

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΜΕΣΟΛΟΓΓΙΟΥ

ΠΑΡΑΡΤΗΜΑ ΝΑΥΠΑΚΤΟΥ

ΤΜΗΜΑ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ & ΔΙΚΤΥΩΝ

## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εφαρμογή διαχείρισης κτιριακών υποδομών με  
ολοκλήρωση ετερογενών συστημάτων

ΣΥΜΕΩΝΙΔΗΣ ΣΠΥΡΙΔΩΝ ΛΑΖΑΡΟΣ

A.M. 0196

**Επιβλέπων:** Ασημακόπουλος Γεώργιος

Ναύπακτος, Μάιος 2013

## Ευχαριστίες

Κατά την διάρκεια της υλοποίησης της παραδοτέας εφαρμογής και της συγγραφής της παρούσης πτυχιακής εργασίας, δύο άνθρωποι συνετέλεσαν στο να έρθει εις πέρας.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κύριο Ασημακόπουλο Γεώργιο για την ευκαρία που μου έδωσε και την εμπιστοσύνη που μου έδειξε με το να μου αναθέσει αυτήν την πτυχιακή εργασία.

Τέλος, θα ήθελα να ευχαριστήσω τον συμφοιτητή μου Δεδούση Δημήτριο, ο οποίος με υπέρμετρο ζήλο και μη φειδόμενος χρόνου, ήταν δίπλα μου από την αρχή της εκπόνησης της εργασίας, μέχρι και την τελευταία ημέρα.

## Πρόλογος

Από την δεκαετία του 1990 και μετά, έχει γεννηθεί η ιδέα του Internet of Things, μία ιδέα που όσο περνάνε τα χρόνια δείχνει να είναι ολοένα και πιο κοντά στην υλοποίησή της. Η τεχνολογία έκτοτε έχει προχωρήσει με πολύ γρήγορους ρυθμούς και αναμένεται να συνεχίσει να προχωράει με τους ίδιους ρυθμούς στο πέρασμα των χρόνων. Η ιδέα αυτή πραγματεύεται την δημιουργία δικτύων συσκευών, που θα μπορούν να επικοινωνούν, να συλλέγουν και να διαχειρίζονται πληροφορίες δίχως την συνεχή παρέμβαση των ανθρώπων, που, εκ των πραγμάτων έχουν πολύ περιορισμένο χρόνο και περιορισμένες δυνατότητες στην επεξεργασία δεδομένων.

Στην παρούσα πτυχιακή εργασία έχει αναπτυχθεί μία αρχική εφαρμογή διαχείρισης κτιριακών υποδομών, χρησιμοποιώντας ως ώρας αισθητήρες φωτεινότητας και θερμοκρασίας και είναι διαθέσιμη στο Διαδίκτυο, καθώς και για smartphones με λειτουργικό Android.

Τέλος χρησιμοποιήθηκε η πλατφόρμα ανοιχτού κώδικα Arduino, η οποία επικοινωνεί με το σύστημα διαχείρισης και είναι ικανή να λάβει μία έξυπνη απόφαση και να πράξει ανάλογα, υλοποιώντας έτσι την έννοια της ετερογένειας.

Μελετήθηκε σε πολύ μεγάλο βαθμό το πρωτόκολλο Zigbee το οποίο είναι και μία από τις βασικές αρχές του Internet of Things. Καλύπτει και δίνει λύσεις για ένα μεγάλο θέμα προς συζήτηση στις μέρες μας, το οποίο είναι η κατανάλωση ενέργειας.

## **Abstract**

Ever since the 1990s, the idea of the Internet of Things, has become increasingly closer to coming to reality. Technology ever since has moved on rapidly, and is expected to do so in the years to follow. This specific idea deals with the creation of networks of devices. These devices are able to communicate, collect data and manage them, without human intervention, which is whatsoever limited, due to human limited time and processing data abilities.

In this graduation thesis, a starting buling management application has been developed, currently using lighting and temperature sensors, and is available via Internet as well as for smartphones using Android as an operating system.

Lastly, one open source platform Arduino was used, which communicates with the building management system, rendering it capable of smart decision making. With the use of this platform, the concept of heterogeinity has been implemented.

The Zigbee communication protocol has been studied to a great extend, as it is one of the basic principles of the Internet of Things. It covers and gives viable solutions to a great problem of our days, which is the energy consumption.

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

<b>ΚΕΦΑΛΑΙΟ 1</b> .....	<b>1</b>
<b>INTERNET OF THINGS</b> .....	<b>1</b>
1.1. Internet of Things – Η ιδέα .....	1
1.2. Πράγματα - Αντικείμενα .....	2
1.3. Τι είναι το Internet of Things .....	3
1.4. Το όραμα του Μελλοντικού Internet .....	4
<b>ΚΕΦΑΛΑΙΟ 2</b> .....	<b>7</b>
<b>Building Energy Management System</b> .....	<b>7</b>
2.1. Ιστορία των BEMS (Building Energy Management Systems) .....	7
2.2. Πλεονεκτήματα BEMS .....	9
2.3. Υποσταθμοί .....	10
2.4. «Εξυπνοι» υποσταθμοί .....	10
2.6. Δομή υποσταθμού .....	11
2.7. Ο κεντρικός σταθμός.....	13
2.7.1. Αισθητήρες στα BEMS .....	13
<b>ΚΕΦΑΛΑΙΟ 3</b> .....	<b>16</b>
<b>ANDROID</b> .....	<b>16</b>
3.1. Τι είναι το Android.....	16
3.2. Ιστορική Αναδρομή.....	16
3.3. Γιατί Android .....	20
3.4. Κινητά τηλέφωνα Android.....	20
3.5. Ταμπλέτες Android .....	21
3.6. Αρχιτεκτονική .....	21
3.6.1. Πυρήνας Linux .....	22
3.6.2. Βιβλιοθήκες .....	22
3.6.3. Χρόνος Εκτέλεσης και η εικονική μηχανή Dalvik.....	23
3.6.4. Πλαίσιο Εφαρμογής (Application Framework).....	23

3.6.5. Εφαρμογές .....	24
3.7. Εργαλεία ανάπτυξης λογισμικού.....	25
3.7.1. Android SDK.....	25
3.7.2. Android NDK .....	25
3.7.3. Eclipse IDE.....	26
3.7.4. Android Development Tools .....	27
<b>ΚΕΦΑΛΑΙΟ 4 .....</b>	<b>28</b>
<b>SunSPOT .....</b>	<b>28</b>
4.1. Sunspots .....	28
4.2. Επεξεργαστής.....	28
4.3. Δίαυλοι επικοινωνίας του Main Board .....	29
4.4. Μνήμη .....	30
<b>ΚΕΦΑΛΑΙΟ 5 .....</b>	<b>32</b>
<b>Zigbee.....</b>	<b>32</b>
5.1. Zigbee.....	32
5.2. Η στοίβα πρωτοκόλλων του ZigBee .....	32
5.3. Τύποι συσκευών .....	34
5.4. Τοπολογίες δικτύων .....	34
5.5. Φυσικό επίπεδο (Physical layer, PHY) .....	35
5.5.1. Χαρακτηριστικά καναλιών και διαμόρφωσης.....	35
5.5.2. Πρόσβαση στο κανάλι.....	36
5.5.3. Υπηρεσίες δεδομένων.....	36
5.6. Επίπεδο ελέγχου πρόσβασης στο μέσο (Medium access control layer, MAC) 37	
5.6.1. Υπηρεσίες δεδομένων.....	37
5.6.2. Πλαίσια MAC .....	38
5.6.2.1. Πεδίο έλεγχου πλαισίου.....	39
5.6.2.2. Πεδίο αριθμού ακολουθίας.....	39
5.6.2.3. Πεδία διευθύνσεων .....	40
5.6.2.4. Ακολουθία ελέγχου πλαισίου (FCS).....	40
5.6.2.5. Ωφέλιμο φορτίο MAC .....	40

5.6.3. Λειτουργίες επιπέδου MAC .....	40
5.7. Επίπεδο δικτύου (Network layer, NWK) .....	41
5.7.1. Πλαίσια επιπέδου δικτύου .....	42
5.7.2. Πίνακες γειτόνων .....	44
5.7.3. Σύνδεση σε ένα δίκτυο και αποχώρηση από αυτό .....	44
5.7.4. Δημιουργία ενός δικτύου .....	46
5.7.5. Δρομολόγηση πλαισίων .....	47
5.7.6. Ασφάλεια .....	48
5.8. Επίπεδο εφαρμογών (Application layer, APL) .....	50
5.8.1. Υποεπίπεδο υποστήριξης εφαρμογών (Application support sublayer, APS).....	50
5.8.1.1. Υπηρεσίες δεδομένων υποεπιπέδου APS .....	51
5.8.1.2. Υπηρεσίες διαχείρισης υποεπιπέδου APS .....	52
5.8.1.3. Πλαίσια υποεπιπέδου APS.....	53
5.8.1.4. Μετάδοση, παραλαβή και επιβεβαίωση πλαισίων.....	54
5.8.1.5. Ασφάλεια στο υποεπίπεδο APS.....	56
5.8.2. Πλαίσιο εφαρμογών (Application framework, AF) .....	56
5.8.2.1. Τα προφίλ του ZigBee .....	57
5.8.2.2. Πλαίσια εντολών AF (Application framework) .....	58
5.8.3. Αντικείμενα συσκευής ZigBee (ZigBee Device Objects, ZDO).....	58
<b>ΚΕΦΑΛΑΙΟ 6 .....</b>	<b>61</b>
<b>ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ WEB ΕΦΑΡΜΟΓΩΝ .....</b>	<b>61</b>
6.1. Apache.....	61
6.2. PHP.....	63
6.3. MySQL.....	64
6.4. PhpMyAdmin .....	66
<b>ΚΕΦΑΛΑΙΟ 7 .....</b>	<b>68</b>
<b>Arduino .....</b>	<b>68</b>
7.1 Τι είναι το Arduino.....	68
7.2. Γιατί Arduino .....	68
7.3. Τεχνικά Χαρακτηριστικά του Arduino Uno .....	69

<b>ΚΕΦΑΛΑΙΟ 8</b> .....	72
<b>ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΑΝΤΙΜΕΤΩΠΙΣΤΗΚΑΝ</b> .....	72
8.1. SunSPOT SDK.....	72
<b>ΚΕΦΑΛΑΙΟ 9</b> .....	74
<b>ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ</b> .....	74
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b> .....	75
<b>ΠΑΡΑΡΤΗΜΑ</b> .....	77
<b>ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ</b> .....	77
Script Python για την διασύνδεση του Arduino Uno με την βάση δεδομένων.....	77
Κώδικας PHP της εφαρμογής web.....	79
Κώδικας Java που υπάρχει στον σταθμό βάσης SunSPOT.....	81
SendDataDemoGuiHostApplication .....	81
sql.java.....	87
Κώδικας που υπάρχει στους αισθητήρες φωτεινότητας SunSPOT .....	87
Κώδικας που υπάρχει στους αισθητήρες θερμοκρασίας SunSPOT.....	90
Κώδικας που υπάρχει στους αισθητήρες φωτεινότητας SunSPOT .....	94
Κώδικας που υπάρχει στο Arduino .....	98
Κώδικας της βάσης δεδομένων .....	99

## ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1.1 Το Internet of Things.....	5
Σχήμα 1.2 Internet of Things – Μία συμβιοτική αλληλεπίδραση μεταξύ του εικονικού, του ψηφιακού και του πραγματικού κόσμου, και των κοινωνιών τους. ....	6
Σχήμα 2.2 Τυπικός υποσταθμός BEMS.....	12
Σχήμα 3.1 Η στοίβα πρωτοκόλλων του Android.....	22



Σχήμα 4.1 Συνδεσμολογία SunSPOT .....	29
Σχήμα 5.1 Η στοίβα πρωτοκόλλων του ZigBee .....	33
Σχήμα 5.2 Τοπολογίες Δικτύων .....	35
Σχήμα 5.3 Δομή πλαισίου MAC .....	39
Σχήμα 5.4 Δομή πλαισίου επιπέδου Δικτύου .....	44
Σχήμα 5.5 Δομή πλαισίου ασφαλείας δικτύου .....	49
Σχήμα 5.6 Δομή βοηθητικής επικεφαλίδας .....	49
Σχήμα 5.7 Δομή πλαισίου ελέγχου ασφαλείας .....	50
Σχήμα 5.8 Μοντέλο αναφοράς APS .....	51
Σχήμα 5.9 Πλαίσιο APS .....	53
Σχήμα 5.10 Έλεγχος πλαισίου .....	53
Σχήμα 5.11 Ασφαλές πλαίσιο APS .....	56
Σχήμα 5.12 Πλαίσιο εντολών AF .....	58
Σχήμα 7.1 Χαρτογράφηση του Arduino Uno .....	70
Σχήμα 7.2 Τα τεχνικά χαρακτηριστικά του Arduino Uno .....	71

# ΚΕΦΑΛΑΙΟ 1

## INTERNET OF THINGS

### 1.1. Internet of Things – Η ιδέα

Η φράση Internet of Things για πρώτη φορά ειπώθηκε στη δύση του 20<sup>ου</sup> αιώνα από τους ιδρυτές του MIT Auto-ID Center. Ο ίδιος ο όρος Auto – ID αναφέρεται σε τεχνολογίες ταυτοποίησης που χρησιμοποιούνται σε οποιαδήποτε βιομηχανία ούτως με σκοπό την αυτοματοποίηση, την ελαχιστοποίηση των λαθών και την βελτιστοποίηση της αποδοτικότητας. Πιο συγκεκριμένα, αυτές οι τεχνολογίες περιλαμβάνουν barcode, έξυπνες κάρτες, αναγνώριση φωνής, αισθητήρες, και βιομετρικά στοιχεία. Αρχής γενομένης όμως από το 2003, ο παραπάνω όρος αναφέρεται κυρίως σε ταυτοποίηση ραδιοσυχνοτήτων, τεχνολογία γνωστή σήμερα, ως RFID.

Η έννοια τώρα του Internet of Things ήρθε στο προσκήνιο το 2005, όταν η Διεθνής Ένωση Τηλεπικοινωνιών (ITU) δημοσίευσε την πρώτη της αναφορά για το θέμα (The Internet of Things, ITU, November 2005). Η αναφορά της ITU υιοθετεί μία κατανοητή και ολοκληρωμένη προσέγγιση στο θέμα, υπαινίσσοντας ότι το Internet of Things θα συνδέσει τον κόσμο των αντικειμένων με έναν τρόπο τόσο αισθητήριο, όσο και ευφυή με το να συνενώνει τεχνολογίες ταυτοποίησης («ετικέτες» αντικειμένων), αισθητήρες και Ασύρματα Δίκτυα Αισθητήρων («αίσθηση» αντικειμένων), ενσωματωμένα συστήματα («σκεπτόμενα» αντικείμενα) και νανοτεχνολογία («συρρίκνωση» αντικειμένων). Στην εν λόγω αναφορά, έγινε λόγος για την νανοτεχνολογία σε συνδυασμό με τις Τεχνολογίες Πληροφορίας και Επικοινωνιών, πράγμα που αγγίζει την έννοια της σύγκλισης στις τεχνολογίες, πράγμα που για πρώτη φορά αναφέρθηκε το 2002.

Ένας πρώιμος ορισμός του Internet of Things από τον συνιδρυτή του MIT Auto-ID Center, Kevin Ashton το 1999 είναι κάπως περιγραφικός και αναφέρεται εδώ:

«Οι ηλεκτρονικοί υπολογιστές σήμερα, και κατ' επέκταση το Διαδίκτυο, είναι σχεδόν πλήρως εξαρτώμενοι από τα ανθρώπινα όντα για τις πληροφορίες. Σχεδόν όλα από τα σχεδόν 50 petabytes δεδομένων διαθέσιμα στο Internet, συλλέχθηκαν ή δημιουργήθηκαν από ανθρώπους – μέσω τυπογράφησης, πατώντας ένα κομβίο εγγραφής, τραβώντας μία ψηφιακή φωτογραφία, ή σαρώνοντας ένα barcode. Τα συμβατικά διαγράμματα στο Internet... αφήνοντας έξω τους πλέον πολυάριθμους και σημαντικούς δρομολογητές όλων – τους ανθρώπους. Το πρόβλημα είναι, ότι οι άνθρωποι, έχουν περιορισμένο χρόνο, προσοχή και ακρίβεια – πράγμα που σημαίνει ότι δεν είναι πολύ καλοί στο να καταγράφουν δεδομένα για αντικείμενα στον πραγματικό κόσμο. Και αυτό είναι πολύ μεγάλη υπόθεση. Έχουμε φυσική υπόσταση, το ίδιο το περιβάλλον μας... Δεν μπορείς να φας bits, να τα κάψεις για να μείνεις ζεστός, ή να τα βάλεις στην δεξαμενή πετρελαίου σου. Οι ιδέες και οι πληροφορίες είναι σημαντικές, αλλά τα αντικείμενα έχουν πολύ μεγαλύτερη σημασία. Κι όμως, σήμερα η Τεχνολογία της Πληροφορίας είναι τόσο εξαρτημένη από τα δεδομένα που προήλθαν από ανθρώπους, που οι ηλεκτρονικοί υπολογιστές γνωρίζουν περισσότερα για ιδέες, παρά για αντικείμενα. Εάν είχαμε ηλεκτρονικούς υπολογιστές που να γνώριζαν όλα αυτά που υπάρχουν για τα αντικείμενα – να χρησιμοποιούν τα δεδομένα που συλλέγουν χωρίς καμία βοήθεια από εμάς – θα είμασταν ικανοί να εντοπίσουμε και να μετρήσουμε οτιδήποτε, και να μειώσουμε σε πάρα πολύ μεγάλο βαθμό τα απόβλητα, τις απώλειες και το κόστος. Θα γνωρίζαμε πότε τα αντικείμενα χρειάζονται αντικατάσταση, επισκευή ή ανάκληση, και πότε είναι φρέσκα ή ληγμένα. Το Internet of Things έχει την δυνατότητα να αλλάξει τον κόσμο, όπως και το Internet έκανε πριν από αυτό. Ίσως και πολλά παραπάνω.

## **1.2. Πράγματα - Αντικείμενα**

Για να ορίσουμε τα αντικείμενα και να αναγνωρίσουμε τι είναι ένα συγκεκριμένο αντικείμενο και τι αντιπροσωπεύει στο υπερσύνολο του μελλοντικού Διαδικτύου απαιτεί μία πολύ προσεκτική ανάλυση.

Μία ανάλυση του τύπου τι είχαν να πουν φιλόσοφοι όπως ο Αριστοτέλης και ο Ιωάννης ο Φιλόππος για το πώς οι φιλοσοφικές τους σκέψεις θα έχουν υπόσταση στο μέλλον.

Ο Αριστοτέλης, στο έργο του «Κατηγορίαι», δίνει μία εμτυπωσιακή ερμηνεία για τα αντικείμενα που είναι δηλαδή τα Όντα. Σύμφωνα με την άποψή του, τα όντα χωρίζονται σε δέκα

διακριτές κατηγορίες. Εμπεριέχουν την ουσία, την ποιότητα, την ποσότητα και την σχέση τους με τα υπόλοιπα. Από αυτές τις κατηγορίες των όντων, η πρώτη, η Ουσία, είναι αυτή στην οποία ο Αριστοτέλης δίνει μία προνομιά θέση.

Διαχωρίζοντας τα αντικείμενα που έρχονται από την φύση, από τα αντικείμενα που προέρχονται από άλλες αιτίες. Ο Ιωάννης ο Φιλόπονος, σχολιάζοντας αυτήν την διάκριση, πρώτα χωρίζει τα αντικείμενα που έρχονται από την φύση σε άψυχα και έμψυχα.

Η κανονική φύση των έμψυχων πραγμάτων, είναι η μορφή τους, η οποία, σύμφωνα με τον Φιλόπονο είναι ταυτοποιημένη με την ψυχή, τον εγγενή τους ξενιστή. Από τον φιλοσοφικό ορισμό των πραγμάτων ο καθένας μπορεί να καταλήξει στο ότι η ίδια η λέξη δεν περιορίζεται στην ερμηνεία της στα υλικά αντικείμενα. Τουναντίον, ο ορισμός εφαρμόζεται σε εικονικά αντικείμενα και γεγονότα τα οποία είναι συνδεδεμένα σε «αντικείμενα».

Στο περίγραμμα του «Internet of Things», ένα αντικείμενο θα μπορούσε να οριστεί σαν μία πραγματική, με φυσική υπόσταση ή εικονική ψυφιακή υπόσταση η οποία υπάρχει και κινείται στον χωροχρόνο και είναι ικανή να αναγνωρίζεται και να ταυτοποιείται. Τα αντικείμενα είναι κοινώς ταυτοποιήσιμα είτε με το να τους ανατίθεται ταυτοτικός αριθμός, όνομα ή / και διεύθυνση.

### **1.3. Τι είναι το Internet of Things**

Το Internet of Things είναι ένα μέρος του Μελλοντικού Διαδικτύου και μπορεί να ορισθεί ως μία παγκόσμια, δυναμική, δικτυακή υποδομή με αυτορυθμιζόμενες δυνατότητες, βασισμένη σε διαλειτουργικά και τυποποιημένα πρωτόκολλα όπου τα φυσικά και εικονικά αντικείμενα έχουν ταυτότητα, φυσικά χαρακτηριστικά και εικονικές προσωπικότητες, χρησιμοποιούν διεπαφές με ευφύια, και είναι απρόσκοπτα ενσωματωμένα στο δίκτυο πληροφοριών.

Στο Internet of Things, τα αντικείμενα αναμένονται να γίνουν ενεργοί συμμετοχοί στις επιχειρήσεις, τις πληροφορίες και τις κοινωνικές διεργασίες, όπου και δύνανται να αλληλεπιδρούν μεταξύ τους και με το περιβάλλον τους με το να εναλλάσσουν δεδομένα και πληροφορίες που «ένιωσαν» στο περιβάλλον τους. Ενώ αντιδρούν αυτόνομα σε γεγονότα του «πραγματικού κόσμου», επηρεάζουν το περιβάλλον τους, αφού τρέχουν διεργασίες που θέτουν σε ενέργεια δράσεις και δημιουργούν υπηρεσίες με ή χωρίς την ανθρώπινη παρέμβαση. Διεπαφές

με τη μορφή υπηρεσιών διευκολύνουν τη διαδραστικότητα με αυτά τα «έξυπνα αντικείμενα» στο Διαδίκτυο, εξετάζοντας και αλλάζοντας την κατάσταση του και οποιαδήποτε πληροφορία άμεσα συσχετισμένη με αυτά, λαμβάνοντας υπόψη την ασφάλεια και θέματα ιδιωτικότητας.

#### **1.4. Το όραμα του Μελλοντικού Internet**

Το όραμα του Μελλοντικού Internet είναι βασισμένο σε τυποποιημένα πρωτόκολλα επικοινωνίας και πραγματεύεται την συγχώνευση των Δικτύων Υπολογιστών, του Internet of Media (IoM), του Internet of Services (IoS), και του Internet of Things σε μία κοινή παγκόσμια πλατφόρμα απρόσκοπτων δικτύων και δικτυωμένων αντικειμένων.

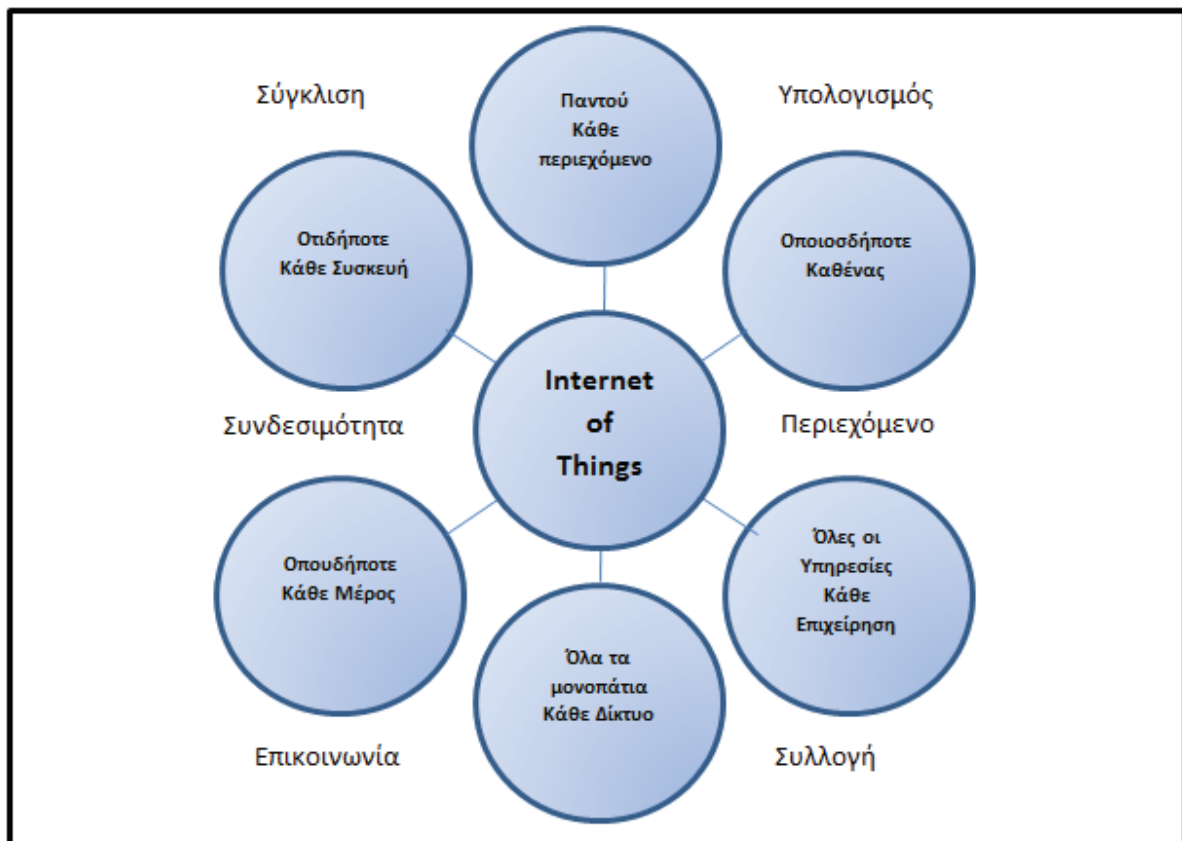
Το IoS υποδηλώνει ένα συστατικό από πλευράς λογισμικού το οποίο θα διανέμεται από διαφορετικά δίκτυα και το Διαδίκτυο, το οποίο θα βελτιώνει την συνεργασία μεταξύ παρόχους υπηρεσιών και καταναλωτές.

Το IoM θα αντιμετωπίσει τις προκλήσεις στην κλιμακωτή κωδικοποίηση βίντεο και επεξεργασία τρισδιάστατων (3D) βίντεο δυναμικά, και προσαρμοσμένα στην εκάστοτε κατάσταση του δικτύου, πράγμα που θα δώσει το έναυσμα για την ανάπτυξη καινοτόμων εφαρμογών.

Αυτό το μελλοντικό δίκτυο δικτύων θα εξαπλωθεί σαν μία υποδομή και δυναμικά θα επεκτείνεται και θα βελτιώνεται από τις ανάγκες που δημιουργούν τα αντικείμενα που συνδέονται το ένα με το άλλο. Στην πραγματικότητα, στις επικοινωνίες του Internet of Things δεν θα συμμετέχουν μόνο οι άνθρωποι, αλλά και το περιβάλλον στο οποίο βρίσκονται.

Η επικοινωνία θα γίνεται περισσότερο ανάμεσα στα τερματικά και τα κέντρα δεδομένων (data centre) παρά μεταξύ των κόμβων στα τρέχοντα δίκτυα. Η αύξηση της χωρητικότητας σε χαμηλότερο κόστος θα αποτελέσει την διάθεση των πληροφοριών των χρηστών ή των αντικειμένων, σε τοπικό επίπεδο. Αυτό, σε συνδυασμό με τις ανεπτυγμένες δυνατότητες επεξεργασίας και την δυνατότητα συνεχούς και αδιάκοπης σύνδεσης, θα κάνει τα τερματικά να αποκτήσουν τον κύριο ρόλο στις επικοινωνίες. Τα τελευταία θα έχουν την ικανότητα να δημιουργήσουν ένα τοπικό δίκτυο επικοινωνίας θα εξυπηρετούν σαν γέφυρα μεταξύ δικτύων επικοινωνίας, συνεπώς σε αστικά περιβάλλοντα, η χωρητικότητα των δικτύων επικοινωνίας, θα επεκτείνεται. Έτσι πολύ πιθανώς να καθοριστεί μία διαφορετική οπτική γωνία των αρχιτεκτονικών δικτύων. Το Μελλοντικό Διαδίκτυο επιδεικνύει υψηλά επίπεδα ετερογένειας, αφού πραγματεύεται τόσο με αντικείμενα, φυσικά και εικονικά, όσο και με επικοινωνιακά

πρωτόκολλα, συσκευές και μοντέλα συσκευών, γνωστικές (cognitive) ικανότητες και πολλά άλλα. Δηλαδή τελείως διαφορετικά αντικείμενα σε όρους λειτουργικότητας, τεχνολογίας και τομείς εφαρμογής, αναμένεται να ανήκουν στο ίδιο επικοινωνιακό περιβάλλον.

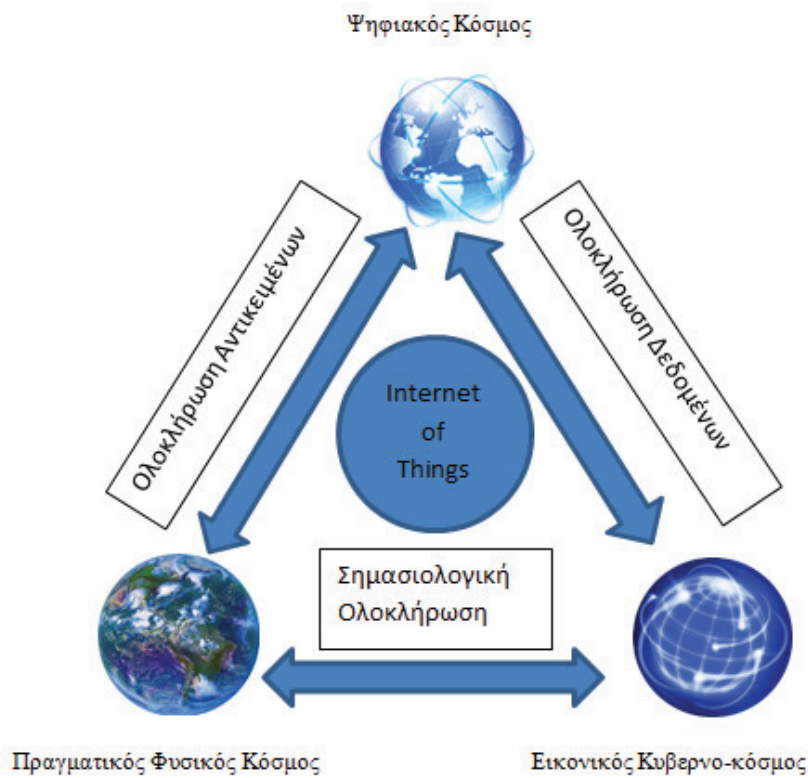


Σχήμα 1.1 Το Internet of Things

Το Internet of Things συνεπάγεται μία συμβιοτική διαδραστικότητα μεταξύ των πραγματικών / φυσικών, εικονικών / ψηφιακών κόσμων· οι φυσικές οντότητες έχουν ψηφιακά ομόλογά τους και εικονική αναπαράσταση. Τα αντικείμενα γίνονται γνωστικά και πλέον μπορούν να αισθάνονται, να επικοινωνούν, να αλληλεπιδρούν, να ανταλλάζουν δεδομένα, πληροφορία και γνώση. Μέσω της χρήσης ευφυών αλγορίθμων λήψης αποφάσεων σε εφαρμογές λογισμικού, κατάλληλες και γρήγορες απαντήσεις δύνανται να δωθούν σε φυσικά φαινόμενα, βασισμένες σε δεδομένα που συλλέχθηκαν από φυσικές οντότητες όπως αισθητήρες. Έτσι, δημιουργούνται νέες ευκαιρίες για εφαρμογές που ανταποκρίνονται σε επιχειρησιακές απαιτήσεις και δημιουργούνται νέες υπηρεσίες βασισμένες σε φυσικά, πραγματικά δεδομένα.

Οτιδήποτε, από άτομα, ομάδες ατόμων, κοινωνίες, αντικείμενα, προϊόντα, υπηρεσίες, διεργασίες, θα διασυνδέεται με το Internet of Things. Η συνδεσιμότητα θα είναι ένα είδος ευκολίας, διαθέσιμο σε όλους σε πολύ μικρό κόστος και δεν θα είναι ιδιοκτησία οποιασδήποτε ιδιωτικής οντότητας.

Υπό αυτό το όραμα και κάνοντας χρήση της ευφυίας στο σε μία δικτυακή υποδομή, τα αντικείμενα θα έχουν την ικανότητα ακόμα και να αντλούν την ενέργεια που τους χρειάζεται. Θα μπορούν να ρυθμιστούν αυτόνομα όταν εκτίθενται σε ένα νέο περιβάλλον, και να επιδείξουν μία ευφυή και γνωστική συμπεριφορά όταν αντιμετωπίσουν κάποια απρόβλεπτη κατάσταση.



Σχήμα 1.2 Internet of Things – Μία συμβιοτική αλληλεπίδραση μεταξύ του εικονικού, του ψηφιακού και του πραγματικού κόσμου, και των κοινωνιών τους.

## ΚΕΦΑΛΑΙΟ 2

### Building Energy Management System

#### 2.1. Ιστορία των BEMS (Building Energy Management Systems)

Τα BEMS έχουν εξελιχθεί παράλληλα με την εξέλιξη της μικροηλεκτρονικής και την επανάσταση που έχει γίνει στο χώρο των υπολογιστών. Αυτό έγινε επειδή τα BEMS είναι απλά μικροϋπολογιστικά συστήματα που χρησιμοποιούνται για τον έλεγχο και την παρακολούθηση κτιριακών αναγκών και υπηρεσιών.

Ο προκάτοχος των BEMS ήταν τα καλωδιωμένα κεντρικά συστήματα. Εμφανίστηκαν στις αρχές του 1960 και χρησιμοποιήθηκαν σε μεγάλα κτίρια. Το κεντρικό σύστημα ήταν ουσιαστικά μια επέκταση των συμβατικών καλωδίων σε μια κεντρική κονσόλα με διακόπτες, λυχνίες ένδειξης και καταγραφείς διαγραμμάτων τα οποία επέτρεπαν σε ένα χειριστή στην κονσόλα να παρακολουθεί απομακρυσμένες βιομηχανικές εγκαταστάσεις και τις ενδείξεις θερμοκρασίας. Δεν χρησιμοποιούνταν καθόλου υπολογιστές ή μικροηλεκτρονική και βασιζόνταν μόνο στον χειριστή να αλλάξει όποιες ρυθμίσεις χρειαζόνταν.

Αυτά τα καλωδιωμένα συστήματα βελτιώθηκαν όταν με τη χρήση της τεχνολογίας της τηλεφωνίας ήταν εφικτό ανεξάρτητα τμήματα της βιομηχανικής εγκατάστασης να συνδεθούν, μέσω πινάκων ελέγχου που συγκέντρωναν δεδομένα τοπικά στην εγκατάσταση, σε ένα κεντρικό πολύκλωνο καλώδιο που ξεκινούσε από την κεντρική κονσόλα και το οποίο διέτρεχε όλη την εγκατάσταση. Αυτή η σύνδεση εξοικονομούσε καλώδιο καθώς χρησιμοποιούσε το ίδιο πολύκλωνο καλώδιο για πολλούς πίνακες ελέγχου.

Το πρώτο BEMS βασισμένο σε υπολογιστή ξεπρόβαλε όταν η αλματώδης ανάπτυξη της μικροηλεκτρονικής επέτρεπε σε ένα τσιπ  $5 \text{ mm}^2$  να περιέχει εκατοντάδες τρανζίστορ. Τα πρώτα BEMS βασισμένα σε υπολογιστή εμφανίστηκαν το 1970 στις Η.Π.Α και ήταν κεντρικά συστήματα ενεργειακής διαχείρισης. Ο κεντρικός σταθμός ήταν βασισμένος σε ένα μικροϋπολογιστή, ο οποίος περιείχε όλη την υπολογιστική δύναμη ή «ευφυΐα» στο σύστημα,



ενώ οι υποσταθμοί ήταν «χαζοί» και αποτελούνταν από πίνακες παρόμοιους με τους προηγούμενους πίνακες BEMS που συγκέντρωναν δεδομένα, οι οποίοι συνδέονταν με αισθητήρες και διακόπτες.

Αυτά τα συστήματα κόστιζαν πολύ και χρησιμοποιούνταν μόνο σε μεγάλες εγκαταστάσεις. Παρόλο που αρχικά ήταν συσχετισμένα μόνο με τον έλεγχο και την παρακολούθηση της μονάδας κλιματισμού και εξαερισμού των εγκαταστάσεων μπορούσαν ακόμα να ελέγχουν τον φωτισμό, τους ανελκυστήρες και να παρακολουθούν την ασφάλεια του κτιρίου και τους συναγερμούς πυρκαγιάς. Στην πραγματικότητα τα BEMS χρησιμοποιούνταν μόνο στην διαχείριση μεγάλων και πολύπλοκων κτιριακών εγκαταστάσεων χωρίς να γίνεται μέσα από αυτά εξοικονόμηση ενέργειας. Αυτά τα πρώτα BEMS υπήρχαν πριν από την ενεργειακή κρίση του 1973.

Παρόλο που τα πρώτα BEMS είχαν την δυνατότητα να παρακολουθούν και να ελέγχουν συναγερμούς πυρκαγιάς και ασφάλειας σπάνια χρησιμοποιούνταν γι' αυτό καθώς ξεχωριστά συστήματα ήταν αφοσιωμένα για τον συγκεκριμένο σκοπό. Υπάρχουν προβλήματα στο να μπορέσουν να ενσωματωθούν συστήματα συναγερμού πυρκαγιάς και ασφάλειας στα BEMS ακόμα και σήμερα, κυρίως λόγω των διαφορετικών προτύπων και των κατασκευαστικών διαφορών που υπάρχουν στις εταιρίες που παράγουν τέτοια συστήματα, και όχι εξαιτίας της τεχνολογίας. Από το 1980 η αλματώδης εξέλιξη των LSI και VLSI οδήγησε στο να μπορέσουν να τοποθετούνται χιλιάδες τρανζίστορ σε ένα τσιπ κάνοντας τα και παράλληλα πιο οικονομικά. Αποτέλεσμα αυτής της εξέλιξης ήταν οι υποσταθμοί να περιέχουν μικροεπεξεργαστές κερδίζοντας έτσι σε υπολογιστική δυνατότητα κάνοντας τους έτσι «ευφυής». Αυτό επιτρέπει να λειτουργούν μόνοι τους ή να γίνουν ανεξάρτητοι υποσταθμοί οι οποίοι εξαρτώνται από τον κεντρικό σταθμό μόνο για ένα μικρό χρονικό διάστημα της λειτουργίας τους. Αυτοί οι υποσταθμοί έχουν σημαντικά μεγαλύτερες δυνατότητες ελέγχου σε σχέση με τους παλιούς υποσταθμούς, οι οποίοι είχαν μόνο δυνατότητα παρακολούθησης και καταγραφής. Πραγματικά αυτοί οι υποσταθμοί μπορούσαν από μόνοι τους να ελέγχουν ένα μικρό κτίριο ενώ ήταν και οικονομική η εγκατάσταση τους σε μικρά και μεσαία κτίρια.

Οι κεντρικοί υποσταθμοί μπορούν να επικοινωνούν με πολλούς υποσταθμούς όταν χρειάζεται είτε μέσω τοπικών επικοινωνιακών δικτύων ή με απομακρυσμένους υποσταθμούς μέσω τηλεφωνικών καλωδίων, LANs και διαδικτύου.

Καθώς οι μικροεπεξεργαστές γίνονται όλο και πιο ισχυροί και λιγότερο ακριβοί έτσι και οι υποσταθμοί γίνονται πιο μικροί σε μέγεθος και οικονομικοί ώστε να μπορούν να ελέγχουν ο καθένας επιμέρους τμήματα της εγκατάστασης. Οι κατασκευαστές BEMS προμηθεύουν κατασκευαστές εγκαταστάσεων με μικρούς υποσταθμούς ώστε να μπορέσουν να τους εντάξουν στην κατασκευή των εγκαταστάσεων τους κάνοντας πιο εύκολη και απλή την μετέπειτα σύνδεση τους με το σύστημα επικοινωνίας των BEMS όταν παραδοθεί η εγκατάσταση.

Δίκτυα επικοινωνίας έχουν αναπτυχθεί σε τέτοιο σημείο που επιτρέπουν την επικοινωνία και τον έλεγχο διάφορων εξοπλισμών του κτιρίου σε επίπεδο διακόπτων φωτισμού και ηλεκτρικής πρίζας.

## **2.2. Πλεονεκτήματα BEMS**

Ένα από τα κύρια πλεονεκτήματα των BEMS είναι η συνεχής παρακολούθηση της εγκατάστασης και η δυνατότητα για μετέπειτα μελέτη των δεδομένων που προέρχονται από αυτήν. Αυτό επιτρέπει στους μηχανικούς και στους τεχνικούς την δυνατότητα να κατανοήσουν καλύτερα τα κτίρια και τις εγκαταστάσεις και οδηγεί συχνά σε βελτίωση αυτών και σε εξοικονόμηση ενέργειας. Αυτή η εξοικονόμηση ενέργειας μπορεί να ελεγχθεί καθώς τα BEMS μπορούν να παρακολουθούν και να καταγράφουν τα δεδομένα από μετρητές καυσίμων και ηλεκτρικού ρεύματος.

Ένα επιπλέον πλεονέκτημα των BEMS είναι ότι η πληροφορία μπορεί να μεταφερθεί μέσω τηλεφωνικών γραμμών από απομακρυσμένες εγκαταστάσεις στον κεντρικό σταθμό. Έτσι εξοικονομείται σημαντικός χρόνος και κόπος στην περίπτωση που παρουσιαστεί κάποιο πρόβλημα καθώς τότε θα έπρεπε κάποιος τεχνικός να επισκεφτεί τις απομακρυσμένες εγκαταστάσεις και να τις ελέγξει.

Ακόμα τα BEMS μπορούν να μειώσουν το ανθρώπινο δυναμικό που χρειαζόταν παλιότερα καθώς ένας χειριστής τώρα μπορεί από τον κεντρικό σταθμό να ελέγχει όλη την εγκατάσταση αφού έχει την δυνατότητα οποιοδήποτε πρόβλημα δημιουργηθεί σε κάποιο υποσταθμό να μπορέσει να το αντιληφθεί αμέσως, καθώς και να το διορθώσει σε πολλές περιπτώσεις. Επίσης μέσω των BEMS έχουμε καλύτερη παρακολούθηση όλης της εγκατάστασης με αποτέλεσμα την καλύτερη συντήρηση αυτής.

### 2.3. Υποσταθμοί

Ο υποσταθμός καθορίζεται ως η μονάδα με εισόδους και εξόδους που ελέγχουν τις εγκαταστάσεις αλλά δεν έχει πληκτρολόγιο και οθόνη ενώ μπορεί να έχει μια μικρή οθόνη ένδειξης και πλήκτρα με τοξάκια που επιτρέπουν περιορισμένη πρόσβαση. Οι είσοδοι του υποσταθμού προέρχονται από αισθητήρες και διακόπτες. Ο υποσταθμός χρησιμοποιεί αυτές τις ενδείξεις αναλόγως είτε για να εκτελέσει βρόγχους ελέγχου και βελτιστοποίησης είτε για άλλους υπολογισμούς. Η έξοδος μετά τον έλεγχο που κάνει ο υποσταθμός πηγαίνουν στο τμήμα εξόδου αυτού για να αλλάξει πιθανώς το άνοιγμα μιας βαλβίδας ή να αύξηση τις στροφές σε ένα κινητήρα.

### 2.4. «Έξυπνοι» υποσταθμοί

Οι παλιότεροι υποσταθμοί ονομάζονταν DGP (Data Gathering Panel) ή FPU (Field Processing Unit) και ο ρόλος τους ήταν να δέχονται σήματα εισόδου και εξόδου να κάνουν μια μικρή επεξεργασία και να τα προωθούν στον κεντρικό σταθμό. Οι σημερινοί υποσταθμοί έχουν δικό τους μικροεπεξεργαστή και μπορούν να θεωρηθούν «έξυπνοι» υποσταθμοί. Αυτοί οι υποσταθμοί μπορούν να έχουν μόνο ένα κεντρικό κύκλωμα ή πολλά τυπωμένα κυκλώματα ή κάρτες συνδεδεμένα σε μια βάση μέσα σε ένα μεταλλικό πινάκα όπως φαίνεται στο.

Στους πίνακες οι οποίοι περιέχουν πολλά τυπωμένα κυκλώματα κάθε κύκλωμα έχει μια συγκεκριμένη λειτουργία για εισόδους, για εξόδους, για επικοινωνία κ.α. Οι πίνακες αυτοί δίνουν την δυνατότητα επέκτασης στον χρήστη προσθέτοντας κυκλώματα για να μπορεί να λαμβάνει περισσότερες εισόδους καθώς και την δυνατότητα να μπορεί να αλλάζει συγκεκριμένα κυκλώματα και όχι όλο τον πίνακα σε περίπτωση που παρουσιαστεί κάποια βλάβη.

Οι υποσταθμοί με μόνο ένα κύκλωμα έχουν περιορισμένο αριθμό εισόδων-εξόδων και δεν υπάρχει η δυνατότητα επέκτασης τους. Ορισμένοι όμως είναι πολλοί μικροί και φτηνοί και μπορούν να χρησιμοποιηθούν αποκλειστικά για συγκεκριμένες λειτουργίες όπως ο έλεγχος ενός λέβητα.

Οι μονοί πίνακες συνήθως περιέχονται μέσα σε μεταλλικά κουτιά τα οποία ενσωματώνονται μέσα σε ηλεκτρικούς πίνακες ή πίνακες ελέγχου μαζί με τους ανάλογους διακόπτες και τις ανάλογες συνδέσεις. Οι πίνακες που περιέχουν πολλά τυπωμένα κυκλώματα είναι μέσα σε δικούς τους ξεχωριστούς ηλεκτρικούς πίνακες. Ανεξαρτήτως πάντως από τον τύπο

του υποσταθμού όλα τα κυκλώματα και τα ηλεκτρονικά του πρέπει να προστατεύονται από την σκόνη και την υγρασία μέσα στις εγκαταστάσεις, για να αποτραπεί τυχόν βραχυκύκλωμα το οποίο μπορεί να καταστρέψει τα τυπωμένα κυκλώματα. Για να έχουμε μια εκτίμηση του βαθμού προστασίας που παρέχει ένα περίβλημα ή ένας μεταλλικός πίνακας χρησιμοποιείται ένας κωδικός που αρχίζει με τα γράμματα IP και ακολουθούν δυο ψηφία. Το πρώτο ψηφίο έχει αρίθμηση από το 0 μέχρι το 6 και δηλώνει την προστασία που παρέχει το περίβλημα από την σκόνη ( 0 = καμία προστασία και 6 = τελείως αεροστεγές ), ενώ το δεύτερο ψηφίο έχει αρίθμηση από το 0 μέχρι το 8 και δηλώνει την προστασία έναντι στην διείσδυση νερού ( 0 = καμία προστασία και 8 = προστασία σε περίπτωση βύθισης ).

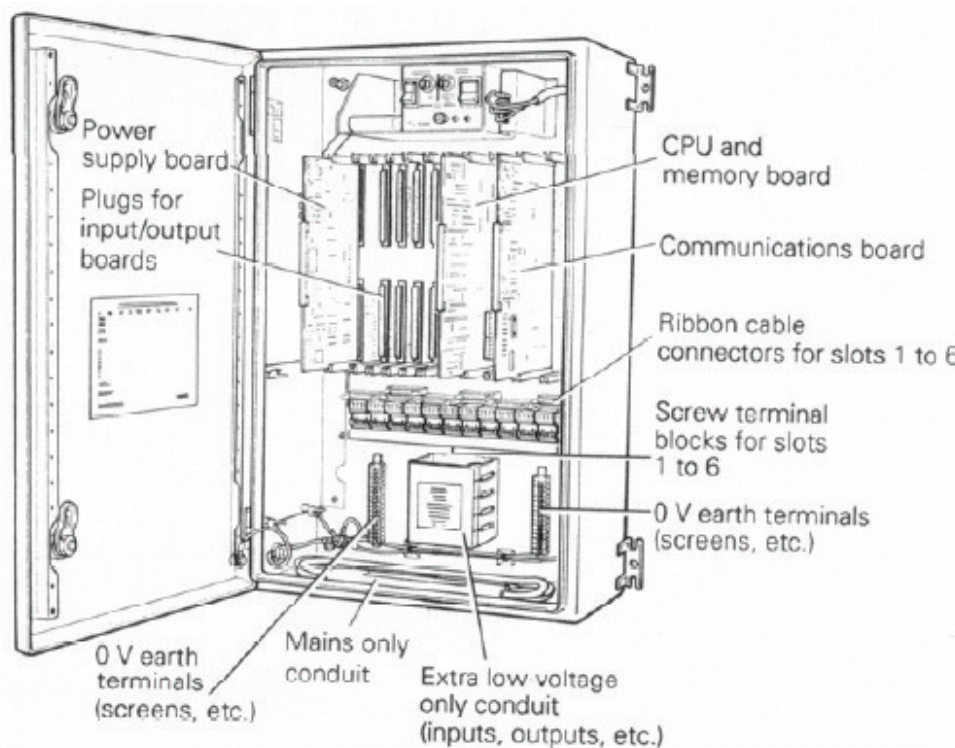
Οι μεταλλικοί πίνακες προσφέρουν προστασία έναντι των ηλεκτρομαγνητικών παρεμβολών και του θορύβου το οποίο βοηθάει την απόδοση των μικροεπεξεργαστών οι οποίοι λειτουργούν με μικρά σήματα τάσης. Αυτό είναι πολύ σημαντικό όταν οι υποσταθμοί είναι εγκαταστημένοι σε χώρο που υπάρχουν κινητήρες, λαμπτήρες εκκένωσης οι οποίοι παράγουν σημαντική ηλεκτρομαγνητική ακτινοβολία.

Ορισμένοι μικροί υποσταθμοί έχουν μικρά πληκτρολόγια ή τηλεχειριστήρια και οθόνες ένδειξης ενός ή δυο γραμμών ώστε να επιτρέπουν περιορισμένη πρόσβαση στα δεδομένα του υποσταθμού από το προσωπικό που βρίσκεται στο χώρο. Η πρόσβαση στα δεδομένα και στα προγράμματα του υποσταθμού μπορεί να περιοριστεί με την χρήση κωδικών ασφαλείας.

## **2.6. Δομή υποσταθμού**

Στο σχήμα 2.2 φαίνεται ένα απλοποιημένο διάγραμμα ενός υποσταθμού με ένα μόνο κεντρικό κύκλωμα χωρίς το κάλυμμα του ώστε να φαίνονται τα βασικά στοιχεία του

τυπωμένου κυκλώματος. Ο υποσταθμός παίρνει εναλλασσόμενο ρεύμα 240 V και με εσωτερικό μετασχηματιστή το μετατρέπει σε 24 V συνεχές για να τροφοδοτήσει τα κυκλώματα του. Σε περίπτωση διακοπή του ρεύματος ή κλεισίματος της κεντρικής



Σχήμα 2.2 Τυπικός υποσταθμός BEMS

παροχής για εργασία συντήρησης ο υποσταθμός τροφοδοτείται από εφεδρική μπαταρία. Χωρίς την εφεδρική μπαταρία όλα τα προγράμματα που υπάρχουν στον υποσταθμό θα χαθούν. Γι' αυτό είναι σημαντικό η μπαταρία να είναι αξιόπιστη, να έχει μεγάλη διάρκεια ζωής και να έχει μεγάλη χωρητικότητα ώστε να μπορεί να τροφοδοτεί τον υποσταθμό για αρκετό χρονικό διάστημα μέχρι να αποκατασταθεί η κεντρική παροχή ρεύματος. Η θύρα επικοινωνίας είναι το σημείο στο οποίο μπορεί να διασυνδέσουμε τον υποσταθμό με ένα μόντεμ ή ένα LAN. Για λόγους επικοινωνίας με άλλους υποσταθμούς κάθε υποσταθμός έχει τον δικό του μοναδικό αριθμό και ονομασία. Αν ο υποσταθμός δεν έχει κάποιο χειριστήριο ή κάποια οθόνη ένδειξης μια επιπλέον θύρα υπάρχει στον πίνακα για την σύνδεση του με κάποιο φορητό υπολογιστή για προγραμματισμό ή αλλαγή των λειτουργιών του.

## 2.7. Ο κεντρικός σταθμός

Ο κεντρικός σταθμός είναι η καρδιά των BEMS και το βασικό κανάλι επικοινωνίας με τον χειριστή του BEMS. Περιέχει το λογισμικό και όλα τα δεδομένα που σχετίζονται με τις λειτουργίες της εγκατάστασης. Η εξέλιξη των VLSI έκανε δυνατή την αντικατάσταση των παλιών ακριβών και μεγάλων σε μέγεθος υπολογιστών με μικρή υπολογιστική δυνατότητα που χρησιμοποιούνταν παλιότερα στα BEMS με τους τυπικούς ηλεκτρονικούς υπολογιστές που έχουμε σήμερα σπίτι μας. Σε σύγκριση με τους υποσταθμούς ο κεντρικός σταθμός με τον υπολογιστή έχει μεγαλύτερη επεξεργαστική ισχύ, μεγαλύτερη μνήμη και μεγαλύτερο αποθηκευτικό χώρο. Επίσης όπως και μερικοί υποσταθμοί έτσι και ο κεντρικός σταθμός έχει την δυνατότητα επέκτασης μέσω καρτών PCI που συνδέονται στην μητρική του υπολογιστή ενώ έχει και οθόνη, πληκτρολόγιο, εκτυπωτή. Ακόμα μπορεί να συνδεθεί όπως και οι υποσταθμοί με μόντεμ ή σε δίκτυο LAN ώστε να μπορεί να επικοινωνεί με τους απομακρυσμένους υποσταθμούς.

### 2.7.1. Αισθητήρες στα BEMS

Ο υποσταθμός ενός BEMS είναι βασικά ένας μικροεπεξεργαστής ο οποίος επεξεργάζεται ψηφιακά ηλεκτρικά σήματα. Όμως το μεγαλύτερο μέρος του εξοπλισμού μιας κτιριακής εγκατάστασης την οποία ελέγχει ένα BEMS βασίζεται σε παραμέτρους όπως η θερμοκρασία, η πίεση ή η ροή και σπάνια σε ηλεκτρικά σήματα. Οπότε πρέπει οι παράμετροι αυτοί να μετρηθούν και να μετατραπούν σε ηλεκτρικά σήματα. Λειτουργία ενός αισθητήρα είναι η μέτρηση και μετατροπή των παραμέτρων αυτών σε ηλεκτρικά σήματα. Μερικοί από τους αισθητήρες που χρησιμοποιούνται είναι οι εξής:

- Αισθητήρες φωτεινότητας

Οι αισθητήρες φωτεινότητας χρησιμοποιούνται για την μέτρηση του στιγμιαίου επιπέδου μέτρησης του φωτός. Διαθέτουν φωτοδίοδο δίοδο υψηλής ευαισθησίας η οποία αντιδρά άμεσα στις επικρατούσες συνθήκες φωτός, ημιφωτός και σε ορισμένες περιπτώσεις σε υπέρυθρη ακτινοβολία.

Χρησιμοποιούνται κατά κόρον σε BEMS, μετεωρολογικούς σταθμούς, θερμοκήπια και συστήματα παρακολούθησης.

- Αισθητήρες θερμοκρασίας

Είναι οι πιο συνηθισμένοι αισθητήρες που χρησιμοποιούνται σε BEMS. Οι περισσότεροι αισθητήρες θερμοκρασίας χρησιμοποιούν θερμοζεύγος, θερμίστορ και RTD.

Οι αισθητήρες που χρησιμοποιούν θερμοζεύγος βασίζονται στην αρχή ότι όταν δύο διαφορετικά μέταλλα έρχονται σε επαφή, δημιουργείται ένα ρεύμα του οποίου η ένταση είναι ανάλογη της θερμοκρασίας ένωσης.

Οι αισθητήρες που χρησιμοποιούν θερμίστορ βασίζονται στην αρχή ότι στους ημιαγωγούς η ηλεκτρική τους αντίσταση αλλάζει με την θερμοκρασία. Η ηλεκτρική τους αντίσταση μειώνεται όταν αυξάνεται η θερμοκρασία.

Οι αισθητήρες που χρησιμοποιούν RTD βασίζονται στην ίδια αρχή με τα θερμίστορ μόνο που τώρα η αντίσταση τους αυξάνεται όταν αυξάνει η θερμοκρασία.

- Αισθητήρες υγρασίας ή υγρόμετρα

Χρησιμοποιούνται για να μετρήσουν την σχετική υγρασία ή το σημείο δρόσου. Υπάρχουν δύο τύποι: τα μηχανικά υγρόμετρα και τα ηλεκτρικά υγρόμετρα.

Τα μηχανικά υγρόμετρα βασίζονται στην αρχή ότι όταν ένα υγροσκοπικό υλικό π.χ ένα υγρό ευαίσθητο νάιλον, όταν εκτίθεται σε υδρατμούς συγκρατεί την υγρασία και διαστέλλεται.

Τα ηλεκτρικά υγρόμετρα χρησιμοποιούν είτε στοιχεία στα οποία αλλάζει η αντίσταση τους, όπως ένα αγωγίμο πλέγμα το οποίο περιβάλλεται από μια ουσία που απορροφά το νερό και η αγωγιμότητα του οποίου είναι ανάλογη του νερού που απορροφά, είτε στοιχεία στα οποία αλλάζει η χωρητικότητα τους, όπως μια λεπτή μεμβράνη από μη αγωγίμο υλικό στις δύο άκρες της οποίας είναι τοποθετημένα μεταλλικά ηλεκτρόδια και όλο μαζί είναι τοποθετημένο μέσα σε μια πλαστική κάψουλα. Η αλλαγή στην χωρητικότητα του αισθητήρα σε σχέση με την σχετική υγρασία είναι μη γραμμική.

- Αισθητήρες πίεσης

Ο αισθητήρας πίεσης συνήθως αντιδρά στη διαφορά πίεσης του μετρούμενο μέσου (νερό ή αέρα) και μιας πίεσης αναφοράς. Η πίεση αναφοράς μπορεί να είναι το απόλυτο κενό, η ατμοσφαιρική πίεση ή η πίεση σε ένα γειτονικό σημείο. Οι αισθητήρες που



χρησιμοποιούνται χωρίζονται σε δύο κατηγορίες: αισθητήρες υψηλής πίεσης και αισθητήρες χαμηλής πίεσης. Οι αισθητήρες υψηλής πίεσης χρησιμοποιούν συνήθως σωλήνες Bourdon και διαφράγματα ενώ οι αισθητήρες χαμηλής πίεσης χρησιμοποιούν εύκαμπτους μεταλλικούς σωλήνες ή μεγάλα διαφράγματα.

- Αισθητήρες ανίχνευσης κίνησης

Οι αισθητήρες ανίχνευσης κίνησης εντοπίζουν αν μέσα σε ένα δωμάτιο υπάρχουν άνθρωποι. Αποτέλεσμα αυτής της ανίχνευσης μπορεί να είναι το κλείσιμο του κλιματισμού και του φωτισμού για εξοικονόμηση ενέργειας, όταν ο χώρος δεν απασχολείται από ανθρώπους. Υπάρχουν δύο τύποι τέτοιων αισθητήρων : οι υπερηχητικοί και οι υπέρυθροι. Οι υπερηχητικοί αισθητήρες στέλνουν ένα χαμηλό υπερηχητικό σήμα για να ανιχνεύσουν την κίνηση ενώ οι υπέρυθροι αισθητήρες λαμβάνουν την θερμότητα που εκπέμπουν οι άνθρωποι όταν αυτοί κινούνται.



## ΚΕΦΑΛΑΙΟ 3

### ANDROID

#### 3.1. Τι είναι το Android

Το Android είναι μια στοίβα λογισμικού για κινητές συσκευές η οποία περιλαμβάνει λειτουργικό σύστημα, ενδιάμεσο λογισμικό (middleware) και βασικές εφαρμογές. Το Android τρέχει τον πυρήνα του λειτουργικού Linux και μέσω της δικιά του εργαλειοθήκης ανάπτυξης συστήματος λογισμικού (Software Development Kit), επιτρέπει στους κατασκευαστές να δημιουργούν πρωτοποριακές εφαρμογές. Αρχικά αναπτύχθηκε από την Google και αργότερα συνεχίστηκε σε συνεργασία με την Open Handset Alliance (OHA). Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού OHA, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής υλικού, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές ανοιχτής τηλεφωνίας.

#### 3.2. Ιστορική Αναδρομή

Η πρώτη έκδοση του Android SDK τον Νοέμβριο του 2007, χαρακτηρίστηκε από τους κατασκευαστές του σαν μια πρώτη ματιά στο SDK του Android, κάτι το οποίο πολλοί παράβλεψαν και βιάστηκαν να κατακρίνουν το Android σαν ένα προβληματικό σύστημα. Στην ουσία όμως το Android δεν παρουσίαζε προβλήματα τα οποία δεν παρουσιάζει οποιοδήποτε σύστημα σε τέτοια πρώιμη φάση. Έτσι το Σεπτέμβριο του 2008, η T-Mobile ανακοινώνει την διαθεσιμότητα του T-Mobile G1, του πρώτου έξυπνου τηλεφώνου (smartphone), βασισμένο στην πλατφόρμα του Android. Λίγες μέρες αργότερα (Οκτώβριο 2008), η Google ανακοινώνει την απελευθέρωση του SDK Release Candidate 1.0. Ακολούθησε τον Φεβρουάριο του 2009 η έκδοση 1.1 σαν μια ανανεωμένη έκδοση του 1.0. Μέχρι τότε το Android δεν υποστήριζε ακόμη την χρήση κουμπιών αφής, παρά μόνο την χρήση των κλασικών 'σκληρών' κουμπιών της συσκευής. Τον Μάιο του 2009 είχαμε την έκδοση Android 1.5, εν ονόματι 'Cupcake'.

Το 'Cupcake' εισάγει κάποια καινούργια χαρακτηριστικά και ανανεώσεις στην διεπιφάνεια χρήστη (User Interface): Ικανότητα για καταγραφή και παρακολούθηση βίντεο μέσα από την λειτουργία της βιντεοκάμερας, μεταφόρτωση βίντεο στο YouTube και φωτογραφιών στο Picasa απευθείας από το τηλέφωνο, καινούργιο μαλακό πληκτρολόγιο (αφής) με πρόβλεψη κειμένου Υποστήριξη προτύπου Bluetooth A2DP και AVRCP. Ικανότητα αυτόματης σύνδεσης σε μικροσυσκευή Bluetooth από μια συγκεκριμένη απόσταση. Καινούργια widgets και φάκελοι που μπορούν να δημοσιευτούν στην αρχική οθόνη. Κινούμενες μεταβάσεις οθόνης.

Το 'Donut', Android 1.6, ήρθε τον Σεπτέμβριο του 2009. Η έκδοση αυτή εισάγει κάποια καινούργια χαρακτηριστικά όπως:

- Βελτιωμένο Android Market
- Ενσωματωμένη φωτογραφική μηχανή, βιντεοκάμερα και διεπαφή (interface) γκαλερί
- Η γκαλερί επιτρέπει πλέον στους χρήστες την επιλογή πολλαπλών φωτογραφιών για διαγραφή
- Ανανεωμένη αναζήτηση με φωνή, με ταχύτερη απόκριση και βαθύτερη ολοκλήρωση
- με εγγενής (native) εφαρμογές, συμπεριλαμβανομένης της δυνατότητας να καλούμε επαφές
- Ανανεωμένη αναζήτηση με την δυνατότητα αναζήτησης σελιδοδεικτών, ιστορικού, επαφών και στο διαδίκτυο από την αρχική οθόνη
- Ανανεωμένη υποστήριξη τεχνολογιών για CDMA/EVDO, 802.1x, VPNs και με μηχανή μετατροπής κειμένου σε ομιλία (text-to-speech)
- Υποστήριξη για ανάλυση οθονών WVGA
- Βελτιώσεις στην ταχύτητα για αναζήτηση και για εφαρμογές φωτογραφικής μηχανής

Το Android 2.1 (Eclair MR1).

Ανάμεσα στις άλλες αλλαγές είναι και:

- Βέλτιστη ταχύτητα υλικού
- Υποστήριξη για περισσότερες οθόνες και αναλύσεις
- Βελτιωμένη διεπιφάνεια χρήστη
- Καινούργια διεπιφάνεια χρήσης για την μηχανή αναζήτησης και υποστήριξη του

- προτύπου HTML5
- Καινούργιες λίστες επαφών
- Καλύτερος λόγος άσπρου – μαύρου για φόντα
- Βελτιωμένοι χάρτες Google (google maps) 3.1.2
- Υποστήριξη Microsoft Exchange
- Ενσωματωμένη υποστήριξη flash για την Camera
- Ψηφιακή μεγέθυνση (zoom)
- Κλάση MotionEvent βελτιωμένη ώστε οι κατασκευαστές να μπορούν να παρακολουθούν αποτελεσματικότερα τα γεγονότα πολλαπλής αφής
- Ανανεωμένο εικονικό πληκτρολόγιο
- Bluetooth 2.1

Η έκδοση FROYO ανάμεσα σε άλλες αλλαγές περιλαμβάνει:

- Βελτιστοποιήσεις στην ταχύτητα γενικά του λειτουργικού συστήματος, στην μνήμη και στην απόδοση
- Ενσωμάτωση στην μηχανή αναζήτησης, της μηχανής Javascript του Chrome V8
- Αυξημένη υποστήριξη Microsoft Exchange (σε πολιτικές ασφαλείας, συγχρονισμού ημερολογίου, auto – discovery, GAL look-up, remote wipe)
- Βελτιωμένος προωθητής εφαρμογής (application launcher), με συντομεύσεις προς τις εφαρμογές τηλεφώνου και εφαρμογές της Μηχανής Αναζήτησης
- Πρόσδεση USB και λειτουργία δυναμικής ζώνης (hotspot) WiFi
- Ανανεωμένη εφαρμογή Αγοράς (Market) με αυτόματη ανανέωση
- Επιλογή για απαγόρευση πρόσβασης δεδομένων πάνω από ένα δίκτυο κινητής τηλεφωνίας
- Γρήγορη εναλλαγή ανάμεσα σε πολλαπλές γλώσσες του πληκτρολογίου και των λεξικών τους
- Φωνητική κλήση και διαμοιρασμός επαφών με Bluetooth
- Υποστήριξη για αριθμητικούς και αλφαριθμητικούς κωδικούς
- Η μηχανή αναζήτησης μπορεί να αποτυπώσει κινούμενα GIFs
- Υποστήριξη για πεδία μεταφόρτωσης αρχείων στην μηχανή αναζήτησης

- Υποστήριξη για εγκατάσταση εφαρμογών στην επεκτάσιμη μνήμη
- Το Android “Gingerbread”
- με την επανέκδοση του σε Android 2.3.3 τον Φεβρουάριο του 2011.
- Οι αλλαγές που έχουν γίνει είναι οι ακόλουθες:
- Βελτιωμένο UI για απλότητα και ταχύτητα
- Πιο γρήγορη, πιο διαισθητική εισαγωγή κειμένου
- Επιλογή λέξεων και αντιγραφή/επικόλληση με ένα άγγιγμα
- Βελτιωμένη ενεργειακή διαχείριση
- υποστήριξη NFC (Near Field Communication)
- Υποστήριξη video κλήσης
- Υποστήριξη του πρωτόκολλου WebM για αναπαραγωγή video

Το Android ‘Hoenycomb’ παρέχει επιπλέον:

- Διπύρηνους και τετραπύρηνους επεξεργαστές
- Βελτιωμένη υποστήριξη των ταμπλετών  
ανάπτυξη λογισμικού (scripting) για 3D, σε γλώσσα η οποία καλείται "Renderscript"
- Video chat μέσω Google Talk
- Google eBooks
- "Ιδιωτική περιήγηση"

Το Android ‘Ice Cream Sandwich’

- Face Unlock
- Android BeamVideo chat μέσω Google Talk
- Video chat μέσω Google Talk

Το Android ‘Jelly Bean’

- Νέα Widget
- Αναδιάταξη αρχικής οθόνης
- Ευέλικτα notifications
- Πληκτρολόγιο με δυνατότητα auto-complete των λέξεων

### 3.3. Γιατί Android

1. Είναι μια πραγματικά ανοιχτή, ελεύθερη πλατφόρμα ανάπτυξης, βασισμένη στο Linux.
2. Διαθέτει αρχιτεκτονική βασισμένη σε δομικά στοιχεία τα οποία μπορούν να τροποποιηθούν, να ολοκληρωθούν και να προσαρμοστούν στις ανάγκες κάθε κατασκευαστή και κατά συνέπεια χρήστη.
3. Πάμπολλες ενσωματωμένες υπηρεσίες που μπορούν να κάνουν την εμπειρία του χρήστη μοναδική, όπως υπηρεσίες βασισμένες στην τοποθεσία, πανίσχυρη SQL βάση δεδομένων, μηχανή αναζήτησης και χάρτες.
4. Αυτόματη διαχείριση του κύκλου ζωής μιας εφαρμογής, με πολλαπλές δικλίδες ασφαλείας ανάμεσα στα προγράμματα. Βελτιστοποιήσεις στον τομέα διαχείρισης μνήμης και χαμηλής κατανάλωσης σε τέτοιο βαθμό που δεν έχει ξανασυναντηθεί σε άλλο smartphone.
5. Υψηλής ποιότητας γραφικά και ήχος.
6. Φορητότητα ανάμεσα σε ένα ευρύ φάσμα ήδη υπάρχοντος υλικού αλλά και μελλοντικού. Αυτό έρχεται σαν απόρροια του γεγονότος ότι όλα τα προγράμματα γράφονται σε Java και εκτελούνται από την εικονική μηχανή Dalvik. Επιπρόσθετα οι οθόνες μπορούν να τροποποιηθούν κατάλληλα για να υποστηρίξουν οποιαδήποτε ανάλυση, μέγεθος και προσανατολισμό οθόνης.

Στην παρούσα πτυχιακή εργασία επιλέχθηκε το Android για την ανάπτυξη της εφαρμογής για τον απλούστατο λόγο ότι είναι ευρέως διαδεδομένο στην σημερινή αγορά εργασίας, είναι ανοιχτού κώδικα, και εφάπτεται με πλήθος μαθημάτων που διδάσκονται στο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Μεσολογίου, και πιο συγκεκριμένα στο Τμήμα Τηλεπικοινωνιακών Συστημάτων και Δικτύων.

### 3.4. Κινητά τηλέφωνα Android

Η πρώτη συσκευή που “έτρεχε” το λειτουργικό σύστημα Android ήταν το HTC Dream, γνωστό και ως T-Mobile G1, που έκανε την εμφάνιση του στα ράφια των καταστημάτων προς το τέλος του 2008.

Στη συνέχεια, πολλά νέα Android κινητά κάνουν την εμφάνισή τους με ταξύ αυτών, το Motorola DROID (έκδοση CDMA), το οποίο από το Νοέμβριο του 2009 διατίθεται αποκλειστικά από την Αμερικανική εταιρία τηλεπικοινωνιών, Verizon, ενώ η GSM έκδοση του, το Motorola Milestone, από διάφορους παροχείς κινητής τηλεφωνίας ανά τον κόσμο. Στην αυγή του 2010, η Google τάραξε τα νερά στον χώρο των τηλεπικοινωνιών με την ανακοίνωση του Google Nexus One, μιας καινοτόμου συσκευής που τρέχει την νεότερη έκδοση του Android (2.1 που θεωρείται κυρίως βελτίωση της 2.0 ή Eclair). Κατασκευασμένο από την HTC, το Nexus One ξεχωρίζει για τα Live Wallpapers (κινούμενα backgrounds νέου σχεδιασμού που αλληλεπιδρούν με τις κινήσεις δαχτύλου του χρήστη) και τη νέα αναβαθμισμένη Έκθεση εικόνων και βίντεο (gallery). Η συσκευή έχει, επίσης, το πλεονέκτημα της άμεσης και απροβλημάτιστης συνεργασίας με όλες τις υπηρεσίες της Google (Gmail, Google Contacts, Google Calendar κ.α.).

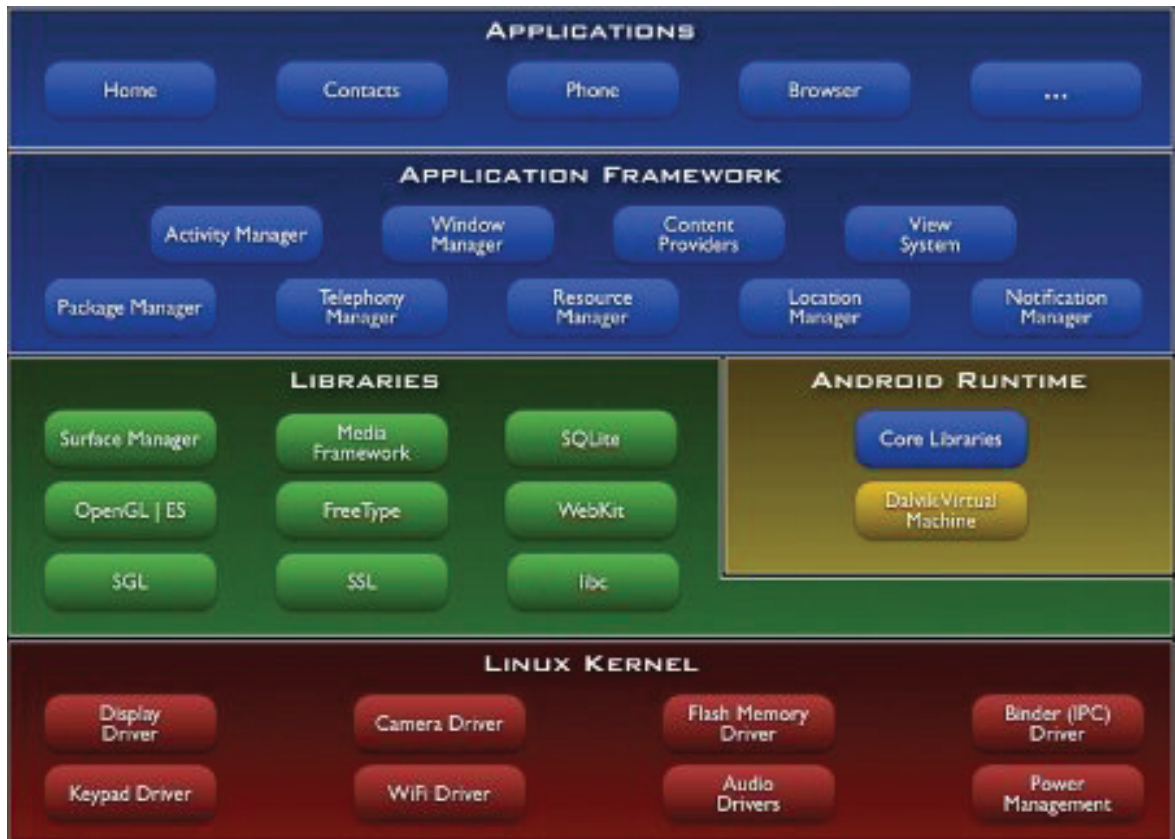
Από τότε και μέχρι σήμερα έχει εμφανιστεί στην αγορά μία πληθώρα συσκευών με λειτουργικό σύστημα Android.

### **3.5. Ταμπλέτες Android**

Τι είναι οι ταμπλέτες; Μία ταμπλέτα είναι ένας Η/Υ που αντί για ποντίκι και πληκτρολόγιο, διαθέτει ως κύρια μορφή εισαγωγής δεδομένων μια οθόνη αφής. Οι ταμπλέτες υπάρχουν από το 2001, αλλά έγιναν δημοφιλέστερες μετά την έλευση του Apple iPad, κερδίζοντας ένα σημαντικό μερίδιο του ενδιαφέροντος του κοινού. Ειδικά με την έλευση των Android ταμπλετών, που πλέον κυριαρχούν στην αγορά αναμένεται στο άμεσο μέλλον να αντικαθιστήσουν πλήρως τα netbooks, μπαίνοντας για τα καλά στην καθημερινότητά μας. Έρευνα της εταιρίας Strategy Analytics έδειξε πως ως το 2015 η αγορά θα έχει κέρδος από τις πωλήσεις των tablets 49 δισεκατομμύρια δολάρια. Η αγορά ταμπλετών τελευταίας τεχνολογίας θα είναι η τρίτη σε δύναμη αγορά, μετά από αυτή της τηλεόρασης και των προσωπικών ηλεκτρονικών υπολογιστών. Η εταιρία ερευνών ανέφερε πως, σύμφωνα με τις προβλέψεις, θα έχουν πουληθεί 149 εκατομμύρια μονάδες ως το 2015, δηλαδή οι πωλήσεις θα αυξηθούν κατά 6,5 φορές από 2010.

### **3.6. Αρχιτεκτονική**

Το Android δεν είναι απλά ένα Λειτουργικό Σύστημα. Είναι μια στοίβα λογισμικού η οποία αποτελείται από 4 επίπεδα και 5 ομάδες, τον πυρήνα Linux, τις βιβλιοθήκες, τον χρόνο εκτέλεσης και το πλαίσιο εφαρμογής.



Σχήμα 3.1 Η στοίβα πρωτοκόλλων του Android

### 3.6.1. Πυρήνας Linux

Αυτός είναι ο πυρήνας του λειτουργικού στον οποίο βασίζεται το Android και βρίσκεται στο χαμηλότερο επίπεδο. Ο πυρήνας υποστηρίζει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος όπως τη διαχείριση μνήμης, τη διαχείριση διεργασιών, τις λειτουργίες δικτύου, την ασφάλεια του λειτουργικού και ένα σύνολο οδηγών υλικού τους οποίους χρειάζεται για να τρέξει το σύστημα, όπως της οθόνης, της κάμερας κ.α.

### 3.6.2. Βιβλιοθήκες

Στο αμέσως ψηλότερο επίπεδο βρίσκουμε τις Βιβλιοθήκες. Όλες αυτές είναι γραμμένες



στην γλώσσα προγραμματισμού C και C++ και μεταγλωττίστηκαν για τη συγκεκριμένη αρχιτεκτονική υλικού που χρησιμοποιείται από τη συσκευή. Οι βιβλιοθήκες αυτές δεν είναι εφαρμογές που μπορούν να σταθούν από μόνες τους. Υπάρχουν για να μπορούν να κληθούν από προγράμματα υψηλότερου επιπέδου. Παραδείγματος χάριν, η SQLite βιβλιοθήκη παρέχει υποστήριξη έτσι ώστε μια εφαρμογή να χρησιμοποιήσει την αποθήκευση δεδομένων, η Webkit βιβλιοθήκη παρέχει λειτουργίες για το διαδικτυακό σερφάρισμα.

### **3.6.3. Χρόνος Εκτέλεσης και η εικονική μηχανή Dalvik**

Στο ίδιο επίπεδο με τις βιβλιοθήκες βρίσκεται ο χρόνος εκτέλεσης, ένα σύνολο βασικών βιβλιοθηκών που επιτρέπουν στους προγραμματιστές να γράψουν εφαρμογές χρησιμοποιώντας JAVA. Στην Java ως γνωστόν υπάρχει η λεγόμενη Java Virtual Machine στην οποία εκτελείτε ο κώδικας bytecode των εφαρμογών. Στο Android υπάρχει μια υλοποίηση αυτής της εικονικής μηχανής η οποία ονομάζεται Dalvik.

Μέσω της εικονικής μηχανής Dalvik τρέχουν οι εφαρμογές του Android καθε μία με τη δικιά της διεργασία και για αυτό το λόγο καμία εφαρμογή δεν έχει επαφή με άλλη, ενώ εκτελούνται ταυτόχρονα. Η Dalvik τρέχει .dex αρχεία, τα οποία είναι bytecodes που προέρχονται από πηγαίο κώδικα και βιβλιοθήκες της Java. Τα αρχεία dex ουσιαστικά αποτελούν συμπιεσμένα δεδομένα για εξοικονόμηση χώρου κατά την εκτέλεση.

Το Android είναι από τη φύση του multitasking λειτουργικό σύστημα και για αυτό επιτρέπει στις εφαρμογές του να τρέχουν σε πολλά νήματα ταυτόχρονα και να απασχολούν πολλές διαδικασίες εάν αυτό είναι αναγκαίο. Για να γίνει αυτό εφικτό η μηχανή Dalvik είναι σχεδιασμένη για να έχει ελάχιστο αντίκτυπο στη χρήση της μνήμης. Χάρη στον λιτό της σχεδιασμό, το σύστημα είναι σε θέση να τρέχει πολλές εικονικές μηχανές ταυτόχρονα.

### **3.6.4. Πλαίσιο Εφαρμογής (Application Framework)**

Πάνω από τις βιβλιοθήκες και το χρόνο εκτέλεσης Android, είναι το πλαίσιο εφαρμογής. Αυτό το επίπεδο εκθέτει διάφορες δυνατότητες του Android στους προγραμματιστές των εφαρμογών ώστε να τις χρησιμοποιήσουν στις εφαρμογές τους. Αυτό το πλαίσιο είναι προεγκατεστημένο στο Android, αλλά είναι επεκτάσιμο, αφού ο κάθε κατασκευαστής μπορεί να το συμπληρώσει με δικά του κομμάτια.



Τα σημαντικότερα δομικά στοιχεία του πλαισίου αυτού είναι:

- Διαχειριστής δραστηριοτήτων - Activity Manager: Υπεύθυνο για τον έλεγχο του χρόνου ζωής των εφαρμογών και για την διατήρηση μιας στοίβας που επιτρέπει την πλοήγηση του χρήστη σε προηγούμενες οθόνες.
- Παροχέας Περιεχομένου - Content Providers: Αυτά τα αντικείμενα περιέχουν δεδομένα που μπορούν να διαμοιραστούν μεταξύ εφαρμογών.
- Διαχειριστής Πόρων - Resource Manager: Οι πόροι, είναι οτιδήποτε υπάρχει σε ένα πρόγραμμα και δεν είναι κώδικας. Για παράδειγμα μπορεί να είναι κωδικοί χρημάτων, αλφαριθμητικοί χαρακτήρες ή ακόμα και έτοιμα σχεδιαγράμματα οθονών φτιαγμένα σε XML, τα οποία μπορεί το πρόγραμμα να καλεί.
- Διαχειριστής Τοποθεσίας - Location Manager: Χρησιμοποιείται για να μπορεί να ξέρει η συσκευή που βρίσκεται ανά πάσα στιγμή.
- Διαχειριστής Κοινοποιήσεων - Notification Manager: Ιδανικός τρόπος για να ενημερώνεται ο χρήστης για γεγονότα που συμβαίνουν, διακριτικά χωρίς να διακόπτεται η εργασία του.

### **3.6.5. Εφαρμογές**

Στο πιο υψηλό επίπεδο βρίσκονται οι εφαρμογές δηλαδή τα προγράμματα που καταλαμβάνουν ολόκληρη την οθόνη και αλληλεπιδρούν με το χρήστη. Οι εφαρμογές είναι που βλέπουν οι χρήστες χωρίς να γνωρίζουν την υπόλοιπη στοίβα του λειτουργικού. Μερικές εφαρμογές έρχονται προεγκατεστημένες με την Android συσκευή (όπως επαφές, μουσική κ.α.), ενώ τις περισσότερες τις εγκαθιστά ο χρήστης ανάλογα με τις ανάγκες του. Για τη διευκόλυνση της διαδικασίας εύρεσης και εγκατάστασης εφαρμογών, σε κάθε Android συσκευή υπάρχει διαθέσιμη και προεγκαταστημένη η εφαρμογή Google Play. Αυτή η εφαρμογή επιτρέπει στους χρήστες να κατεβάζουν εφαρμογές που έχουν δημιουργηθεί από τρίτους προγραμματιστές. Το Σεπτέμβριο του 201 υπήρχαν περισσότερες από 675.000 εφαρμογές διαθέσιμες και ο αριθμός των εγκατεστημένων ξεπέρασε τα 25 δισεκατομμύρια.

## **3.7. Εργαλεία ανάπτυξης λογισμικού**

Για την ευκολότερη ανάπτυξη εφαρμογών για συσκευές με Android, η Google διαθέτει δωρεάν μια ομάδα εργαλείων. Τα εργαλεία αυτά βοηθάνε τον προγραμματιστή σε όλη τη διαδικασία ανάπτυξης της εφαρμογής, από την μεταγλώττιση του πηγαίου κώδικα μέχρι την ψηφιακή υπογραφή της εφαρμογής για τη διανομή της.

### **3.7.1. Android SDK**

Το πρώτο βήμα στην πορεία ανάπτυξης της εφαρμογής είναι η εγκατάσταση και ρύθμιση του Android SDK. Το Android SDK παρέχει τα εργαλεία που χρειάζονται για την ανάπτυξη εφαρμογών στην πλατφόρμα Android. Περιλαμβάνει παραδείγματα εφαρμογών με τον πηγαίο τους κώδικα, εργαλεία ανάπτυξης, ένα εξομοιωτή και τις απαιτούμενες βιβλιοθήκες για την ανάπτυξη των εφαρμογών στη γλώσσα προγραμματισμού Java. Ακόμη αναλαμβάνει τη μεταγλώττιση του πηγαίου κώδικα ώστε να τρέχει στην εικονική μηχανή Dalvik.

Ο εξομοιωτής του Android χρησιμοποιεί τα Android Virtual Device (AVD). Τα AVD επιτρέπουν τον ορισμό διαφόρων πτυχών του υλικού του μιμούμενου τηλεφώνου και επιτρέπουν τη δημιουργία πολλών συνθέσεων ώστε να υπάρχει ποικιλία παραλλαγών υλικού διασφαλίζοντας την ομαλή λειτουργία της εφαρμογής σε περισσότερες συσκευές Android. Όταν η εφαρμογή λειτουργεί με τον εξομοιωτή, μπορεί να χρησιμοποιήσει τις υπηρεσίες της πλατφόρμας Android για να επικαλεστεί άλλες εφαρμογές, να έχει πρόσβαση στο δίκτυο, να αναπαράγει ήχο και βίντεο, να αποθηκεύει και να ανακτά δεδομένα, να ενημερώνει το χρήστη και άλλα. Ο εξομοιωτής περιλαμβάνει επίσης μια ποικιλία από δυνατότητες εντοπισμού σφαλμάτων, όπως μια κονσόλα στην οποία καταγράφονται τα μηνύματα εξόδου του πυρήνα, προσομοίωση τυχόν διακοπών της εφαρμογής (όπως όταν φθάνουν μηνύματα SMS ή τηλεφωνικές κλήσεις), καθώς και απορρίψεις στο δίκτυο δεδομένων.

### **3.7.2. Android NDK**

Το NDK είναι ένα εργαλείο που επιτρέπει τη μεταγλώττιση κάποιων μερών μίας android εφαρμογής χρησιμοποιώντας native γλώσσες προγραμματισμού όπως η C και η C++. Για ορισμένους τύπους εφαρμογών, αυτό μπορεί να είναι χρήσιμο, καθώς με αυτόν τον τρόπο μπορούν να επαναχρησιμοποιηθούν ορισμένες υπάρχουσες βιβλιοθήκες που ενδέχεται να έχουν αυξημένη απόδοση. Η χρήση του NDK, ως επί των πλείστον, δεν θα ωφελήσει τις περισσότερες εφαρμογές. Για σωστό αποτέλεσμα, ένας προγραμματιστής θα πρέπει να εξισορροπήσει τα οφέλη και τα μειονεκτήματά αυτής της χρήσης. Αξίζει να σημειωθεί ότι, η χρήση native κώδικα στις εφαρμογές Android, τις περισσότερες φορές δεν οδηγεί σε αισθητή βελτίωση των επιδόσεων, αλλά αυξάνει την πολυπλοκότητα της εφαρμογής. Σε γενικές γραμμές, θα πρέπει η χρήση του NDK να γίνεται μόνο επειδή ταιριάζει στη φύση της εφαρμογής που αναπτύσσεται και όχι απλά επειδή ο προγραμματιστής, της εν λόγω εφαρμογής, προτιμά γλώσσες σαν τη C ή τη C++.

Τυπικά, εφαρμογές που αναπτύσσονται καλύτερα με χρήση NDK, είναι εφαρμογές που είναι αυτόνομες και που δε χρησιμοποιούν πολλή μνήμη. Κάποια τέτοια παραδείγματα είναι η επεξεργασία σήματος και η προσομοίωση της φυσικής. Κατά την εξέταση του κατά πόσο θα πρέπει ή όχι να αναπτυχθεί η εφαρμογή σε native κώδικα, πρέπει να ληφθούν σοβαρά υπόψη οι απαιτήσεις της καθώς, επίσης, και το αν τα Android APIs, που θα χρησιμοποιηθούν, παρέχουν τη λειτουργικότητα που χρειάζεται.

### 3.7.3. Eclipse IDE

Οι περισσότερες εφαρμογές στο Android βασίζονται στην γλώσσα προγραμματισμού Java. Έτσι, ο κάθε προγραμματιστής μπορεί να χρησιμοποιήσει έναν οποιονδήποτε text editor για να γράψει τον κώδικα και μετέπειτα να μεταγλωττίσει τα αρχεία μέσω γραμμής εντολών. Ο συγκεκριμένος τρόπος ανάπτυξης δεν είναι ιδιαίτερα φιλικός προς το χρήστη γι' αυτό συνίσταται η χρήση ενός IDE (Integrated Development Environment) που να υποστηρίζει Java, όπως το Eclipse ή το Netbeans.

Το Eclipse είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού με δυνατότητα εύκολης επεκτασιμότητας χάρη στο σύστημα αρθρωμάτων που διαθέτει. Είναι γραμμένο κυρίως στη γλώσσα προγραμματισμού Java και μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε Java και μέσω των διαφόρων αρθρωμάτων μπορεί να υποστηρίξει και επιπλέον γλώσσες προγραμματισμού όπως C, C++, Perl, PHP, Python, Ruby και άλλες. Το Eclipse είναι

μια εφαρμογή Ελεύθερου Λογισμικού καθώς διατίθεται κάτω από την άδεια Eclipse Public License. Αναπτύσσεται ραγδαία από την παγκόσμια κοινότητα του Ελεύθερου Λογισμικού ενώ χρηματοδοτείται από το ίδρυμα Eclipse και από διάφορες δωρεές.

#### **3.7.4. Android Development Tools**

Η Google υποστηρίζει επίσημα το Eclipse και έχει αναπτύξει ειδικά για αυτό ένα επιπρόσθετο άρθρωμα, το Android Development Tools, το οποίο συνδέει το Eclipse με το Android SDK και όλες τις δυνατότητες του. Επίσης το άρθρωμα παρέχει σύνδεση με τον διαχειριστή εικονικών συσκευών του Android SDK, για τη διαχείριση και εκκίνηση των εικονικών συσκευών μέσω γραφικής διεπαφής. Έτσι διευκολύνει τις δοκιμές σε διάφορους τύπους συσκευών καθώς και στην αποσφαλμάτωση τους.

## ΚΕΦΑΛΑΙΟ 4

### SunSPOT

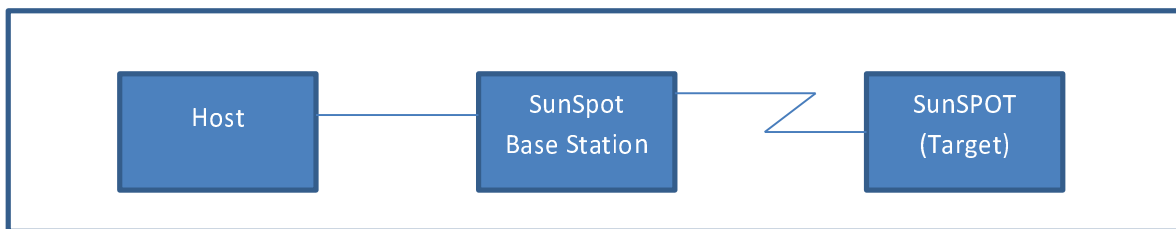
#### 4.1. Sunspots

Τα SunSPOT (Small Programable Object Technology) είναι μικρές ασύρματες συσκευές με ενσωματωμένους αισθητήρες που λειτουργούν με μπαταρίες και είναι προγραμματιζόμενες σχεδόν εξολοκλήρου σε Java δίνοντας την δυνατότητα σε απλούς προγραμματιστές να αναπτύξουν εύκολα λογισμικό χωρίς να έχουν εξειδικευμένες γνώσεις σε embedded συστήματα. Επίσης περιλαμβάνει αρκετούς ενσωματωμένους αισθητήρες και την δυνατότητα να διασυνδεθεί με εξωτερικές συσκευές. Τα SunSPOTs χρησιμοποιούν μια υλοποίηση της Java ME που λέγεται Squawk και υποστηρίζει CLDC 1.1 και MIDP 1.0. Τα SunSPOTs δεν έχουν κάποιο λειτουργικό σύστημα, αλλά τρέχει την Squawk Virtual Machine απευθείας πάνω στον επεξεργαστή, και η VM παρέχει τις βασικότερες λειτουργίες ενός OS. Επίσης όλοι οι drivers των συσκευών είναι γραμμένοι σε Java. Το αναπτυξιακό σύστημα που δίνεται από την SUN είναι η έκδοση v3(Purple) και περιλαμβάνει δύο eSPOTS ένα basestation, ένα USB καλώδιο για την σύνδεση των SPOTs/basestation στο υπολογιστή και ένα cd με το sdk για την ανάπτυξη και εγκατάσταση εφαρμογών στα SPOTS.

#### 4.2. Επεξεργαστής

Πρόκειται για τον ARM920T ARM Thumb processor της ATMEL που περιέχεται σε ένα SOC (System On Chip) κύκλωμα το AT91RM9200. Σε κανονική λειτουργία καταναλώνει 44mW και η μέγιστη ταχύτητα του ρολογιού φτάνει τα 180MHz. Το SOC ενσωματώνει 16Kbyte cache εντολών, και 64-way associative 16Kbyte cache δεδομένων. Η MMU (ARMv4) έχει ένα TLB buffer 64 στοιχείων για δεδομένα και άλλον ένα TLB 64 στοιχείων για μετάφραση εντολών. Η πρόσβαση στην εξωτερική μνήμη(flash, pSRAM) γίνεται από το EBI διάυλο, ο ελεγκτής του διαύλου είναι ρυθμισμένος ώστε να εκκινεί το σύστημα(διαδικασία boot) από την flash όπου βρίσκεται η Squawk VM. Επίσης το SOC περιλαμβάνει μια μεγάλη συλλογή από interfaces για περιφερειακές συσκευές όπως θύρες USB host/device , ethernet MAC,

προγραμματιζόμενος ελεγκτής I/O (PIO), ελεγκτές SPI/USART/I2C/I2S, και 3 16-bit χρονιστές/μετρητές. Επιπλέον ενσωματώνεται και ένας DMA controller (PDC) για άμεσες και γρήγορες εγγραφές στην μνήμη και στους διαύλους USART/I2S/SPI. Λόγο του μικρού μεγέθους της συσκευής οι USB host και η μια USART θύρες δεν χρησιμοποιούνται όπως και τα TWI/I2S/Ethernet MAC interfaces. Επειδή όμως όλα τα σήματα υπάρχουν στο βύσμα του main board που το διασυνδέει με την κάρτα επέκτασης(eDEMO board), μπορούμε να χρησιμοποιήσουμε τα παραπάνω interfaces αν προσθέσουμε τις κατάλληλες φυσικές διασυνδέσεις και γράψουμε τους αντίστοιχους drivers.



Σχήμα 4.1 Συνδεσμολογία SunSPOT

### 4.3. Δίαυλοι επικοινωνίας του Main Board

Η επικοινωνία μεταξύ των SPOTs και workstation γίνεται κυρίως μέσω του διαύλου USB και για την διασύνδεση υπάρχει μια υποδοχή mini USB τύπου B. Η USB client συσκευή στα SPOTs είναι συμβατή με τα πρότυπα USB 1.1 και USB 2.0 και υποστηρίζει ACM modem για την σειριακή μετάδοση. Για την επικοινωνία ανάμεσα σε εσωτερικές συσκευές του main board και μεταξύ του main board και του eDEMO board(κάρτα επέκτασης/αισθητήρων) χρησιμοποιείται το SPI και το PIO. Το SPI είναι ένας σειριακός διάυλος για την επικοινωνία με τον ασύρματο πομποδέκτη IC CC2420, τον power controller και τον έλεγχο των LEDs του eDEMO board. Το PIO interface ελέγχει το activity LED που βρίσκεται αριστερά της mini USB υποδοχής καθώς και τα σήματα ελέγχου και κατάστασης του ασύρματου πομποδέκτη, όπως για παράδειγμα ότι το κανάλι είναι ελεύθερο για μετάδοση ή ότι η RX ουρά είναι πλήρης. Τέλος μέσω του PIO μεταφέρονται τα σήματα ελέγχου του κυκλώματος που ρυθμίζει την τροφοδοσία ρεύματος στην USB θύρα.

## 4.4. Μνήμη

Η μνήμη στο Main Board είναι η Spansion S71PL032J40, και αποτελείται από 4Mbyte NOR flash και 512Kbyte pSRAM(pseudo-SRAM) που βρίσκονται στο ίδιο chip. Ο χρόνος πρόσβασης(access time) για την pSRAM είναι 70nsec και για την Flash 65nsec και έχουν 16-bit δίαυλο δεδομένων. Και οι δυο χρησιμοποιούν τροφοδοσία 3Volt και σε κανονικές συνθήκες λειτουργίας η κατανάλωση είναι 25ma για την pSRAM και 22ma για την Flash. Τα δεδομένα στην pSRAM διατηρούνται όσο το SPOT είναι συνδεδεμένο σε κάποια τροφοδοσία ή μπαταρία. Όταν στο SPOT είναι σε κατάσταση deep-sleep, που τα περισσότερα υποσυστήματα δεν τροφοδοτούνται για εξοικονόμηση ενέργειας η pSRAM καταναλώνει περίπου 8μΑ για την διατήρηση των δεδομένων της ενώ η flash απενεργοποιείται. Η flash είναι προγραμματισμένη ήδη από το εργοστάσιο και περιέχει τον bootloader, την Squawk VM, τις βασικές βιβλιοθήκες και μια προ εγκατεστημένη εφαρμογή (bounce demo).

Ένας κόμβος αισθητήρα (sensor node) αποτελείται από έναν πομποδέκτη RF ο οποίος είναι υπεύθυνος για την επικοινωνία, έναν μικροεπεξεργαστή (CPU), μνήμη τυχαίας προσπέλασης (RAM) για επεξεργασία δεδομένων (προαιρετικά μπορεί να φέρει και μνήμη μόνιμης αποθήκευσης δεδομένων) και μια μικρή μπαταρία που του παρέχει μια περιορισμένη ενεργειακή αυτονομία. Επίσης περιέχει τους κατάλληλους αισθητήρες (sensing unit ή sensors) μέσω των οποίων συλλέγει τα ανάλογα δεδομένα (π.χ. θερμοκρασία, υγρασία, ανίχνευση αντικειμένων). Ενσωματώνει μονάδα μετατροπής ADC (Analog to Digital) η οποία μετατρέπει τα δεδομένα που συλλέγονται από τους αισθητήρες από την αναλογική μορφή σε ψηφιακή, για την περαιτέρω επεξεργασία από τον επεξεργαστή του κόμβου πριν την αποστολή τους προς μία κεντρική βάση δεδομένων (Sink) μέσω του ασύρματου δικτύου. Επιπρόσθετα μπορεί ένας κόμβος να φέρει μία μονάδα εύρεσης γεωγραφικού στίγματος (GPS) η οποία βοηθάει στην εύρεση του κόμβου μέσα στην περιοχή ανάπτυξης του δικτύου και στην χαρτογράφηση της ακριβής θέσης του, μονάδα κίνησης (mobile unit) ώστε να μπορεί να μετακινείται και μια εναλλακτική μορφή ενέργειας (π.χ. φωτο-βολταϊκά στοιχεία για συλλογή ηλιακής ενέργειας) ώστε να έχει την δυνατότητα επαναφόρτισης της μπαταρίας του. Τα WSNs είναι αυτο-οργανούμενα, κατά περίπτωση, (ad-hoc) δίκτυα στα οποία γίνονται ασύρματες ζεύξεις μεταξύ γειτονικών κόμβων. Χαρακτηριστικό τέτοιων δικτύων είναι ότι κόμβοι που βρίσκονται εκτός

εμβέλειας μεταξύ τους μπορούν να επικοινωνήσουν ασύρματα χρησιμοποιώντας ενδιάμεσους κόμβους για την προώθηση των μηνυμάτων . Βασικός στόχος στα Wireless Sensor Networks (WSN) είναι η μεταφορά το συλλεγμένων δεδομένων σε ένα κεντρικό σημείο ελέγχου (sink). Το sink ή base station μπορεί να ελέγχεται απ' ευθείας από το χρήστη μέσω ενός τερματικού ή από ένα απομακρυσμένο τερματικό μέσω του διαδικτύου, δορυφορικής σύνδεσης ή την χρήση κάποιας άλλης τεχνολογίας.



## ΚΕΦΑΛΑΙΟ 5

### Zigbee

#### 5.1. Zigbee

Το ZigBee είναι μία τεχνολογία στο χώρο των ασύρματων δικτύων προσωπικού χώρου (WPANs). Προήλθε από τη συνεργασία της εταιρίας ZigBee Alliance με την επιτροπή IEEE 802.15.4 και παρέχει τη δυνατότητα για συνδέσεις συσκευών με χαμηλό ρυθμό μετάδοσης, χαμηλό κόστος και χαμηλή κατανάλωσης ισχύος. Η επιτροπή IEEE 802.15.4 ασχολήθηκε με τις προδιαγραφές των δύο χαμηλότερων επιπέδων (φυσικό και ελέγχου πρόσβασης στο μέσο), ενώ η ZigBee Alliance με τα ανώτερα επίπεδα (δικτύου και εφαρμογών). Το Μάιο του 2003 εγκρίθηκαν οι προδιαγραφές της επιτροπής IEEE 802.15.4 και το Δεκέμβριο του 2004 οι προδιαγραφές της ZigBee Alliance για την έκδοση 1.0 του ZigBee.

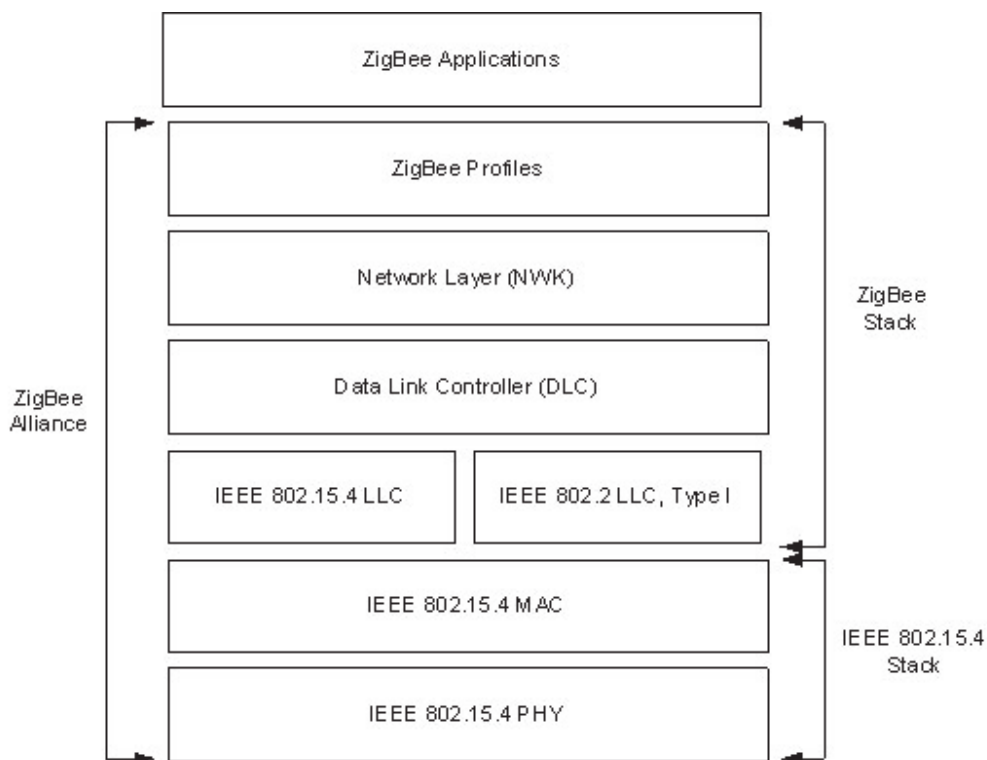
#### 5.2. Η στοίβα πρωτοκόλλων του ZigBee

Η στοίβα πρωτοκόλλων του ZigBee αποτελείται από 4 επίπεδα. Κάθε επίπεδο εκτελεί ένα συγκεκριμένο σύνολο λειτουργιών και παρέχει τις υπηρεσίες του στο ανώτερο επίπεδο μέσω μιας διεπαφής που ονομάζεται σημείο πρόσβασης υπηρεσιών (service access point, SAP). Τα 4 επίπεδα της στοίβας πρωτοκόλλων του ZigBee είναι τα παρακάτω:

- Το φυσικό επίπεδο (Physical layer, PHY) που είναι υπεύθυνο για την ενεργοποίηση και απενεργοποίηση του πομποδέκτη, τη μετάδοση και λήψη δεδομένων, την ανίχνευση ενέργειας στο κανάλι, την εκτίμηση της κατάστασης των καναλιών για την πολλαπλή πρόσβαση με ανίχνευση φέροντος και με αποφυγή συγκρούσεων (CSMA-CA) και τη μέτρηση της ποιότητας των λαμβανομένων πακέτων.
- Το επίπεδο ελέγχου πρόσβασης στο μέσο (Medium access control layer, MAC) που παρέχει υπηρεσίες μεταφοράς δεδομένων και διαχείρισης. Είναι υπεύθυνο για την πρόσβαση στο κανάλι, για τη διαχείριση των χρονοσχισμών και για την παροχή μιας αξιόπιστης σύνδεσης μεταξύ δύο επιπέδων MAC. Επιπρόσθετα παρέχει τα μέσα για την εφαρμογή διαφόρων μηχανισμών ασφάλειας. Το επίπεδο δικτύου (Network layer, NWK) που είναι υπεύθυνο για τη

δημιουργία του δικτύου, για την είσοδο και την έξοδο μία συσκευής από ένα δίκτυο, για την ασφάλεια και για τη δρομολόγηση των μεταδιδόμενων πακέτων.

- Το επίπεδο εφαρμογών (Application layer, APL) που περιλαμβάνει το υποεπίπεδο υποστήριξης εφαρμογών (Application support sublayer, APS), το πλαίσιο εφαρμογών (Application framework, AF), τα αντικείμενα συσκευής ZigBee (ZigBee Device Objects, ZDO) και τις καθορισμένες από τον κατασκευαστή εφαρμογές. Το υποεπίπεδο APS είναι υπεύθυνο για τη σύνδεση δύο συσκευών βάσει των αναγκών και των υπηρεσιών τους και για την αποστολή δεδομένων μεταξύ τους. Τα ZDO είναι αυτά που καθορίζουν το ρόλο της κάθε συσκευής στο δίκτυο και το επίπεδο ασφάλειας. Επίσης συμβάλλουν στην ανίχνευση των συσκευών σε ένα δίκτυο και στον προσδιορισμό των υπηρεσιών που αυτές παρέχουν. Το πλαίσιο εφαρμογών είναι το περιβάλλον στο οποίο φιλοξενούνται οι εφαρμογές μέσα σε μία συσκευή ZigBee.



Σχήμα 5.1 Η στοίβα πρωτοκόλλων του ZigBee

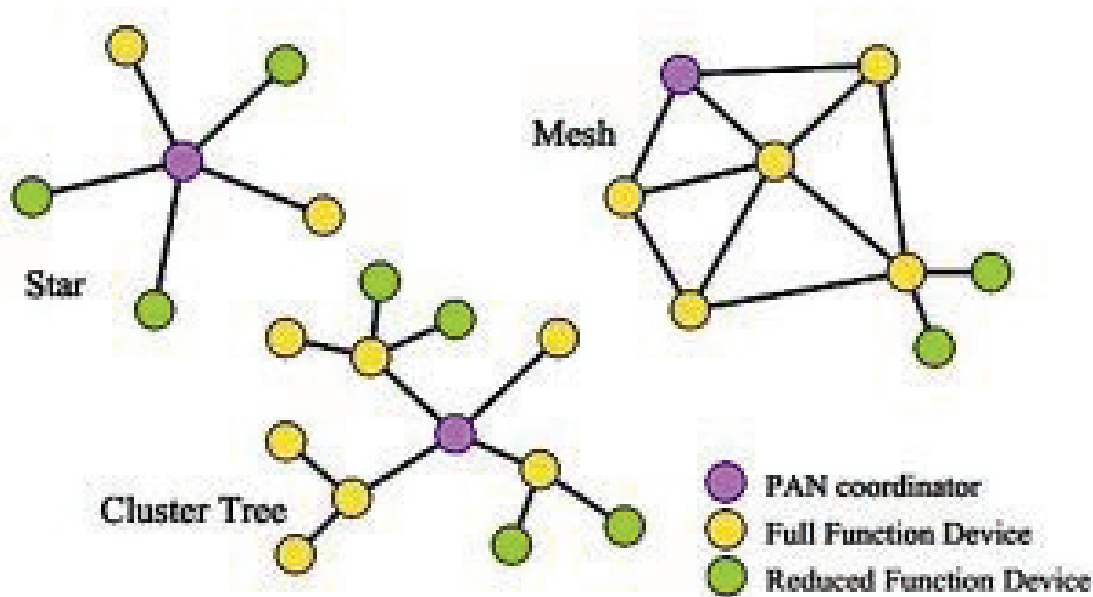
### 5.3. Τύποι συσκευών

Σε ένα δίκτυο που λειτουργεί με βάση τη στοίβα πρωτοκόλλων του ZigBee, οι συσκευές μπορούν να διακριθούν σε πλήρους λειτουργίας (full-function device , FFD) και μειωμένης λειτουργίας (reduced-function device, RFD). Σε κάθε δίκτυο υπάρχει πάντοτε μία FFD που έχει το ρόλο του συντονιστή του δικτύου (PAN coordinator). Οι FFDs μπορούν να επικοινωνήσουν με όλες τις άλλες συσκευές, ενώ οι RFDs μόνο με μία FFD. Οι RFDs χρησιμοποιούνται σε πολύ απλές εφαρμογές.

### 5.4. Τοπολογίες δικτύων

Ανάλογα με τις απαιτήσεις των εφαρμογών, το ZigBee μπορεί να υποστηρίξει δύο τοπολογίες δικτύων (σχήμα 5.2). Ανεξαρτήτως τοπολογίας, κάθε συσκευή έχει μία μοναδική διεύθυνση με μήκος 64 bits. Αυτή μπορεί να χρησιμοποιηθεί για την επικοινωνία μέσα σε ένα δίκτυο ή να χρησιμοποιηθεί από το συντονιστή για να χορηγήσει μία συντομευμένη διεύθυνση (16 bits) στη συσκευή. Για κάθε δίκτυο που δημιουργείται, ο συντονιστής επιλέγει μία ταυτότητα (16 bits) που προσδιορίζει μοναδικά το συγκεκριμένο δίκτυο. Ο συνδυασμός ταυτότητας δικτύου και διεύθυνσης συσκευής επιτρέπει την επικοινωνία μεταξύ συσκευών. Κάθε δίκτυο μπορεί να έχει μέχρι και 255 συσκευές. Οι τοπολογίες αυτές είναι:

- Τοπολογία σε σχήμα αστεριού(star): Σε αυτή υπάρχει ο συντονιστής(pan coordinator) του δικτύου, ο οποίος εγκαθιστά συνδέσεις σημείου προς σημείο με άλλες συσκευές. Επίσης ο συντονιστής λειτουργεί και ως δρομολογητής για τη μεταφορά των δεδομένων μεταξύ των άλλων συσκευών, αφού αυτές δεν μπορούν να επικοινωνήσουν απευθείας.
- Τοπολογία σημείου προς σημείο: Κάθε συσκευή εγκαθιστά συνδέσεις σημείου προς σημείο με άλλες συσκευές που βρίσκονται μέσα στην εμβέλεια της. Με αυτό τον τρόπο δημιουργούνται δίκτυα που έχουν τη μορφή δένδρου(cluster tree) ή πλέγματος(mesh). Με τη βοήθεια αλγορίθμων δρομολόγησης, όλες οι συσκευές μπορούν να επικοινωνήσουν μεταξύ τους. Πολλά τέτοια δίκτυα μπορούν να ενωθούν μεταξύ τους και να σχηματίσουν ένα μεγαλύτερο. Στο μεγαλύτερο δίκτυο υπάρχει μόνο ένας συντονιστής δικτύου(pan coordinator), ενώ κάθε μικρότερο δίκτυο έχει από ένα δρομολογητή.



Σχήμα 5.2 Τοπολογίες Δικτύων

## 5.5. Φυσικό επίπεδο (Physical layer, PHY)

Το φυσικό επίπεδο παρέχει δύο ειδών υπηρεσίες, δεδομένων και διαχείρισης, στο επίπεδο MAC. Η πρόσβαση σε αυτές επιτυγχάνεται μέσω δύο διαφορετικών σημείων πρόσβασης υπηρεσιών (SAPs), των PHY dataSAP (PD-SAP) και PLME-SAP. Επίσης στο επίπεδο αυτό υπάρχει και μία βάση δεδομένων (PHY PIB) που περιέχει πληροφορίες σχετικές με τη λειτουργία του (κανάλι που χρησιμοποιείται, κανάλια που υποστηρίζονται, εκπεμπόμενη ισχύς, τρόπος ανίχνευσης ελεύθερου καναλιού).

### 5.5.1. Χαρακτηριστικά καναλιών και διαμόρφωσης

Προσφέρονται συνολικά 27 κανάλια, αριθμημένα από 0 ως 26 και είναι διαθέσιμα σε 3 ζώνες συχνοτήτων. Ένα κανάλι υπάρχει μεταξύ 868 και 868,6 MHz, 10 κανάλια μεταξύ 902 και 928 MHz και 16 κανάλια μεταξύ 2,4 και 2,4835 GHz. Κάθε συσκευή θα πρέπει να υποστηρίζει όλα τα κανάλια, εκτός και αν κάποια από αυτά δεν είναι ελεύθερα στην περιοχή που λειτουργεί. Ο πομπός, και στις 3 ζώνες συχνοτήτων, έχει ελάχιστη ισχύ εκπομπής ίση με -3 dBm, ενώ η μέγιστη περιορίζεται από τη νομοθεσία κάθε περιοχής. Ο κάθε πομπός έχει τη δυνατότητα να ρυθμίζει την ισχύ εκπομπής του, έτσι ώστε να εκπέμπει πάντοτε με την ελάχιστη απαιτούμενη. Η ευαισθησία του δέκτη είναι -85 dBm ή καλύτερη στη ζώνη μεταξύ 2,4-2,4835 GHz και -92

dBm ή καλύτερη στις άλλες δύο. Η ευαισθησία ορίζεται ως η ελάχιστη απαιτούμενη ισχύς για να έχουμε ποσοστό λανθασμένων πλαισίων μικρότερο του 1% για πλαίσια με 20 bytes ωφέλιμο φορτίο.

### **5.5.2. Πρόσβαση στο κανάλι**

Οι μεταδόσεις σε κάθε κανάλι μπορούν να γίνουν με δύο τρόπους. Στον πρώτο τρόπο χρησιμοποιείται ο μηχανισμός πολλαπλής πρόσβασης με ανίχνευση φέροντος, αποφυγή συγκρούσεων και χωρίς χρονοσχισμές (unslotted CSMA-CA). Κάθε συσκευή πριν να μεταδώσει, ανιχνεύει το κανάλι. Αν είναι αδρανές, αρχίζει να μεταδίδει. Αν είναι κατειλημμένο, ο αποστολέας αναβάλλει τη μετάδοση μέχρι το κανάλι να γίνει αδρανές.

Στο δεύτερο τρόπο χρησιμοποιείται το υπερπλαίσιο, το οποίο οριοθετείται από τα αναγνωριστικά σήματα και χωρίζεται σε 16 χρονοσχισμές ίσης διάρκειας. Το αναγνωριστικό σήμα στέλνεται πάντοτε στην πρώτη χρονοσχισμή. Τα σήματα αυτά χρησιμοποιούνται για να περιγράψουν τη δομή του υπερπλαισίου, για το συγχρονισμό των συσκευών του δικτύου και για τον προσδιορισμό του δικτύου. Το υπερπλαίσιο μπορεί να έχει ένα ενεργό και ένα μη ενεργό μέρος. Κατά τη διάρκεια του μη ενεργού μέρους, ο συντονιστής δεν αλληλεπιδρά με το δίκτυο του και λειτουργεί με χαμηλή κατανάλωση ισχύος. Το ενεργό μέρος αποτελείται από την περίοδο πρόσβασης με ανταγωνισμό (CAP) και από την περίοδο χωρίς ανταγωνισμό (CFP). Στην CAP κάθε συσκευή που μεταδίδει χρησιμοποιεί το μηχανισμό πολλαπλής πρόσβασης με ανίχνευση φέροντος, αποφυγή συγκρούσεων και χρονοσχισμές (slotted CSMA-CA). Αυτό είναι το ίδιο με το unslotted CSMA-CA, με τη διαφορά ότι οι μεταδόσεις ξεκινούν πάντοτε στην αρχή κάποιας χρονοσχισμής και σε περίπτωση κατειλημμένου καναλιού, ο αποστολέας αναβάλλει τη μετάδοση για τυχαίο αριθμό χρονοσχισμών. Όταν ένα κανάλι είναι αδρανές, η μετάδοση πραγματοποιείται μόνο αν υπάρχει αρκετός χρόνος για να ολοκληρωθεί πριν τη λήξη της CAP. Η CFP περιέχει κρατημένες χρονοσχισμές και βρίσκεται ακριβώς πριν το μη ενεργό μέρος. Οι χρονοσχισμές αυτές μπορούν να χρησιμοποιηθούν για την εξυπηρέτηση εφαρμογών που απαιτούν συγκεκριμένο εύρος ζώνης.

### **5.5.3. Υπηρεσίες δεδομένων**

Το σημείο πρόσβασης υπηρεσιών για δεδομένα (PD-SAP) υποστηρίζει τη μεταφορά των μονάδων δεδομένων πρωτοκόλλου του επιπέδου MAC (MPDUs). Για την πραγματοποίηση της μεταφοράς αυτής ορίζονται τρεις στοιχειώδεις υπηρεσίες:

- PD-DATA.request: Το επίπεδο MAC ζητά από το φυσικό επίπεδο την αποστολή μίας MPDU.
- PD-DATA.confirm: Το φυσικό επίπεδο επιβεβαιώνει, θετικά ή αρνητικά, στο επίπεδο MAC την αποστολή μίας MPDU.
- PD-DATA.indication: Το επίπεδο MAC ενημερώνεται από το φυσικό επίπεδο για την άφιξη δεδομένων.

Η μεταφορά των MPDUs γίνεται με τα πακέτα φυσικού επιπέδου. Το κάθε πακέτο αποτελείται από τα πεδία επικεφαλίδα συγχρονισμού (SHR, synchronization header), επικεφαλίδα φυσικού επιπέδου (PHR, PHY header) και ωφέλιμο φορτίο φυσικού επιπέδου (PHY payload). Η επικεφαλίδα συγχρονισμού χρησιμοποιείται για το συγχρονισμό του δέκτη και για να προσδιορίζει την έναρξη των δεδομένων. Η επικεφαλίδα φυσικού επιπέδου προσδιορίζει το μήκος του ωφέλιμου φορτίου, ενώ το πεδίο ωφέλιμο φορτίο φυσικού επιπέδου περιέχει την MPDU.

## **5.6. Επίπεδο ελέγχου πρόσβασης στο μέσο (Medium access control layer, MAC)**

Το επίπεδο MAC παρέχει τις υπηρεσίες του στο επίπεδο δικτύου. Η πρόσβαση στις υπηρεσίες δεδομένων γίνεται μέσω του MLDE-SAP, ενώ για τις υπηρεσίες διαχείρισης υπάρχει το MLME-SAP. Επίσης στο επίπεδο αυτό υπάρχει και μία βάση δεδομένων (MAC PIB) που περιέχει πληροφορίες σχετικές με τη λειτουργία του.

### **5.6.1. Υπηρεσίες δεδομένων**

Το σημείο πρόσβασης υπηρεσιών για τα δεδομένα (MLDE-SAP) υποστηρίζει τη μεταφορά των μονάδων δεδομένων πρωτοκόλλου του επιπέδου δικτύου (NPDUs). Για την πραγματοποίηση της μεταφοράς αυτής ορίζονται οι παρακάτω στοιχειώδεις υπηρεσίες:

- MLDE-DATA.request: Το επίπεδο δικτύου ζητά από το επίπεδο MAC την αποστολή μίας NPDU.
- MLDE-DATA.confirm: Το επίπεδο MAC επιβεβαιώνει, θετικά ή αρνητικά, στο επίπεδο δικτύου την αποστολή μίας NPDU.
- MLDE-DATA.indication: Το επίπεδο δικτύου ενημερώνεται από το επίπεδο MAC για την άφιξη δεδομένων.
- MLDE-PURGE.request: Χρησιμοποιείται όταν το επίπεδο δικτύου θέλει να διαγράψει ένα μήνυμά του από την ουρά αναμονής του MAC.
- MLDE-PURGE.confirm: Απαντάει στην προηγούμενη αίτηση.

### 5.6.2. Πλαίσια MAC

Ορίζονται τέσσερις διαφορετικοί τύποι πλαισίων, τα δεδομένα, οι επιβεβαιώσεις, οι εντολές και τα αναγνωριστικά σήματα. Η γενική δομή ενός πλαισίου MAC φαίνεται στο σχήμα 5.3. Αποτελείται από τα πεδία επικεφαλίδας MAC (MHR), ωφέλιμου φορτίου MAC και υποσημείωσης MAC (MFR). Το MHR περιέχει τα πεδία έλεγχου πλαισίου, αριθμού ακολουθίας και διευθύνσεων. Το πεδίο ωφέλιμου φορτίου MAC περιέχει διάφορες πληροφορίες ανάλογα με τον τύπο του πλαισίου. Το MFR περιέχει την ακολουθία ελέγχου του πλαισίου (FCS). Το συνολικό μήκος του πλαισίου δεν πρέπει να ξεπερνά τα 127 bytes (μέγιστο μήκος ωφέλιμου φορτίου του πακέτου φυσικού επιπέδου).

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination PAN identifier	Destination address	Source PAN identifier	Source address	Frame payload	FCS
		Addressing fields					
MHR						MAC payload	MFR

## Σχήμα 5.3 Δομή πλαισίου MAC

### 5.6.2.1. Πεδίο έλεγχου πλαισίου

Έχει μήκος 2 bytes και η δομή του αποτελείται από τα παρακάτω υποπεδία:

- Τύπος πλαισίου (frame type) που καθορίζει τον τύπο του πλαισίου (δεδομένα, επιβεβαίωση, εντολή ή αναγνωριστικό σήμα).
- Ασφάλεια (security enabled) που καθορίζει αν το πλαίσιο είναι κρυπτογραφημένο.
- Πλαίσιο που εκκρεμεί (frame pending) που καθορίζει εάν ο αποστολέας έχει και άλλα δεδομένα να στείλει στον παραλήπτη.
- Αίτημα επιβεβαίωσης (acknowledgment request) που καθορίζει εάν ο αποστολέας θέλει να λάβει επιβεβαίωση για το πλαίσιο.
- Intra-PAN που καθορίζει εάν το πλαίσιο θα σταλεί μέσα στο ίδιο PAN ή σε διαφορετικό.
- Μορφή διεύθυνσης προορισμού (destination addressing mode) που καθορίζει τον τύπο, την ύπαρξη και το μέγεθος των υποπεδίων προορισμού στα πεδία διευθύνσεων.
- Μορφή διεύθυνσης πηγής (source addressing mode) που καθορίζει τον τύπο, την ύπαρξη και το μέγεθος των υποπεδίων πηγής στα πεδία διευθύνσεων.

Υπάρχουν συνολικά και 5 bits κρατημένα για μελλοντική χρήση, 3 μετά το Intra-PAN και 2 μετά τη μορφή διεύθυνσης προορισμού.

### 5.6.2.2. Πεδίο αριθμού ακολουθίας

Έχει μήκος 8 bits και παίρνει μία τυχαία τιμή. Χρησιμοποιείται για να αντιστοιχίζεται ένα πλαίσιο επιβεβαίωσης με το πλαίσιο εντολής ή δεδομένων. Έτσι, όταν φθάσει μία επιβεβαίωση, ο παραλήπτης γνωρίζει ότι το πλαίσιο εντολής ή δεδομένων που έχει τον ίδιο αριθμό ακολουθίας με αυτή, έχει φθάσει στον προορισμό του. Στην περίπτωση των αναγνωριστικών σημάτων ο αριθμός ακολουθίας προσδιορίζει ένα συγκεκριμένο σήμα.



### **5.6.2.3. Πεδία διευθύνσεων**

Η ύπαρξη των πεδίων αυτών καθώς και το μήκος τους εξαρτώνται από τον τύπο του πλαισίου. Στα πλαίσια επιβεβαιώσεων δεν υπάρχουν καθόλου. Αποτελούνται από τα παρακάτω υποπεδία:

- Ταυτότητα δικτύου προορισμού (destination PAN identifier) που προσδιορίζει την ταυτότητα του δικτύου στο οποίο ανήκει ο παραλήπτης.
- Διεύθυνση προορισμού (destination address) που προσδιορίζει τη διεύθυνση του παραλήπτη.
- Ταυτότητα δικτύου πηγής (source PAN identifier) που προσδιορίζει την ταυτότητα του δικτύου στο οποίο ανήκει ο δημιουργός του πλαισίου.
- Διεύθυνση πηγής (source address) που προσδιορίζει τη διεύθυνση του δημιουργού του πλαισίου.

### **5.6.2.4. Ακολουθία ελέγχου πλαισίου (FCS)**

Έχει μήκος 2 bytes. Περιέχει κώδικα CRC για ανίχνευση λαθών. Υπολογίζετε για τα πεδία επικεφαλίδα MAC (MHR) και ωφέλιμο φορτίο MAC.

### **5.6.2.5. Ωφέλιμο φορτίο MAC**

Περιέχει διάφορες πληροφορίες ανάλογα με τον τύπο του πλαισίου. Μπορεί να είναι κρυπτογραφημένο. Σε ένα αναγνωριστικό σήμα περιέχει πληροφορίες σχετικά με τη δομή του υπερπλαισίου, σε ένα πλαίσιο δεδομένων περιέχει τα δεδομένα του ανωτέρου επιπέδου, σε μία επιβεβαίωση δεν υπάρχει καθόλου και σε ένα πλαίσιο εντολών περιέχει την εντολή που μεταδίδεται.

## **5.6.3. Λειτουργίες επιπέδου MAC**

Το επίπεδο MAC έχει μία σειρά από λειτουργίες. Δημιουργεί αναγνωριστικά σήματα εάν η συσκευή είναι συντονιστής του δικτύου τα οποία χρησιμοποιούνται για την περιγραφή της

δομή του υπερπλαισίου, για το συγχρονισμό των συσκευών του δικτύου και για τον προσδιορισμό του δικτύου.

Επιπλέον υποστηρίζει τη δημιουργία και τη διατήρηση δικτύων. Ζητά από το φυσικό επίπεδο να ανιχνεύσει σε ορισμένα κανάλια και να παραδώσει τα αποτελέσματα στο επίπεδο δικτύου. Έτσι ανιχνεύονται τα αναγνωριστικά σήματα που εκπέμπουν οι συντονιστές των δικτύων και μπορεί να ζητήσει τη σύνδεση σε κάποιο δίκτυο. Επιπλέον, το επίπεδο MAC μπορεί να ζητήσει από το φυσικό επίπεδο και τη μέτρηση της ενέργειας σε ένα σύνολο καναλιών και έτσι να επιλέξει το κατάλληλο κανάλι για τη δημιουργία ενός δικτύου.

Μία ακόμα λειτουργία του είναι η υποστήριξη υπηρεσιών ασφάλειας. Το επίπεδο MAC είναι σε θέση να παρέχει στα ανώτερα του επίπεδα κάποιες βασικές υπηρεσίες ασφαλείας, όπως είναι ο έλεγχος πρόσβασης, η κρυπτογράφηση των δεδομένων και η ακεραιότητα των πλαισίων όταν αυτό του ζητηθεί. Τα κλειδιά και οι λίστες που απαιτούνται για αυτές, του παρέχονται από τα ανώτερα επίπεδα.

## **5.7. Επίπεδο δικτύου (Network layer, NWK)**

Το επίπεδο δικτύου παρέχει τις υπηρεσίες του, δεδομένων και διαχείρισης, στο επίπεδο εφαρμογών και επιπλέον εξασφαλίζει τη σωστή λειτουργία του επιπέδου MAC. Οι υπηρεσίες δεδομένων παρέχονται στο επίπεδο εφαρμογών μέσω του σημείου πρόσβασης υπηρεσιών NLDE (Network Layer Data Entity), ενώ οι υπηρεσίες διαχείρισης μέσω του σημείου πρόσβασης υπηρεσιών NLME (Network Layer Management Entity). Επίσης στο επίπεδο αυτό υπάρχει και μία βάση δεδομένων που περιέχει πληροφορίες σχετικές με τη λειτουργία του.

Οι υπηρεσίες δεδομένων περιλαμβάνουν την παραλαβή των δεδομένων από το ανώτερο επίπεδο, την τοποθέτησή τους σε κατάλληλα πλαίσια και την αποστολή τους στην κατάλληλη συσκευή, είτε απευθείας είτε μέσω κάποιας άλλης. Για την πραγματοποίηση αυτών ορίζονται οι στοιχειώδεις υπηρεσίες, NLDE-DATA.request, NLDE-DATA.confirm και NLDE-DATA.indication.

Οι υπηρεσίες διαχείρισης περιλαμβάνουν τη διαμόρφωση της λειτουργίας μίας συσκευής ανάλογα με το ρόλο της στο δίκτυο, τη δημιουργία ενός δικτύου, τη σύνδεση σε ένα δίκτυο, την αποχώρηση από αυτό, τη διευθυνσιοδότηση των συσκευών του δικτύου, τη δυνατότητα να

ανακαλύπτουν γειτονικές συσκευές, τη δυνατότητα να ανακαλύπτουν και να καταγράφουν διαδρομές για την αποτελεσματική αποστολή των μηνυμάτων στο δίκτυο και τη δυνατότητα να ελέγχουν τη λειτουργία του δέκτη.

### 5.7.1. Πλαίσια επιπέδου δικτύου

Κάθε πλαίσιο αποτελείται από τα πεδία επικεφαλίδα (NWK Header) και ωφέλιμο φορτίο (NWK Payload). Η επικεφαλίδα του πλαισίου αποτελείται από το πεδίο έλεγχος πλαισίου (Frame Control) και τα πεδία δρομολόγησης (Routing Fields).

Το πεδίο έλεγχου πλαισίου (Frame Control, 2 bytes) έχει τη δομή που φαίνεται στο και αποτελείται από τα παρακάτω υποπεδία:

- Τύπος πλαισίου (Frame type) που προσδιορίζει τον τύπο του πλαισίου (δεδομένα ή εντολή).
- Έκδοση πρωτοκόλλου (Protocol version) που προσδιορίζει την έκδοση του πρωτοκόλλου που χρησιμοποιείται.
- Ανακάλυψη διαδρομής (Discover route) που χρησιμοποιείται για να ελέγχει τις λειτουργίες ανακάλυψης διαδρομής για τη μετάδοση του πλαισίου.
- Ασφάλεια (Security) που δείχνει αν έχουν χρησιμοποιηθεί οι υπηρεσίες ασφάλειας που παρέχει το επίπεδο δικτύου.

Υπάρχουν και 7 bits κρατημένα για μελλοντική χρήση, ένα μετά την ανακάλυψη διαδρομής και έξι μετά την ασφάλεια.

Τα πεδία δρομολόγησης αποτελούνται από τα παρακάτω υποπεδία:

- Διεύθυνση προορισμού (Destination Address) που προσδιορίζει την διεύθυνση προορισμού του πλαισίου.

- Διεύθυνση πηγής (Source Address) που προσδιορίζει την διεύθυνση της συσκευής που δημιούργησε το πλαίσιο.
- Ακτίνα (Radius) που προσδιορίζει τον αριθμό των βημάτων που ακολούθησε ένα πλαίσιο μέχρι να φθάσει στον προορισμό.
- Αριθμός ακολουθίας (Sequence Number) που είναι ένας μετρητής που αυξάνεται κατά 1 κάθε φορά που δημιουργείται ένα καινούριο πλαίσιο.

Το πεδίο ωφέλιμο φορτίο έχει μεταβλητό μήκος και περιέχει συγκεκριμένες πληροφορίες για κάθε τύπο πλαισίου. Στο πλαίσιο δεδομένων περιέχονται τα δεδομένα που θέλει το ανώτερο επίπεδο να στείλει. Στο πλαίσιο εντολών περιέχεται η ταυτότητα της εντολής (1 byte), που προσδιορίζει τον τύπο της, και το ωφέλιμο φορτίο της εντολής. Στις προδιαγραφές του ZigBee ορίζονται οι παρακάτω τύποι εντολών:

- Route request: Με αυτή μία συσκευή μπορεί να ζητήσει από τις γειτονικές της να ψάξουν για ένα συγκεκριμένο παραλήπτη. Με τις πληροφορίες που παίρνει, μπορεί να επιλέξει κατάλληλα δρομολόγια για τη μετάδοση πλαισίων προς όλες τις συσκευές του δικτύου.
- Route reply: Με αυτή η συσκευή προορισμού ενημερώνει τον αρχικό αποστολέα της Route request ότι έλαβε τη συγκεκριμένη αίτηση. Επίσης επιστρέφει πληροφορίες που αφορούν τη διαδρομή που ακολούθησε η αντίστοιχη αίτηση.
- Route error: Χρησιμοποιείται όταν μία συσκευή δεν μπορεί να προωθήσει ένα πλαίσιο δεδομένων. Με αυτή ενημερώνεται ο αποστολέας ότι το πλαίσιο δεν έφθασε στον προορισμό του.
- Leave: Χρησιμοποιείται από μία συσκευή για να ενημερώσει ότι αποχωρεί από το δίκτυο ή για να ζητήσει από κάποια συσκευή να αποχωρήσει.

Octets: 2	2	2	1	1	Variable
Frame Control	Destination Address	Source Address	Radius <sup>a</sup>	Sequence Number <sup>b</sup>	Frame Payload
	Routing Fields				
NWK Header					NWK Payload

Σχήμα 5.4 Δομή πλαισίου επιπέδου Δικτύου

### 5.7.2. Πίνακες γειτόνων

Κάθε συσκευή διατηρεί στη βάση δεδομένων της έναν πίνακα με πληροφορίες για τις συσκευές που βρίσκονται μέσα στην ακτίνα δράσης της. Στον πίνακα αυτό, για κάθε γειτονική συσκευή υπάρχει μία εγγραφή που περιέχει την ταυτότητα του δικτύου της, τον τύπο της, τη διεύθυνση της μέσα στο δίκτυο που ανήκει, τη σχέση που υπάρχει μεταξύ των δύο συσκευών και την εκτεταμένη της διεύθυνση εφόσον οι δύο συσκευές είναι συνδεδεμένες. Στον πίνακα αυτό μπορεί να υπάρχουν και διάφορες άλλες προαιρετικές πληροφορίες, όπως το πόσο συχνά εκπέμπει αναγνωριστικά σήματα, αν αποδέχεται τις αιτήσεις για σύνδεση, μία εκτίμηση της ποιότητας της ζεύξης, το λογικό κανάλι στο οποίο λειτουργεί και το πόσα βήματα απέχει από το συντονιστή του δικτύου της.

### 5.7.3. Σύνδεση σε ένα δίκτυο και αποχώρηση από αυτό

Κάθε συσκευή ZigBee έχει τη δυνατότητα να συνδέεται σε ένα δίκτυο και να αποχωρεί από αυτό. Μία σχέση γονέα-παιδιού αναπτύσσεται κάθε φορά που μία συσκευή ενός δικτύου επιτρέπει σε μία άλλη να γίνει μέλος του δικτύου. Η νέα συσκευή είναι το παιδί, ενώ η πρώτη είναι ο γονέας. Ένα παιδί μπορεί να προστεθεί σε ένα δίκτυο με 2 τρόπους.

Στον πρώτο τρόπο, η διαδικασία ξεκινά από τη νέα συσκευή (παιδί). Μία εφαρμογή της ζητά αρχικά να γίνει ανίχνευση κάποιων καναλιών. Η αίτηση αυτή μεταβιβάζεται από το ανώτερο προς τα κατώτερα επίπεδα. Στη συνέχεια, τα αποτελέσματα της ανίχνευσης, τα διαθέσιμα δίκτυα και τα χαρακτηριστικά τους, ακολουθώντας την αντίθετη πορεία παραδίδονται στο επίπεδο εφαρμογών, το οποίο και αποφασίζει σε ποίο από τα διαθέσιμα δίκτυα θέλει να

συνδεθεί. Στέλνει μία αίτηση στο επίπεδο δικτύου για σύνδεση με το συγκεκριμένο δίκτυο. Το επίπεδο δικτύου ψάχνει στον πίνακα γειτόνων για μία συσκευή που να ανήκει σε αυτό το δίκτυο και αν αυτή αποδέχεται τις αιτήσεις για σύνδεση τότε χρησιμοποιεί τη διεύθυνση της συσκευής που βρίσκεται στον πίνακα, στέλνει μία αίτηση για σύνδεση και περιμένει για την απάντηση. Αν δεν βρει κάποια κατάλληλη, απαντά ότι η σύνδεση με το συγκεκριμένο δίκτυο δεν επιτρέπεται και το επίπεδο δικτύου ψάχνει για μία δεύτερη κατάλληλη συσκευή και επαναλαμβάνει την αίτηση για σύνδεση. Η διαδικασία συνεχίζεται μέχρι να έρθει κάποια θετική απάντηση ή μέχρι να μη βρίσκει κάποια άλλη κατάλληλη συσκευή, οπότε και το ανώτερο επίπεδο ενημερώνεται για την αδυναμία σύνδεσης στο συγκεκριμένο δίκτυο. Η διαδικασία αυτή μπορεί να ξεκινήσει μόνο από μία συσκευή που δεν ανήκει σε κάποιο δίκτυο.

Στην πλευρά του γονέα ακολουθείται η εξής διαδικασία. Η αίτηση για σύνδεση παραδίδεται ιεραρχικά στο επίπεδο δικτύου. Εκεί απορρίπτεται σε περίπτωση που ο πιθανός γονέας δεν είναι συντονιστής του δικτύου ή δρομολογητής ή αν δεν υπάρχει διαθέσιμη διεύθυνση για να δοθεί στη νέα συσκευή. Εφόσον δεν συντρέχει λόγος απόρριψης, ο πιθανός γονέας ελέγχει μήπως η συσκευή που έστειλε την αίτηση ανήκει ήδη στο δίκτυο. Ο έλεγχος αυτός γίνεται με σύγκριση της εκτεταμένης διεύθυνσης της συσκευής, που περιέχεται στην αίτηση, με τις διευθύνσεις που υπάρχουν στον πίνακα γειτόνων του. Αν βρεθεί μία εγγραφή, ο γονέας απαντά θετικά στην αίτηση με την διεύθυνση που βρίσκεται στον πίνακα. Σε διαφορετική περίπτωση, διαθέτει μία νέα διεύθυνση, η οποία είναι μοναδική στο δίκτυο, και στη συνέχεια απαντά και πάλι θετικά.

Στο δεύτερο τρόπο, η διαδικασία ξεκινά από τον πιθανό γονέα, ο οποίος θα πρέπει να είναι συντονιστής του δικτύου ή δρομολογητής και να γνωρίζει την εκτεταμένη διεύθυνση της συσκευής, την οποία επιθυμεί να εντάξει στο δίκτυο. Αρχικά το επίπεδο εφαρμογών ζητά από το επίπεδο δικτύου να συνδεθεί με τη συγκεκριμένη συσκευή. Το επίπεδο δικτύου συγκρίνει την εκτεταμένη διεύθυνση που του στάλθηκε με αυτές που υπάρχουν στον πίνακα γειτόνων. Αν βρεθεί στον πίνακα, ενημερώνει το επίπεδο εφαρμογών ότι η συγκεκριμένη συσκευή ανήκει ήδη στο δίκτυο αλλιώς διαθέτει μία νέα διεύθυνση για τη συσκευή και ενημερώνει το επίπεδο εφαρμογών ότι η συσκευή προστέθηκε στο δίκτυο.

Στην πλευρά του παιδιού το επίπεδο εφαρμογών ζητά από το επίπεδο δικτύου να κάνει εκπομπές μηνυμάτων προς όλα τα δίκτυα που λειτουργούν στην γύρω περιοχή. Τα μηνύματα

αυτά περιέχουν την εκτεταμένη διεύθυνση της συσκευής, ενώ ως διεύθυνση προορισμού έχουν τη διεύθυνση εκπομπής. Μόλις ένα τέτοιο μήνυμα φθάσει στη συσκευή του πιθανού γονέα, το επίπεδο δικτύου θα καταλάβει ότι πρόκειται για μία συσκευή που ανήκει στο δίκτυο και θα τις στείλει τη διεύθυνση την οποία θα πρέπει να χρησιμοποιεί για τις μεταδόσεις της στο δίκτυο.

Η αποχώρηση μίας συσκευής από ένα δίκτυο μπορεί και αυτή να γίνει με δύο τρόπους. Στον πρώτο τρόπο, το ίδιο το παιδί ζητά από το γονέα να αποχωρήσει. Ο γονέας εγκρίνει και η σύνδεση διακόπτεται. Στην περίπτωση όμως που το παιδί έχει και αυτό άλλα παιδιά, πρέπει πρώτα να ζητήσει από αυτά να φύγουν και στη συνέχεια να διακόψει τη σύνδεση. Στο δεύτερο τρόπο, ο γονέας ζητά από το παιδί να φύγει από το δίκτυο. Μόλις ένα παιδί λάβει αυτή την εντολή, ενεργεί όπως στον πρώτο τρόπο.

#### **5.7.4. Δημιουργία ενός δικτύου**

Μόνο συσκευές ZigBee που έχουν τη δυνατότητα να γίνουν συντονιστές δικτύων μπορούν να ξεκινήσουν τη διαδικασία δημιουργίας δικτύου. Αρχικά, το επίπεδο εφαρμογών ζητά από το επίπεδο δικτύου να δημιουργήσει ένα καινούριο δίκτυο με συγκεκριμένα χαρακτηριστικά. Η διαδικασία συνεχίζεται με την εκτέλεση ανίχνευσης ενέργειας σε ένα σύνολο καναλιών. Η ανίχνευση αυτή πραγματοποιείται από το επίπεδο δικτύου, με τη βοήθεια των κατώτερων επιπέδων. Με τη λήψη των αποτελεσμάτων, το επίπεδο δικτύου κατατάσσει τα κανάλια σε μία σειρά, ξεκινώντας από το κανάλι στο οποίο μετρήθηκε η χαμηλότερη ενέργεια. Επίσης απορρίπτει τα κανάλια στα οποία η ενέργεια ξεπερνά ένα αποδεκτό επίπεδο, το οποίο καθορίζεται από την εφαρμογή για την αποφυγή παρεμβολών.

Στη συνέχεια το επίπεδο δικτύου, με τη βοήθεια των κατώτερων, εκτελεί ενεργή ανίχνευση των αποδεκτών καναλιών, ψάχνοντας για συσκευές. Τα αποτελέσματα της ανίχνευσης παραδίδονται στο επίπεδο δικτύου, το οποίο προσδιορίζει τον αριθμό και τις ταυτότητες των δικτύων που λειτουργούν σε κάθε κανάλι. Ως κανάλι λειτουργίας του δικτύου επιλέγεται εκείνο με το μικρότερο αριθμό δικτύων. Έπειτα επιλέγεται η ταυτότητα του δικτύου. Αυτή έχει μήκος 16 bits, δεν πρέπει να είναι μία από τις κρατημένες για ειδικούς σκοπούς και πρέπει να προσδιορίζει μοναδικά το δίκτυο στη συγκεκριμένη περιοχή. Η αδυναμία εύρεσης κατάλληλου καναλιού ή μοναδικής ταυτότητας, οδηγεί τη διαδικασία σε τερματισμό.

Αφού επιλεγεί η ταυτότητα, το επίπεδο δικτύου του συντονιστή πρέπει να προσδιορίσει τη διεύθυνση του μέσα στο δίκτυο. Αυτή έχει μήκος 16 bits και την τιμή 0. Έπειτα μέσω του σημείου πρόσβασης υπηρεσιών διαχείρισης του επιπέδου MAC (MLME-SAP), το επίπεδο δικτύου διαμορφώνει τη λειτουργία του MAC, ανάλογα με τις απαιτήσεις της εφαρμογής. Τελικά, το επίπεδο εφαρμογών ενημερώνεται για την επιτυχία της αίτησης και ο συντονιστής είναι πλέον έτοιμος να δεχθεί νέες συσκευές στο δίκτυο με τη διαδικασία που περιγράφηκε στην προηγούμενη παράγραφο.

### **5.7.5. Δρομολόγηση πλαισίων**

Κάθε συντονιστής δικτύου και όλοι οι δρομολογητές έχουν από έναν πίνακα δρομολόγησης. Κάθε εγγραφή σε αυτόν περιέχει τη διεύθυνση προορισμού, την κατάσταση της διαδρομής και τη διεύθυνση του επόμενου βήματος. Εκτός του πίνακα δρομολόγησης, στις συσκευές των συντονιστών και των δρομολογητών υπάρχει και ένας πίνακας ανακάλυψης διαδρομής. Αυτός περιέχει την ταυτότητα του αιτήματος διαδρομής, τη διεύθυνση της συσκευής που έκανε την αίτηση, τη διεύθυνση του αποστολέα του αιτήματος, το κόστος της διαδρομής από τη συσκευή που έκανε την αίτηση ως τη συγκεκριμένη συσκευή, το κόστος της διαδρομής από τη συγκεκριμένη συσκευή ως τον τελικό προορισμό και το χρόνο λήξης.

Μόλις το επίπεδο εφαρμογών μιας συσκευής συντονιστή ή δρομολογητή στείλει ένα πλαίσιο για μετάδοση στο επίπεδο δικτύου, τότε αυτό ελέγχει τη διεύθυνση προορισμού του πλαισίου. Αν η συσκευή προορισμού είναι ένα από τα παιδιά του, τότε γίνεται απευθείας μετάδοση. Στην περίπτωση αυτή η διεύθυνση του επόμενου βήματος και η διεύθυνση προορισμού στον πίνακα δρομολόγησης είναι ίδιες. Αν η διεύθυνση προορισμού είναι η διεύθυνση εκπομπής, τότε στο πεδίο διεύθυνση προορισμού της επικεφαλίδας του πλαισίου MAC τοποθετείται η διεύθυνση εκπομπής, ενώ στο πεδίο ταυτότητα δικτύου προορισμού η ταυτότητα του δικτύου στο οποίο ανήκει η συσκευή.

Στην περίπτωση που δεν συμβαίνει κάτι από τα παραπάνω, το επίπεδο δικτύου ψάχνει στον πίνακα δρομολόγησης για μία εγγραφή που να έχει τη συγκεκριμένη διεύθυνση προορισμού. Αν υπάρχει, το πλαίσιο παραδίδεται στο επίπεδο MAC για αποστολή. Στο πεδίο διεύθυνση προορισμού της επικεφαλίδας του πλαισίου MAC τοποθετείται η διεύθυνση του



επόμενου βήματος. Το πεδίο διεύθυνση πηγής περιέχει τη διεύθυνση της συσκευής που έστειλε το πλαίσιο.

Στο επόμενο βήμα το πλαίσιο φθάνει σε μία ενδιάμεση συσκευή. Εκεί γίνεται δεκτό από το επίπεδο MAC της συσκευής, αφού έχει ως διεύθυνση προορισμού τη διεύθυνση της συγκεκριμένης συσκευής. Εφόσον δεν ανιχνευθεί κάποιο λάθος, το πλαίσιο παραδίδεται στο επίπεδο δικτύου. Από το πεδίο διεύθυνση προορισμού της επικεφαλίδας δικτύου (NWK Header), καταλαβαίνει ότι πρόκειται για ένα πλαίσιο που δεν προορίζεται για τη συγκεκριμένη συσκευή. Ψάχνει στον πίνακα δρομολόγησης για μία εγγραφή που να έχει τη συγκεκριμένη διεύθυνση προορισμού. Αν υπάρχει, το πλαίσιο παραδίδεται στο επίπεδο MAC για αποστολή. Στο πεδίο διεύθυνση προορισμού της επικεφαλίδας του πλαισίου MAC τοποθετείται η διεύθυνση του επόμενου βήματος από τον πίνακα δρομολόγησης, ενώ στο πεδίο διεύθυνση πηγής περιέχεται η διεύθυνση της ενδιάμεσης συσκευής. Με αυτά τα διαδοχικά βήματα, το πλαίσιο φθάνει στον τελικό προορισμό του και παραδίδεται στο επίπεδο εφαρμογών.

Υπάρχει βέβαια και το ενδεχόμενο ο πίνακας δρομολόγησης του αποστολέα ή κάποιας ενδιάμεσης συσκευής να μην περιέχει εγγραφή για την συγκεκριμένη διεύθυνση προορισμού. Στην περίπτωση αυτή, το πλαίσιο αποθηκεύεται προσωρινά και ξεκινά η διαδικασία αναζήτησης διαδρομής που περιγράφεται στην επόμενη παράγραφο. Με το τέλος αυτής, συμπληρώνεται ο πίνακας δρομολόγησης και συνεχίζεται η αποστολή του πλαισίου προς τον τελικό προορισμό.

Η παραπάνω διαδικασία ισχύει μόνο όταν η συσκευή έχει δυνατότητες δρομολόγησης. Οι συσκευές που δεν έχουν τέτοιες δυνατότητες, μπορούν να επικοινωνήσουν μόνο με τα παιδιά και το γονέα τους. Τα πλαίσια που έχουν προορισμό είτε το γονέα είτε κάποιο από τα παιδιά, μεταδίδονται απευθείας. Τα πλαίσια με προορισμό μία συσκευή που είναι απόγονος κάποιου από τα παιδιά, μεταδίδονται μέσω του κατάλληλου παιδιού. Όταν ο προορισμός του πλαισίου είναι μία τυχαία συσκευή, η μετάδοση του πλαισίου γίνεται μέσω του γονέα. Αν και αυτός δεν έχει δυνατότητες δρομολόγησης, η μετάδοση συνεχίζεται προς το δικό του γονέα. Η διαδικασία αυτή συνεχίζεται μέχρι το πλαίσιο να φθάσει σε μία συσκευή με δυνατότητες δρομολόγησης, οπότε και μεταδίδεται προς τον τελικό προορισμό.

#### **5.7.6. Ασφάλεια**

Για την ασφάλεια των μεταδιδόμενων πλαισίων, το επίπεδο δικτύου χρησιμοποιεί το πλαίσιο που φαίνεται στο σχήμα 5.5. Το πλαίσιο αποτελείται από την πλήρη επικεφαλίδα δικτύου (Full NWK header) και από το ασφαλές ωφέλιμο φορτίο δικτύου (Secured NWK payload). Η πλήρης επικεφαλίδα περιέχει την επικεφαλίδα του γενικού πλαισίου δικτύου και τη βοηθητική επικεφαλίδα πλαισίου (Auxiliary frame header).

Octets: Variable	14	Variable	
Original NWK Header ([B3], Clause 7.1)	Auxiliary frame header	Encrypted Payload	Encrypted Message Integrity Code (MIC)
		Secure frame payload = Output of CCM*	
Full NWK header		Secured NWK payload	

Σχήμα 5.5 Δομή πλαισίου ασφαλείας δικτύου

Η βοηθητική επικεφαλίδα πλαισίου έχει τη δομή του σχήματος 5.6. Αποτελείται από τα πεδία έλεγχου ασφάλειας (Security control), μετρητή πλαισίων (Frame Counter), διεύθυνσης πηγής (Source Address) και αριθμού ακολουθίας κλειδιού (Key Sequence Number).

Octets: 1	4	0/8	0/1
Security control	Frame Counter	Source Address	Key Sequence Number

Σχήμα 5.6 Δομή βοηθητικής επικεφαλίδας

Το πεδίο έλεγχου ασφάλειας (σχήμα 5.7) αποτελείται από τα παρακάτω υποπεδία:

- Επίπεδο ασφάλειας (Security level) που προσδιορίζει τις μεθόδους που έχουν χρησιμοποιηθεί στο πλαίσιο για την παροχή ασφάλειας.
- Ταυτότητα κλειδιού (Key identifier) που προσδιορίζει τον τύπο του κλειδιού κρυπτογράφησης που χρησιμοποιείται.
- Το επόμενο bit (Extended Nonce) δείχνει αν υπάρχει το πεδίο διεύθυνσης πηγής στη βοηθητική επικεφαλίδα.

Τέλος, τα επόμενα 2 bits είναι κρατημένα για μελλοντική χρήση.

Bit: 0-2	3-4	5	6-7
Security level	Key identifier	Extended Nonce	Reserved

Σχήμα 5.7 Δομή πλαισίου ελέγχου ασφαλείας

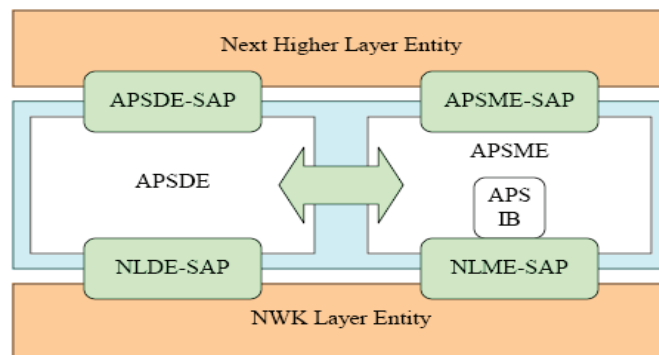
Το πεδίο διεύθυνσης πηγής περιέχει την εκτεταμένη διεύθυνση της συσκευής που είναι υπεύθυνη για την ασφάλεια του πακέτου ενώ το πεδίο μετρητή πλαισίων χρησιμοποιείται για την αρίθμηση των πλαισίων. Το πεδίο αριθμού ακολουθίας κλειδιού προσδιορίζει ακριβώς το κλειδί κρυπτογράφησης που χρησιμοποιείται. Τέλος, το πεδίο ασφαλούς ωφέλιμου φορτίου δικτύου έχει μεταβλητό μήκος και αποτελείται από το κρυπτογραφημένο ωφέλιμο φορτίο και τον κρυπτογραφημένο κωδικό ακεραιότητας μηνυμάτων.

## 5.8. Επίπεδο εφαρμογών (Application layer, APL)

Το επίπεδο εφαρμογών αποτελείται από το υποεπίπεδο υποστήριξης εφαρμογών (Application support sublayer, APS), το πλαίσιο εφαρμογών (Application framework, AF), τα αντικείμενα συσκευής ZigBee (ZigBee Device Objects, ZDO) και από τις 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. Το επίπεδο εφαρμογών έχει φτιαχτεί εξ' ολοκλήρου από τη Zigbee Alliance.

### 5.8.1. Υποεπίπεδο υποστήριξης εφαρμογών (Application support sublayer, APS)

Το υποεπίπεδο APS χρησιμοποιεί τις υπηρεσίες που του παρέχονται από το επίπεδο δικτύου μέσω των σημείων πρόσβασης υπηρεσιών NLDE (Network Layer Data Entity) και NLME (Network Layer Management Entity), ενώ ταυτόχρονα παρέχει τις υπηρεσίες του στις εφαρμογές και στα ZDO. Αυτές οι υπηρεσίες παρέχονται διαμέσου των οντοτήτων δεδομένων (APSDE) και διαχείρισης (APSME) και των αντίστοιχων σημείων πρόσβασης υπηρεσιών (APSDE-SAP και APSME-SAP). Επίσης στο υποεπίπεδο αυτό υπάρχει και μία βάση δεδομένων (APSIB), που περιέχει πληροφορίες σχετικές με τη λειτουργία του.



Σχήμα 5.8 Μοντέλο αναφοράς APS.

### 5.8.1.1. Υπηρεσίες δεδομένων υποεπιπέδου APS

Η υπηρεσία δεδομένων που παρέχει το υποεπίπεδο APS είναι η δημιουργία των μονάδων δεδομένων πρωτοκόλλου APS (APS-PDUs) από τα δεδομένα των εφαρμογών και η μεταφορά των δεδομένων από τη μία συσκευή στην άλλη. Οι στοιχειώδεις υπηρεσίες που υποστηρίζει το APSDE-SAP είναι οι:

- APSDE-DATA.request. Με αυτή μία εφαρμογή ενημερώνει το υποεπίπεδο APS ότι έχει δεδομένα για μεταφορά.
- APSDE-DATA.confirm. Με αυτή το υποεπίπεδο APS ενημερώνει την εφαρμογή για την έκβαση της αίτησης.

- APSDE-DATA.indication. Με αυτή το υποεπίπεδο APS ενημερώνει την εφαρμογή ότι έχουν φθάσει δεδομένα από μία άλλη συσκευή.

#### **5.8.1.2. Υπηρεσίες διαχείρισης υποεπιπέδου APS**

Οι υπηρεσίες διαχείρισης επιτρέπουν σε μία εφαρμογή να αλληλεπιδρά με τη στοίβα πρωτοκόλλων. Οι υπηρεσίες που παρέχονται είναι η σύνδεση δύο ή και περισσότερων συσκευών του ίδιου δικτύου, η διαχείριση της βάσης δεδομένων και η ασφάλεια των δεδομένων. Οι στοιχειώδεις υπηρεσίες που υποστηρίζει το APSME-SAP είναι οι:

- APSME-BIND. Με τη request, μία εφαρμογή ζητά από το υποεπίπεδο APS να συνδεθεί με μία εφαρμογή μίας άλλης συσκευής. Με την confirm, το υποεπίπεδο APS ενημερώνει την εφαρμογή για την έκβαση της αίτησης.
- APSME-GET. Με τη request, μία εφαρμογή ζητά από το υποεπίπεδο APS να διαβάσει την τιμή μιας ιδιότητας της βάσης δεδομένων. Με την confirm, το υποεπίπεδο APS επιστρέφει την τιμή της ιδιότητας.
- APSME-SET. Με τη request, μία εφαρμογή ζητά από το υποεπίπεδο APS να αλλάξει την τιμή μιας ιδιότητας της βάσης δεδομένων. Με την confirm, το υποεπίπεδο APS ενημερώνει την εφαρμογή για την έκβαση της αίτησης.
- APSME-UNBIND. Με τη request, μία εφαρμογή ζητά από το υποεπίπεδο APS να τερματίσει μία σύνδεση. Με την confirm, το υποεπίπεδο APS ενημερώνει την εφαρμογή για την έκβαση της αίτησης.

### 5.8.1.3. Πλαίσια υποεπιπέδου APS

Το πλαίσιο του υποεπιπέδου APS (σχήμα 5.9) αποτελείται από την επικεφαλίδα APS (APS header) και από το ωφέλιμο φορτίο APS (APS payload). Η επικεφαλίδα περιέχει το πεδίο έλεγχος πλαισίου και τα πεδία διευθύνσεων.

Octets: 1	0/1	0/1	0/2	0/1	Variable
Frame control	Destination end-point	Cluster Identifier	Profile Identifier	Source endpoint	Frame payload
	Addressing fields				
APS header					APS payload

Σχήμα 5.9 Πλαίσιο APS

Το πεδίο έλεγχος πλαισίου (σχήμα 5.10) αποτελείται από τα παρακάτω υποπεδία:

- Τύπος πλαισίου (Frame type) που προσδιορίζει τον τύπο του πλαισίου.
- Τρόπος παράδοσης (Delivery mode) που καθορίζει τον τρόπο παράδοσης των δεδομένων στις εφαρμογές.
- Τρόπος έμμεσης μετάδοσης (Indirect address mode) που καθορίζει αν υπάρχουν τα υποπεδία σημείο τερματισμού προορισμού και σημείο τερματισμού πηγής στην επικεφαλίδα του πλαισίου.
- Ασφάλεια (Security) που προσδιορίζει αν το πλαίσιο είναι κρυπτογραφημένο.
- Αίτημα επιβεβαίωσης (Ack. Request) που προσδιορίζει αν το συγκεκριμένο πλαίσιο απαιτεί επιβεβαίωση.

Το επόμενο bit είναι κρατημένο για μελλοντική χρήση.

Bits: 0-1	2-3	4	5	6	7
Frame type	Delivery mode	Indirect address mode <sup>a</sup>	Security	Ack. request	Reserved

Σχήμα 5.10 Έλεγχος πλαισίου.

Τα πεδία διευθύνσεων αποτελούνται τα παρακάτω υποπεδία:

- Σημείο τερματισμού στον προορισμό (destination endpoint) που προσδιορίζει την εφαρμογή στην οποία πρέπει να παραδοθούν τα δεδομένα.
- Ταυτότητα συμπλέγματος (cluster identifier) που καθορίζει τη μορφή που έχουν τα μεταφερόμενα δεδομένα.
- Ταυτότητα προφίλ (profile identifier) που προσδιορίζει το προφίλ στο οποίο απευθύνεται το πλαίσιο.
- Σημείο τερματισμού στην πηγή (source endpoint) που προσδιορίζει την εφαρμογή που στέλνει τα δεδομένα.

Τα πεδία διευθύνσεων δεν περιέχονται στα πλαίσια εντολών. Σε αυτά οι διευθύνσεις προορισμού και πηγής περιέχονται σε υποπεδία στο ωφέλιμο φορτίο. Το πεδίο ωφέλιμο φορτίο έχει μεταβλητό μήκος και περιέχει συγκεκριμένες πληροφορίες για κάθε τύπο πλαισίου. Στα πλαίσια δεδομένων, περιέχει τα δεδομένα των εφαρμογών. Στα πλαίσια εντολών, περιέχει την ταυτότητα της εντολής και την εντολή ενώ στα πλαίσια επιβεβαιώσεων δεν υπάρχει.

#### **5.8.1.4. Μετάδοση, παραλαβή και επιβεβαίωση πλαισίων**

Μία συσκευή μπορεί να μεταδώσει ένα πλαίσιο APS μόνο αν ανήκει στο δικτύου. Η μετάδοση μπορεί να είναι είτε άμεση είτε έμμεση. Στην άμεση, τα πλαίσια περιλαμβάνουν τα υποπεδία σημείο τερματισμού στον προορισμό και σημείο τερματισμού στην πηγή. Όλες οι συσκευές που διαθέτουν πίνακα συνδέσεων κάνουν άμεσες μεταδόσεις πλαισίων προς κάθε προορισμό.

Στην περίπτωση των έμμεσων μεταδόσεων, τα πλαίσια περιλαμβάνουν μόνο ένα από τα παραπάνω υποπεδία και έχουν την τιμή 10 στο υποπεδίο τρόπος παράδοσης. Όλες οι συσκευές που δεν διαθέτουν πίνακα συνδέσεων, όταν κάνουν μία έμμεση μετάδοση θα πρέπει να την κατευθύνουν προς τη συσκευή του συντονιστή που περιέχει τον αντίστοιχο πίνακα. Το συγκεκριμένο πλαίσιο δεν θα περιέχει το υποπεδίο σημείο τερματισμού στον προορισμό. Ο συντονιστής με τη βοήθεια του πίνακα συνδέσεων, θα βρει τη διεύθυνση προορισμού και το αντίστοιχο σημείο τερματισμού. Στη συνέχεια, θα μεταδώσει το πλαίσιο προς τον τελικό προορισμό χωρίς το υποπεδίο σημείο τερματισμού στην πηγή.

Η έμμεση μετάδοση χρησιμοποιείται κυρίως από πολύ απλές συσκευές που δεν έχουν τη δυνατότητα να αποθηκεύουν μεγάλο όγκο δεδομένων. Για μία άμεση μετάδοση, η συσκευή θα πρέπει να γνωρίζει τη διεύθυνση προορισμού, το σημείο τερματισμού στον προορισμό και την ταυτότητα του συμπλέγματος. Σε μία έμμεση μετάδοση, μόνο η ταυτότητα του συμπλέγματος απαιτείται. Και στις δύο περιπτώσεις, έμμεση και άμεση μετάδοση, η μεταφορά του πλαισίου από την πηγή στον προορισμό επιτυγχάνεται μέσω των υπηρεσιών που παρέχουν τα κατώτερα στρώματα.

Στην πλευρά του δέκτη, το υποεπίπεδο APS παραλαμβάνει τα πλαίσια από το επίπεδο δικτύου. Όταν το πλαίσιο περιέχει και τα δύο υποπεδία, το APS το παραδίδει στην εφαρμογή που καθορίζεται από το σημείο τερματισμού στον προορισμό. Στην περίπτωση των έμμεσων μεταδόσεων, όπου το υποπεδίο τρόπος παράδοσης έχει την τιμή 10, ο τρόπος χειρισμού του πλαισίου είναι διαφορετικός. Αν ένα πλαίσιο περιέχει μόνο το υποπεδίο σημείο τερματισμού στην πηγή και φθάσει σε μία συσκευή που διαθέτει πίνακα συνδέσεων, τότε θα βρεθεί στον πίνακα η αντίστοιχη εγγραφή και το πλαίσιο θα αποσταλεί προς τον τελικό του προορισμό χωρίς το υποπεδίο σημείο τερματισμού στην πηγή. Αν η συσκευή δεν διαθέτει πίνακα ή ο πίνακας δεν περιέχει την κατάλληλη εγγραφή, το πλαίσιο θα απορριφθεί. Επίσης, το πλαίσιο θα απορριφθεί και στην περίπτωση που η συσκευή δεν διαθέτει πίνακα και δεν υπάρχει το υποπεδίο σημείο τερματισμού στον προορισμό.

Εκτός από την άμεση και έμμεση μετάδοση, υπάρχει και η εκπομπή προς όλες τις εφαρμογές που περιέχονται σε μία συσκευή. Στην περίπτωση αυτή, το υποπεδίο τρόπος παράδοσης έχει την τιμή 01 και το πλαίσιο APS έχει τα υποπεδία ταυτότητα συμπλέγματος, ταυτότητα προφίλ και σημείο τερματισμού στην πηγή (με τιμή 255). Ως διεύθυνση προορισμού στην επικεφαλίδα δικτύου χρησιμοποιείται η διεύθυνση εκπομπής στο δίκτυο.

Σε κάθε πλαίσιο APS υπάρχει το υποπεδίο αίτημα επιβεβαίωσης (Ack request). Όταν αυτό έχει την τιμή 1, ο δημιουργός του πλαισίου απαιτεί από τον παραλήπτη να του επιβεβαιώσει την ορθή λήψη. Εφόσον η επιβεβαίωση δεν έρθει μέσα σε ένα ορισμένο χρονικό διάστημα, το πλαίσιο μεταδίδεται ξανά. Η παραπάνω διαδικασία επαναλαμβάνεται είτε μέχρι να έρθει μία επιβεβαίωση είτε μέχρι να συμπληρωθεί ο μέγιστος αριθμός επαναμεταδόσεων (3 επαναμεταδόσεις). Στην περίπτωση των έμμεσων μεταδόσεων, η ενδιάμεση συσκευή που αναλαμβάνει να μεταδώσει το πλαίσιο στον τελικό προορισμό, έχει την υποχρέωση να στείλει



μία επιβεβαίωση στο δημιουργό του πλαισίου. Στη συνέχεια στέλνει το πλαίσιο προς τον τελικό προορισμό και απαιτεί από αυτόν να της στείλει επιβεβαίωση.

#### 5.8.1.5. Ασφάλεια στο υποεπίπεδο APS

Το υποεπίπεδο APS είναι υπεύθυνο για όλες τις διαδικασίες που απαιτούνται για την εγκαθίδρυση ασφαλών συνδέσεων. Μέσω των υπηρεσιών του επιτυγχάνεται η δημιουργία των διαφόρων κλειδιών, η μεταφορά τους από τη μία συσκευή στην άλλη, η αλλαγή τους και η είσοδος μίας συσκευής σε ένα δίκτυο που παρέχει ασφάλεια. Όλες οι παραπάνω διαδικασίες ξεκινούν κατόπιν εντολής του ZigBee Device Object (ZDO), το οποίο καθορίζει το επίπεδο ασφαλείας που θα χρησιμοποιηθεί και διαχειρίζεται τα κλειδιά.

Για την ασφάλεια των μεταδιδόμενων πλαισίων, το υποεπίπεδο APS χρησιμοποιεί το πλαίσιο του σχήματος 5.11. Αποτελείται από την πλήρη επικεφαλίδα APS (Full APS header) και από το ασφαλές ωφέλιμο φορτίο APS (Secured APS payload). Η πλήρης επικεφαλίδα περιέχει την επικεφαλίδα του γενικού πλαισίου APS και τη βοηθητική επικεφαλίδα πλαισίου (Auxiliary frame header).

Octets: variable	5 or 6	Variable	
Original APS Header {[B7], Clause 7.1}	Auxiliary frame header	Encrypted Payload	Encrypted Message Integrity Code (MIC)
		Secure frame payload = Output of CCM*	
Full APS header		Secured APS payload	

Σχήμα 5.11 Ασφαλές πλαίσιο APS

Το πεδίο ασφαλές ωφέλιμο φορτίο APS έχει μεταβλητό μήκος και αποτελείται από το κρυπτογραφημένο ωφέλιμο φορτίο και τον κρυπτογραφημένο κωδικό ακεραιότητας μηνυμάτων. Για την κρυπτογράφηση χρησιμοποιείται ο κώδικας CCM.

#### 5.8.2. Πλαίσιο εφαρμογών (Application framework, AF)

Το πλαίσιο εφαρμογών είναι το περιβάλλον μέσα στο οποίο φιλοξενούνται όλες οι εφαρμογές σε μία συσκευή ZigBee. Μέσα σε αυτό στέλνουν και λαμβάνουν δεδομένα διαμέσου του APSDE-SAP. Υπεύθυνα για τον έλεγχο και τη διαχείριση των εφαρμογών είναι τα ZigBee Device Objects (ZDO). Έως και 240 εφαρμογές μπορούν να υπάρξουν στο πλαίσιο εφαρμογών. Για το διαχωρισμό τους, καθεμία έχει το δικό της σημείο τερματισμού (αριθμημένα από 1 ως 240). Το σημείο τερματισμού 0 χρησιμοποιείται για τη μεταφορά δεδομένων προς τα ZDO, ενώ το 255 χρησιμοποιείται για την εκπομπή δεδομένων προς όλες τις εφαρμογές. Τα σημεία τερματισμού 241-254 είναι κρατημένα για μελλοντική χρήση.

#### **5.8.2.1. Τα προφίλ του ZigBee**

Κάθε συσκευή ZigBee μπορεί να υποστηρίζει ένα ή και περισσότερα προφίλ. Δύο συσκευές μπορούν να επικοινωνήσουν μόνο αν υποστηρίζουν ένα συγκεκριμένο προφίλ. Το κάθε προφίλ έχει τη δική του ταυτότητα (profile identifier, 2 bytes), η οποία είναι μοναδική. Τα προφίλ καθορίζουν τη φύση των εφαρμογών, τα χαρακτηριστικά των συσκευών και τα συμπλέγματα δεδομένων που χρησιμοποιούνται στην επικοινωνία.

Τα συμπλέγματα δεδομένων έχουν και αυτά τη δική τους ταυτότητα (cluster identifier, 1 byte), η οποία είναι μοναδική μέσα σε ένα συγκεκριμένο προφίλ. Καθορίζουν τις δομές των δεδομένων που χρησιμοποιούνται στην επικοινωνία μεταξύ των σημείων τερματισμού. Κάθε σύμπλεγμα αποτελείται από ιδιότητες. Κάθε ιδιότητα έχει μία μοναδική ταυτότητα (attribute identifier, 2 bytes) μέσα σε ένα συγκεκριμένο σύμπλεγμα.

Τα χαρακτηριστικά των συσκευών ZigBee καθορίζονται από τους παρακάτω περιγραφείς:

- Περιγραφέας κόμβου. Περιέχει πληροφορίες σχετικά με τις συσκευές που υπάρχουν σε αυτόν, τις ζώνες συχνότητας που λειτουργεί, τον κωδικό του κατασκευαστή και το μέγιστο μέγεθος των δεδομένων που παραδίδονται στο υποεπίπεδο APS ή παραλαμβάνονται από αυτό.
- Περιγραφέας ισχύος κόμβου. Περιέχει πληροφορίες σχετικά με την κατάσταση λειτουργίας του δέκτη, τις διαθέσιμες πηγές ενέργειας, την πηγή ενέργειας που χρησιμοποιείται και το επίπεδο φόρτισης.

- Απλός περιγραφέας. Περιέχει τον αριθμό του σημείου, την ταυτότητα του προφίλ που υποστηρίζει το συγκεκριμένο σημείο, την ταυτότητα των συσκευών που υποστηρίζονται, τον αριθμό και τη λίστα των εισερχόμενων και εξερχόμενων συμπλεγμάτων δεδομένων που υποστηρίζονται.
- Σύνθετος περιγραφέας. Είναι προαιρετικός και περιέχει πληροφορίες για τη συσκευή, όπως το όνομα του κατασκευαστή, τον αριθμό της και το URL στο οποίο βρίσκονται όλες οι πληροφορίες σχετικά με αυτή.
- Περιγραφέας χρήστη. Είναι και αυτός προαιρετικός και περιέχει το όνομα που δίνει ο χρήστης στη συσκευή.

#### 5.8.2.2. Πλαίσια εντολών AF (Application framework)

Το γενικό πλαίσιο εντολών AF φαίνεται στο σχήμα 5.12 και αποτελείται από τα παρακάτω πεδία:

- Μέτρηση συναλλαγών (Transaction count) που καθορίζει τον αριθμό των συναλλαγών που περιέχονται στο πλαίσιο.
- Τύπος πλαισίου (Frame type) που καθορίζει τον τύπο της υπηρεσίας που χρησιμοποιείται.
- Συναλλαγές (Transactions) που αποτελούνται από τον αριθμό ακολουθίας (8 bits) και από το πεδίο των δεδομένων.

Bits: 4	4	Variable	Variable	Variable
Transaction count	Frame type	Transaction 1	...	Transaction <i>n</i>

Σχήμα 5.12 Πλαίσιο εντολών AF

#### 5.8.3. Αντικείμενα συσκευής ZigBee (ZigBee Device Objects, ZDO)

Τα αντικείμενα συσκευής ZigBee (ZigBee Device Objects, ZDO) βρίσκονται στο επίπεδο εφαρμογών. Επικοινωνούν με το υποεπίπεδο APS και το επίπεδο δικτύου μέσω των APSME-SAP και NLME-SAP αντίστοιχα. Η μεταφορά των δεδομένων σε αυτά γίνεται από το υποεπίπεδο APS μέσω του APSDE-SAP και του σημείου τερματισμού 0. Επικοινωνούν με τις άλλες εφαρμογές της συσκευής μέσω μίας κοινής διεπαφής και επιπλέον μπορούν να επικοινωνήσουν με οποιαδήποτε άλλη συσκευή ZigBee χρησιμοποιώντας το προφίλ συσκευής. Είναι υπεύθυνα για τις παρακάτω διαδικασίες:

- Αρχικοποίηση του υποεπιπέδου APS και του επιπέδου δικτύου.
- Ανακάλυψη της ταυτότητας των άλλων συσκευών του δικτύου.
- Ανακάλυψη υπηρεσιών. Με αυτή, μία συσκευή μπορεί να ανακαλύψει όλες τις εφαρμογές που είναι διαθέσιμες στα σημεία τερματισμού μίας άλλης, να ζητήσει τον περιγραφέα κόμβου ή τον περιγραφέα ισχύος μιας συγκεκριμένης συσκευής ή και τον απλό περιγραφέα ενός συγκεκριμένου σημείου τερματισμού της.
- Διαχείριση ασφάλειας. Τα ZDO είναι υπεύθυνα για την ενεργοποίηση ή όχι της ασφάλειας. Όταν η ασφάλεια είναι ενεργοποιημένη, πρέπει να καθορίσουν το επίπεδο στο οποίο θα εφαρμόζεται, να δημιουργήσουν τα κλειδιά που είναι απαραίτητα για την κρυπτογράφηση των δεδομένων, να φροντίσουν για τη μεταφορά αυτών στις υπόλοιπες συσκευές που συμμετέχουν στην επικοινωνία και να πιστοποιούν την ταυτότητα των συσκευών που στέλνουν τα πλαίσια.
- Διαχείριση δικτύου. Τα ZDO ζητούν από το επίπεδο δικτύου να ανιχνεύσει ένα συγκεκριμένο σύνολο καναλιών, επιλέγουν το κατάλληλο κανάλι για τη δημιουργία ενός δικτύου ή για την ένταξη σε ένα δίκτυο με βάση τα αποτελέσματα των ανιχνεύσεων, ξεκινούν τη διαδικασία για αποχώρηση ή ένταξη σε ένα δίκτυο και επιτρέπουν ή απαγορεύουν σε άλλες συσκευές να συνδεθούν.
- Διαχείριση συνδέσεων μεταξύ σημείων τερματισμού. Τα ZDO διαχειρίζονται τις εγγραφές σε έναν πίνακα συνδέσεων και εξυπηρετούν τις αιτήσεις των εφαρμογών για σύνδεση με κάποια άλλη εφαρμογή σε μία απομακρυσμένη συσκευή.

- Διαχείριση κόμβου. Με αυτή, οι συντονιστές και οι δρομολογητές μπορούν να επιτύχουν τη διαχείριση μιας απομακρυσμένης συσκευής τερματισμού.

## ΚΕΦΑΛΑΙΟ 6

### ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ WEB ΕΦΑΡΜΟΓΩΝ

#### 6.1. Apache

Ο Apache HTTP server, συχνά αναφερόμενος απλά σαν Apache, είναι ένας web server ο οποίος διαδραμάτισε καίριο ρόλο στην αρχική ανάπτυξη του παγκόσμιου ιστού. Το 2009 έγινε ο web server που ξεπέρασε το όριο των εκατό εκατομμυρίων σελίδων στο διαδίκτυο. Ο Apache ήταν η πρώτη βιώσιμη εναλλακτική λύση απέναντι στον Netscape Corporation web server (γνωστό σήμερα ως Sun Java System web server), και από τότε εξελίχθηκε σε υπολογίσιμο αντίπαλο άλλων web server που βασίζονται σε Unix όσον αφορά την λειτουργικότητα και τις επιδόσεις.

Ο Apache αναπτύσσεται και συντηρείται από μια ανοικτή κοινότητα προγραμματιστών υπό την αιγίδα του Apache Software Foundation. Η εφαρμογή είναι διαθέσιμη για μια μεγάλη ποικιλία λειτουργικών συστημάτων στα οποία περιλαμβάνονται τα Unix, GNU, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X, Microsoft Windows, OS/2, TPF και eComStation. Ο Apache χαρακτηρίζεται ως ένα λογισμικό ανοικτού κώδικα. Από τον Απρίλιο του 1996 και μετά, ο Apache είναι ο πιο δημοφιλής HTTP server του διαδικτύου. Επίσης μετά από μέτρηση που πραγματοποιήθηκε τον Αύγουστο του 2009, ο Apache εξυπηρετεί το 54,32% όλων των σελίδων του διαδικτύου και το 66% από τις 1.000.000 πιο δημοφιλείς. Υπάρχουν δύο επεξηγήσεις όσον αφορά το όνομα του project. Σύμφωνα με το Apache Foundation, το όνομα επελέγη από σεβασμό στην φυλή των αυτοχθόνων Αμερικανών Apache οι οποίοι ήταν γνωστοί για την αντοχή και τις ικανότητες τους στην μάχη. Παρ' όλα αυτά, κατά την περίοδο 1996-2001, η επεξήγηση που έδινε η ιστοσελίδα του Apache project ήταν ότι επειδή πρόκειται για έναν server ο οποίος δημιουργήθηκε βασισμένος σε προσθήκες (patches), ονομάστηκε patchy server και με τον καιρό κατέληξε να αποκαλείται Apache.

Ο Apache υποστηρίζει μία πολύ μεγάλη ποικιλία χαρακτηριστικών και δυνατοτήτων. Πολλά από αυτά προσαρτώνται στον πυρήνα με την μορφή modules επεκτείνοντας τις δυνατότητες του. Αυτά περιλαμβάνουν από υποστήριξη server-side γλωσσών προγραμματισμού έως και αλγόριθμους αυθεντικοποίησης. Κάποιες από τις δημοφιλείς γλώσσες που υποστηρίζονται είναι οι Perl, Python, Tcl και PHP. Κάποια από τα δημοφιλή modules αυθεντικοποίησης που υποστηρίζονται είναι τα mod\_access, mod\_auth, mod\_digest και mod\_auth\_digest.

Κάποιες από τις άλλες δυνατότητες περιλαμβάνουν υποστήριξη των πρωτοκόλλων SSL και TLS(mod\_ssl), ένα proxy module, ένα URL rewriter (mod\_rewrite), παραμετροποιημένες καταγραφές συμβάντων (mod\_log\_config) καθώς και υποστήριξη φίλτρων (mod\_ext\_filter). Μια δημοφιλείς μέθοδος συμπίεσης που χρησιμοποιείται στον Apache είναι το external extension module(mod\_gzip) το οποίο βοηθά στον να μειωθεί το μέγεθος των ιστοσελίδων που εξυπηρετούνται μέσω HTTP. Επίσης δημοφιλείς είναι και το ModSecurity το οποίο είναι μια μηχανή ανοιχτού κώδικα που εντοπίζει και εμποδίζει εισβολές σε διαδικτυακές εφαρμογές. Το ιστορικό του Apache μπορεί να διαχειριστεί μέσω ενός web browser χρησιμοποιώντας ελεύθερες εφαρμογές όπως AWStats/W3Perl ή το Visitors.

Επιπλέον χαρακτηριστικά του Apache είναι το Virtual Hosting, που επιτρέπει σε πολλές διαφορετικές ιστοσελίδες να εξυπηρετούνται από μία μόνο εγκατάσταση του server, παραμετροποιήσιμα μηνύματα σφάλματος, Βάσεις δεδομένων βασισμένες σε αυθεντικοποίηση DBMS, διαχείριση περιεχομένου και υποστήριξη διαφόρων GUIs(Graphical User Interfaces).

Ο Apache χρησιμοποιείται κυρίως για την εξυπηρέτηση στατικών και δυναμικών σελίδων στο διαδίκτυο. Πολλές διαδικτυακές εφαρμογές σχεδιάζονται με βάση το περιβάλλον και τα χαρακτηριστικά που προσφέρει ο Apache. Ο συγκεκριμένος server αποτελεί κομμάτι της δημοφιλούς ομάδας εφαρμογών LAMP την οποία αποτελούν ο Apache, το λειτουργικό Linux, το σύστημα διαχείρισης βάσεων δεδομένων MySQL και οι γλώσσες προγραμματισμού PHP/Perl/Python. Ο Apache αποτελεί βασικό κομμάτι πολλών πακέτων εφαρμογών όπως : Oracle Database, IBM WebSphere application server, WebObject application server, Mac OS X, Novell NetWare6.5 καθώς και σε πολλές διανομές του λειτουργικού συστήματος Linux.

## 6.2. PHP

Η PHP είναι μια γλώσσα προγραμματισμού που σχεδιάστηκε για τη δημιουργία δυναμικών σελίδων στο διαδίκτυο και είναι επισήμως γνωστή ως: HyperText preprocessor. Είναι μια server-side (εκτελείται στον διακομιστή) scripting γλώσσα που γράφεται συνήθως πλαισιωμένη από HTML, για μορφοποίηση των αποτελεσμάτων. Αντίθετα από μια συνηθισμένη HTML σελίδα η σελίδα PHP δεν στέλνεται άμεσα σε έναν πελάτη (client), άντ' αυτού πρώτα αναλύεται και μετά αποστέλλεται το παραγόμενο αποτέλεσμα. Τα στοιχεία HTML στον πηγαίο κώδικα μένουν ως έχουν, αλλά ο PHP κώδικας ερμηνεύεται και εκτελείται. Ο κώδικας PHP μπορεί να θέσει ερωτήματα σε βάσεις δεδομένων, να δημιουργήσει εικόνες, να διαβάσει και να γράψει αρχεία, να συνδεθεί με απομακρυσμένους υπολογιστές, κ.ο.κ. Σε γενικές γραμμές οι δυνατότητες που μας δίνει είναι απεριόριστες. Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP. Η PHP 5 ήταν ένα μεγάλο βήμα μπροστά για τη γλώσσα, αν και όχι τόσο μεγάλο όσο η μετάβαση από την PHP 3 στην PHP 4. Η PHP 5 προσφέρει scripts για αντικειμενοστραφή προγραμματισμό (object-oriented). Επίσης, υπάρχει μια μεγάλη ποικιλία από συναρτήσεις για αντικείμενα (objects) που τα κάνει πολύ πιο ευέλικτα και εύκολα στη χρήση τους. Ακόμη, τα αντικείμενα αντιμετωπίζονται πάντα ως αναφορές (references) ώστε να βοηθηθούν οι προγραμματιστές που δυσκολεύονται να εργαστούν με τα αντικείμενα. Η PHP χρησιμοποιεί μια μίξη από διερμηνευση (interpretation)



και μεταγλώττιση (compilation) έτσι ώστε να μπορέσει να δώσει στους προγραμματιστές τον καλύτερο δυνατό συνδυασμό απόδοσης και ευελιξίας. Στο παρασκήνιο, η PHP μεταγλωττίζει το script σε μια σειρά από εντολές (instructions), που είναι γνωστές με τον όρο opcodes, οι οποίες εντολές εκτελούνται μία-μία μέχρι να τελειώσει το script. Αυτό είναι κάτι διαφορετικό από τις παραδοσιακές γλώσσες που μεταγλωττίζονται, όπως είναι η C++, όπου ο κώδικας μεταγλωττίζεται σε εκτελέσιμο κώδικα μηχανής, ενώ η PHP μεταγλωττίζει εκ νέου το script κάθε φορά που αυτό απαιτείται. Αυτή η συνεχής μεταγλώττιση μπορεί να φαίνεται ως απώλεια χρόνου, αλλά δεν είναι καθόλου κακή καθώς δεν χρειάζεται να κάνουμε συνέχεια εμείς τη μεταγλώττιση των scripts όταν γίνονται κάποιες αλλαγές σ' αυτά.

### 6.3. MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) το οποίο μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά του από την κόρη του Μόντυ Βιντένιους, την Μάι. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Η βάση δεδομένων MySQL έχει γίνει η πιο δημοφιλής βάση δεδομένων ανοιχτού λογισμικού εξαιτίας της σταθερά υψηλής απόδοσής της, της αξιοπιστίας της και της ευκολίας της χρήσης της. Χρησιμοποιείται παγκοσμίως τόσο από μεμονωμένους δημιουργούς διαδικτυακών χώρων όσο και από πολλούς από τους μεγαλύτερους και τους πιο ραγδαία αναπτυσσόμενους οργανισμούς για την εξοικονόμηση χρόνου και χρήματος. Επίσης, χρησιμοποιείται για τη δημιουργία διαδικτυακών χώρων με μεγάλο όγκο δεδομένων, κρίσιμων συστημάτων για τη λειτουργία εταιρικών εφαρμογών και πακέτων λογισμικού μεγάλων εταιρειών. Η MySQL δεν είναι μόνο η πιο δημοφιλής βάση δεδομένων ανοιχτού λογισμικού, αλλά συγχρόνως έχει γίνει και η επιλεγμένη βάση δεδομένων για τη νέα γενιά εφαρμογών που βασίζεται στο LAMP (Linux, Apache, MySQL, PHP/Perl/Python). Η MySQL τρέχει σε περισσότερες από 20 πλατφόρμες συμπεριλαμβανομένων του Linux, των Windows, του OS/X, του HP-UX, του AIX και του Netware, παρέχοντας στο χρήστη όλη την απαιτούμενη ευελιξία. Η MySQL είναι γραμμένη σε C και C++. Η έκδοση της MySQL Enterprise Server 5.5 διαθέτει τα ακόλουθα χαρακτηριστικά:

- Συναλλαγές ACID για τη δημιουργία αξιόπιστων και ασφαλών κρίσιμων εταιρικών εφαρμογών.
- Αποθηκευμένες διαδικασίες για τη βελτίωση της παραγωγικότητας των

προγραμματιστών.

- Διαδικασίες πυροδότησης για την επιβολή πολύπλοκων επιχειρηματικών κανόνων στο επίπεδο της βάσης δεδομένων.
- Μεθόδους προστασίας των ευαίσθητων δεδομένων.
- Σχήμα Πληροφοριών για την παροχή εύκολης πρόσβασης σε με τα δεδομένα.
- Κατανεμημένες Συναλλαγές για την υποστήριξη πολύπλοκων συναλλαγών εντός πολλαπλών βάσεων δεδομένων.
- Αρχιτεκτονική Μηχανής Αποθήκευσης για την παροχή της μέγιστης δυνατής ευελιξίας.
- Μηχανή Αποθήκευσης Αρχείων για την καταγραφή του ιστορικού και των δεδομένων λογιστικού ελέγχου.
- Ενοποιημένη Μηχανή Αποθήκευσης για τη δημιουργία μίας λογικής βάσης δεδομένων από πολλούς φυσικούς εξυπηρετητές.
- Εκδόσεις με διορθώσεις των πιο σημαντικών σφαλμάτων.

Με τη βοήθεια της PHP μπορούμε να συνδεθούμε σε έναν οποιοδήποτε MySQL Server στον οποίο έχουμε λογαριασμό, να πάρουμε δεδομένα από ήδη υπάρχουσες βάσεις, να εισάγουμε δεδομένα σε πίνακες βάσεων, να ανανεώσουμε κάποια υπάρχοντα δεδομένα, να φτιάξουμε νέες βάσεις και νέους πίνακες και γενικά να κάνουμε οτιδήποτε γίνεται με μια MySQL βάση δεδομένων. Επομένως, μέσα από τις Web σελίδες μας μπορούμε να διαχειριστούμε εύκολα μια MySQL βάση δεδομένων και έτσι οι σελίδες μας να αποκτήσουν πολλές άλλες δυνατότητες που απαιτούν οι σύγχρονες απαιτήσεις των χρηστών δηλαδή να γίνουν δυναμικές, ελκυστικές και ανταγωνιστικές. Μια τυπική διαδικτυακή συναλλαγή βάσεων δεδομένων αποτελείται από τις παρακάτω φάσεις:

- Ο web browser ενός χρήστη κάνει μια HTTP αίτηση για μια συγκεκριμένη διαδικτυακή σελίδα.
- Ο διαδικτυακός διακομιστής (Apache Server) λαμβάνει την αίτηση για τη σελίδα, ανακαλεί το αρχείο και το περνά στη μηχανή PHP για επεξεργασία.
- Η μηχανή PHP αρχίζει την ανάλυση του script. Μέσα στο script, υπάρχει μια εντολή που συνδέει τη βάση δεδομένων και εκτελεί ένα ερώτημα. Η PHP ανοίγει μια σύνδεση με το MySQL διακομιστή (server) και στέλνει το κατάλληλο ερώτημα.
- Ο MySQL διακομιστής (server) λαμβάνει το ερώτημα της βάσης δεδομένων, το

επεξεργάζεται και στέλνει τα αποτελέσματα ξανά στη μηχανή PHP.

- Η μηχανή PHP σταματά την εκτέλεση του script, που συνήθως περιλαμβάνει τη μορφοποίηση των αποτελεσμάτων του ερωτήματος σε HTML. Επιστρέφει μετά την τελική HTML σελίδα στον web διακομιστή (Apache Server).
- Ο διαδικτυακός διακομιστής (Apache Server) περνά την HTML σελίδα ξανά στο browser, όπου ο χρήστης μπορεί να δει τα αποτελέσματα που ζήτησε.

## 6.4. PhpMyAdmin

Το phpMyAdmin είναι ένα σύνολο από php scripts με το οποίο διαχειριζόμαστε τις βάσεις δεδομένων που έχουμε μέσω web. Το phpMyAdmin μπορεί να διαχειριστεί ένα ολόκληρο mysql server ή ακόμα και απλές βάσεις δεδομένων όπου ο κάθε χρήστης έχει ένα λογαριασμό και μπορεί να δημιουργήσει και να διαχειριστεί τις δικές του βάσεις δεδομένων. Υποστηρίζει 47 γλώσσες μεταξύ των οποίων και τα Ελληνικά και είναι λογισμικό ανοιχτού κώδικα.

Οι δυνατότητες του PhpMyAdmin είναι οι εξής:

- Δημιουργεί και να διαγράφει βάσεις δεδομένων.
- Δημιουργεί, τροποποιεί, διαγράφει, αντιγράφει και μετονομάζει πίνακες.
- Κάνει συντήρηση της βάσης.
- Προσθέτει, διαγράφει και τροποποιεί πεδία πινάκων.
- Εκτελεί Sql ερωτήματα, ακόμα και ομαδικά (batch).
- Διαχειρίζεται κλειδιά σε πεδία.
- Φορτώνει αρχεία κειμένου σε πίνακες.
- Δημιουργεί και διαβάζει πίνακες (που προέρχονται από dump βάσης).
- Εξάγει δεδομένα σε μορφή CVS, Latex, XML.
- Διαχειρίζεται πολλούς διακομιστές.
- Διαχειρίζεται τους χρήστες MySQL και τα δικαιώματά τους.
- Ελέγχει την αναφορική ακεραιότητα των δεδομένων των MyISAM πινάκων.
- Δημιουργεί PDF γραφικών του layout της βάσης δεδομένων.
- Εκτελεί αναζητήσεις σε όλη τη βάση ή μέρος αυτής.
- Υποστηρίζει πίνακες InnoDB και ξένα κλειδιά.
- Υποστηρίζει MySQLi, μια βελτιωμένη επέκταση της MySQL.



## ΚΕΦΑΛΑΙΟ 7

### Arduino

#### 7.1 Τι είναι το Arduino

Το Arduino είναι ένα εργαλείο που επιτρέπει σε ένα πλήθος υπολογιστικών μηχανών, τα οποία μπορούν να αντιληφθούν, να αισθανθούν και να ελέγξουν τον φυσικό κόσμο, πολύ περισσότερο από έναν Η/Υ. Είναι μία πλατφόρμα ανοιχτού κώδικα, βασισμένη σε έναν απλό μικροελεγκτή και ένα περιβάλλον ανάπτυξης με το οποίο δίνεται η δυνατότητα να γραφτεί κώδικας πάνω στην πλακέτα.

Μπορεί να χρησιμοποιηθεί για την ανάπτυξη διαδραστικών αντικειμένων, την συλλογή δεδομένων από μία πληθώρα αισθητήρων και διακοπών, και τον έλεγχο μηχανών, φωτών και πολλών άλλων εξόδων στον φυσικό κόσμο.

Η γλώσσα που χρησιμοποιεί είναι μία εφαρμογή της Wiring, μίας παρόμοιας υπολογιστικής πλατφόρμας, η οποία χρησιμοποιεί το περιβάλλον προγραμματισμού πολυμέσων Processing.

#### 7.2. Γιατί Arduino

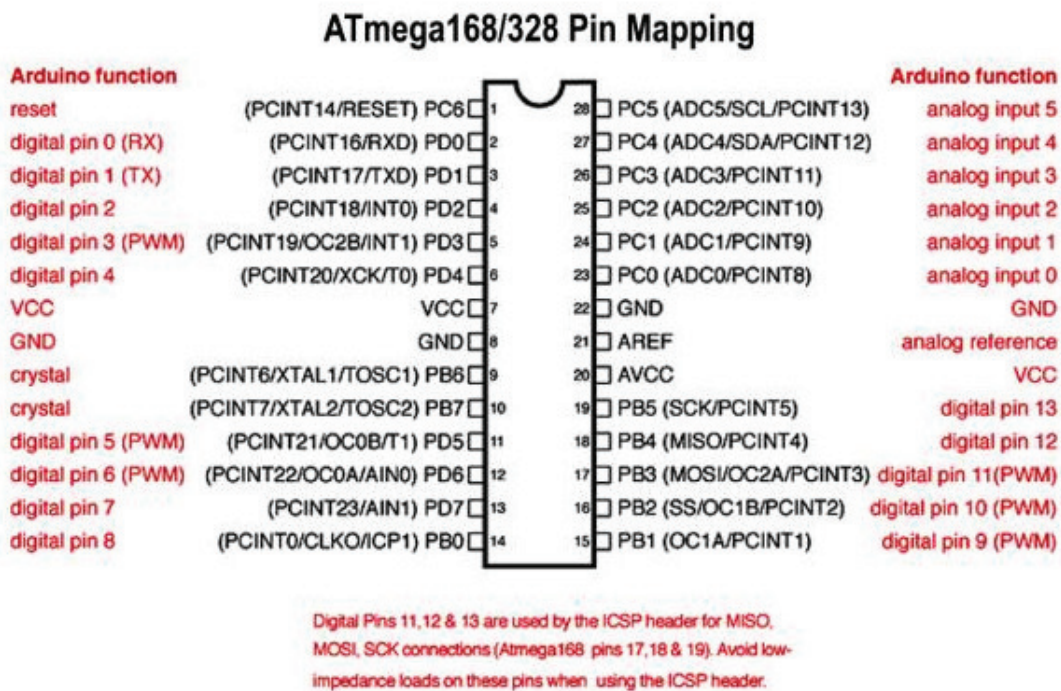
Υπάρχουν πάρα πολλοί μικροελεγκτές και πλατφόρμες μικροελεγκτών στην αγορά, όπως . Parallax Basic Stamp, Netmedia's BX-24, Phidgets, και το Handyboard του MIT, και πολλά άλλα. Καθένα από αυτά τα εργαλεία παίρνουν όλες τις ακατάστατες λεπτομέρειες που δημιουργούνται από τον προγραμματισμό ενός μικροελεγκτή και τις συμμαζεύουν σε ένα πακέτο ιδιαίτερα εύκολο στην χρήση. Το Arduino απλοποιεί μεν τον προγραμματισμό του μικροελεγκτή, αλλά παρέχει και πολλές ευκολίες δε. Όπως οι παρακάτω:

- Ανέξοδο – Οι πλακέτες Arduino είναι σχετικά ανέξοδες σε σύγκριση με τις υπόλοιπες πλατφόρμες μικροελεγκτών. Η πιο φτηνή έκδοσή τους, μπορεί να συναρμολογηθεί με το χέρι, ενώ και οι ήδη συναρμολογημένες δεν κοστίζουν παραπάνω από 50 ευρώ.
- Cross-platform – Το λογισμικό του Arduino τρέχει σε: Windows, Macintosh OSX, και Linux λειτουργικά συστήματα, σε αντίθεση με τα περισσότερα συστήματα μικροελεγκτών, που περιορίζονται στα Windows.
- Απλό και καθαρό περιβάλλον προγραμματισμού – Το περιβάλλον προγραμματισμού του Arduino είναι εύκολο στη χρήση, ακόμα και για αρχάριους, ενώ παραμένει αρκούντως ευλύγιστο ούτως ώστε οι έμπειροι χρήστες να επωφεληθούν από αυτό. Για τους εκπαιδευτές, είναι κατά κάποιο τρόπο βολικά βασισμένο στο Processing προγραμματιστικό περιβάλλον, πράγμα που επιτρέπει στους μαθητευόμενους να έχουν ήδη την εξοικίωση ενός παρόμοιου περιβάλλοντος.
- Ανοιχτός κώδικας και επεκτάσιμο λογισμικό – Το λογισμικό του Arduino είναι δημοσιευμένο σαν εργαλείο ανοιχτού κώδικα, διαθέσιμο για πειραματισμό και επέκταση από έμπειρους χρήστες. Η γλώσσα προγραμματισμού μπορεί να επεκταθεί από C++ βιβλιοθήκες, και ανθρώπους που επιθυμούν να καταλάβουν το τις τεχνικές λεπτομέρειες ούτως ώστε να προσφέρουν στο Arduino το άλμα προς την AVR C γλώσσα προγραμματισμού, πάνω στην οποία είναι βασισμένο.
- Ανοιχτός κώδικας και επεκτάσιμο υλικό - Το Arduino είναι βασισμένο στους μικροελεγκτές Atmel ATMEGA8 και ATMEGA168. Οποιοδήποτε σχέδιο για τα ηλεκτρονικά κυκλώματα που χρησιμοποιεί, μεταξύ αυτών και οι προαναφερθέντες μικροελεγκτές, είναι δημοσιοποιημένο, ούτως ώστε έμπειροι σχεδιαστές κυκλωμάτων να μπορούν να κάνουν την δική τους έκδοση του μοντέλου, επεκτείνοντας και βελτιώνοντάς το. Ακόμα και σχετικά αρχάριοι χρήστες μπορούν να αναπτύξουν μία τέτοια πλακέτα σε μία πλακέτα ανάπτυξης κυκλωμάτων, γνωστή και ως breadboard.

### 7.3. Τεχνικά Χαρακτηριστικά του Arduino Uno

Το Arduino Uno έχει 14 εισόδους / εξόδους, 6 από τις οποίες μπορούν να χρησιμοποιηθούν για διαμόρφωση Pulse Width Modulation. Έχει 6 αναλογικές εισόδους, είσοδο USB, τροφοδοσία, ενώ επιτυγχάνει ταχύτητα ρολογιού 16 MHz με την βοήθεια ενός κρυσταλλικού ταλαντωτή.

Διαφέρει από τους πρόγονούς του με το να μην διαθέτει FTDI USB σε σειριακό οδηγό. Τουναντίον, ο μικροελεγκτής Atmega 16U2 είναι προγραμματισμένος έτσι, ώστε να είναι μετατροπέας από USB σε σειριακό.



### Mapping ATmega168/328 to Arduino pins

Σχήμα 7.1 Χαρτογράφηση του Arduino Uno

<b>Microcontroller</b>	<b>ATmega328</b>
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega 328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Σχήμα 7.2 Τα τεχνικά χαρακτηριστικά του Arduino Uno



## ΚΕΦΑΛΑΙΟ 8

### ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΑΝΤΙΜΕΤΩΠΙΣΤΗΚΑΝ

#### 8.1. SunSPOT SDK

Από τις πρώτες κι όλας ημέρες κατά τη διάρκεια της εκπόνησης της παρούσης πτυχιακής εργασίας, ήταν εμφανή κάποια προβλήματα, ένα από αυτά ήταν και η εγκατάσταση του SDK των αισθητήρων SunSPOT. Ο κύριος λόγος, ήταν ότι δεν υπάρχει κάποιο ενημερωμένο documentation ούτε στον επίσημο ιστότοπο και forum των SunSPOT, ούτε και σε κάποιο ανεπίσημο.

Η πρώτη προσπάθεια εγκατάστασης του εν λόγω SDK είχε γίνει με τον παρακάτω τρόπο:

1. Σε native λειτουργικό περιβάλλον Windows 7 Home Premium 64bit, με μνήμη RAM 4 GB, επεξεργαστή Intel Core I5.
2. Με την χρήση εικονικού μηχανήματος Ubuntu Linux 10.10

Το αποτέλεσμα ήταν ανεπιτυχές λόγω ασυμβατότητας του JDK του εικονικού μηχανήματος.

Έγινε προσπάθεια για αναβάθμιση του εικονικού μηχανήματος, μέχρι και το λειτουργικό σύστημα Ubuntu Linux 12.04 όπου και τελικά βρέθηκε η λύση.

Ο λόγος είναι ότι στα Ubuntu Linux 12.04 μαζί με μία πληθώρα αλλαγών, έγιναν αλλαγές και στον τρόπο εμφάνισης των λαθών. Εν τέλει το πρόβλημα αποδείχθηκε πως ήταν ότι κάθε έκδοση Ubuntu Linux έρχεται μαζί με το openjdk αδιάφορο έκδοσης. Το openjdk όμως, είναι ασύμβατο με το SDK των SunSPOT, πράγμα που δεν ήταν ευχιάκριτο μέχρι τότε.

Η λύση ήταν σχετικά απλή, εφόσον το μόνο που χρειάστηκε να γίνει ήταν να κατέβει στο εικονικό μηχάνημα η έκδοση (τελευταία ή όχι) του JDK της εταιρίας Oracle πλέον, και να αντικαταστήσει το μέχρι εκείνη την ώρα εγκατεστημένο openjdk των Ubuntu Linux.

Αξίζει να σημειωθεί ότι ακόμα και αυτή τη στιγμή που μιλάμε δεν υπάρχει κατάλληλο documentation σε καμία από τις προαναφερθείσες τοποθεσίες, καθώς επίσης και οποιαδήποτε καθοδήγηση από τον ιστότοπο της εταιρίας Oracle και τους θυγατρικούς αυτού, δεν υποστηρίζεται πλέον.

## ΚΕΦΑΛΑΙΟ 9

### ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Η παρούσα πτυχιακή εργασία έχει ολοκληρωθεί με πληρότητα, παρόλα αυτά υπάρχουν αρκετά σημεία ενδιαφέροντος, και ιδιαίτερης αξίας και προσοχής. Ορισμένα από αυτά δημιουργήθηκαν κατά την πάροδο του χρόνου, άλλα προέκυψαν σαν ιδέες, και παραθέτονται εδώ:

1. Δημιουργία tutorial για την ορθή εγκατάσταση του SunSPOT SDK.

Θεωρώ ότι είναι μείζωνος σημασίας η δημιουργία ενός τέτοιου tutorial, καθώς στο Τμήμα Τηλεπικοινωνιακών Συστημάτων και Δικτύων του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Μεσολογίου, υπάρχει πληθώρα αισθητήρων SunSPOT και πουθενά δεν υπάρχει κάποιος σωστός τρόπος να δείξει σε σπουδαστές και δυνητικά ακόμα και καθηγητές το πώς χρησιμοποιούνται.

2. Βελτιστοποίηση του ήδη υπάρχοντος πηγαίου κώδικα.

Αυτή τη στιγμή ο κώδικας μπορεί να υποστηρίξει μεγάλο πλήθος αισθητήρων σαν είσοδο, αλλά με το πέρασμα του χρόνου, πολλά πράγματα θα αλλάξουν με την πλήρη υλοποίηση του IPv6, για όλη την εφαρμογή. Επομένως ο πηγαίος κώδικας θα πρέπει να ετοιμαστεί για αυτήν την μετάβαση.

3. Επέκταση της εφαρμογής με χρήση επιπλέον συσκευών.

Η εφαρμογή αυτή, ως ώρα έχει επιτύχει στο να δέχεται δεδομένα από δύο ειδών αισθητήρες (θερμότητας και φωτεινότητας), να τα αποτυπώνει με δύο διαφορετικούς τρόπους (ιστοσελίδα και εφαρμογή για Android) και να παίρνει μία έξυπνη απόφαση χρησιμοποιώντας το Arduino UNO. Παρόλα αυτά, για να γίνει μία πλήρης εφαρμογή διαχείρισης κτιριακών υποδομών, θα πρέπει να βρεθεί ο τρόπος με τον οποίο θα δύναται να διαχειρίζεται περισσότερα από αυτά. Μία πληθώρα από αισθητήρες μπορεί να χρησιμοποιηθεί, με σκοπό την συλλογή δεδομένων και από άλλες πηγές.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

[1] Harald Sundmaeker Patrick, Guillemin Peter Friess Sylvie Woelfflé, Μάρτιος 2010, “Vision and Challenges for Realising the Internet of Things”

[2] September 2009, “Internet of Things – Strategic Research Roadmap”

[3] Αριστοτέλης, «Κατηγορία»

[4] Ιωάννης ο Φιλόπονος, «Περί των Κατηγοριών»

[5] Adrian A. Hopgood, 2001, Knowledge-Based Systems for Engineers and Scientists

[6] Derek J. Croome 2004, “Intelligent buildings: design, management and operation”

[7] Anshuman Sharma, 2003, “Design of Wireless Sensor Networks for Building Management Systems”

[8] Android Developers: What is Android?

<http://developer.android.com/guide/basics/whatis-android.html>

[9] Android versions comparison

<http://socialcompare.com/en/comparison/android-versionscomparison>

[10] Lauren Darcey & Shane Conder, “Sams Teach Yourself Android Application Development in 24 Hours”

[11] [www.sunspotworld.com](http://www.sunspotworld.com)

[12] ZigBee Document 053474r06, version 1.0, 14 December 2004, ZigBee Alliance, “ZigBee Specification”

[13]William C. Craig, ] “ZigBee: Wireless Control That Simply Works”

[14] Jo Woon Chong, Ho Young Hwang, Chang Yong Jung, and Dan Keun Sung, “Analysis of throughput & energy consumption in a ZigBee network”

[15] Dennis Cox, Emil Jovanov, Aleksandar Milenkovic, “Time Synchronization for ZigBee Networks”,

[16]Paolo Baronti, Prashant Pillai, Vince W.C. Chook ,Stefano Chessa, Alberto Gotta , Y. Fun Hu, “Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards”

[17] [www.arduino.cc](http://www.arduino.cc)

## ΠΑΡΑΡΤΗΜΑ

### ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ

#### **Script Python για την διασύνδεση του Arduino Uno με την βάση δεδομένων.**

```
#!/usr/bin/python

import MySQLdb

import time

import serial

ser = serial.Serial('/dev/ttyUSB0', 115200)

while True:

    db = MySQLdb.connect("83.212.107.16","test","test","spot" )

    cursor = db.cursor()

    sql = "SELECT * FROM new_table \
        ORDER BY id DESC LIMIT 0,1"

    cursor.execute(sql)

    results = cursor.fetchall()

    for row in results:

        id = row[0]

        node = row[1]

        value = row[2]

        date = row[3]

        print "id=%s,node=%s,value=%d,date=%s" % \
```

```

(id, node, value, date )
db.close()
if value < 20:
    ser.write('c')
    print "ooo"
    while value < 20:
        db = MySQLdb.connect("83.212.107.16","test","test","spot" )
        cursor = db.cursor()
        sql = "SELECT * FROM new_table \
ORDER BY id DESC LIMIT 0,1"
        cursor.execute(sql)
        results = cursor.fetchall()
        for row in results:
            id = row[0]
            node = row[1]
            value = row[2]
            date = row[3]
            print "value=%d" % \
(value)
        db.close()
    if value >= 20:
        break
    time.sleep(5)
ser.write('f')
time.sleep(10)

```

## Κώδικας PHP της εφαρμογής web

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="refresh" content="10" >

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Heterogeneous Systems</title>

<style>

body {

}

h1 {

text-align: center;

font-size: 35px;

font-family:"Times New Roman", Times, serif;

font-style:italic;

letter-spacing:2px

}

table {

margin-top: 1%;

background-color: #ffffff;

width: 80%;

}

div.scrollWrapper{

}

td {
```



```

padding: 10px;

border: none;

text-align: center;

}

</style>

<body>

<h1>Real-Time Values From SunSpots</h1>

<?php

    $dbcnx = mysql_connect("83.212.107.16","test","test");

    if (!$dbcnx)

        die('Could not connect: ' . mysql_error());

    if (!mysql_select_db("spot" )

        die('Αδύνατος ο εντοπισμός της βάσης δεδομένων. ' . mysql_error());

    mysql_query("SET NAMES 'utf8'", $dbcnx);

    $sql="SELECT * FROM new_table ORDER BY id DESC LIMIT 0,15";

    $result = mysql_query($sql);

    if (!$result) {

        echo("<P> Λάθος στην εκτέλεση του query : " . mysql_error() . "</P>");

        exit();

    }

?>

<div align="center" class="scrollWrapper">

<table align="center" border="2">

```

```

<tbody>
<tr>
    <td>Αριθμός Συμβάντος</td>
    <td>Κόμβος</td>
    <td>Τιμή</td>
    <td>Ημερομηνία</td>
<?php
    while ( $row = mysql_fetch_array($result) ) {
        echo "<tr><td>" . $row["id"] . "</td><td>" . $row["node"] . "</td><td>" . $row["value"] . "</td><td>" .
        $row["date"] . "</td>";

        if($row["value"]<20){
            echo '<td><div style="color: red; text-decoration: blink;">Value is really low</div></td>';
        }
        echo "</tr>";
    }
    echo "</tbody></table></div>";
?>
<br \><br \><br \>
<p align="right">&#169 Spiros Simeonidis 2013</p>
</body>
</html>

```

## **Κώδικας Java που υπάρχει στον σταθμό βάσης SunSPOT**

### **SendDataDemoGuiHostApplication**

```
package org.sunspotworld.demo;
```

```

import com.sun.spot.io.j2me.radiogram.*;

import com.sun.spot.peripheral.ota.OTACommandServer;
import com.sun.spot.util.IEEEAddress;
import javax.microedition.io.*;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

/**
 * This application is the 'on Desktop' portion of the SendDataDemo.
 * This host application collects sensor samples sent by the 'on SPOT'
 * portion running on neighboring SPOTs and graphs them in a window.
 *
 * @author Vipul Gupta
 * modified Ron Goldman
 */
public class SendDataDemoGuiHostApplication {

    // Broadcast port on which we listen for sensor samples

    private static final int HOST_PORT = 67;

    private JTextArea status;

    private long[] addresses = new long[8];

```

```

private DataWindow[] plots = new DataWindow[8];

sql sql=null;

private void setup() {

    JFrame fr = new JFrame("Send Data Host App");

    status = new JTextArea();

    JScrollPane sp = new JScrollPane(status);

    fr.add(sp);

    fr.setSize(360, 200);

    fr.validate();

    fr.setVisible(true);

    for (int i = 0; i < addresses.length; i++) {

        addresses[i] = 0;

        plots[i] = null;

    }

}

private DataWindow findPlot(long addr) {

    for (int i = 0; i < addresses.length; i++) {

        if (addresses[i] == 0) {

            String ieee = IEEEAddress.toDottedHex(addr);

            status.append("Received packet from SPOT: " + ieee + "\n");

            addresses[i] = addr;

            plots[i] = new DataWindow(ieeee);

            final int ii = i;

```

```

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            plots[ii].setVisible(true);
        }
    });
    return plots[i];
}
if (addresses[i] == addr) {
    return plots[i];
}
}
return plots[0];
}

```

```

private void run() throws Exception {
    RadiogramConnection rCon;
    Radiogram dg;
    try {
        // Open up a server-side broadcast radiogram connection
        // to listen for sensor readings being sent by different SPOTs
        rCon = (RadiogramConnection) Connector.open("radiogram://:" + HOST_PORT);
        dg = (Radiogram)rCon.newDatagram(rCon.getMaximumLength());
    } catch (Exception e) {
        System.err.println("setUp caught " + e.getMessage());
        throw e;
    }
}

```

```
}
```

```
status.append("Listening...\n");
```

```
// Main data collection loop
```

```
while (true) {
```

```
    try {
```

```
        // Read sensor sample received over the radio
```

```
        rCon.receive(dg);
```

```
        DataWindow dw = findPlot(dg.getAddressAsLong());
```

```
        long time = dg.readLong();    // read time of the reading
```

```
        int val = dg.readInt();    // read the sensor value
```

```
        dw.addData(time, val);
```

```
        String ieee = IEEEAddress.toDottedHex(dg.getAddressAsLong());
```

```
        //System.out.println(dg.getAddressAsLong());
```

```
        long inside = 5716365247801778L;
```

```
        long car = 5716365247796868L;
```

```
        long office = 5716365247804354L;
```

```
        long outside = 5716365247799709L;
```

```
        if (dg.getAddressAsLong() == inside){
```

```
            ieee = "Inside Sensor";}
```

```
        else if (dg.getAddressAsLong() == car){
```

```
            ieee = "Inside Lighting Sensor";
```

```
        }else if (dg.getAddressAsLong() == office){
```

```

ieee = "Outside Lighting Sensor";

}else if (dg.getAddressAsLong()==outside){

ieee= "Outside Sensor";

}

else {

    ieee="No Sensor Found";

}

sql db=new sql(ieee, val);

}catch (Exception e) {

    System.err.println("Caught " + e + " while reading sensor samples.");

    throw e;

}

}

}

/**
 * Start up the host application.
 *
 * @param args any command line arguments
 */
public static void main(String[] args) throws Exception {

    // register the application's name with the OTA Command server & start OTA running
    OTACommandServer.start("SendDataDemo-GUI");

```

```
SendDataDemoGuiHostApplication app = new SendDataDemoGuiHostApplication();  
app.setup();  
app.run();  
}  
}
```

**sql.java**

## **Κώδικας που υπάρχει στους αισθητήρες φωτεινότητας SunSPOT**

```
package org.sunspotworld.demo;
```

```
import com.sun.spot.io.j2me.radiogram.*;  
import com.sun.spot.resources.Resources;  
import com.sun.spot.resources.transducers.ITriColorLED;  
import com.sun.spot.resources.transducers.ILightSensor;  
import com.sun.spot.util.Utils;  
import javax.microedition.io.*;  
import javax.microedition.midlet.MIDlet;  
import javax.microedition.midlet.MIDletStateChangeException;
```

```
/**
```

```
* This application is the 'on SPOT' portion of the SendDataDemo. It
```

```
* periodically samples a sensor value on the SPOT and transmits it to
```



```

* a desktop application (the 'on Desktop' portion of the SendDataDemo)
* where the values are displayed.
*
* @author: Vipul Gupta
* modified: Ron Goldman
*/
public class SensorSampler extends MIDlet {

    private static final int HOST_PORT = 67;

    private static final int SAMPLE_PERIOD = 10 * 1000; // in milliseconds

    protected void startApp() throws MIDletStateChangeException {

        RadiogramConnection rCon = null;

        Datagram dg = null;

        String ourAddress = System.getProperty("IEEE_ADDRESS");

        ILightSensor lightSensor = (ILightSensor)Resources.lookup(ILightSensor.class);

        ITriColorLED led = (ITriColorLED)Resources.lookup(ITriColorLED.class, "LED7");

        System.out.println("Starting sensor sampler application on " + ourAddress + " ...");

        // Listen for downloads/commands over USB connection
        new com.sun.spot.service.BootloaderListenerService().getInstance().start();

    try {

        // Open up a broadcast connection to the host port

```

```

// where the 'on Desktop' portion of this demo is listening

rCon = (RadiogramConnection) Connector.open("radiogram://broadcast:" + HOST_PORT);

dg = rCon.newDatagram(50); // only sending 12 bytes of data
} catch (Exception e) {

    System.err.println("Caught " + e + " in connection initialization.");

    notifyDestroyed();

}

while (true) {

    try {

        // Get the current time and sensor reading

        long now = System.currentTimeMillis();

        int reading = lightSensor.getValue();

        // Flash an LED to indicate a sampling event

        led.setRGB(255, 255, 255);

        led.setOn();

        Utils.sleep(50);

        led.setOff();

        // Package the time and sensor reading into a radio datagram and send it.

        dg.reset();

        dg.writeLong(now);

        dg.writeInt(reading);

        rCon.send(dg);

```

```

        System.out.println("Light value = " + reading);

        // Go to sleep to conserve battery
        Utils.sleep(SAMPLE_PERIOD - (System.currentTimeMillis() - now));
    } catch (Exception e) {
        System.err.println("Caught " + e + " while collecting/sending sensor sample.");
    }
}

protected void pauseApp() {
    // This will never be called by the Squawk VM
}

protected void destroyApp(boolean arg0) throws MIDletStateChangeException {
    // Only called if startApp throws any exception other than MIDletStateChangeException
}
}

```

## **Κώδικας που υπάρχει στους αισθητήρες θερμοκρασίας SunSPOT**

```

package org.sunspotworld.demo;

import com.sun.spot.io.j2me.radiogram.*;
import com.sun.spot.resources.Resources;
import com.sun.spot.resources.transducers.ITriColorLED;
import com.sun.spot.resources.transducers.ILightSensor;

```

```

import com.sun.spot.resources.transducers.ITemperatureInput;

import com.sun.spot.sensorboard.EDemoBoard;

import com.sun.spot.util.Utils;

import javax.microedition.io.*;

import javax.microedition.midlet.MIDlet;

import javax.microedition.midlet.MIDletStateChangeException;

/**
 * This application is the 'on SPOT' portion of the SendDataDemo. It
 * periodically samples a sensor value on the SPOT and transmits it to
 * a desktop application (the 'on Desktop' portion of the SendDataDemo)
 * where the values are displayed.
 *
 * @author: Vipul Gupta
 * modified: Ron Goldman
 */
public class SensorSampler extends MIDlet {

    private static final int HOST_PORT = 67;

    private static final int SAMPLE_PERIOD = 10 * 1000; // in milliseconds

    protected void startApp() throws MIDletStateChangeException {

        RadiogramConnection rCon = null;

        Datagram dg = null;

        String ourAddress = System.getProperty("IEEE_ADDRESS");

```

```

//ILightSensor lightSensor = (ILightSensor)Resources.lookup(ILightSensor.class);

ITriColorLED led = (ITriColorLED)Resources.lookup(ITriColorLED.class, "LED7");

ITemperatureInput tempSensor = EDemoBoard.getInstance().getADCTemperature();

System.out.println("Starting sensor sampler application on " + ourAddress + " ...");

    // Listen for downloads/commands over USB connection
    new com.sun.spot.service.BootloaderListenerService().getInstance().start();

try {
    // Open up a broadcast connection to the host port
    // where the 'on Desktop' portion of this demo is listening
    rCon = (RadiogramConnection) Connector.open("radiogram://broadcast:" + HOST_PORT);
    dg = rCon.newDatagram(50); // only sending 12 bytes of data
} catch (Exception e) {
    System.err.println("Caught " + e + " in connection initialization.");
    notifyDestroyed();
}

while (true) {
    try {
        // Get the current time and sensor reading
        long now = System.currentTimeMillis();

        //int reading = lightSensor.getValue();

        int tempCelsius = (int) tempSensor.getCelsius();
    }
}

```

```

// Flash an LED to indicate a sampling event

led.setRGB(255, 255, 255);

led.setOn();

Utils.sleep(50);

led.setOff();

// Package the time and sensor reading into a radio datagram and send it.

dg.reset();

dg.writeLong(now);

// dg.writeInt(reading);

dg.writeInt(tempCelsius);

rCon.send(dg);

System.out.println("Temperature = " + tempCelsius);

// Go to sleep to conserve battery

Utils.sleep(SAMPLE_PERIOD - (System.currentTimeMillis() - now));

} catch (Exception e) {

    System.err.println("Caught " + e + " while collecting/sending sensor sample.");

}

}

}

protected void pauseApp() {

    // This will never be called by the Squawk VM

```

```
}  
  
protected void destroyApp(boolean arg0) throws MIDletStateChangeException {  
    // Only called if startApp throws any exception other than MIDletStateChangeException  
}  
}
```

## **Κώδικας που υπάρχει στους αισθητήρες φωτεινότητας SunSPOT**

```
/*  
 * SensorSampler.java  
 *  
 * Copyright (c) 2008-2010 Sun Microsystems, Inc.  
 *  
 * Permission is hereby granted, free of charge, to any person obtaining a copy  
 * of this software and associated documentation files (the "Software"), to  
 * deal in the Software without restriction, including without limitation the  
 * rights to use, copy, modify, merge, publish, distribute, sublicense, and/or  
 * sell copies of the Software, and to permit persons to whom the Software is  
 * furnished to do so, subject to the following conditions:  
 *  
 * The above copyright notice and this permission notice shall be included in  
 * all copies or substantial portions of the Software.  
 *  
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
```

```
* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
* FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
* DEALINGS IN THE SOFTWARE.
*/
```

```
package org.sunspotworld.demo;
```

```
import com.sun.spot.io.j2me.radiogram.*;
```

```
import com.sun.spot.resources.Resources;
```

```
import com.sun.spot.resources.transducers.ITriColorLED;
```

```
import com.sun.spot.resources.transducers.ILightSensor;
```

```
import com.sun.spot.util.Utils;
```

```
import javax.microedition.io.*;
```

```
import javax.microedition.midlet.MIDlet;
```

```
import javax.microedition.midlet.MIDletStateChangeException;
```

```
/**
```

```
* This application is the 'on SPOT' portion of the SendDataDemo. It
* periodically samples a sensor value on the SPOT and transmits it to
* a desktop application (the 'on Desktop' portion of the SendDataDemo)
* where the values are displayed.
```

```
*
```

```
* @author: Vipul Gupta
```

```
* modified: Ron Goldman
```

```
*/
```



```

public class SensorSampler extends MIDlet {

    private static final int HOST_PORT = 67;

    private static final int SAMPLE_PERIOD = 10 * 1000; // in milliseconds

    protected void startApp() throws MIDletStateChangeException {

        RadiogramConnection rCon = null;

        Datagram dg = null;

        String ourAddress = System.getProperty("IEEE_ADDRESS");

        ILightSensor lightSensor = (ILightSensor)Resources.lookup(ILightSensor.class);

        ITriColorLED led = (ITriColorLED)Resources.lookup(ITriColorLED.class, "LED7");

        System.out.println("Starting sensor sampler application on " + ourAddress + " ...");

        // Listen for downloads/commands over USB connection
        new com.sun.spot.service.BootloaderListenerService().getInstance().start();

    try {

        // Open up a broadcast connection to the host port

        // where the 'on Desktop' portion of this demo is listening

        rCon = (RadiogramConnection) Connector.open("radiogram://broadcast:" + HOST_PORT);

        dg = rCon.newDatagram(50); // only sending 12 bytes of data

    } catch (Exception e) {

        System.err.println("Caught " + e + " in connection initialization.");

        notifyDestroyed();
    }
}

```

```
}
```

```
while (true) {
```

```
    try {
```

```
        // Get the current time and sensor reading
```

```
        long now = System.currentTimeMillis();
```

```
        int reading = lightSensor.getValue();
```

```
        // Flash an LED to indicate a sampling event
```

```
        led.setRGB(255, 255, 255);
```

```
        led.setOn();
```

```
        Utils.sleep(50);
```

```
        led.setOff();
```

```
        // Package the time and sensor reading into a radio datagram and send it.
```

```
        dg.reset();
```

```
        dg.writeLong(now);
```

```
        dg.writeInt(reading);
```

```
        rCon.send(dg);
```

```
        System.out.println("Light value = " + reading);
```

```
        // Go to sleep to conserve battery
```

```
        Utils.sleep(SAMPLE_PERIOD - (System.currentTimeMillis() - now));
```

```
    } catch (Exception e) {
```

```

        System.err.println("Caught " + e + " while collecting/sending sensor sample.");
    }
}

protected void pauseApp() {
    // This will never be called by the Squawk VM
}

protected void destroyApp(boolean arg0) throws MIDletStateChangeException {
    // Only called if startApp throws any exception other than MIDletStateChangeException
}
}

```

## **Κώδικας που υπάρχει στο Arduino**

```

/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:

```

```

void setup() {
    // initialize the digital pin as an output.
    pinMode(led, OUTPUT);
    Serial.begin(115200);
}

// the loop routine runs over and over again forever:
void loop() {
    if (Serial.available() > 0){
        char value = Serial.read();
        if (value == 'c'){
            digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
        }else{ // wait for a second
            digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
            // wait for a second
        }
    }
}

```

## **Κώδικας της βάσης δεδομένων**

```

CREATE TABLE IF NOT EXISTS `new_table` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `node` varchar(45) DEFAULT NULL,
  `value` int(45) DEFAULT NULL,
  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=687 ;

```