-hoc), χωρίς όμως την υπάρξει προϋπάρχουσας υποδομής όπως συμβαίνει με τα υπόλοιπα συμβατά δίκτυα.

Το Mote tier παρέχει όλους τους αλγόριθμους για τον σχηματισμό μιας σταθερής και αξιόπιστης επικοινωνίας μεταξύ των κόμβων και στους σταθμούς βάσης με ένα βασικό πλεονέκτημα να μηδενίζεται το ενεργειακό κόστος. Πρόκειται για βιβλιοθήκες λογισμικού TinyOS, γραμμένες σε Nesc, οι οποίες μέσω του *Programmer's Notepad - PN* αποθηκεύονται στον σταθμό βάσης και τους κόμβους του δικτύου. Το πρωτόκολλο αυτό βασίζεται στα πρότυπα ZigBee και 802.15.4.

▼ *Server tier*

Σε αυτή τη βαθμίδα δικτύου πραγματοποιείται η διαχείριση και επεξεργασία των δεδομένων τα οποία έχουν σταλεί από τον σταθμό βάσης, μέσω του Xserve. Αποτελείται από την γέφυρα επικοινωνίας ανάμεσα στο ασύρματο δίκτυο και τον χρήστη. Με την βοήθεια ενός client-PC δίνεται η δυνατότητα να προωθηθούν άμεσα τα δεδομένα στο χρήστη προς και από το παρόν δίκτυο. Ωστόσο μπορεί η μεταφορά να πραγματοποιηθεί μέσω κάποιων routers (stargates) στο internet. Ωστόσο θα πρέπει να σημειωθεί ότι ο πυρήνας του XServe παρέχει

τα προαναφερόμενα σύμφωνα με τις ρυθμίσεις που έχουν πραγματοποιηθεί σε αυτόν καθώς και στον τρόπο εγκατάστασης του.

Με την υπηρεσία Serial Forwarder το XServe δέχεται και στέλνει δεδομένα σε ακατέργαστη μορφή, χωρίς κάποιο είδους επεξεργασίας ή μετατροπής. Στην περίπτωση αυτή το XServe είναι εγκατεστημένο σε συσκευές δικτύου (stargates) που είναι συνδεδεμένες διαδοχικά και γίνεται η προώθηση των δεδομένων, μέσω internet.

Με την υπηρεσία Application Server το XServe μπορεί να λειτουργήσει ως Serve παρέχοντας υπηρεσίες υψηλότερου επιπέδου όπως επεξεργασία και μετατροπή δεδομένων σε πραγματικό χρόνο. Ο χρήστης έχει την δυνατότητα να αλληλεπιδράσει με το XServer μέσω ενός προγράμματος βασισμένο σε περιβάλλον UNIX, το Cygwin. Στο terminal interface του Cygwin εισάγονται εντολές που αφορούν τις ρυθμίσεις του XServe, την απεικόνιση των δεδομένων σε πραγματικό χρόνο και τη μετατροπή και αποθήκευση των δεδομένων σε XML αρχείο.

▼ *Client Tier*

Σε αυτή την βαθμίδα του δικτύου πραγματοποιείται η απεικόνιση των δεδομένων στην οθόνη του ηλεκτρονικού υπολογιστή, σε περιβάλλον φιλικό για το χρήστη. Το πρόγραμμα που χρησιμοποιείται στην περίπτωση αυτή είναι το MoteView, το οποί επιτρέπει την απεικόνιση των δεδομένων με την μορφή τιμών ή με την μορφή γραφικών παραστάσεων τα δεδομένα σε μορφή SQL για την διαχείριση των δεδομένων μέσα σε συγκεκριμένες βάσεις δεδομένων που αντιστοιχούν σε κάθε πλακέτα αισθητήρων.

5.3.MOTE TIER

Σύμφωνα με τον Κωνσταντίνου Γ., (2012) το επίπεδο Mote, όπου τρέχει το XMesh, είναι ένα λογισμικό που τρέχει πάνω από το ασύρματο δίκτυο αισθητήρων σχηματίζοντας ένα mesh δίκτυο. Το λογισμικό XMesh παρέχει τους δικτυακούς αλγορίθμους που απαιτούνται για να σχηματιστεί μια αξιόπιστη επικοινωνία που θα συνδέει όλους τους κόμβους αισθητήρων με τον εξυπηρετητή. Έχει αναπτυχθεί από την εταιρεία Crossbow και τρέχει πάνω από το λειτουργικό σύστημα του TinyOS που είναι εγκατεστημένο στους αισθητήρες. Το XMesh είναι ένα πλήρες multi-hop, ad-hoc δικτυακό πρωτόκολλο mesh που όπως

αναφέραμε πιο πάνω αναπτύχθηκε από την Crossbow, ειδικά για ασύρματα δίκτυα αισθητήρων.

Ένα XMesh δίκτυο αποτελείται από κόμβους (Motes) που επικοινωνούν ασύρματα μεταξύ τους και είναι ικανά να μεταπηδούν ράδιο-μηνύματα σε ένα κεντρικό σταθμό (base station) όπου και μεταφέρονται σε ένα ηλεκτρονικό υπολογιστή. Η μεταπήδηση επεκτείνει την απόσταση της ασύρματης επικοινωνίας αποδοτικά και μειώνει την ενέργεια που απαιτείται για να αποστέλλονται τα δεδομένα των αισθητήρων. Με την μεταπήδηση των δεδομένων με αυτό τον τρόπο το XMesh προφέρει δύο σημαντικά οφέλη και είναι τα εξής:

- Βελτιωμένη κάλυψη
- Αξιοπιστία.

Ένα μήνυμα μπορεί να παραδοθεί σε ένα ή περισσότερους κόμβους ενδιάμεσα που θα δρομολογήσουν τα δεδομένα, οπότε δύο κόμβοι δεν χρειάζεται να είναι σε κατευθείαν επικοινωνία μεταξύ τους για να παραλάβουν ένα πακέτο. Με αυτό τον τρόπο, σε περίπτωση βλάβης κάποιου ενδιάμεσου αισθητήρα ή αδυναμία επικοινωνίας λόγω παρεμβολής ή άλλων εμποδίων, μπορούν να αντιμετωπιστούν με το να γίνει αναδρομολόγηση των πακέτων γύρω από την περιοχή με την κακή σύνδεση. Επίσης οι αισθητήρες είναι συνήθως σε sleep mode (95% του χρόνου) ώστε να έχουν μεγαλύτερη διάρκεια ζωής.

Το XMesh είναι μια βιβλιοθήκη λογισμικού το οποίο χρησιμοποιεί το TinyOS λειτουργικό σύστημα και τρέχει επάνω στους κόμβους αισθητήρων. Ολόκληρο το XMesh δίκτυο αποτελείται από:

- Τους κόμβους αισθητήρων
- Τον βασικό σταθμό, όπως προαναφέρθηκε ο αισθητήρας αυτός είναι ακόμη ένας αισθητήρας όπως τους υπόλοιπους με τη διαφορά ότι ενώνεται με ένα Crossbow MIB520 interface board που καταλήγει ένα ηλεκτρονικό υπολογιστή. Επίσης ο αισθητήρας αυτός προγραμματίζεται έτσι ώστε να διαχειρίζεται το δίκτυο προωθώντας και λαμβάνοντας πακέτα από αυτό.
- Ένα υπολογιστή που λαμβάνει τα δεδομένα ή αποστέλλει εντολές προς το δίκτυο.

5.3.1. Χαρακτηριστικά του XMesh

Το XMesh διαθέτει πολλές λειτουργίες και γνωρίσματα σύμφωνα με τον Μανωλόπουλο Χ., (2010) και τον Κωνσταντίνου Γ., (2012) τα βασικά είναι τα ακόλουθα:

✓ TrueMesh

Η τεχνολογία TrueMesh αναφέρεται στην ικανότητα των κόμβων αισθητήρων να αναζητούν δυναμικά νέα δρομολόγια για να παραδίδουν πακέτα όταν κάποια κομμάτια του δικτύου βγουν εκτός λειτουργίας λόγω παρεμβολής σημάτων ή λόγω ενέργειας. Οι κόμβοι ανακαλύπτουν ο ένας τον άλλο και χτίζουν ένα δέντρο δρομολόγησης βασισμένο στις εκτιμήσεις κόστους των μονοπατιών για το συγκεκριμένο περιβάλλον στο οποίο ανήκουν. Επομένως οι κόμβοι μέσα στο XMesh δίκτυο είναι αυτό διοργανωμένοι και αυτό ρυθμισμένοι σε περίπτωση βλάβης.

✓ Πολλαπλές Υπηρεσίες Μεταφοράς

Το XMesh προσφέρει πολλαπλές υπηρεσίες επικοινωνίας μεταξύ των κόμβων αισθητήρων οι οποίες είναι:

- Upstream: Προώθηση πακέτων από ένα κόμβο αισθητήρα στο base station.
- Downstream: Προώθηση πακέτων από το base station σε ένα ή περισσότερους κόμβους αισθητήρων.
- Single Hop: Προώθηση πακέτων σε γειτονικούς κόμβους μόνο.

✓ Πολλαπλά είδη Ποιότητας Υπηρεσιών (QoS)

Το XMesh διαθέτει πολλά είδη ποιότητας υπηρεσιών τα οποία είναι:

- Best Effort: Link level acknowledgement όπου οι κόμβοι θα δοκιμάσουν πολλαπλές φορές να στείλουν ένα μήνυμα στον γειτονικό τους κόμβο.
- Guaranteed Delivery: Επιτρέπει end-to-end acknowledgement όταν το μήνυμα αποστέλλεται μέσω του πλέγματος στον βασικό σταθμό ή αντίστροφα και ένα acknowledge message τότε στέλλεται πίσω στον αποστολέα.

✓ Πολλαπλά είδη Ενέργειας

Το XMesh μπορεί να ρυθμιστεί για να τρέχει ένα από τα δύο είδη power mode τα οποία είναι:

- High Power (HP): Το HP mode έχει τα εξής χαρακτηριστικά:
 - Δυνατότητα TrueMesh
 - Κάθε κόμβος στο δίκτυο μπορεί να δρομολογήσει δεδομένα
 - Ψηλό Bandwidth, χαμηλή καθυστέρηση (πλήρης χρήση καναλιού)
 - Οι αντένες των κόμβων είναι πάντοτε ενεργοποιημένες

- Low Power (LP): Το LP mode έχει τα εξής χαρακτηριστικά:
 - Δυνατότητα TrueMesh
 - Κάθε κόμβος στο δίκτυο μπορεί να δρομολογήσει δεδομένα
 - Χαμηλό Bandwidth, ψηλή καθυστέρηση (ιδανικό για εφαρμογές με χαμηλό data rate)
 - Οι αντένες των κόμβων είναι κανονικά σε sleep mode με μικρή κατανάλωση ενέργειας και ξυπνούν περιοδικά για να ελέγξουν για διακίνηση δεδομένων

- Extended Low Power (ELP): Το ELP mode έχει τα εξής χαρακτηριστικά:
 - Χρησιμοποιείται μόνο για τους τελικούς κόμβους του δικτύου
 - Οι κόμβοι αυτοί δεν δρομολογούν δεδομένα
 - Χρησιμοποιεί υβριδική star mesh ρύθμιση

✓ Διαγνωστικά Υγείας

Μέσα στο δίκτυο XMesh, οι κόμβοι μπορούν αυτόματα να αποστέλλουν πληροφορίες σχετικά με την υγεία τους στο base station. Τα δεδομένα υγείας αφορούν στο πόσο καλά ένας κόμβος λειτουργεί στο δίκτυο όσο αφορά την αποστολή δεδομένων, τα αποθέματα μπαταρίας και το radio signal strength indicator (RSSI) του πατέρα τους.

✓ Συγχρονισμός Χρόνου

Στο Low Power (LP) mode υποστηρίζεται ένας ενιαίος συγχρονισμός χρόνου στο ± 1 msec. Η χρονοσήμανση χρησιμοποιείται για να συγχρονίσει τα μηνύματα αλλά είναι διαθέσιμο και στους χρήστες για να συγχρονίσουν τις μετρήσεις των αισθητήρων.

✓ Προγραμματισμό εξ αέρος (ασύρματα)

Το XMesh υποστηρίζει προγραμματισμό εξ αέρος (ασύρματα), επιτρέποντας στους χρήστες να επαναπρογραμματίζουν όλους τους κόμβους στο πλέγμα με νέο κώδικα.

5.3.2. Σχηματισμός του δικτύου

Για την δημιουργία του ασυρμάτου δικτύου εκτελούνται δύο παράλληλες λειτουργίες που είναι οι εξής:

✓ 1^η Λειτουργία: Εκτίμηση Σήματος (*Link Estimation*)

Η εκτίμηση σήματος πραγματοποιείται καθώς ένας κόμβος συλλέγει πληροφορίες για την εκτίμηση της κατάστασης και της κίνησης του δικτύου στην περιοχή του. Οι πληροφορίες που συλλέγονται χρησιμοποιούνται με στόχο να συμπληρωθεί ο πίνακας routing table, του οποίου το μέγεθος καθορίζεται από τις 16 διευθύνσεις για τους κόμβους του δικτύου, ενώ για τον σταθμό βάσης ορίζονται 40 διευθύνσεις.

Σε περίπτωση που ένας κόμβος αναγνωρίσει περισσότερους από 16 γείτονες τότε αναγκάζεται να διαγράψει κάποιους, η επιλογή γίνεται από τον πίνακα διαγράφωντας αυτούς με το χαμηλότερα ποιοτικό σήμα. Παρακολουθώντας τις τιμές της μεταβλητής Sequence Number της MHR κεφαλίδας του Mac Protocol Data Unit (MPDU) υπολογίζεται η εκτίμηση σήματος, διαιρώντας τον αριθμό των μηνυμάτων που έχουν ληφθεί προς το συνολικό άθροισμα.

Η εκτίμηση σήματος μπορεί να δοθεί με την εξής σχέση:

$$\text{New_Estimate} = 255 * \text{received} / (\text{received} + \text{missed}) \quad (5.1)$$

Όπου :

Received = μηνύματα που ελήφθησαν

Missed= μηνύματα που απέτυχαν – χάθηκαν

Θα πρέπει να σημειωθεί ότι η εκτίμηση σήματος παίρνει τιμές στο διάστημα [0-255] με στόχο την κανονικοποίηση της εκτίμησης λήψης (RE).

✓ 2^η Λειτουργία: Επιλογή Γονέα (*Parent Selection*)

Αφού ο κόμβος έχει συμπληρώσει τον πίνακα (neighborhood table) με τις διεύθυνσεις των γειτονικών κόμβων, επιλέγει ως γονέα (parent) ένα γειτονικό κόμβο σύμφωνα με το ελάχιστο κόστος επικοινωνίας. Ο γειτονικός κόμβος δεν πρέπει να είναι σε κατάσταση λειτουργίας ELP και να μην είναι απόγονος (child node) του αρχικού, για τα τρία τελευταία RUI (Route Update Intervals) έτσι ώστε να αποφεύγονται βρόχοι επανάληψης κατά τον σχηματισμό δικτύου.

Το RUI είναι το χρονικό διάστημα (36 secs για HP mode, 360 secs για LP mode) που μεσολαβεί κατά την διαδοχική αποστολή δυο Route Update Messages (RUM), τα οποία περιχούν πληροφορίες των διαθέσιμων διαδρομών στο δίκτυο, όπως:

- *Parent ID (ταυτότητα γονέα)*: Αν ο κόμβος δεν έχει εισέλθει στο δίκτυο, η τιμή της μεταβλητής αυτής είναι 0xFFFF.
- *Cost*: Παρέχει ενημέρωση των γειτονικών κόμβων ποιο είναι το κόστος αποστολής ενός μηνύματος προς τον σταθμό βάσης.
- *Hop Count*: Ο αριθμός των hops από κόμβο σε κόμβο που κάνει το μήνυμα για να φτάσει στον σταθμό βάσης.

Ωστόσο, οι προαναφερόμενες πληροφορίες καθορίζονται από κάποιες μεταβλητές οι οποίες είναι οι παρακάτω:

- *Εκτίμηση λήψης RE*: είναι η ποιότητα λήψης από ένα γειτονικό mote και υπολογίζεται χρησιμοποιώντας τον αλγόριθμο EWMA (Exponentially Weighted Moving Average) στην παρακάτω σχέση:

$$Re = (1-\alpha) * RE + \alpha * \text{New_Estimate} \quad (5.2)$$

Όπου:

Re= εκτίμηση λήψης το οποίο κυμαίνεται από 0 έως 255

alpha= EWMA παράγοντας, της οποίας οι τιμές βρίσκονται στο διάστημα από 0 - 1 και καθορίζει το βάρος της νέας εκτιμώμενης τιμής.

- *Εκτίμηση εκπομπής SE*: είναι η ποιότητα μετάδοσης ενός μηνύματος σε ένα γειτονικό κόμβο και υπολογίζεται μέσω του RUM μηνύματος που προέρχεται από αυτόν. Όταν ένας κόμβος εκπέμπει RUM μήνυμα περιλαμβάνει σε αυτό την εκτίμηση λήψης RE των γειτόνων του, έτσι ώστε οι γειτονικοί κόμβοι να γνωρίζουν την εκτίμηση της εκπομπής SE του κόμβου.

Ωστόσο, είναι σημαντικό να αναλυθούν τα κόστοι που παρουσιάζονται και αυτά είναι τα εξής:

- LC: Είναι το κόστος σύνδεσης ενός κόμβου με έναν γειτονικό. Αν οι μεταβλητές RE και SE είναι γνωστές τότε το LC υπολογίζεται από την σχέση:

$$LC = (1 \ll 18) / (SE * RE) \quad (5.3)$$

- NC: Είναι το Κόστος αποστολής μηνύματος σε γειτονικό κόμβο. Προκύπτει από τον RUM μήνυμα του συγκεκριμένου γειτονικού κόμβου.
- OC: Είναι το κόστος αποστολής ενός μηνύματος προς τον σταθμό βάσης. Ένα node επιλέγει ένα γειτονικό ως parent, όταν αυτό έχει την μικρότερη τιμή OC η οποία υπολογίζεται από την σχέση:

$$OC = LC + NC \quad (5.4)$$

Το XMesh εκτελεί την διαδικασία επιλογής γονέα κάθε 8 χρονικά διαστήματα Route Update Intervals (RUI), χρόνος αρκετός για έναν κόμβο να συλλέξει πληροφορίες από τους γείτονες του και να πάρει τις κατάλληλες αποφάσεις. Όταν ένας κόμβος τεθεί σε λειτουργία ή έχασε τον κόμβο – γονέα του (parent), εκτελεί έναν γρήγορο αλγόριθμο επιλογής γονέα και επιλέγει έναν τυχαίο και όχι βέλτιστο κόμβο, με σκοπό να εισέλθει στο δίκτυο όσο το δυνατόν ταχύτερα.

Η επιλογή του κατάλληλου κόμβου για γονέα πραγματοποιείται αργότερα με βάση την παραπάνω διαδικασία εκτίμησης. Ο χρόνος χρειάζεται ένας κόμβος να εισέλθει στο δίκτυο καθορίζεται από τα χρονικά διαστήματα RUI και τον αριθμό των hops από κόμβο σε κόμβο. Ένας κόμβος που στέλνει μήνυμα στον σταθμό βάσης μέσω ενός hop, χρειάζεται έναν RUI για να εισέλθει στο δίκτυο. Αυτός που στέλνει ένα μήνυμα στον σταθμό βάσης μέσω δυο hops, χρειάζεται δύο RUIs κ.ο.κ. μέχρι και 8 RUIs, όπου και το δίκτυο σταθεροποιείται

5.3.3. Προγραμματισμός κόμβων και σταθμού βάσης

Το Xmesh εκτός από τα προαναφερόμενα χαρακτηριστικά που δίνουν την δυνατότητα πολλών επιλογών στον χρήστη, περιέχει και εφαρμογές οι οποίες είναι έτοιμες και γραμμένες σε κώδικα NesC, για όλους τους τύπους διάταξης των αισθητήρων της εταιρίας Crossbow. Ο

πίνακας 5.1 καταγράφει τα ονόματα της κάθε διάταξης αισθητήρων που μπορεί να τοποθετηθεί πάνω σε κάθε κόμβο, σε αντιστοιχία με όνομα και το path της εφαρμογής.

Σε αρκετές περιπτώσεις γίνεται χρήση του MTS400 sensor board, οπότε μέσω της αντίστοιχης εφαρμογής *XMTS400* ο κάθε κόμβος μπορεί να κάνει μετρήσεις θερμοκρασίας, υγρασίας, φωτεινότητας, βαρομετρικής πίεσης και δόνησης στους X-Y άξονες.

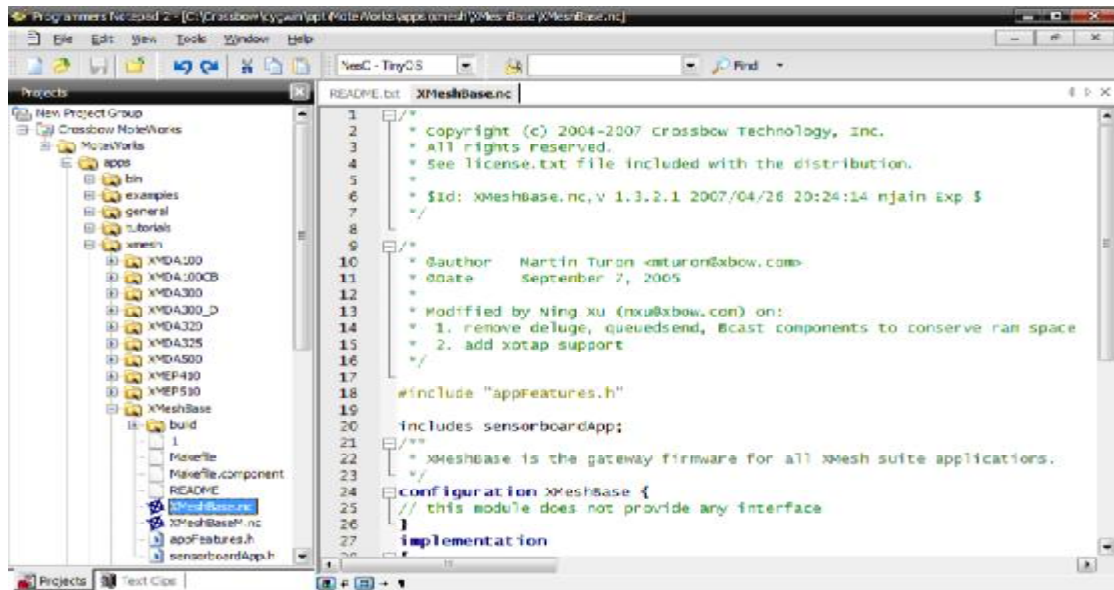
Sensor and Data Acquisition Boards	Application Name	Location of Driver Folder
MDA100CA	XMDA100	MoteWorks/apps/XMesh/XMDA100
MDA100CB	XMDA100CB	MoteWorks/apps/XMesh/XMDA100CB
MDA300	XMDA300	MoteWorks/apps/XMesh/XMDA300
MDA 300 (Differential)	XMDA300_D	MoteWorks/apps/XMesh/XMDA300_D
MDA320	XMDA320	MoteWorks/apps/XMesh/XMDA320
MDA325	XMDA325	MoteWorks/apps/XMesh/XMDA325
MDA500	XMDA500	MoteWorks/apps/XMesh/XMDA500
MEP410	XMEP410	MoteWorks/apps/XMesh/XMEP410
MEP510	XMEP510	MoteWorks/apps/XMesh/XMEP510
MSP410	XMSP410	MoteWorks/apps/XMesh/XMSP410
MTS101	XMTS101	MoteWorks/apps/XMesh/XMTS101
MTS300CA	XMTS300	MoteWorks/apps/XMesh/XMTS300
MTS300CB	XMTS300CB	MoteWorks/apps/XMesh/XMTS300CB
MTS310CA	XMTS310	MoteWorks/apps/XMesh/XMTS310
MTS310CB	XMTS310CB	MoteWorks/apps/XMesh/XMTS310CB
MTS400	XMTS400	MoteWorks/apps/XMesh/XMTS400
MTS410	XMTS410	MoteWorks/apps/XMesh/XMTS410
MTS420CA/MTS420CB	XMTS420	MoteWorks/apps/XMesh/XMTS420
MTS420CC	XMTS420CC	MoteWorks/apps/XMesh/XMTS420CC
MTS450	XMTS450	MoteWorks/apps/XMesh/XMTS450
MTS4510	XMTS510	MoteWorks/apps/XMesh/XMTS510

Πίνακας 5.1. Διατάξεις αισθητήρων με τα αντίστοιχα αρχεία εφαρμογών
(Μανωλόπουλο Χ., 2010)

Για τον σταθμό βάσης χρησιμοποιείται η εφαρμογή *XMeshBase*, η οποία αποτελεί τον δίαυλο επικοινωνίας ανάμεσα στο ασύρματο XMesh δίκτυο και στο XServe. Για την υλοποίηση του προγραμματισμού βάσης εκτελούνται με τα κάτωθι βήματα.

▼ Βήμα 1^ο :

Αφού ανοίξει το programmer's notepad, επιλέγεται το NesC αρχείο **XMeshBase.nc** (όχι το XMeshBaseM.nc) που βρίσκεται στο path MoteWorks/apps/xmesh/XMeshBase όπως φαίνεται στα αριστερά της οθόνης (Εικόνα 5.1).



Εικόνα 5.1: Άνοιγμα του αρχείου XMeshBase.nc

▼ **Βήμα 2^ο :**

Επιλέγεται tools > shell και στο παράθυρο που θα εμφανιστεί εισάγεται η εντολή:

```
make micaz install,0 mib520,com4
```

Το 0 αντιπροσωπεύει το ID του σταθμού βάσης, το mib520 το αντίστοιχο USB board προγραμματισμού, ενώ το com4 την εικονική θύρα προγραμματισμού.

▼ **Βήμα 3^ο :**

Στην συνέχεια αφού έχει ολοκληρωθεί το πρόγραμμα του σταθμού βάσης, στο output window κα πρέπει να εμφανιστεί το εξής μήνυμα:

```
process exit code: 0
```

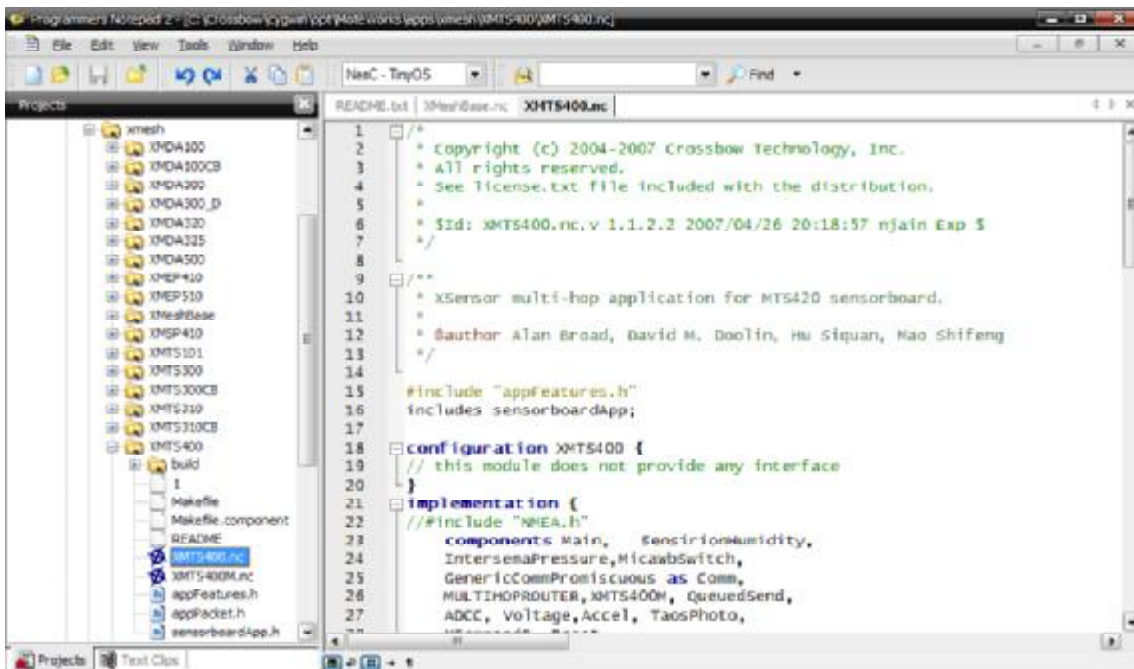


Εικόνα 5.2: Εμφάνιση επιτυχούς μηνύματος στο παράθυρο εξόδου

Για τον προγραμματισμό ενός κόμβου ακολουθείται η ίδια διαδικασία. Στην συνέχεια καταγράφονται τα βήματα που θα πρέπει να πραγματοποιηθούν.

▼ Βήμα 1^ο :

Τοποθετείται πάνω στο USB board MIB520 ένα MICAZ MPR2600 και επιλέγουμε το NesC αρχείο XMTS400.nc που βρίσκεται στο path MoteWorks/apps/xmesh/Xmesh Base, όπως φαίνεται αριστερά της οθόνης (Εικόνα 5.3).



Εικόνα 5.3: Άνοιγμα του αρχείου XMTS400.nc

▼ Βήμα 2^ο :

Επιλέγεται tools > shell και στο παράθυρο που θα εμφανιστεί εισάγεται η εντολή:

```
make micaz install,6495 mib520,com4
```

Ο αριθμός 6495 αντιστοιχεί στον ID αριθμό του συγκεκριμένου κόμβου. Κάθε κόμβος έχει και διαφορετικό ID.

▼ Βήμα 3^ο :

Το output window εμφανίζει την επιτυχή ολοκλήρωση της διαδικασίας.



```
Output
Uploading: flash
Firmware Version: 1.8
Fuse High Byte set to 0xd9
Fuse Extended Byte set to 0xff
rm -f build/micaz/main.exe.out-6495 build/micaz/main.srec.out-6495
> Process Exit Code: 0
> Time Taken: 00:17

[24]: 100 | ANSI | CR+LF | DTS | Project file: C:\Crossbow\cygwin\opt\Note\works\apps\ymesh\XMTS400\XMTS400.nc
```

Εικόνα 5.4: Εμφάνιση επιτυχούς μηνύματος στο παράθυρο εξόδου

Θα πρέπει να σημειωθεί ότι για τον προγραμματισμό των υπόλοιπων κόμβων ακολουθείται ακριβώς η ίδια διαδικασία, δίνοντας έμφαση πάντα στην σωστή καταγραφή του ID αριθμού του κάθε κόμβου. Στο σημείο αυτό προγραμματισμός του ασύρματου δικτύου WSN έχει ολοκληρωθεί (Μανωλόπουλο X., 2010).

5.4. SERVER TIER

Το Server Tier είναι μια συνεχής διεργασία που χειρίζεται την μετάφραση και την εναποθήκευση των δεδομένων που προέρχονται από το ασύρματο δίκτυο αισθητήρων. Παρέχει την γεφύρωση μεταξύ των ασύρματων κόμβων και των διαδικτυακών πελατών. Το XServe είναι εφαρμογή στο Server Tier που τρέχει σε υπολογιστή και παίζει τον ρόλο του gateway μεταξύ του πλέγματος του ασύρματου δικτύου και των εφαρμογών που αλληλεπιδρούν με το πλέγμα.

Στον πυρήνα του το XServe παρέχει υπηρεσίες που δρομολογούν πακέτο από και προς το mesh δίκτυο με υψηλού επιπέδου υπηρεσίες για να αναλύσει, μετατρέψει και να επεξεργαστεί τα δεδομένα καθώς μεταφέρονται ανάμεσα στο πλέγμα και τις εφαρμογές. Οι υψηλές επιπέδου υπηρεσίες μπορούν να ρυθμιστούν χρησιμοποιώντας XML σύνταξη.

Το XServe μπορεί να ενωθεί με το ασύρματο δίκτυο αισθητήρων που χρησιμοποιούν το πρωτόκολλο XMesh μέσω του base station στο οποίο τρέχει το XMeshBase που προσφέρει η Crossbow. Παρέχει διαφορετικά επίπεδα υπηρεσιών ανάλογα με το πώς έχει ρυθμιστεί και εγκατασταθεί όπως προαναφέρθηκε και στην αρχή του κεφαλαίου (Κωνσταντίνου Γ., 2012).

5.4.1. Εντολές του Xserve

Το XServe είναι το πρόγραμμα το οποίο συνδέει απευθείας με το ασύρματο δίκτυο, μέσω του σταθμού βάσης. Στο terminal screen του Cygwin, ο χρήστης μέσω ενός μεγάλου φάσματος εντολών έχει την δυνατότητα να επιλέγει διάφορες ρυθμίσεις και λειτουργίες που αφορούν το XServe.

Θα πρέπει να σημειωθεί ότι αρκετές από τις εντολές που παρουσιάζονται στην συνέχεια αφορούν τα εξής:

- Τις ρυθμίσεις σχετικά με την θύρα επικοινωνίας (COM, LAN),
- Τον τρόπο απεικόνισης των μηνυμάτων (raw, parsed, converted format)
- Την μετατροπή / αποθήκευση τους σε XML αρχείο.

```
Usage: xserve <-?|r|a|p|c|xr|xp|xc|dbxmlr|xmlp|xmlc|v|alert|m>
<-l=tablename>
<-dbserver=servername> <-dbport=portnum>
<-dbname=database name> <-dbuser=username>
<-dbpasswd=password>
<-h=path,hostname,portnum,config_file>
<-m=com,baud,protocol,slaveaddress,defaultregistervaluesas>
<-xmlfile=filename> <-xmlport=portnum>
[<-sf=hostname:port> | <-fsf=hostname:port> | <-device=dev>]
<-port=num> <-baud=num> <-platform=plt>
<-debug=level>
<-configfiles=filename:filename:>
<-loadparsers=filename:filename:...>
<-loaddatasinks=filename:filename:...>
<-heartbeat=<num missed>

-?      = display help [help]
-r      = raw display of tos packets [raw]
-a      = ascii display of tos packets [ascii]
-p      = parsed display of tos packets [parsed]
-c      = converted display of tos packets [converted]
-xr     = raw tos packets xported to file [export raw]
-xp     = parsed tos packets exported to file [export parsed]
-xc     = converted tos packets exported to file [export converted]
-db     = parsed tos packets exported to db [database parsed]
-dbserver = database server name (default=localhost)
-dbport  = database server port number (default=5432)
-dbname  = database name (default=MoteView db)
```

Εικόνα 5.5.α: Εντολές του XServe


```

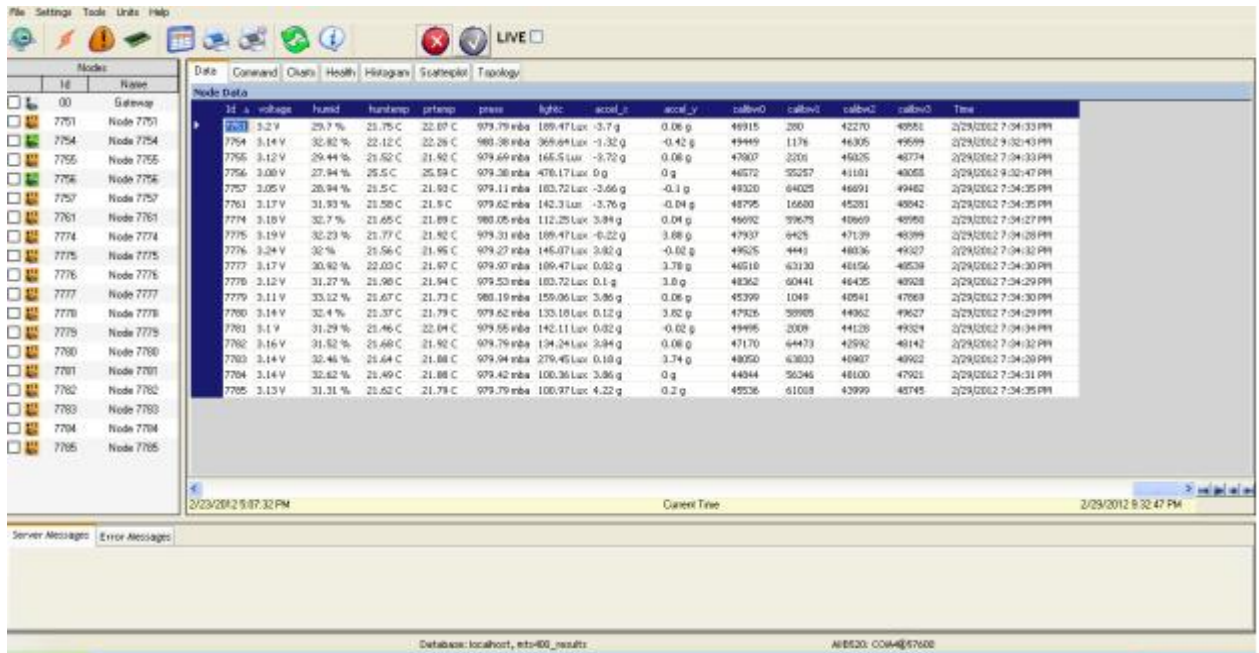
-dbuser      = database user (default=MoteView user)
-dbpasswd    = database user password (default=MoteView user password)
-l          = parsed tos packets exported to db
              (deprecated) [database parsed]
-xmlr       = raw tos packets exported to xml [xml raw]
-xmlp       = parsed tos packets exported to xml [xml parsed]
-xmlc       = converted tos packets exported to xml [xml converted]
-xmlfile    = file name to store exported xml (default=screen)
-xmlport    = port number to start the xml server
-v          = show version of all modules
-h          = display data through web server
-m          = export data using modbus
-port       = set server port <default = 9001>
-sf         = connect to unframed serial forwarder
-sfs        = connect to framed serial forwarder
-device     = connect to serial device <default = /dev/ttyS0>
-baud       = set serial baud rate <default = 57600>
-platform   = set platform. <default = mica2>
              values=mica2dot|mica2|mica|telos|micaz
-debug      = set debug level. <default = DBG_WARNING>
-alert      = alert when data values are above/below specified ranges
-daemon     = run in daemon mode
-nomonitor  = run without a system monitor
-configfiles = load xml configuration files.
-loadparsers = load only the listed parsers files.
              (default=all files are loaded)
-loaddatasinks = load only the listed datasinks files.
              (default=all files are loaded)

```

Εικόνα 5.5.β: Εντολές του XServe

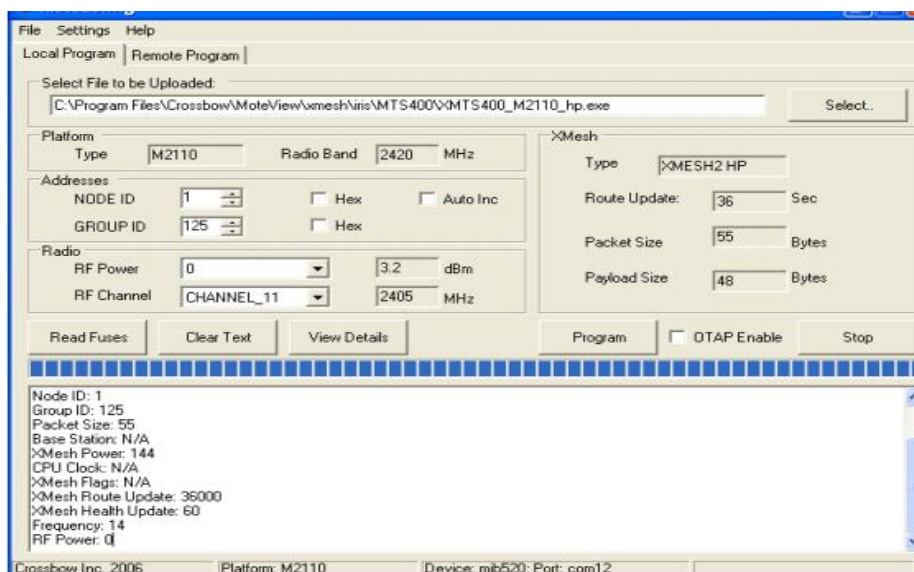
5.5.CLIENT TIER

Το Client Tier παρέχει το λογισμικό της διεπιφάνειας του χρήστη και την γραφική διαπροσωπεία για τον έλεγχο του δικτύου. Η Crossbow παρέχει το λογισμικό MoteView που δένει τα λογισμικά όλων των επιπέδων. Μέσω του λογισμικού MoteView γίνεται εύκολα η παρακολούθηση και ο έλεγχος των sensor, η σύνδεση στη βάση δεδομένων, η ανάλυση και η παρουσίαση των μετρήσεων των αισθητήρων. Η διαπροσωπεία του MoteView φαίνεται στην Εικόνα 5.6.



Εικόνα 5.6: Διαπροσωπεία MoteView

Σημειώνεται ότι το MoteView μπορεί να έχει ακόμα ένα εγκατεστημένο ακόμα ένα λογισμικό – το MoteConfig. Το MoteConfig είναι ένα εργαλείο που τρέχει πάνω σε Windows OS και επιτρέπει τον προγραμματισμό των IRIS motes. Παρέχει την διαπροσωπεία που φαίνεται στην Εικόνα 5.7 για την ρύθμιση και την εγκατάσταση pre-compiled XMesh/TinyOS λογισμικά πάνω στους αισθητήρες. Το MoteConfig επιτρέπει στους χρήστες να καθορίσουν το Mote ID, το Group ID, το RF κανάλι και τη RF ισχύ.



Εικόνα 5.7: Διαπροσωπεία MoteConfig

ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΑ

Η παρούσα πτυχιακή εργασία είχε σκοπό την ανάλυση του τρόπου λήψης των μηνυμάτων που στέλνονται από κόμβους ενός ασύρματου δικτύου αισθητήρων WSN. Πραγματοποίησε εκτενέστερη περιγραφή των συστημάτων που χρησιμοποιούνται σε τέτοιες περιπτώσεις με στόχο την κάλυψη του θεωρητικού υποβάθρου τους.

Τα ασύρματα δίκτυα και οι μικροελεγκτές μπορούν να εφαρμοστούν σε συστήματα απλών καθημερινών εργασιών μέχρι και συστήματα που απαιτούν πλήρη ασφάλεια και γρήγορη εκτέλεση. Οι τομές που εφαρμόζονται τα προαναφερόμενα συστήματα είναι οι περιβαλλοντικές εφαρμογές, γεωργικές εφαρμογές, εφαρμογές πρόληψης καταστροφών και παροχής βοήθειας, οικιακές εφαρμογές, επιτήρηση μηχανών και βιομηχανικές εφαρμογές, επιτήρηση αντικειμένων, εφαρμογές ασφαλείας, στρατιωτικές εφαρμογές, τηλεματική - έλεγχος μεταφορών και συγκοινωνιών, ιατρικές εφαρμογές και υγιεινή και διαφορές άλλες εμπορικές εφαρμογές.

Η πληθώρα εφαρμογών δίνει την δυνατότητα της ελαχιστοποίησης του χρόνου μιας εργασίας αλλά και την ασφάλεια της υλοποίησης της. Με την πάροδο των χρόνων και την ανάπτυξη της τεχνολογία τα συστήματα αυτά εξιδανικεύονται για κάθε εφαρμογή και δίνονται βελτιωμένες λύσεις διαρκώς.

Τα συμπεράσματα που αποκομίσθηκαν από αυτή την εργασία είναι ότι δίνεται στον χρήστη να επιλέξει και να ολοκληρώσει ένα σύστημα ανάλογα με την εφαρμογή που επιθυμεί να εκτέλεση. Επιπλέον, δίνεται η δυνατότητα να καθορίζονται με ακρίβεια τα αποτελέσματα που μπορεί να συλλέξει καθώς και να υπάρχει η πλήρη ασφάλεια αυτών κατά την μεταφορά τους. Ωστόσο, είναι σημαντικό να σημειωθεί ότι μπορούν να πραγματοποιηθούν βελτιώσεις σε κάθε ολοκληρωμένο σύστημα.

Τέλος, με την παρούσα πτυχιακή εργασία δίνεται βήμα στους Ηλεκτρολόγους Μηχανικούς Τ.Ε του Τ.Ε.Ι Δυτικής Ελλάδας να πραγματοποιήσουν και να υλοποιήσουν κάποια εφαρμογή έχοντας την ανάλυση και την πλήρη περιγραφή του θεωρητικού υποβάθρου των ασύρματων αισθητήρων και μικροελεγκτών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Ελληνική Βιβλιογραφία

1. Κατσαούνης Γ., 2013, *Ανάπτυξη ασύρματου δικτύου αισθητήρων και ελεγκτών στο Εργαστήριο Γενικής Ηλεκτροτεχνίας, Διπλωματική Εργασία, Εκδόσεις Πανεπιστήμιο Πατρών, Πάτρα*
2. Κωνσταντίνου Γ., 2012, *FIREWATCH: Ανίχνευση και πρόβλεψη δασικών πυρκαγιών με τη χρήση ασύρματου δικτύου αισθητήρων και γεωγραφικών δεδομένων, Εκδόσεις Πανεπιστήμιο Κύπρου, Λευκωσία*
3. Μακρή Χ., 2011, *Μελέτη Συνύπαρξης Ασύρματων Δικτύων Αισθητήρων και Δικτύων Wi-Fi σε Πραγματικό περιβάλλον, Εκδόσεις Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα*
4. Μανωλόπουλος Χ., 2010, *Ασύρματοι αισθητήρες και μικροελεγκτές, Εκδόσεις Πανεπιστήμιο Πατρών, Πάτρα*
5. Παπαβασιλείου Σ., 2010, *Δίκτυα Αισθητήρων Πρωτόκολλα προσπέλασης μέσου (MAC layer), Εκδόσεις Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα.*
6. Πλευράκη Α., 2011, *Σχεδίαση και υλοποίηση συστήματος συλλογής, επεξεργασίας και απεικόνισης περιβαλλοντικών δεδομένων από δίκτυο ασύρματων αισθητήρων, Εκδόσεις Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα*
7. Σπένδας Α, 2012, *Ασύρματα δίκτυα αισθητήρων και ζητήματα ασφάλειας, Εκδόσεις Πανεπιστήμιο Μακεδονίας, Θεσσαλονίκη.*

Ξένη Βιβλιογραφία

1. Asvest T., 2014, *ZigBee (zigbee/zigbee.txt), Εκδόσεις Εφαρμογές τηλεπικοινωνιακών διατάξεων.*
2. Crossbow, 2007, *Xserve – User’s Manual & Moteworks – Getting started Guide*
3. David Gay, Philip Levis, David Culler, and Eric Brewer, 2003, *nesC 1.1 Language Reference Manual.*
4. Dvorak J., 2005, *IEEE 802.15.4 and Zigbee Overview*

Διαδίκτυο

1. <http://brain.ee.auth.gr>
2. <http://www.slideshare.net>
3. <http://www.analog.com>
4. <http://www.ti.com>
5. <http://www.farnell.com>
6. <http://vivliothmmy.ee.auth.gr>
7. <http://artemis-new.cslab.ece.ntua.gr>
8. <http://nemertes.lis.upatras.gr/jspui>
9. <https://dspace.lib.uom.gr>
10. <http://conta.uom.gr>
11. <http://microcontrollerslab.com>
12. <http://www.cse.wustl.edu/~jain/cse574-08/ftp/medical>
13. <http://mobliezigbee.blogspot.gr>
14. <http://usn-pioneer.tistory.com>
15. <http://www.regent-elec.com>
16. <http://neobenblog.blogspot.gr>
17. <http://old.ceid.upatras.gr>
18. <http://docplayer.gr>
- 19.** <http://www.xbow.com>
20. <http://www.chipcon.com>
21. <http://www.memsic.com>
22. <http://www.datasheetlib.com>