

Ατομική Διπλωματική Εργασία

**FIREWATCH: ΑΝΙΧΝΕΥΣΗ ΚΑΙ ΠΡΟΒΛΕΨΗ ΔΑΣΙΚΩΝ
ΠΥΡΚΑΓΙΩΝ ΜΕ ΤΗ ΧΡΗΣΗ ΑΣΥΡΜΑΤΟΥ ΔΙΚΤΥΟΥ
ΑΙΣΘΗΤΗΡΩΝ ΚΑΙ ΓΕΩΓΡΑΦΙΚΩΝ ΔΕΔΟΜΕΝΩΝ**

Γιώργος Κωνσταντίνου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2012

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**FireWatch: Ανίχνευση και Πρόβλεψη Δασικών Πυρκαγιών με τη χρήση
ασύρματου δικτύου αισθητήρων και γεωγραφικών δεδομένων**

Γιώργος Κωνσταντίνου

Επιβλέπων Καθηγητής
Γιώργος Σαμάρας

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του
Πανεπιστημίου Κύπρου

Μάιος 2012

Ευχαριστίες

Καταρχάς θα ήθελα να ευχαριστήσω τον καθηγητή Γιώργο Σαμάρα, για την ευκαιρία που μου έδωσε να ασχοληθώ με μια πολύ ενδιαφέρον διπλωματική εργασία κάτω από την επίβλεψή του.

Θα ήθελα επίσης να ευχαριστήσω όλους όσους δουλεύουν στο Εργαστήριο Κινητού και Ασύρματου Υπολογισμού. Ειδικότερα θέλω να ευχαριστήσω τον μεταδιδακτορικό φοιτητή του τμήματος μας Παναγιώτη Ανδρέου για την συνεχής και καρποφόρα συνεργασία και επίβλεψη, τον διδακτορικό φοιτητή Μάριο Μπέλκ για την βοήθεια σε σχέση με τις τεχνολογίες διαδικτύου που χρησιμοποιήθηκαν κατά την υλοποίηση της διπλωματικής εργασίας και τέλος, τον συμφοιτητή μου Μάριο Κωνσταντινίδη για την συνεργασία που είχαμε κατά της εκπόνηση των διπλωματικών μας εργασιών.

Περίληψη

Η έγκαιρη ανίχνευση πυρκαγιών μπορεί να σώσει όχι μόνο τον φυσικό πλούτο μιας χώρας αλλά και ζωές. Η ανίχνευση και πρόβλεψη πιθανής πυρκαγιάς σε μηδαμινό χρόνο είναι αναγκαία για την έγκαιρη λήψη δραστικών μέτρων με σκοπό την καταστολή τους. Η μελέτη μας έδειξε ότι τα υπάρχοντα συστήματα, αν και αρκετά αποτελεσματικά, δεν είναι τόσο αποδοτικά αφού χρειάζονται κάποιο χρόνο μέχρι να ανιχνεύσουν την ύπαρξη πυρκαγιάς. Στα πλαίσια της παρούσας διπλωματικής θα παρουσιαστεί το καινοτόμο σύστημα FireWatch¹ το οποίο στοχεύει να ξεπεράσει τα μειονεκτήματα που παρουσιάζουν οι σημερινές τεχνολογίες ανίχνευσης πυρκαγιών, χρησιμοποιώντας ένα κράμα από τεχνολογίες όπως τα Ασύρματα Δίκτυα Αισθητήρων, την αυτοματοποιημένη Συνεργατική Εργασία μεταξύ των εμπλεκόμενων με την βοήθεια υπολογιστή, ανάλυση και ψηφιακή μοντελοποίηση εδάφους και Σύστημα Γεωγραφικών Πληροφοριών.

Με τη χρήση Ασύρματου Δικτύου Αισθητήρων η ανίχνευση πυρκαγιών γίνεται πιο ακριβής εφόσον περιβαλλοντικές μετρήσεις (θερμοκρασία, υγρασία κτλ) θα συλλέγονται σε πραγματικό χρόνο και θα αποθηκεύονται σε μια εξειδικευμένη βάση δεδομένων για μετέπειτα, υψηλού επιπέδου ανάλυση και εξαγωγή συμπερασμάτων. Οι Ασύρματοι Αισθητήρες θα τοποθετηθούν στο δάσος με τέτοιο τρόπο που θα καλύπτουν ένα μεγάλο μέρος της περιοχής με αυξημένο κίνδυνο πυρκαγιών, σχηματίζοντας ένα Ασύρματο Δίκτυο. Επίσης γεωγραφικά δεδομένα της ελεγχόμενης περιοχής αποθηκεύονται σε μια γεωγραφική βάση δεδομένων. Το Σύστημα Συνεργασίας αλληλεπιδρά με τις βάσεις δεδομένων, αναλύει τα δεδομένα και παρουσιάζει στους χρήστες μέσω ενός web application τις συνθήκες που υπάρχουν στο δάσος. Επίσης σε περίπτωση ανίχνευσης πυρκαγιάς το Σύστημα Συνεργασίας προειδοποιεί τις αρμόδιες αρχές ώστε να λάβουν τα απαραίτητα μέτρα.

Με την χρήση του FireWatch η ανίχνευση και πρόβλεψη δασικών πυρκαγιών γίνεται σε πραγματικό χρόνο. Οι αρμόδιες αρχές θα γνωρίζουν πιο γρήγορα αν τυχόν πάει κάτι λάθος, αυξάνοντας την ταχύτητα που παίρνουν δράση. Η διαφύλαξη του περιβάλλοντος (χλωρίδα και πανίδα) θα είναι γεγονός αφού το σύστημα θα μειώσει την ζημιά από τις πυρκαγιές. Τέλος, το σύστημα FireWatch είναι χρηματοδοτημένο

¹ Το Σύστημα FireWatch, <http://firewatch.cs.ucy.ac.cy/>

από το Ίδρυμα Προώθησης Έρευνας και πρόκειται να τεθεί σε εφαρμογή στο Τμήμα Δασών στην Κύπρο.

Περιεχόμενα

Κεφάλαιο 1	1
Εισαγωγή	1
1.1 Ορισμός προβλήματος	1
1.2 Αντιμετώπιση προβλήματος	3
1.3 Σκοπός και Στόχοι εργασίας	4
Κεφάλαιο 2	6
Ασύρματοι Αισθητήρες	6
2.1 Συσκευές ασύρματων αισθητήρων	6
2.1.1 Αρχιτεκτονική και δυνατότητες	7
2.2 Ασύρματο δίκτυο αισθητήρων.....	9
2.2.1 Σχεδίαση Ασύρματων Δικτύων Αισθητήρων	10
2.2.2 Κάλυψη και εγκατάσταση ασύρματων δικτύων αισθητήρων.....	13
2.2.3 Στρατηγικές εγκατάστασης ασύρματων δικτύων αισθητήρων	13
Κεφάλαιο 3	14
Σχετική Εργασία	14
3.1 Εισαγωγή	14
3.2 Αρχικές μέθοδοι ανίχνευσης πυρκαγιάς	14
3.3 Συστήματα ανίχνευσης πυρκαγιών με τη χρήση δορυφόρων.....	15
3.4 Συστήματα ανίχνευσης πυρκαγιών με τη χρήση καμερών.....	17
3.5 Συστήματα ανίχνευσης πυρκαγιών με τη χρήση ασύρματου δικτύου αισθητήρων	19
3.6 Σύγκριση συστημάτων.....	21
Κεφάλαιο 4	23
Απαιτήσεις Συστήματος	23
4.1 Εισαγωγή	23
4.2 Απαιτήσεις συστήματος FireWatch.....	24
4.2.1 Σενάρια χρήσης	24
4.3 Απαιτήσεις επιμέρους συστημάτων.....	29
4.3.1 Απαιτήσεις ασύρματου δικτύου αισθητήρων	29
4.3.2 Απαιτήσεις συστήματος γεωγραφικών δεδομένων.....	31
4.3.3 Απαιτήσεις συστήματος συνεργασίας.....	33
Κεφάλαιο 5	35
Αρχιτεκτονική Συστήματος.....	35
5.1 Γενική αρχιτεκτονική.....	35
5.1.1 Επίπεδο-1: Back-End	36
5.1.2 Επίπεδο-2: Front-End.....	37

Κεφάλαιο 6	39
Σχεδίαση Συστήματος	39
6.1 Εισαγωγή	39
6.2 Σχεδίαση ασύρματου δικτύου αισθητήρων	40
6.2.1 Συστατικά ασύρματου δικτύου αισθητήρων.....	40
6.2.2 Τοπολογία δικτύου	49
6.2.3 Πλαίσιο λογισμικού για το ασύρματο δίκτυο αισθητήρων.....	52
6.2.4 Δομή πακέτου αισθητήρων	58
6.2.5 Μετασχηματισμός ακατέργαστων μετρήσεων αισθητήρων σε παγκόσμιες μονάδες μέτρησης	61
6.2.6 Βάση Δεδομένων Ασύρματου Δικτύου Αισθητήρων	67
6.3 Σχεδίαση συστήματος γεωγραφικών δεδομένων.....	70
6.3.1 Λειτουργίες γεωγραφικού συστήματος δεδομένων	70
6.3.2 Επιμέρους συστατικά γεωγραφικού συστήματος δεδομένων.....	71
6.4 Σχεδίαση συστήματος συνεργασίας.....	71
6.4.1 Βάση δεδομένων συστήματος συνεργασίας	72
6.4.2 Συγχρονισμός βάσης δεδομένων αισθητήρων με βάση δεδομένων συστήματος συνεργασίας.....	78
6.4.3 Διαπροσωπεία FireWatch	79
6.4.4 Υπηρεσία επικοινωνίας	80
6.4.5 Παραγωγή ροής εργασιών.....	83
Κεφάλαιο 7	84
Υλοποίηση Συστήματος	84
7.1 Εισαγωγή	84
7.2 Τεχνικά χαρακτηριστικά υπολογιστή ανάπτυξης του FireWatch.....	85
7.3 Υλοποίηση συστήματος ασύρματου δικτύου αισθητήρων.....	85
7.3.1 Εργαλεία ανάπτυξης.....	86
7.3.2 Εκκίνηση/Τερματισμός καταγραφής μετρήσεων από το δίκτυο αισθητήρων	87
7.3.3 Προγραμματισμός κόμβων αισθητήρων	88
7.3.4 Υλοποίηση λειτουργιών βάσης δεδομένων ασύρματου δικτύου αισθητήρων	88
7.4 Υλοποίηση συστήματος γεωγραφικών δεδομένων	90
7.4.1 Εργαλεία ανάπτυξης.....	90
7.4.2 Κατασκευή χάρτη.....	90
7.4.3 Πρόσθεση επιπέδων στον χάρτη	90
7.4.4 Πρόσθεση σημείων ενδιαφέροντος στον χάρτη.....	91
7.5 Υλοποίηση συστήματος συνεργασίας	91
7.5.1 Εργαλεία ανάπτυξης.....	91
7.5.2 Υλοποίηση λειτουργιών βάσης δεδομένων συστήματος συνεργασίας.....	93
7.5.3 Αλγόριθμος ανίχνευσης φωτιάς	95
7.5.4 Μεθοδολογία ανάπτυξης και παρουσίαση εφαρμογής	101
Κεφάλαιο 8	108

Συμπεράσματα.....	108
8.1 Ανασκόπηση συστήματος.....	108
8.2 Μελλοντική εργασία.....	109
Βιβλιογραφία	111
Παράρτημα Α	113
Παράρτημα Β	1
Παράρτημα Γ	1
Παράρτημα Δ	1
Παράρτημα Ε	1

Σχήματα

Εικόνα 1: FireWatch-Υψηλού Επιπέδου Αρχιτεκτονική.....	4
Εικόνα 2: Παραδείγματα Ασύρματων συσκευών αισθητήρων	7
Εικόνα 3: Αρχιτεκτονική Συσκευής Ασύρματου Αισθητήρα	8
Εικόνα 4: Wireless Sensor Network.....	10
Εικόνα 5: FIRE LOOKOUT TOWER (LA-California)	15
Εικόνα 6: Osborne Fire Finder	15
Εικόνα 7: Φωτιές στην κεντρική Αφρική (Aqua).....	16
Εικόνα 8: Κάμερα FIRE WATCH (Γερμανία)	18
Εικόνα 9: The IR Radar (Αυστραλία)	18
Εικόνα 10: FireBug εγκατεστημένο σε φυτό	20
Εικόνα 11: Libelium interface	21
Εικόνα 12: Αρχιτεκτονική Συστήματος	35
Εικόνα 13: Κύρια συστατικά Ασύρματου Δικτύου Αισθητήρων.....	40
Εικόνα 14: Επεξεργαστής IRIS M2110	41
Εικόνα 15: Connector (με άσπρο χρώμα) του επεξεργαστή IRIS M2110 με μπαταριοθήκη	42
Εικόνα 16: Διάγραμμα Chip XM2110CA	43
Εικόνα 17: MTS400 Environmental Sensor Board	45
Εικόνα 18: Άνω όψη MIB520CB.....	48
Εικόνα 19: Βασικός σταθμός (Base Station)	48
Εικόνα 20: Τοπολογία Star.....	49
Εικόνα 21: Τοπολογία Mesh	50
Εικόνα 22: Τοπολογία Star Mesh.....	52
Εικόνα 23: Software framework για το ασύρματο δίκτυο αισθητήρων.....	52
Εικόνα 24: Διαπροσωπεία MoteView.....	57
Εικόνα 25: Διαπροσωπεία MoteConfig	58
Εικόνα 26: Πίνακες επαφών	74
Εικόνα 26: Πίνακες membership	76
Εικόνα 26: Συστατικά Υπηρεσίας Επικοινωνίας	81
Εικόνα 27: Linked Servers	92
Εικόνα 28: Microsoft Visual Studio 2010 Screenshot	93
Εικόνα 29: Σχηματική Αναπαράσταση Αλγόριθμου Ανίχνευσης Φωτιάς FWI	95
Εικόνα 30: Screenshot Εισαγωγής στο Σύστημα FireWatch	102
Εικόνα 31: Screenshot Dashboard	102
Εικόνα 32: Legend.....	103
Εικόνα 33: Screenshot WSN Monitoring – WSN Data	103
Εικόνα 34: Screenshot WSN Monitoring – WSN Chart	104
Εικόνα 35: Screenshot Enhanced G.I.S.	105
Εικόνα 36: Screenshot Collaboration-Search.....	105
Εικόνα 37: Screenshot Collaboration-Send Notification.....	106
Εικόνα 38: Screenshot Collaboration-Add New.....	106
Εικόνα 39: Screenshot Account Settings-Edit Profile.....	107

Κεφάλαιο 1

Εισαγωγή

1.1	Ορισμός προβλήματος	1
1.2	Αντιμετώπιση προβλήματος	3
1.3	Σκοπός και Στόχοι εργασίας	4

1.1 Ορισμός προβλήματος

Οι δασικές πυρκαγιές αποτελούν την κύρια απειλή και ένα από τα σημαντικότερα προβλήματα για τα δάση της Κύπρου και γενικότερα της Ευρώπης. Κάθε καλοκαίρι, οι πυρκαγιές προκαλούν τεράστιες καταστροφές στα δασικά οικοσυστήματα αφήνοντας στο πέρασμά τους στάχτες και αποκαΐδια.

Τα ευρωπαϊκά δάση έχουν γίνει ένας ευαίσθητος πόρος, ο οποίος έχει επηρεαστεί σοβαρά από τις πυρκαγιές που φθάνουν σε μεγαλύτερες κλίμακες με το πέρασμα των χρόνων. Συγκεκριμένα, ο Ευρωπαϊκός Οργανισμός Περιβάλλοντος (ΕΟΠ) αναφέρει ότι ο αριθμός και οι επιπτώσεις των καταστροφών έχουν αυξηθεί στην Ευρώπη κατά την περίοδο 1998-2009. Κατά την περίοδο που καλύπτει η έκθεση, οι καταστροφές προκάλεσαν περίπου 100.000 θανάτους, έχουν πληγεί περισσότερο από 11 εκατομμύρια άνθρωποι και οδήγησε σε οικονομικές απώλειες περίπου 150 δισ. ευρώ. Πυρκαγιές μεγάλης κλίμακας αποτελούν επαναλαμβανόμενα φαινόμενα σε όλη την ΕΕ, με αποτέλεσμα θανάτους, σημαντικές οικονομικές απώλειες και μακροχρόνιες επιπτώσεις στις κοινωνίες. Για παράδειγμα, στην Ισπανία το 2009, πάνω από 15.000 δασικές πυρκαγιές κόστισαν περισσότερο από 160 εκατ. ευρώ και επηρεάστηκαν 110.000 εκτάρια, εκ των οποίων πάνω από το 50% της απώλειας προκλήθηκε από 34 μεγάλες πυρκαγιές. Στην Ελλάδα το 2007, 84 άνθρωποι σκοτώθηκαν από 3.000 πυρκαγιές που κατέστρεψαν 2.700 km² των καλλιεργήσιμων εκτάσεων γης και πάνω από 1.000 κτίρια. Ένα από τα βασικά αποτελέσματα της έκθεσης αυτής είναι ότι παρόλο που ορισμένες πολιτικές της ΕΕ έχουν ήδη εγκριθεί ή ξεκινήσει, απαιτούνται

περαιτέρω προσπάθειες για την υλοποίηση μιας προσέγγισης Ολοκληρωμένης Διαχείρισης Κινδύνου (IRM) που περιλαμβάνει την πρόληψη, ετοιμότητα, αντιμετώπιση και αποκατάσταση για όλους τους κινδύνους στην Ευρώπη.

Υπάρχει ένα ευρύ φάσμα παραγόντων που συμβάλει στην αύξηση του κινδύνου δασικών πυρκαγιών όπως είναι οι ψηλές θερμοκρασίες και η παρατεταμένη ξηρασία, οι δυνατοί άνεμοι, οι μεγάλες κλίσεις των δασικών εδαφών και η εύφλεκτη ξηροφυτική βλάστηση [1]. Ένας άλλος επίσης σημαντικός παράγοντας είναι και η συσσώρευση βιομάζας, λόγω της εγκατάλειψης πολλών αγροτικών περιοχών. Ο αυξανόμενος τουρισμός και η μαζική έξοδος των κατοίκων των πόλεων για αναψυχή στα δάση, συμβάλουν επίσης σημαντικά στην αύξηση του κινδύνου πρόκλησης πυρκαγιών.

Οι επιπτώσεις των δασικών πυρκαγιών είναι τεράστιες, αφού επηρεάζουν όχι μόνο το δάσος, αλλά επίσης την οικονομία, την κοινωνία και τον άνθρωπο. Καταστρέφονται σπίτια και άλλες περιουσίες, υποβαθμίζεται το περιβάλλον, διαταράσσεται η ισορροπία του οικοσυστήματος, διαβρώνεται το έδαφος, δημιουργούνται χείμαρροι και πλημμύρες, επηρεάζεται το μικροκλίμα, καταστρέφεται η χλωρίδα και η πανίδα του τόπου και χάνεται η φυσική ομορφιά.

Όπως αρχικά αναφέραμε, τα συστήματα που χρησιμοποιούνται στο παρόν στάδιο αδυνατούν να προσφέρουν ακριβής πρόβλεψη για πυρκαγιές. Επίσης πριν την ανίχνευση μιας πυρκαγιάς χρειάζεται χρόνος ώστε να είναι ορατή. Τότε όμως, είναι ήδη αργά. Η ανάγκη για μια καινοτόμα εφαρμογή παρμένη από τον χώρο της Πληροφορικής και γενικότερα της τεχνολογίας είναι αναγκαία για την έγκαιρη πρόβλεψη και ανίχνευση πυρκαγιών. Σκοπός μας είναι να μειώσουμε στο ελάχιστο τις ζημιές μιας επικείμενης πυρκαγιάς.

Αναφορικά, οι διεθνείς εκδηλώσεις (π.χ., Παγκόσμιο Κύπελλο της FIFA) συνήθως προσελκύουν εκατομμύρια συμμετέχοντες μέσα σε πολύ μικρό χρονικό διάστημα. Η ανάπτυξη έξυπνων δικτύων αισθητήρων (π.χ. αισθητήρες, ενεργοποιητές, RFID) σε κτίρια μπορεί να συμβάλει στη βελτίωση της εμπειρίας του επισκέπτη, παρέχοντας τα μέσα για την εύκολη αλληλεπίδραση με το περιβάλλον. Για παράδειγμα, κατά τη διάρκεια αυτών των εφήμερων μεγάλων γεγονότων, αισθητήρες μέτρησης της ευημερίας (π.χ., ήχου, εγγύτητα) μπορούν να σχηματίσουν λογικές ομάδες, προκειμένου να κτιστεί μια ένωση των πόρων που παρέχει σε πραγματικό χρόνο ένα χάρτη αφίξεων επισκεπτών στα διάφορα περίπτερα και θέσεις και να προτείνει στους επισκέπτες μιαν ιδανική περιήγηση, έτσι ώστε να μεγιστοποιήσουν την εμπειρία και την

ικανοποίηση τους. Επιπλέον, αν συμβεί μια κατάσταση κρίσης, αυτή η ένωση των πόρων μπορεί επίσης να βοηθήσει τους ανθρώπους να εντοπίσουν και να συντονίσει τις αρχές για τη διάσωση. Επιπλέον, οι εφαρμογές μπορούν να χρησιμοποιηθούν σε συνδυασμό με το δίκτυα έξυπνων κινητών συσκευών, προκειμένου να δημιουργήσει ευκαιριακά κοινωνικά δίκτυα που σχηματίζονται αυθόρμητα σύμφωνα με τις σχέσεις, οι οποίες είναι σαφείς (π.χ., τη φιλία, την ιδιοκτησία) και / ή έμμεσες (π.χ., θέση, την ενέργεια, τις δυνατότητες).

Τα τελευταία χρόνια τα Ασύρματα Δίκτυα Αισθητήρων χρησιμοποιούνται ευρέως για πολλών ειδών εφαρμογές. Οι μικρές αυτές συσκευές έχουν αναπτυχθεί τόσο πολύ τεχνολογικά που παρέχουν δυνατότητες να καταγράφουν μετρήσεις όπως θερμοκρασία, υγρασία, πίεση, επίπεδα φωτισμού κτλ, σε πραγματικό χρόνο και να επικοινωνούν αποτελεσματικά μεταξύ τους ώστε να συγκεντρώνουν τις μετρήσεις αυτές σε κεντρικά σημεία.

Το σύστημα του FireWatch στηρίζεται κυρίως στο ασύρματο δίκτυο αισθητήρων, που με την αποδοτική του χρήση θα παρέχει ένα τεχνολογικό εργαλείο στους υπαλλήλους του Τμήματος Δασών.

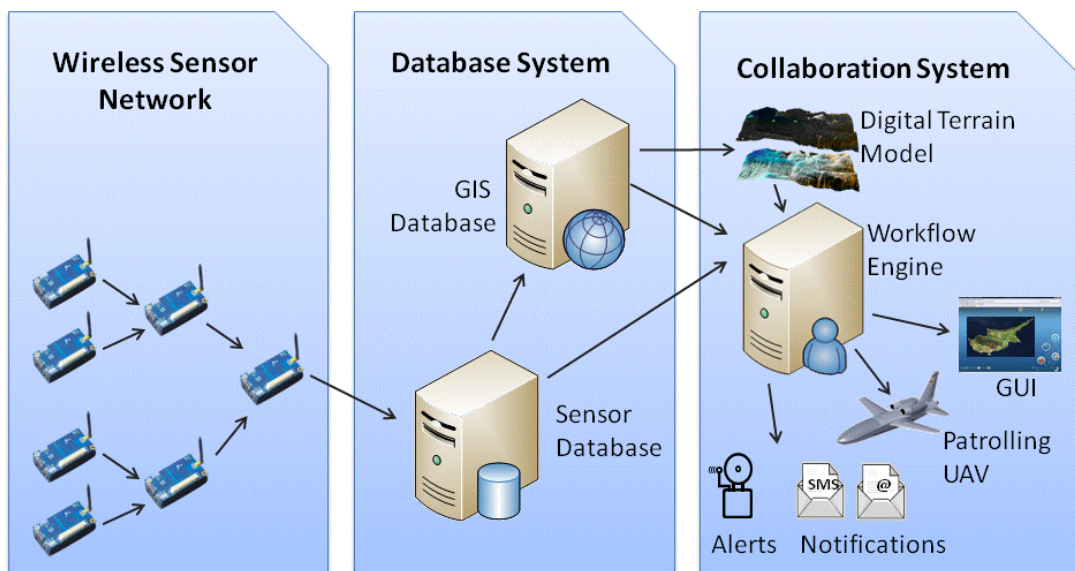
1.2 Αντιμετώπιση προβλήματος

Με βάση το πιο πάνω πρόβλημα που υπάρχει στα δάση της Ευρώπης και ειδικότερα της Κύπρου, αποφασίστηκε να αναπτυχθεί μια εφαρμογή που θα μπορούσε αποδοτικά να προβλέπει και να αντιμετωπίζει τις δασικές πυρκαγιές κάνοντας χρήση ασύρματων αισθητήρων.

Οι αισθητήρες θα εξαπλώνονται στο δάσος υπό παρακολούθηση, ώστε να λαμβάνουν σε πραγματικό χρόνο περιβαλλοντικές μετρήσεις. Τα δεδομένα αυτά, μέσω ενός πρωτοκόλλου δρομολόγησης, θα συλλέγονται σε μια εξειδικευμένη βάση δεδομένων.

Λαμβάνοντας επιπρόσθετες πληροφορίες από βάση γεωγραφικών δεδομένων, τα δεδομένα των αισθητήρων επεξεργάζονται, αναλύονται και παρουσιάζονται σε μορφή που οι χρήστες μπορούν να κατανοήσουν. Με την εξαγωγή πληροφορίας υψηλού επιπέδου θα μπορεί να γίνει πιο εύκολα η επίβλεψη και ο έλεγχος των δασικών περιοχών, παρέχοντας τις κατάλληλες προειδοποιήσεις και μηνύματα στους χρήστες μέσω μιας διαδικτυακής εφαρμογής. Ένα αρχικό σχέδιο του συστήματος FireWatch φαίνεται στην Εικόνα 1. Πολλοί κόμβοι-αισθητήρες θα τοποθετούνται έξυπνα σε μια

προκαθορισμένη, μικρού μεγέθους, δασική περιοχή σχηματίζοντας έτσι ένα ασύρματο δίκτυο αισθητήρων. Επίσης θα συλλεχθούν γεωγραφικά δεδομένα για την επιλεγμένη δασική περιοχή και θα αποθηκεύονται σε μια γεωγραφική βάση δεδομένων. Οι ασύρματοι κόμβοι αισθητήρων θα καταγράφουν συνεχώς περιβαλλοντικά φαινόμενα και οι μετρήσεις θα μεταδίδονται σε μια βάση δεδομένων αποκλειστικά προσαρμοσμένη για το ασύρματο δίκτυο αισθητήρων, όπου και θα αποθηκεύονται. Το σύστημα συνεργασίας θα αλληλεπιδρά με τις δύο βάσεις δεδομένων και θα δίνει τη δυνατότητα στους χρήστες να επιβλέπουν την κατάσταση που επικρατεί στο περιβάλλον για την επιλεγμένη περιοχή μέσω μιας διαδικτυακής εφαρμογής. Σε περίπτωση ανίχνευσης πιθανού κινδύνου το σύστημα θα ενημερώνει αμέσως τους εμπλεκόμενους φορείς. Επίσης σε περιπτώσεις ανίχνευσης ή πρόβλεψης θα παρέχονται στους χρήστες ασφαλή και αποδοτικά μονοπάτια για να προσεγγίσουν την περιοχή.



Εικόνα 1: FireWatch-Υψηλού Επιπέδου Αρχιτεκτονική

1.3 Σκοπός και Στόχοι εργασίας

Το FireWatch έχει ως γενικό σκοπό να δημιουργήσει ένα καινοτόμο, ολοκληρωμένο σύστημα ανίχνευσης και πρόβλεψης δασικών πυρκαγιών σε πραγματικό χρόνο με την χρήση Ασύρματου Δικτύου Αισθητήρων. Η χρήση γεωγραφικών δεδομένων και των δεδομένων ανάλυσης του εδάφους θα προσφέρουν στους χρήστες καλύτερες επιλογές για την δράση που θα ληφθεί σε περίπτωση ανίχνευση πυρκαγιάς. Επιπλέον οι αρχές θα γνωρίζουν ποιες περιοχές είναι πιο ριψοκίνδυνες για πυρκαγιές,

καθώς και πως μπορεί μελλοντικά να αναπτυχθεί η πυρκαγιά με βάση την κατεύθυνση του ανέμου.

Με την χρήση του FireWatch θα ενδυναμωθεί η παρούσα υποδομή του τμήματος δασών και θα βοηθήσει στη διατήρηση του περιβάλλοντος.

Οι τεχνολογικοί στόχοι της εργασίας είναι:

- Αναγνώριση μεθοδολογίας τοποθέτησης των αισθητήρων και αλγορίθμων δρομολόγησης για την αποδοτική κάλυψη της δασικής περιοχής. Αυτό θα επιτευχθεί με την βοήθεια των γεωγραφικών δεδομένων.
- Χρησιμοποίηση αλγορίθμων ανίχνευσης και πρόβλεψης πυρκαγιών για την ανάπτυξη ενός καινοτόμου συστήματος επιτήρησης.
- Ανάλυση Εδάφους και ψηφιακή μοντελοποίηση των δασικών περιοχών της Κύπρου.
- Σχεδίαση και ανάπτυξη συστήματος πλοήγησης εκτός οικιστικών δρόμων (χωματόδρομοι) που θα βοηθούν τους πυροσβέστες να βρίσκουν τα βέλτιστα μονοπάτια ώστε να προσεγγίσουν αποτελεσματικά την πυρκαγιά.
- Σχεδίαση και ανάπτυξη ενός συστήματος που είναι πανταχού παρόν για την ανίχνευση/πρόβλεψη των δασικών πυρκαγιών. Δηλαδή επιτρέπει σε πραγματικό χρόνο τη επίβλεψη των δασών μειώνοντας τις ζημιές από πυρκαγιές, το κόστος συντήρησης ενώ παράλληλα θα διατηρείται το περιβάλλον και η άγρια ζωή.
- Παροχή μηχανισμών αυτόματης ενημέρωσης εμπλεκομένων και συντονισμός της προσπάθειας για κατάσβεση της πυρκαγιάς.

Κεφάλαιο 2

Ασύρματοι Αισθητήρες

2.1	Συσκευές ασύρματων αισθητήρων	6
2.1.1	Αρχιτεκτονική και δυνατότητες	7
2.2	Ασύρματο δίκτυο αισθητήρων.....	9
2.2.1	Σχεδίαση Ασύρματων Δικτύων Αισθητήρων	10
2.2.2	Κάλυψη και εγκατάσταση ασύρματων δικτύων αισθητήρων.....	13
2.2.3	Στρατηγικές εγκατάστασης ασύρματων δικτύων αισθητήρων	13

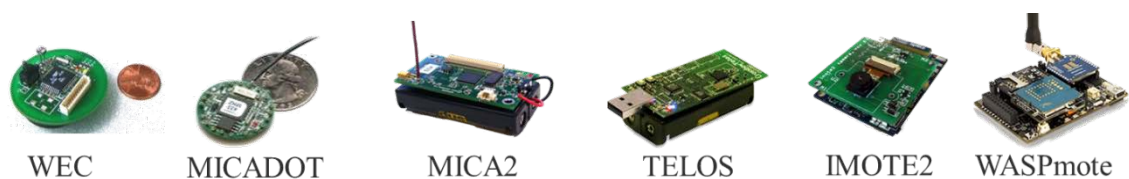
2.1 Συσκευές ασύρματων αισθητήρων

Οι συσκευές Ασύρματων Αισθητήρων αποτελούν μικρές συσκευές με μεγάλες δυνατότητες που μπορούν να χρησιμοποιηθούν για την ανάπτυξη συστημάτων επιτήρησης του περιβάλλοντος. Αυτό έγινε εφικτό με την ραγδαία ανάπτυξη των ενσωματωμένων επεξεργαστών, των συστατικών για τους αισθητήρες και η ανάπτυξη ασύρματης επικοινωνίας με μικρό κόστος επικοινωνίας.

Κάθε αισθητήρας είναι μια συσκευή, η οποία απαρτίζεται από:

- συστατικό επικοινωνίας (radio transceiver), η οποία του δίνει τη δυνατότητα να επικοινωνεί με τους υπόλοιπους αισθητήρες.
- πηγή ισχύος (μπαταρίες AA συνήθως), η οποία του δίνει την απαραίτητη ενέργεια για να λειτουργεί.
- μικροεπεξεργαστή (microcontroller), ο οποίος είναι υπεύθυνος για τη γενική λειτουργία του κόμβου-αισθητήρα.
- μονάδα μνήμης για την αποθήκευση δεδομένων (δυναμική και στατική),
- έναν ή περισσότερους αισθητήρες,
- Analog-to-digital converters (μετατροπείς ADCs).

Στην Εικόνα 2 παρουσιάζονται μερικά παραδείγματα ασύρματων συσκευών αισθητήρων που αναπτύχθηκαν την τελευταία δεκαετία όπως ο αισθητήρας WEC το 1998, ο MICADOT το 2000, ο MICA2 το 2002, ο TELOS το 2004, ο IMOTE2 το 2008 και ο WASPmote το 2010. Τα κύρια πλεονεκτήματα αυτών των συσκευών είναι ότι δεν κοστίζουν ακριβά, κυμαίνονται μεταξύ 10\$ και 200\$, έχουν χαμηλή κατανάλωση και υποστηρίζουν προγραμματισμό εξ αέρος. Δηλαδή μπορούμε να εγκαταστήσουμε νέες εφαρμογές πάνω στους αισθητήρες αφού έχει τεθεί σε εφαρμογή το ασύρματο δίκτυο. Στη συνέχεια αυτής της ενότητας θα δούμε μερικές από τις δυνατότητες και των συσκευών αυτών.



Εικόνα 2: Παραδείγματα Ασύρματων συσκευών αισθητήρων

2.1.1 Αρχιτεκτονική και δυνατότητες

Η αρχιτεκτονική των ασύρματων αισθητήρων μπορεί να χωριστεί σε 2 επίπεδα όπως φαίνεται στην Εικόνα 3. Το ασύρματο μέρος αποτελείται από τα συστατικά για επεξεργασία, αποθήκευση και επικοινωνία, ενώ η πλακέτα αισθητήρων αποτελείται από ένα σύνολο μικροαισθητήρων που καταγράφουν διάφορες μετρήσεις.



Εικόνα 3: Αρχιτεκτονική Συσκευής Ασύρματου Αισθητήρα

Πιο συγκεκριμένα το κάτω επίπεδο (ασύρματο μέρος), είναι υπεύθυνο για την επεξεργασία και την επικοινωνία. Τροφοδοτείται συνήθως από δύο μπαταρίες AA και διαθέτει μνήμη για αποθήκευση δεδομένων και μετρήσεων.

Το πάνω επίπεδο αποτελείται από ένα σύνολο αισθητήρων που είναι υπεύθυνα για την καταγραφή μετρήσεων όπως θερμοκρασία, υγρασία, φως κτλ. Οι μετρήσεις αυτές είναι αναλογικές και μετατρέπονται σε ψηφιακές μέσω ενός analog-to-digital converter.

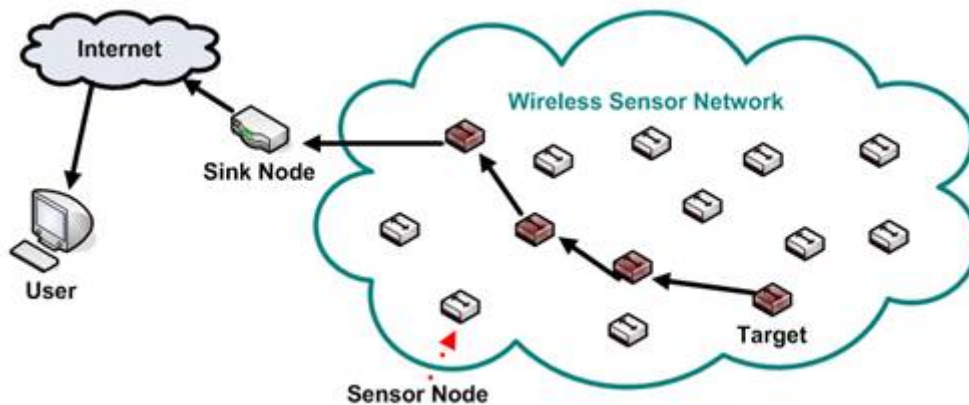
Η ασύρματη επικοινωνία μεταξύ των αισθητήρων είναι κάτι που χρειάζεται προσοχή αφού είναι ο κύριος παράγοντας για την εξάντληση της μπαταρίας. Διαφορετικά πρωτόκολλα επικοινωνίας (όπως IEEE 802.11, WLAN, IEEE 802.15, ZigBee 802.15.4) προσφέρουν διαφορετική κατανάλωση ενέργειας αλλά και διαφορετική ακτίνα επικοινωνίας. Κάθε αισθητήρας έχει πιθανές καταστάσεις επικοινωνίας: μετάδοση (transmit mode), λήψη (receive mode), αδρανής κατάσταση (idle mode) και κατάσταση ύπνου (sleep mode). Η κατάσταση μετάδοσης είναι αυτή που έχει και την πιο μεγάλη κατανάλωση ενώ η κατάσταση ύπνου την μικρότερη. Οι περισσότεροι αισθητήρες σήμερα το 95% του χρόνου τους είναι σε sleep mode.

Λειτουργικά συστήματα που τρέχουν σε ασύρματους αισθητήρες υπάρχουν πολλά. Κύριο στόχο έχουν όλα την επεκτασιμότητα, την ετερογένεια και την ελάχιστη δυνατή κατανάλωση ενέργειας για την λειτουργία του αισθητήρα ώστε να μεγιστοποιείται η διάρκεια ζωής του. Πολλές φορές η αντικατάσταση μπαταριών στους αισθητήρες δεν αποτελεί επιλογή αφού μπορεί να έχουν εγκατασταθεί σε δύσβατες ή απροσέγγιστες περιοχές. Επομένως η μέγιστη διάρκεια ζωής είναι το σημαντικότερο μέτρο σύγκρισης που υπολογίζεται κατά την σχεδίαση ενός λειτουργικού συστήματος που προορίζεται για αισθητήρες.

Ένα ευρέως open source, γνωστό και χρησιμοποιούμενο λειτουργικό σύστημα για ασύρματους αισθητήρες είναι το TinyOS [2] που αναπτύχθηκε για πρώτη φορά στο University of California-Berkeley και από τότε έχουν υλοποιηθεί πολλές εκδόσεις του. Στηρίζεται σε ασύγχρονα γεγονότα (event based), για την ελαχιστοποίηση του κόστους λειτουργίας. Διασφαλίζει την εύχρηστη πρόσβαση στους πόρους της ασύρματης συσκευής και υποστηρίζει την ταυτόχρονη πρόσβαση σε αυτούς με την ελάχιστη χρήση hardware.

2.2 Ασύρματο δίκτυο αισθητήρων

Η ικανότητα των Συσκευών Ασύρματων Αισθητήρων να επικοινωνούν μεταξύ τους ασύρματα επιτρέπει το σχηματισμό των ad hoc Ασύρματων Δικτύων Αισθητήρων (ΑΔΑ) που μπορούν να εγκατασταθούν για την παρακολούθηση φυσικών φαινομένων σε περιβάλλοντα με διάφορες συνθήκες. Οι αισθητήρες αναδιοργανώνονται με τη χρήση κάποιου αλγορίθμου σε ασύρματα δίκτυα και τα δεδομένα τα οποία λαμβάνονται από αυτούς αναμεταδίδονται σε γειτονικούς αισθητήρες μέχρι να φτάσουν στον επιθυμητό προορισμό, όπως φαίνεται στην Εικόνα 4. Η χρήση των Ασύρματα Δίκτυα Αισθητήρων έχει ήδη αναπτυχθεί σε διάφορες εφαρμογές όπως η περιβαλλοντική παρακολούθηση, σε βιομηχανικούς αυτοματισμούς, για παρακολουθήσεις κτιρίων και υγείας, διαχείριση κυκλοφοριακής συμφόρησης κτλ. Σε αυτές τις εφαρμογές, οι χρήστες μπορούν να θέσουν επερωτήματα προς το δίκτυο και να λάβει μετρήσεις των αισθητήρων σε πραγματικό χρόνο.



Εικόνα 4: Wireless Sensor Network

2.2.1 Σχεδίαση Ασύρματων Δικτύων Αισθητήρων

Κατά τον σχεδιασμό ενός ασύρματου δικτύου αισθητήρων κύριος στόχος είναι η μεγιστοποίηση της διάρκειας ζωής του δικτύου ώστε να ελαχιστοποιείται το κόστος συντήρησης που πολλές φορές είναι αδύνατη να επιτευχθεί σε δύσβατα περιβάλλοντα. Αυτό οφείλεται στο γεγονός ότι οι ασύρματοι αισθητήρες είναι εφοδιασμένοι με μπαταρίες τύπου AA που σε κάποια φάση θα εξαντληθούν. Προκειμένου να παραταθεί η διάρκεια ζωής του δικτύου, μεγάλο μέρος έρευνας έχει αφιερωθεί στο σχεδιασμό και την υλοποίηση ενεργειακά αποδοτικών συστημάτων υλικού, όπως η χρήση χαμηλής συχνότητας επεξεργαστές, χαμηλής ισχύος μονάδες επικοινωνίας και χαμηλής κατανάλωσης εξαρτήματα αισθητήρα. Για την αποτελεσματική χρήση του υλικού αυτού και για την περαιτέρω μείωση της κατανάλωσης ενέργειας χρειάζονται επίσης ενεργειακά αποδοτικούς αλγόριθμους και εφαρμογές ώστε να παραταθεί η συνολική διάρκεια ζωής.

Πιο κάτω δίνονται τα κύρια χαρακτηριστικά ενός Ασύρματου Δικτύου Αισθητήρων.

Διαχείριση περιορισμένης ενέργειας και πόρων:

Τα ασύρματα δίκτυα αισθητήρων, περιορισμένα σε ενέργεια και πόρων όπως η CPU και η μνήμη, μπορούν να αναπτυχθούν σε σκληρό και δύσβατο περιβάλλον. Σε περιπτώσεις όπου η φυσική επαφή για αντικατάσταση ή συντήρηση τους είναι αδύνατη, το ασύρματα μέσο, είναι ο μόνος τρόπος για απομακρυσμένη πρόσβαση. Ως εκ τούτου, το ενδιάμεσο λογισμικό (middleware) θα πρέπει να παρέχει μηχανισμούς για την αποτελεσματική χρήση του επεξεργαστή και της μνήμης, ενώ να επιτρέπει χαμηλότερη

κατανάλωση ενέργειας για την επικοινωνία. Ένας κόμβος αισθητήρα θα πρέπει να μπορεί να πραγματοποιεί τις τρεις βασικές λειτουργίες του - αίσθηση, επεξεργασία δεδομένων, επικοινωνία - χωρίς να εξαντλεί τους πόρους. Σε ενεργειακά αποδοτικά middleware, για παράδειγμα, τα περισσότερα μέρη της συσκευής μπορούν να απενεργοποιηθούν τις περισσότερες φορές ανάλογα με την εφαρμογή αποθηκεύοντας ενέργεια.

Επεκτασιμότητα, κινητικότητα, και δυναμική τοπολογία του δικτύου:

Η επεκτασιμότητα ορίζεται ως εξής: αν μια εφαρμογή μεγαλώνει, το δίκτυο θα πρέπει να είναι αρκετά ευέλικτο ώστε να επιτρέπει αυτή την ανάπτυξη, οποτεδήποτε και οπουδήποτε, χωρίς να επηρεάζεται η απόδοση του δικτύου. Αποτελεσματικές υπηρεσίες ενδιάμεσου λογισμικού πρέπει να είναι σε θέση να διατηρήσει αποδεκτά το επίπεδο επίδοσης, καθώς το δίκτυο μεγαλώνει. Η τοπολογία του δικτύου υπόκειται σε συχνές αλλαγές εξαιτίας παραγόντων όπως η δυσλειτουργία, βλάβη της συσκευής, κινούμενα εμπόδια, την κινητικότητα, και παρεμβολές. Το ενδιάμεσο λογισμικό θα πρέπει να υποστηρίζει την αξιόπιστη λειτουργία των αισθητήρων που αποτελούν το δίκτυο, παρά τις δυναμικές αλλαγές με την αναπροσαρμογή του περιβάλλοντος του δικτύου. Επίσης το ενδιάμεσο λογισμικό, θα πρέπει να υποστηρίζει μηχανισμούς για ανοχή λαθών και αυτορρύθμιση/αυτοσυντήρηση των αισθητήρων.

Ετερογένεια:

Το ενδιάμεσο λογισμικό θα πρέπει να παρέχει χαμηλού επιπέδου προγραμματιστικά μοντέλα για να ανταποκριθεί στην μεγάλη πρόκληση της γεφύρωσης του χάσματος μεταξύ των δυνατοτήτων της τεχνολογίας του υλικού και τις απαραίτητες γενικές δραστηριότητες, όπως η αναδιάρθρωση, η εκτέλεση και η επικοινωνία. Θα πρέπει να θεσπίσει μηχανισμούς του συστήματος που διασυνδέονται με τους διάφορους τύπους του υλικού και των δικτύων, που υποστηρίζονται μόνο από κατανεμημένες, βασικές αφαιρετικότητες του λειτουργικού συστήματος.

Δυναμική οργάνωση δικτύου:

Σε αντίθεση με τα παραδοσιακά δίκτυα, τα δίκτυα ασύρματων αισθητήρων πρέπει να διαχειριστούν με πόρους που είναι δυναμικοί, όπως η ενέργεια, το εύρος ζώνης, και την επεξεργαστική ισχύ. Επίσης τα δίκτυα αισθητήρων πρέπει να

υποστηρίζουν εφαρμογές που τρέχουν για μεγάλο χρονικό διάστημα, έτσι τα πρωτόκολλα δρομολόγησης πρέπει να είναι σχεδιασμένα έτσι ώστε να μπορέσει το δίκτυο να λειτουργεί για όσο πιο μεγάλο χρονικό διάστημα είναι δυνατό. Επειδή η γνώση του δικτύου είναι απαραίτητη για να λειτουργήσει σωστά, το ενδιάμεσο λογισμικό θα πρέπει να παρέχει ad hoc ανακάλυψη πόρων. Ένας κόμβος αισθητήρας θα πρέπει να γνωρίζει τη θέση του στο δίκτυο και σε όλη την τοπολογία του δικτύου. Σε ορισμένες περιπτώσεις, η τοποθέτηση κάθε κόμβου μέσω GPS είναι αδύνατη, ανέφικτη, ή ακριβή. Σημαντικά θέματα παραμέτρων του συστήματος, όπως το μέγεθος και η πυκνότητα του δικτύου ανά τετραγωνικό μίλι, επηρεάζουν το συμβιβασμό μεταξύ της καθυστέρησης, της αξιοπιστίας και της ενέργειας.

Ενσωμάτωση στον πραγματικό κόσμο:

Οι περισσότερες εφαρμογές ασύρματων δικτύων αισθητήρων είναι φαινόμενα πραγματικού χρόνου, όπου ο χρόνος και ο χώρος είναι εξαιρετικά σημαντικά. Ως εκ τούτου, το ενδιάμεσο λογισμικό θα πρέπει να παρέχει σε πραγματικό χρόνο τις υπηρεσίες να προσαρμόζονται στις αλλαγές και να παρέχει συνεπή δεδομένα.

Συνάθροιση δεδομένων:

Οι περισσότερες εφαρμογές ασύρματων δικτύων αισθητήρων περιλαμβάνουν κόμβους που περιέχουν περιττά στοιχεία και βρίσκονται σε συγκεκριμένη τοπική περιοχή. Αυτά τα γνωρίσματα άνοιξαν τη δυνατότητα για τη συνάθροιση των δεδομένων εντός δικτύου από διαφορετικές πηγές, την εξάλειψη των επιπλέον δεδομένων και μείωση του αριθμού των μεταδόσεων προς το βασικό σταθμό (base station). Η συγκέντρωση αυτή εξοικονομεί σημαντική ενέργεια και πόρους, δεδομένου ότι το κόστος επικοινωνίας είναι πολύ υψηλότερο από το κόστος υπολογισμού.

Ασφάλεια:

Τα ΑΔΑ έχουν ευρέως αναπτυχθεί σε τομείς που περιλαμβάνουν ευαίσθητες πληροφορίες, για παράδειγμα, υγειονομική περίθαλψη και διάσωση. Η μεγάλη ανάπτυξη των ΑΔΑ σε αντίξοα περιβάλλοντα αυξάνει την έκθεσή τους σε κακόβουλες εισβολές και επιθέσεις, όπως denial of service. Επιπλέον, το ασύρματο μέσο διευκολύνει υποκλοπές και παραλλαγή του data πακέτου ώστε να θέσει σε κίνδυνο την λειτουργία του δικτύου. Όλοι αυτοί οι παράγοντες καθιστούν εξαιρετικά σημαντική την

ασφάλεια. Επιπλέον, οι κόμβοι έχουν πόρους με περιορισμένη ισχύ και δυνατότητες επεξεργασίας, έτσι σταθεροί μηχανισμοί ασφάλειας, οι οποίοι έχουν μεγάλη κατανάλωση πόρων, κρίνονται ακατάλληλοι. Οι προκλήσεις αυτές αυξάνουν την ανάγκη για την ανάπτυξη ολοκληρωμένων και ασφαλών λύσεων που συμβάλλουν στην επίτευξη προστασίας, διατηρώντας παράλληλα την επιθυμητή απόδοση του δικτύου.

2.2.2 Κάλυψη και εγκατάσταση ασύρματων δικτύων αισθητήρων

Για την δυνατότητα να αισθάνονται τα ΑΔΑ πιθανή πυρκαγιά θα πρέπει να υπάρχουν αισθητήρες αρκετά κοντά που μπορεί να πραγματικά να ανιχνεύουν το γεγονός. Έτσι, η περιοχή κάλυψης ενός WSN (δηλαδή, ποια σημεία της εν λόγω περιοχής είναι αρκετά κοντά στους αισθητήρες, έτσι ώστε ένα γεγονός που πραγματοποιείται σε αυτό το σημείο να μπορεί να γίνει αισθητό) και η εγκατάσταση των αισθητήρων στον χώρο αυτό (δηλαδή, ποιος είναι ο αριθμός των αισθητήρων είναι απαραίτητος και που πρέπει να τοποθετούνται) είναι δύο σημαντικές πτυχές που πρέπει να ληφθούν υπόψη.

2.2.3 Στρατηγικές εγκατάστασης ασύρματων δικτύων αισθητήρων

Οι στρατηγικές εγκατάστασης στα ΑΔΑ παίζουν σημαντικό ρόλο στην επίλυση του προβλήματος κάλυψης του χώρου. Για παράδειγμα, ο επιλεγμένος αριθμός και οι θέσεις των κόμβων αισθητήρων επηρεάζουν την απόδοση της παρακολούθησης και εντοπισμού. Επιπλέον, διάφορες παράμετροι πρέπει να λαμβάνονται υπόψη συμπεριλαμβανομένου της φυσικής γεω-τοποθεσίας των κόμβων αισθητήρων, τη γειτονιά τους, τη δομή του δικτύου, την ισχύ εκπομπής, κλπ.

Κεφάλαιο 3

Σχετική Εργασία

3.1	Εισαγωγή	14
3.2	Αρχικές μέθοδοι ανίχνευσης πυρκαγιάς	14
3.3	Συστήματα ανίχνευσης πυρκαγιών με τη χρήση δορυφόρων.....	15
3.4	Συστήματα ανίχνευσης πυρκαγιών με τη χρήση καμερών.....	17
3.5	Συστήματα ανίχνευσης πυρκαγιών με τη χρήση ασύρματου δικτύου αισθητήρων	19
3.6	Σύγκριση συστημάτων.....	21

3.1 Εισαγωγή

Στο παρόν στάδιο θα μελετηθεί η ανίχνευση και πρόβλεψη δασικών πυρκαγιών με τη χρήση διαφόρων συστημάτων. Συγκεκριμένα θα μελετήσουμε τις αρχικές μεθόδους ανίχνευσης πυρκαγιών, τα συστήματα που χρησιμοποιούν δορυφόρους, κάμερες και τέλος αυτά που χρησιμοποιούν ασύρματους αισθητήρες. Επίσης θα δούμε τα πλεονεκτήματα και τα μειονεκτήματα κάθε προσέγγισης προβάλλοντας μια ποιοτική σύγκριση μεταξύ τους.

3.2 Αρχικές μέθοδοι ανίχνευσης πυρκαγιάς

Από τις αρχές του 20ου αιώνα οι άνθρωποι χρησιμοποιούσαν πύργους επιτήρησης όπως φαίνεται στην Εικόνα 5 για την ανίχνευση δασών. Εγκατεστημένα σε ψηλά σημεία, για μέγιστη κάλυψη πεδίου, οι πύργοι ήταν εφοδιασμένοι συνήθως με 1-2 άτομα για την επίβλεψη του δάσους. Η έγκαιρη σήμανση συναγερμού στηριζόταν στην καλή λειτουργία των ματιών των παρατηρητών οι οποίοι αργότερα χρησιμοποιούσαν εργαλεία όπως κιάλια.



Εικόνα 5: FIRE LOOKOUT TOWER (LA-California)

Σε περίπτωση ανίχνευσης χρησιμοποιούσαν μια συσκευή εύρεσης της πιθανής τοποθεσίας της φωτιάς που έβλεπαν με το όνομα Osborne Fire Finder (Εικόνα 6) ώστε να ενημερώσουν. Σιγά σιγά η πιο πάνω μέθοδος εγκαταλείφθηκε με την ανάπτυξη της τεχνολογίας και νέοι μέθοδοι αναπτύχθηκαν με σκοπό την πιο γρήγορη ανίχνευση πυρκαγιών.



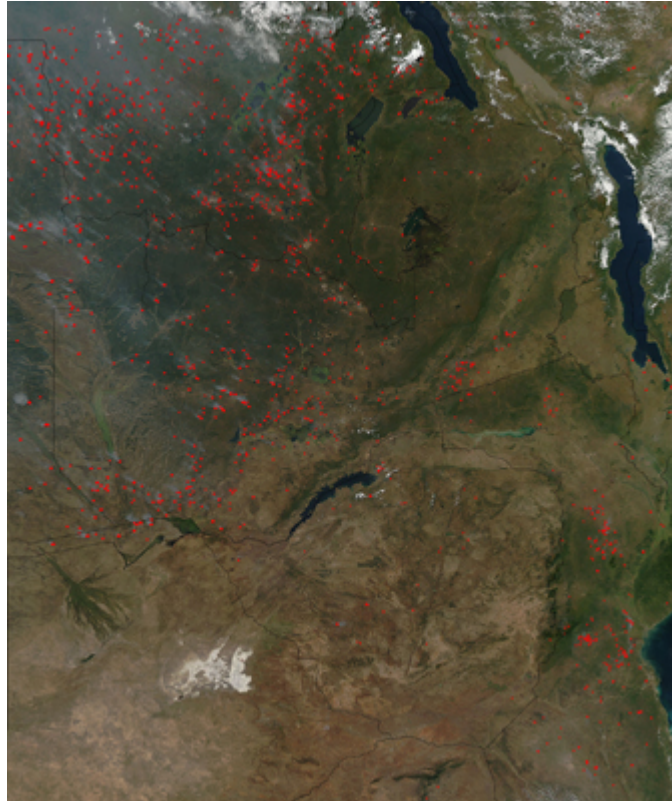
Εικόνα 6: Osborne Fire Finder

Παρόλο που η συγκεκριμένη μέθοδος ανίχνευσης ήταν χαμηλή σε κόστος και οι υπάλληλοι που επάνδρωναν το πύργο κάλυπταν μια σχετικά μεγάλη ακτίνα, μειονεκτεί ως προς την ακρίβεια και την καθυστέρηση στην ανίχνευσης πυρκαγιάς αφού στηρίζεται αποκλειστικά στον ανθρώπινο παράγοντα.

3.3 Συστήματα ανίχνευσης πυρκαγιών με τη χρήση δορυφόρων

Ένα παράδειγμα χρήσης δορυφόρων για την ανίχνευση πυρκαγιών είναι το MODerate resolution Imaging Spectroradiometer (MODIS) [3] που εγκαταστάθηκε από την NASA σε δύο από τους δορυφόρους της, τον Terra και Aqua αντίστοιχα. Η τροχιά του Terra γύρω από τη Γη έχει προγραμματιστεί έτσι ώστε να περνά από το βορρά προς νότο διασχίζοντας τον ισημερινό το πρωί, ενώ το Aqua περνά νότια προς βόρεια πάνω

από τον ισημερινό το απόγευμα. Το σύστημα MODIS παρακολουθεί τις συνεχείς αλλαγές στην ατμόσφαιρα και μπορεί να προβλέπει φυσικές καταστροφές όπως πυρκαγιές (Εικόνα 7) και να παρέχει βοήθεια λήψης αποφάσεων για την προστασία του περιβάλλοντος.



Εικόνα 7: Φωτιές στην κεντρική Αφρική (Aqua)

Παρομοίως με το MODIS, το σύστημα Forest Fire Management and Fire Prevention System (FOMFIS) χρησιμοποιεί δορυφόρους. Ο κύριος στόχος του FOMFIS ήταν η ενσωμάτωση των υφιστάμενων τεχνολογιών κάτω από το ίδιο περιβάλλον δεδομένων και κοινής διεπιφάνειας χρήστη για να παράγουν ένα ολοκληρωμένο υπολογιστικό σύστημα που βασίζεται σε ημι-αυτόματη επεξεργασία δορυφορικών εικόνων (χάρτες καυσίμου), κοινωνικο-οικονομική μοντελοποίηση των κινδύνων και πιθανοτικά μοντέλα που θα χρησιμεύσει ως χρήσιμο εργαλείο για το σχεδιασμό πρόληψης πυρκαγιών των δασών και τη διαχείριση τους. Το σύστημα FOMFIS παρέχει με τη χρήση των πιο πάνω μοντέλων, τον πιθανό κίνδυνο πυρκαγιάς για μια δεδομένη περιοχή.

Τα πιο πάνω συστήματα προσφέρουν μεγάλη κάλυψη αφού έχουν τη δυνατότητα να επιτηρούν τεράστιες περιοχές. Επίσης παρέχουν αυτόματους

μηχανισμούς προειδοποίησης και έχουν την ικανότητα να προβλέπουν δασικές πυρκαγιές. Όμως υποφέρουν από μερικά μειονεκτήματα. Συγκεκριμένα, η καταγραφή των περιβαλλοντικών φαινομένων δεν γινόταν σε πραγματικό χρόνο αλλά παρατηρείται κάποια καθυστέρηση λόγω της τροχιάς των δορυφόρων. Επομένως πιθανή ανίχνευση πυρκαγιών ήταν εκτός των χρονικών περιθωρίων που απαιτούμε και είναι αδύνατη η επίβλεψη συγκεκριμένων δασικών περιοχών σε 24/7. Επίσης η χρήση δορυφόρων δεν είναι το ότι καλύτερο αφού ενδιάμεσα μπορούσε να παρεμβάλλονται καιρικά φαινόμενα (βροχές, σύννεφα, ομίχλη κτλ) που να επηρεάζαν αρνητικά την ακρίβεια του συστήματος.

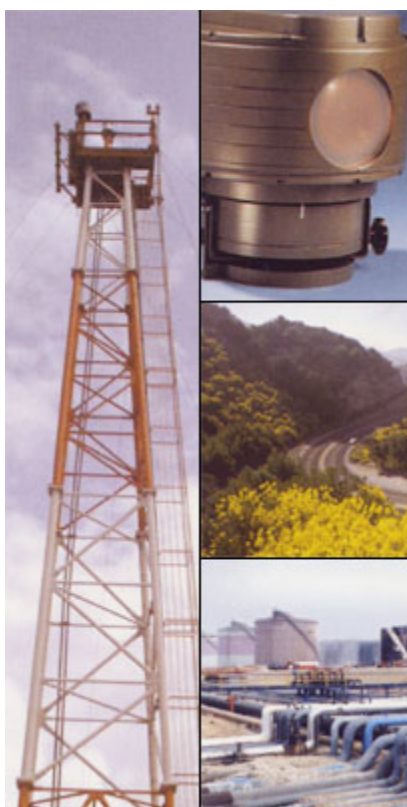
3.4 Συστήματα ανίχνευσης πυρκαγιών με τη χρήση καμερών

Οι κάμερες παρακολούθησης δασών είναι ευρέως γνωστές στην Ευρώπη. Οι κάμερες αυτές τοποθετούνται σε ανυψωμένα σημεία για την αυτόματη ανίχνευση πυρκαγιών και την απομακρυσμένη επιτήρησή τους. Σε περίπτωση ανίχνευσης διαθέτουν μηχανισμούς προειδοποίησης των αρμόδιων αρχών ώστε να λάβουν μέτρα δράσης. Παραδείγματα τέτοιων συστημάτων είναι το FIRE WATCH [5] της Εικόνα 8 στη Γερμανία και το The IR Radar [6] της Εικόνα 9 στην Αυστραλία. Το FIRE WATCH [5] χρησιμοποιεί Optical Sensor Systems (OOS), που είναι εγκατεστημένοι σε πύργους και έχουν τη δυνατότητα να ανιχνεύουν σύννεφα καπνού από μεγάλες αποστάσεις (10km). Κάθε OSS έχει τη δυνατότητα να καλύπτει περιοχή μέχρι και 70.000 ha. Όμως, κάθε OSS περιστρέφεται 360° κάθε 4 με 10 λεπτά σε βήματα των 10°, που σημαίνει ότι παίρνει κάποιο χρόνο μέχρι να ανιχνευτεί πιθανή φωτιά.. Συστήματα υπερύθρων σε εφαρμογές ασφαλείας οδήγησαν τους κατασκευαστές να φτιάξουν IR Radars για να ανιχνεύουν φωτιές όπως και το The IR Radar [6] που εφαρμόστηκε στην Αυστραλία. Το IR Radar σαρώνει μια συγκεκριμένη περιοχή κάθε λίγα δευτερόλεπτα και παράγει πανοραμικές θερμικές εικόνες. Οι εικόνες σαρώνονται για να βρεθούν τυχόν παράξενα ευρήματα, όπως απροσδόκητη αύξηση της θερμοκρασίας. Παρόλο που είναι σχετικά καλή η ακρίβεια μέτρησης της θερμοκρασίας (5 βαθμοί Κελσίου ακρίβεια) επηρεάζεται από καιρικές συνθήκες.



Εικόνα 8: Κάμερα FIRE WATCH (Γερμανία)

Άλλα συστήματα όπως το [4] παρόλο που χρησιμοποιούν συστήματα για πρόβλεψη με τη χρήση διαφόρων πιθανολογικών μοντέλων, η ανίχνευση πυρκαγιών στηρίζεται σε κάμερες. Πέραν από την ανακρίβεια που έχουν οι κάμερες όπως θα δούμε πιο κάτω, το σύστημα υποφέρει από λανθασμένες σημάνσεις συναγερμού κάτω από ορισμένες περιπτώσεις όπως ηλιοβασίλεμα, σύννεφα, τροχαία κίνηση κτλ.



Εικόνα 9: The IR Radar (Αυστραλία)

Αν και η επίβλεψη δασικών περιοχών με τη χρήση καμερών είναι πιο ελεγχόμενη σε σύγκριση με τους δορυφόρους λόγω της συνεχόμενης σάρωσης και της

ανθρώπινης επίβλεψης, εντούτοις παρουσιάζει κάποια μειονεκτήματα. Παρομοίως με τα συστήματα ανίχνευσης πυρκαγιών με δορυφόρους, τα συστήματα με κάμερες υποφέρουν από την καθυστέρηση στην ανίχνευση πυρκαγιών, καθώς παίρνουν κάποιο χρόνο πριν ανιχνευτούν. Επίσης την νύχτα όπου ο ορατότητα είναι περιορισμένη και σε κακές καιρικές συνθήκες δεν μπορούν να δώσουν μεγάλη ακρίβεια στην ανίχνευση πυρκαγιών. Επίσης η λειτουργία των καμερών προϋποθέτει εργοδότηση εξειδικευμένου προσωπικού που θα επιβλέπουν την λειτουργία τους. Το κόστος για την εργοδότηση και εκπαίδευση ατόμων κατάλληλων για την επίβλεψη των καμερών είναι ακόμα ένα μειονέκτημα. Ακόμη τέτοια συστήματα δεν έχουν την δυνατότητα πρόβλεψη πυρκαγιών.

3.5 Συστήματα ανίχνευσης πυρκαγιών με τη χρήση ασύρματου δικτύου αισθητήρων

Πιο πάνω περιγράψαμε 2 μεθόδους ανίχνευσης/πρόβλεψης δασικών πυρκαγιών που χρησιμοποιούνται σήμερα, με κάμερες και δορυφόρους. Όπως είδαμε και τα δύο υποφέρουν από κάποια μειονεκτήματα. Επιπλέον και τα 2 είναι επιρρεπής στον ανθρώπινο παράγοντα που μπορεί να προκαλέσει περισσότερη καθυστέρηση κατά την λήψη αποφάσεων.

Αυτή τη στιγμή τα κύρια συστήματα που χρησιμοποιούνται για την ανίχνευση και πρόβλεψη δασικών πυρκαγιών δεν χρησιμοποιούν τεχνολογίες ασύρματων δικτύων αισθητήρων. Παρόλα αυτά εντοπίσαμε μερικά και τα παραθέτουμε πιο κάτω.

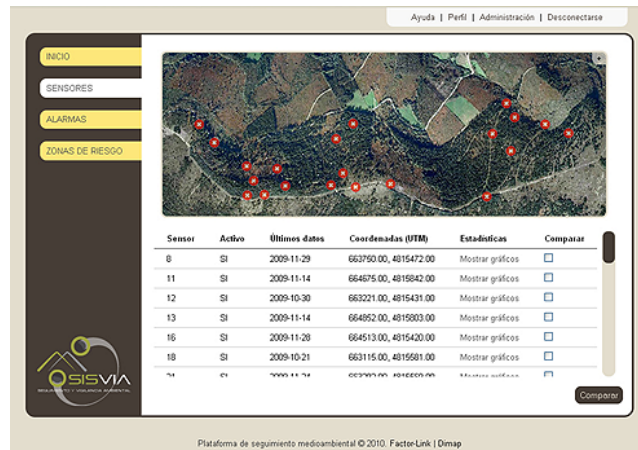
Ένα από τα λιγοστά που υπάρχουν είναι το FireBug project [3] που αναπτύχθηκε από το Πανεπιστήμιο της Καλιφόρνιας Berkeley. Στόχος του συγκεκριμένου project είναι η συλλογή δεδομένων σε πραγματικό χρόνο από τις πυρκαγιές που επιτρέπει την προγνωστική ανάλυση της εξέλιξης της συμπεριφοράς πυρκαγιάς. Για να συλλέγουν τα δεδομένα αυτά χρησιμοποιούν αισθητήρες που είναι εξαπλωμένοι στο δάσος όπως φαίνεται στην Εικόνα 10. Τα FireBugs είναι μικροί ασύρματοι κόμβοι αισθητήρων, που τρέχουν το TinyOS λειτουργικό σύστημα, και είναι εγκατεστημένοι στο δάσος για να συλλέγουν περιβαλλοντικές μετρήσεις. Μπορούν να μετρούν υγρασία, θερμοκρασία, την γεω-τοποθεσία τους, την φωτεινότητα και διαθέτουν επιταχυνσιόμετρο. Έχουν την ικανότητα να αυτοδιοργανώνονται σε ένα δίκτυο ασύρματων κόμβων αισθητήρων. Συγκεκριμένα το σύστημα αποτελείται από το επίπεδο του δικτύου αισθητήρων, το επίπεδο ελέγχου και επεξεργασίας των δεδομένων και ένα επίπεδο αλληλεπίδρασης και επικοινωνίας με το ασύρματο δίκτυο αισθητήρων.

Για την αυτό-διοργάνωση και δρομολόγηση του δικτύου χρησιμοποιούν το πρωτόκολλο MH6 ενώ μέσω του προγράμματος Surge παρουσιάζονται για την τοπολογία του δικτύου. Για την επιτήρηση του συστήματος χρησιμοποιείται μια διαδικτυακή εφαρμογή απ' όπου οι χρήστες μπορούν να παρακολουθούν τις τελευταίες μετρήσεις του δικτύου.



Εικόνα 10: FireBug εγκατεστημένο σε φυτό

Ένα πιο εμπορικό σύστημα που χρησιμοποιεί ασύρματους αισθητήρες είναι το Libelium [4]. Η συγκεκριμένη εφαρμογή χρησιμοποιεί τους αισθητήρες Waspnotes για τη συλλογή δεδομένων σε πραγματικό χρόνο. Ο στόχος του είναι να παρέχει στους διάφορους οργανισμούς μια περιβαλλοντική υποδομή παρακολούθησης, με δυνατότητες διαχείρισης συναγερμού και να παραδίνει στις αρχές συναγερμούς προειδοποίησης. Η επιτήρηση του δικτύου γινόταν μέσω μιας διαδικτυακής εφαρμογής όπως φαίνεται στην Εικόνα 11, όπου οι χρήστες μπορούν να δουν την τοποθεσία και τις μετρήσεις των αισθητήρων. Αυτές οι προειδοποιήσεις προκαλούνταν όταν οι μετρήσεις από τους αισθητήρες ξεπερνούσαν ένα προκαθορισμένο threshold. Κάποια επιπλέον χαρακτηριστικά των αισθητήρων είναι ότι χρησιμοποιούν Solar Panels για μεγάλη διάρκεια ζωής και αυτονομία, ενώ έχουν δυνατότητες κάλυψης μέχρι 40KM. Ακόμη διαθέτουν μηχανισμούς χρήσης GRPS για αποστολή δεδομένων σε κρίσιμες περιπτώσεις.



Εικόνα 11: Libelium interface

Τέτοια συστήματα επιτρέπουν την επιτήρηση των δασικών περιοχών με μεγάλη ακρίβεια και σε πραγματικό χρόνο. Παρόλα αυτά είναι επιρρεπής σε βλάβες, ενώ έχουν περιορισμένη ζωή αφού ακόμα και αυτά που διαθέτουν Solar Panels χρειάζονται να είναι εγκατεστημένα σε περιοχές ηλιόλουστες, κάτι που δεν είναι πάντα εφικτό. Επίσης η κάλυψη εξαρτάται από τον αριθμό των αισθητήρων. Αν και συνήθως οι απλοί αισθητήρες δεν κοστίζουν ακριβά, κάποιοι άλλοι με επιπρόσθετα χαρακτηριστικά όπως οι Waspnotes μπορούν να αυξήσουν το κόστος της εφαρμογής σε μεγάλο βαθμό.

3.6 Σύγκριση συστημάτων

Ο πιο κάτω πίνακας παρουσιάζει συνοπτικά μια ποιοτική σύγκριση μεταξύ των διαφόρων συστημάτων ανίχνευσης πυρκαγιών.

Systems	Accuracy	Human Factor	Real-time detection	G.I.S.	Alert Mechanisms	Collaboration Mechanisms	Fire Prediction	Area Covered
	Other Systems							
Fire Lookout Towers	Low	Y	Low	N	Y	N	N	Medium
	Surveillance Systems							
FIRE WATCH (Germany)	Medium	Y	Low-Medium	N	Y	N	N	15 km
The IR Radar	Medium	Y	Low-Medium	N	Y	N	N	Medium
	Satellite Based Systems							
MODIS	Medium	Y	Low	Y	Y	N	Y	Huge
FOMFIS	Low	Y	Low	Y	Y	N	Y	Large
	Sensor-based Systems							
Fire Bug Project	High	Y	High	N	N	N	N	Depends on sensor usage
Libelium	High	N	High	N	Y	N	N	Depends on sensor usage
FireWatch	High	N	High	Y	Y	Y	Y	Depends on sensor usage

Πίνακας 1: Πίνακας ποιοτικής σύγκρισης μεταξύ των συστημάτων ανίχνευσης φωτιάς

Κεφάλαιο 4

Απαιτήσεις Συστήματος

4.1	Εισαγωγή	23
4.2	Απαιτήσεις συστήματος FireWatch.....	24
4.2.1	Σενάρια χρήσης	24
4.3	Απαιτήσεις επιμέρους συστημάτων.....	29
4.3.1	Απαιτήσεις ασύρματου δικτύου αισθητήρων	29
4.3.2	Απαιτήσεις συστήματος γεωγραφικών δεδομένων.....	31
4.3.3	Απαιτήσεις συστήματος συνεργασίας.....	33

4.1 Εισαγωγή

Έχοντας μια γενική ιδέα του συστήματος ήρθε η ώρα να ορίσουμε τις γενικές απαιτήσεις που διέπουν το σύστημα αλλά και τα επιμέρους υποσυστήματα.

Αρχικά θα ακολουθήσει μια ανάλυση των γενικών απαιτήσεων που αφορούν το σύστημα ως προς την ολότητα του σε σχέση με την αξιοπιστία, την επεκτασιμότητα και την ευχρηστία του.

Ακολούθως θα δούμε τις απαιτήσεις από κάθε σύστημα ξεχωριστά, που θα συνεισφέρουν στις γενικές απαιτήσεις. Συγκεκριμένα θα αναλυθούν οι απαιτήσεις για το ασύρματο δίκτυο αισθητήρων, το σύστημα γεωγραφικών δεδομένων και του συστήματος συνεργασίας.

Αναλύοντας τις απαιτήσεις του συστήματος θα δοθούν οι βάσεις πάνω στις οποίες θα κτιστεί η υλοποίηση του FireWatch. Επιπλέον απαιτήσεις που έχουν εμφανιστεί κατά την διάρκεια της υλοποίησης θα εμφανίζονται σε αυτό το κεφάλαιο.

4.2 Απαιτήσεις συστήματος FireWatch

Συνολικά το τελικό προϊόν του FireWatch θα πρέπει να πληρεί κάποιες γενικές απαιτήσεις. Κύριος στόχος αυτών των απαιτήσεων είναι να διασφαλίσουν την υψηλή ποιότητα λογισμικού για κάθε ένα από τα επιμέρους κομμάτια της αρχιτεκτονικής.

Όλα τα υποσυστήματα θα πρέπει να λειτουργούν με τέτοιο τρόπο ώστε να προσφέρουν απόδοση, ορθότητα, ευρωστία και αξιοπιστία.

Για την εξαγωγή των απαιτήσεων του συστήματος του FireWatch έχουν αναλυθεί τα βασικά σενάρια χρήσης του συστήματος. Τα σενάρια καθορίζουν τη συμπεριφορά του χρήστη μέσω βημάτων ή πράξεις που πραγματοποιούνται για την εκτέλεση των βασικών λειτουργιών του συστήματος.

4.2.1 Σενάρια χρήσης

Στην ενότητα αυτή, παρουσιάζουμε τα σενάρια χρήσης της πλατφόρμας FireWatch. Για κάθε σενάριο παρουσιάζουμε τις καθορισμένες απαιτήσεις του.

Τα σενάρια χρήσης που αναφέρονται παρακάτω:

Σενάριο ανίχνευση πυρκαγιάς και παρακολούθησης:

Η περίπτωση πυρανίχνευσης αφορά το ασύγχρονο γεγονός που μπορεί να συμβεί όταν σε κάποια χρονική στιγμή κάτι ασυνήθιστο παρατηρηθεί από ένα ή περισσότερους κόμβους. (π.χ., θερμοκρασία μεγαλύτερη από ένα threshold). Η εκδήλωση θα μεταδίδεται στη συνέχεια, στις βάσεις δεδομένων και στο σύστημα συνεργασίας για την περαιτέρω επεξεργασία.

Οι απαιτήσεις του πιο πάνω σεναρίου είναι:

- Αποδοτικότητα χρόνου: Τα συμβάντα θα πρέπει να ανιχνεύονται σε πραγματικό χρόνο.
- Αξιοπιστία: Τα δεδομένα των συμβάντων θα πρέπει να είναι ακριβή. Το σύστημα θα πρέπει να αποτρέψει ή να αντιμετωπίσει τις αποτυχίες ή βλάβες.
- Διαθεσιμότητα: Όλα τα συστατικά FireWatch πρέπει να είναι διαθέσιμα (on-line) 365/24/7.
- Ασφάλεια: Όλα τα συμβάντα και οι ειδοποιήσεις πρέπει να αποστέλλονται μέσω ενός ασφαλούς διαύλου επικοινωνίας.

Σενάριο πρόβλεψης φωτιάς:

Σε κάποια χρονική στιγμή, όταν ένα γεγονός πρόβλεψης πυρκαγιάς ανιχνευτεί από τον αλγόριθμο πρόβλεψης θα ειδοποιούνται οι αρχές.

Οι απαιτήσεις του πιο πάνω σεναρίου είναι:

- Ακρίβεια πρόβλεψης: Η πρόβλεψη θα πρέπει να είναι όσο το δυνατόν ακριβής, ανάλογα με τα πληροφοριακά μοντέλα.
- Επικύρωση πρόβλεψης: Η πρόβλεψη θα πρέπει να επικυρωθεί με τα διαθέσιμα μέσα (UAV ή υπάλληλοι που περιπολούσαν στην απειλούμενη περιοχή).
- Αποδοτικότητα χρόνου: Τα συμβάντα θα πρέπει να ανιχνεύονται σε πραγματικό χρόνο.
- Αξιοπιστία: Τα δεδομένα των συμβάντων θα πρέπει να είναι ακριβή. Το σύστημα θα πρέπει να αποτρέψει ή να αντιμετωπίσει τις αποτυχίες ή βλάβες.
- Διαθεσιμότητα: Όλα τα συστατικά FireWatch πρέπει να είναι διαθέσιμα (on-line) 365/24/7.

Σενάριο βλάβης κόμβου αισθητήρα:

Εντός της δασικής περιοχής, μια πυρκαγιά ξεσπά και απλώνεται πάνω από την περιοχή που παρακολουθείται από κόμβους αισθητήρων. Η φωτιά φτάνει σε κόμβο αισθητήρα που προκαλεί τη θερμοκρασία του να αυξηθεί γρήγορα η οποία τελικά οδηγεί σε αποτυχία του κόμβου. Η αποτυχία του αισθητήρα αναφέρεται σε γειτονικούς κόμβους και εναλλακτικά πρωτόκολλα δρομολόγησης ανακτούν την αποτυχία του δικτύου.

Οι απαιτήσεις του πιο πάνω σεναρίου είναι:

- Αποδοτικότητα χρόνου: Βλάβες δρομολογίων και συνδέσεων πρέπει να ανακτούνται σε πραγματικό χρόνο.
- Αξιοπιστία: Αποτυχίες κόμβων αισθητήρων πρέπει να ανιχνεύονται με ακρίβεια. Το σύστημα θα πρέπει να αποτρέψει ή να αντιμετωπίσει τις αποτυχίες ή βλάβες.
- Αποδοτικότητα εγκατάστασης: Πολλαπλές διαδρομές δρομολόγησης (> 2) από έναν κόμβο αισθητήρα στο βασικό σταθμό θα πρέπει να υπάρχουν.

Σενάριο πλοήγησης υπαλλήλων προστασίας δασικών πυρκαγιών:

Σε περίπτωση ανίχνευσης ή πρόβλεψης δασικής πυρκαγιάς θα μεταδίδονται οι πληροφορίες στο σύστημα συνεργασίας για τον προσδιορισμό όλων των πιθανών διαδρομών που μπορεί να ακολουθήσουν οι υπάλληλοι ώστε να πάνε στο σημείο που τους ενδιαφέρει, ή την δυναμική προσαρμογή των μονοπατιών, σε περίπτωση που ένα μονοπάτι καταστεί απροσπέλαστο λόγω φυσικών καταστροφών.

Οι απαιτήσεις του πιο πάνω σεναρίου είναι:

- Αποδοτικότητα χρόνου: Όλες οι πιθανές διαδρομές θα πρέπει να υπολογίζονται και να παρέχονται σε πραγματικό χρόνο. Αναπροσαρμογή των μονοπατιών θα πρέπει να πραγματοποιείται σε πραγματικό χρόνο και αποτελεσματικά.
- Αξιοπιστία: Τα δεδομένα εδαφικής ανάλυσης θα πρέπει να είναι ενημερωμένα και αξιόπιστα, ώστε να παρέχονται οι πιο αποδοτικές διαδρομές.
- Υπευθυνότητα: Ο υπεύθυνος προστασίας δασών από πυρκαγιές πρέπει να είναι καλά εκπαιδευμένος να διαχειρίζονται αποτελεσματικά τις επιλογές των πιθανών διαδρομών που παρέχει το σύστημα και είναι αρμόδιος για την προτεραιότητα των δράσεων που πρέπει να ληφθούν.

Σενάριο ενημέρωσης των αρχών:

Σε κάποια χρονική στιγμή, μπορεί να γίνει πυρανίχνευση ή πρόβλεψη από τους κόμβους αισθητήρων ή από τη βάση δεδομένων της πλατφόρμας FireWatch. Ένα μήνυμα ειδοποίησης θα κατασκευαστεί με όλες τις σημαντικές πληροφορίες του συμβάντος (δηλαδή, χρόνος, τόπος, θερμοκρασία, υγρασία της περιοχής υψηλού κινδύνου πυρκαγιάς) και θα αποστέλλεται στις αντίστοιχες αρχές.

Οι απαιτήσεις του πιο πάνω σεναρίου είναι:

- Αποδοτικότητα χρόνου: Το μήνυμα πρέπει να αποστέλλεται σε πραγματικό χρόνο.
- Αξιοπιστία: Οι πληροφορίες του μηνύματος πρέπει να είναι αξιόπιστες και ακριβείς.
- Διαθεσιμότητα: Όλα τα συστατικά FireWatch πρέπει να είναι διαθέσιμα (on-line) 365/24/7.

Σενάριο συνεργασίας υπαλλήλων:

Το σενάριο συνεργασία υπαλλήλων αναφέρεται στην έγκαιρη επικοινωνία μεταξύ των εμπλεκόμενων αρχών, για μια πιθανής επικίνδυνη κατάσταση, μέσω κάποιας μορφής επικοινωνίας (π.χ. SMS, φωνητικών μηνυμάτων, ηλεκτρονικό ταχυδρομείο). Οι υπάλληλοι μπορούν να επικοινωνούν αποτελεσματικά και ανταλλάζουν μηνυμάτων, ασφαλή μονοπάτια δρόμο και πορεία δράσης.

Οι απαιτήσεις του πιο πάνω σεναρίου είναι:

- Αποδοτικότητα χρόνου: Το μήνυμα πρέπει να αποστέλλεται σε πραγματικό χρόνο.
- Αξιοπιστία: Οι πληροφορίες του μηνύματος πρέπει να είναι αξιόπιστες και ακριβείς.
- Διαθεσιμότητα: Όλα τα συστατικά FireWatch πρέπει να είναι διαθέσιμα (on-line) 365/24/7.
- Ανταπόκριση: Το σύστημα θα πρέπει να ενημερώνει το χρήστη που δημιουργεί και αποστέλλει το μήνυμα σχετικά με την κατάσταση του μηνύματος (παράδοση ή όχι).

Σενάριο διαχείρισης χρηστών

Το σενάριο διαχείρισης χρηστών χρησιμοποιείται από έναν χρήστη με δικαιώματα διαχειριστή για τη δημιουργία νέων χρηστών και για την ενημέρωση και διαγραφή των υπάρχοντων χρηστών. Τα δικαιώματα πρόσβασης χορηγούνται επίσης σε ατομικά σε χρήστη και ομάδα χρηστών. Τα δικαιώματα αυτά προσδιορίζουν τις λειτουργίες που μπορούν να εκτελεστούν μέσω της πλατφόρμας FireWatch.

Οι απαιτήσεις του πιο πάνω σεναρίου είναι:

- Αποδοτικότητα χρόνου: Οι πληροφορίες πρέπει να αποθηκεύονται σε μικρό χρονικό διάστημα.
- Αξιοπιστία: Οι αποθηκευμένες πληροφορίες θα πρέπει να είναι ακριβείς και να ελέγχονται με κανόνες.
- Διαθεσιμότητα: Όλα τα συστατικά FireWatch πρέπει να είναι διαθέσιμα (on-line) 365/24/7.

Σενάριο διαχείρισης παραμέτρων κόμβου αισθητήρα / εδάφους

Το σενάριο αυτό σχετίζεται με τον καθορισμό και την τροποποίηση των ιδιοτήτων, τα όρια και τις παράμετροι των βασικών οντοτήτων του συστήματος που εμπλέκονται στην πυρανίχνευση και παρακολούθηση, καθώς και στο σενάριο πρόβλεψης πυρκαγιάς. Στις ιδιότητες αυτές περιλαμβάνονται τα όρια ευαισθησίας, το όριο εξάπλωσης της φωτιάς κλπ. Αυτές οι αλλαγές αναμένεται να επηρεάσουν τις συνθήκες πρόκλησης του συναγερμού ανιχνεύσεις πυρκαγιάς, τις υποψίες, τις προβλέψεις και τη διαδικασία της αυτοματοποιημένης λήψης αποφάσεων και την υποστήριξη.

Οι απαιτήσεις του πιο πάνω σεναρίου είναι:

- Αποδοτικότητα χρόνου: Οι τιμές των παραμέτρων πρέπει να αποθηκεύονται σε μικρό χρονικό διάστημα.
- Αξιοπιστία: Οι τιμές των παραμέτρων θα πρέπει να είναι εντός ενός προκαθορισμένου πεδίου τιμών.
- Διαθεσιμότητα: Όλα τα συστατικά FireWatch πρέπει να είναι διαθέσιμα (on-line) 365/24/7.

Πιο κάτω παρατίθενται συνοπτικά κάποια γενικά χαρακτηριστικά που θα πρέπει να πληρεί το σύστημα:

Αξιοπιστία:

Η αξιοπιστία του συστήματος αποτελεί σημαντικότερο στοιχείο. Όλα τα επιμέρους συστήματα θα πρέπει να προσφέρουν ένα αξιόπιστο σύστημα καθώς σε αντίθετη περίπτωση θα είναι δύσκολη η διαχείριση του συστήματος και η επαναφορά σε περίπτωση λάθους.

Εφόσον το σύστημα πρόκειται να ειδοποιεί πολλαπλά άτομα, θα πρέπει να είναι 100% αξιόπιστο ως προς τις προειδοποιήσεις. Έγερση συναγερμού από λάθος θα έχει σαν αποτέλεσμα την εκμηδένιση του συστήματος από τους χρήστες.

Επομένως η σταθερή και προβλεπόμενη συμπεριφορά του συστήματος είναι κάτι που επιβάλλεται πριν τεθεί σε λειτουργία.

Επεκτασιμότητα:

Κατά την υλοποίηση του συστήματος θα πρέπει να είναι δυνατή η πρόσθεση νέων στοιχείων στην πλατφόρμα. Η αρχική ανάπτυξη του συστήματος δεν είναι και η τελική.

Νέοι προγραμματιστές θα πρέπει να είναι σε θέση να κατανοήσουν πως λειτουργεί η πλατφόρμα για να μπορούν να την επεκτείνουν και να την τελειοποιήσουν. Μέσω αυτού του εγγράφου θα είναι μια καλή αρχή.

Ευχρηστία:

Το FireWatch προορίζεται για τα μέλη του Τμήματος Δασών. Η χρήση του από τους υπαλλήλους του τμήματος θα πρέπει να είναι εύκολη.

Σε γενικές γραμμές τα διάφορα υποσυστήματα θα πρέπει να επιτρέπουν στους χρήστες να τα ρυθμίζουν και να τα χειρίζονται. Ο χειρισμός θα πρέπει να είναι όσο πιο απλός γίνεται προς όφελος της εμπειρίας του χρήστη.

4.3 Απαιτήσεις επιμέρους συστημάτων

Παρόλο που θέσαμε τις γενικές απαιτήσεις για ολόκληρο το σύστημα, κάθε υποσύστημα θέτει καινούριες απαιτήσεις λόγω της ιδιαιτερότητας και της μοναδικότητας του.

Γι' αυτό το λόγω θα θέσουμε για κάθε ένα από τα υποσυστήματα ποιες είναι οι απαιτήσεις που πρέπει να πληρούν για να προσφέρουν ένα ψηλό επίπεδο ποιότητας λογισμικού.

4.3.1 Απαιτήσεις ασύρματου δικτύου αισθητήρων

Πολλοί είναι οι παράγοντες που πρέπει να ληφθούν υπόψη για το ασύρματο δίκτυο αισθητήρων.

- Το γεγονός ότι οι ασύρματοι κόμβοι αισθητήρων έχουν περιορισμένη εμβέλεια επικοινωνίας απαιτεί την προσεκτική τοποθέτηση των αισθητήρων έτσι ώστε τουλάχιστο ένας κόμβος να είναι στην εμβέλεια επικοινωνίας μαζί με κάποιο άλλο κόμβο.
- Στο δίκτυο αισθητήρων υπάρχει προκαθορισμένη περίοδος επικοινωνίας, ενώ την υπόλοιπη διάρκεια βρίσκονται σε mode (idle ή sleep) που εξασφαλίζει μεγαλύτερη διάρκεια ζωής. Αυτή η προκαθορισμένη περίοδος επικοινωνίας

πρέπει να είναι ρυθμισμένη ώστε να επηρεάζει μόνο θετικά τη διάρκεια ζωής του δικτύου.

- Πρέπει να δίνεται η δυνατότητα να μετατρέπονται οι ακατέργαστες μετρήσεις των αισθητήρων σε κατανοητή προς τον άνθρωπο μορφή.

Ακριβείς μετρήσεις πραγματικού χρόνου:

Οι αισθητήρες θα πρέπει να είναι εφοδιασμένοι με sensor board που να καταγράφουν περιβαλλοντικές μετρήσεις. Με την συνεχή καταγραφή των περιβαλλοντικών μετρήσεων το σύστημα μας θα μπορεί να τις χρησιμοποιήσει και να εξάγει συμπεράσματα όταν υπάρχει πιθανότητα φωτιάς.

Σύμφωνα με τον αλγόριθμο που χρησιμοποιούμε για την ανίχνευση φωτιάς, οι αισθητήρες θα πρέπει να είναι σε θέση να καταγράφουν συγκεκριμένες μετρήσεις. Οι κύριες περιβαλλοντικές μετρήσεις θα πρέπει να είναι:

- 1) Θερμοκρασία
- 2) Υγρασία

Επιπλέον χρήσιμες μετρήσεις είναι:

- 3) Βαρομετρική Πίεση
- 4) Ένταση Φωτός

Οι μετρήσεις αυτές δεν πρέπει να έχουν χαμηλότερη ακρίβεια απ' ό,τι αυτή που αναγράφει ο κατασκευαστής (περιθώριο μηδαμινού λάθους). Για την διασφάλιση της κατάστασης των αισθητήρων και της ακρίβειας τους επιβάλλεται να προστατεύονται εξωτερικά με τη χρήση κάποιας προστατευτικής θήκης. Με αυτό τον τρόπο θα είναι ασφαλής από καιρικές συνθήκες καθώς και από άλλους εξωγενείς παράγοντες (βροχές, σκόνες, ζώα, ψηλές/χαμηλές θερμοκρασίες). Η διατήρηση του υλικού των αισθητήρων θα πρέπει να είναι ένα από τα βασικά μας μελήματα.

Επιπλέον, κατά την τοποθέτηση του κάθε αισθητήρα (εφόσον οι αισθητήρες δεν υποστηρίζουν GPS), θα πρέπει να καταγράφεται η ακριβής του τοποθεσία. Κάτι τέτοιο είναι εξίσου σημαντικό για να είναι γνωστή η ακριβής τοποθεσία της ύπαρξης φωτιάς. Επομένως σε περίπτωση έγερσης συναγερμού οι χρήστες θα γνωρίζουν από πιο αισθητήρα προήλθε και που είναι εγκατεστημένος ο συγκεκριμένος αισθητήρας.

Αποδοτική τοποθέτηση των ασύρματων κόμβων:

Απώτερος μας σκοπός είναι η μεγιστοποίηση της κάλυψης του χώρου από τους αισθητήρες με τέτοιο τρόπο που να επικοινωνούν μεταξύ τους, λαμβάνοντας υπόψη ότι η εμβέλεια του κάθε αισθητήρα είναι περιορισμός.

Επιπλέον, η στρατηγική τοποθέτηση των αισθητήρων θα πετύχει να διατηρήσει διαθέσιμα τα δεδομένα των αισθητήρων ακόμα και στην περίπτωση που κάποιοι από αυτούς καταστραφούν (από πυρκαγιά, καιρικά φαινόμενα κτλ). Η δρομολόγηση πακέτων από πολλαπλά μονοπάτια είναι μια τεχνική που θα ακολουθηθεί για να πετύχουμε αυτό το στόχο.

Μεγιστοποίηση χρόνου ζωής ασύρματου δικτύου αισθητήρων και των κόμβων αισθητήρων:

Οι κόμβοι αισθητήρων είναι εφοδιασμένοι με 2 μπαταρίες τύπου AA, μη επαναφορτιζόμενες. Η κατανάλωση ενέργειας των αισθητήρων πρέπει να είναι όσο πιο χαμηλή γίνεται ώστε να διατηρούνται για μεγαλύτερη χρονική διάρκεια και για να μειώνεται το κόστος συντήρησης.

Επίσης όπως προαναφέρθηκε περιβαλλοντικά φαινόμενα όπως βροχές, σκόνη, σεισμοί μπορούν να βλάψουν το υλικό των κόμβων. Μπορούμε να χρησιμοποιήσουμε προστατευτικές θήκες, χωρίς όμως να επηρεάζονται οι περιβαλλοντικές μετρήσεις ή η αποδοτικότητα της επικοινωνίας.

Επεκτασιμότητα:

Η επεκτασιμότητα του ασύρματου δικτύου αισθητήρων είναι πολύ σημαντική για το σύστημα μας αφού το δίκτυο μπορεί να αποτελείται από εκατοντάδες αισθητήρες.

Επίσης νέοι κόμβοι αισθητήρων μπορούν να ενώνονται στο δίκτυο ή παλιοί κόμβοι να αφαιρούνται από το δίκτυο (σε περίπτωση καταστροφής ή συντήρησης). Το σύστημα θα πρέπει να είναι σε θέση να λειτουργεί αποδοτικά με οποιοδήποτε αριθμό κόμβων αισθητήρων.

4.3.2 Απαιτήσεις συστήματος γεωγραφικών δεδομένων

Οι απαιτήσεις του G.I.S. αφορούν κυρίως την επίδοση, την επεκτασιμότητα, την διεπιφάνεια χρήση και την υποστήριξη.

Επίδοση:

Το σύστημα Συνεργασίας θα αλληλεπιδρά συνεχώς με τη Γεωγραφική Βάση δεδομένων, όπου θα είναι αποθηκευμένο μεγάλο ποσό πληροφορίας σχετικά με τις δασικές περιοχές που βρίσκονται υπό προστασία.

Ο χρόνος καθυστέρησης που υπάρχει κατά την διάδοση των δεδομένων θα πρέπει να μειωθεί στο ελάχιστο ώστε να διασφαλιστεί η ανίχνευση και πρόβλεψη πυρκαγιών σε πραγματικό χρόνο. Επίσης θα πρέπει να γίνεται σε αμελητέο χρόνο η προβολή των σημείων που ζητούν οι χρήστες στον χάρτη.

Επεκτασιμότητα:

Η υπάρχον πλατφόρμα της Geomatic θα πρέπει να έχει δυνατότητες επέκτασης βάσης των χαρακτηριστικών του FireWatch. Πιο συγκεκριμένα οι χάρτες θα πρέπει να μπορούν να υποστηρίξουν επιπλέον στρώματα εκτός από τα προκαθορισμένα.

Παραδείγματα στρωμάτων που θα χρειαστούν κατά την υλοποίηση του FireWatch είναι αυτό των ασύρματων κόμβων, του είδους της δασικής βλάστησης και εδαφικής ανάλυσης της περιοχής όπου είναι εγκατεστημένοι οι αισθητήρες και η εμφάνιση μονοπατιών που μπορεί να είναι όχι απαραίτητα μέρος κάποιου οικιστικού δρόμου.

Η Γεωγραφική Βάση Δεδομένων θα πρέπει να είναι σε θέση να υποστηρίξει τέτοια νέα στρώματα.

Διεπιφάνεια χρήστη:

Μέσω του συστήματος Συνεργασίας οι χρήστες θα μπορούν να αλληλεπιδρούν με την Γεωγραφική Βάση Δεδομένων παρακολουθώντας τα δεδομένα των αισθητήρων σε συνδυασμό με τα γεωγραφικά δεδομένα.

Η αλληλεπίδραση του χρήστη με το σύστημα των γεωγραφικών δεδομένων θα πρέπει να είναι εύκολη στη διαχείριση.

Υποστήριξη:

Σκοπός της χρήσης γεωγραφικών δεδομένων είναι η παροχή πληροφοριών στους εμπλεκόμενους φορείς για την αντιμετώπιση της πυρκαγιάς πιο αποτελεσματική. Με την χρήση γεωγραφικών δεδομένων οι διάφοροι κυβερνητικοί φορείς θα μπορούν να συντονιστούν πιο αποτελεσματικά.

Επίσης οι διάφοροι τρόποι επικοινωνίας θα πρέπει περιλαμβάνουν γεωγραφικές πληροφορίες, ώστε οι εμπλεκόμενοι από το Τμήμα Δασών να μπορούν να καθορίσουν ανάλογα ποιο σχέδιο δράσης θα ακολουθήσουν.

4.3.3 Απαιτήσεις συστήματος συνεργασίας

Οι απαιτήσεις του Συστήματος Συνεργασίας καθοδηγούνται από τους χρήστες του FireWatch και ενδέχεται να προστεθούν επιπλέον απαιτήσεις κατά την υλοποίηση του συστήματος.

Ευχρηστία:

Οι χρήστες θα αλληλεπιδρούν συνεχώς με το FireWatch μέσω της διαδικτυακής εφαρμογής.

Οι χρήστες θα πρέπει να μπορούν εύκολα να πλοηγηθούν και να παρακολουθούν σε πραγματικό χρόνο τα γεωγραφικά δεδομένα και τα δεδομένα των αισθητήρων. Το μεγάλο πλήθος δεδομένων (από τους αισθητήρες και τα γεωγραφικά δεδομένα) θα πρέπει να είναι άρτια οργανωμένο ώστε να μπορεί εύκολα να διαχειρίζεται. Παραδοσιακά υπάρχουν πέντε γενικές αρχές ευχρηστίας:

- *Ευκολία στην εκμάθηση*: ο χρήστης μπορεί γρήγορα να μάθει το σύστημα ώστε αυτό να του φανεί χρήσιμο.
- *Αποδοτικό στη χρήση*: από τη στιγμή που ο χρήστης έχει μάθει το σύστημα, να είναι δυνατή η υψηλή παραγωγικότητα στη χρήση του.
- *Ευκολία στην απομνημόνευση*: να είναι εύκολη για τον περιστασιακό χρήστη η επαναχρησιμοποίηση του συστήματος μετά την πάροδο κάποιου χρόνου αποχής από την χρήση.
- *Λίγα σφάλματα*: περιορισμένος αριθμός λαθών από πλευράς του χρήστη, κατά την χρήση του συστήματος. Εύκολη επαναφορά από τα λάθη και αποτροπή καταστροφικών λαθών.
- *Ευχάριστο στη χρήση*: οι χρήστες είναι υποκειμενικά ευχαριστημένοι από τη χρήση του συστήματος.

Ασφάλεια:

Στη βάση δεδομένων του Συστήματος Συνεργασίας υπάρχουν αποθηκευμένες εμπιστευτικές επαφές και ευαίσθητα στοιχεία προσωπικού απορρήτου. Κρίνεται μεγάλης σημασίας η ασφάλεια πρόσβασης στο σύστημα από τρίτους.

Στη διαδικτυακή εφαρμογή του FireWatch, θα έχουν πρόσβαση μόνο εξουσιοδοτημένα άτομα. Για να γίνει είσοδος στο σύστημα ο κάθε χρήστης θα πρέπει να εισάγει τα διαπιστευτήρια που του έχουν ανατεθεί, ενώ θα χρησιμοποιηθούν πρωτόκολλα ασφαλείας ώστε να αποτρέπονται κακόβουλες προσπάθειες πρόσβασης.

Αξιοπιστία:

Στη βάση δεδομένων του συστήματος θα αποθηκεύονται τα δεδομένα τα οποία λαμβάνονται από το δίκτυο των αισθητήρων, πληροφορίες για τους κόμβους αισθητήρων, την κατάσταση του δικτύου και άλλα δεδομένα που χρειάζονται για τη διαχείριση του συστήματος. Επίσης οι χρήστες θα μπορούν να καταχωρούν νέες πληροφορίες στο σύστημα.

Τα νέα δεδομένα που εισάγονται θα πρέπει να ελέγχονται για την εγκυρότητα τους και να αφήνουν σε συνεπή κατάσταση το σύστημα και τις βάσεις δεδομένων.

Επικοινωνία:

Όπως προαναφέρθηκε το Σύστημα Συνεργασίας θα χρησιμοποιεί διάφορους τρόπους επικοινωνίας για να στέλλει προειδοποιήσεις στους χρήστες όπως SMS, Voice Messages, emails, Screen Notifications και Sound Alarms.

Το σύστημα θα πρέπει να παρέχει όλους τους μηχανισμούς για αποστολή τέτοιων προειδοποιήσεων.

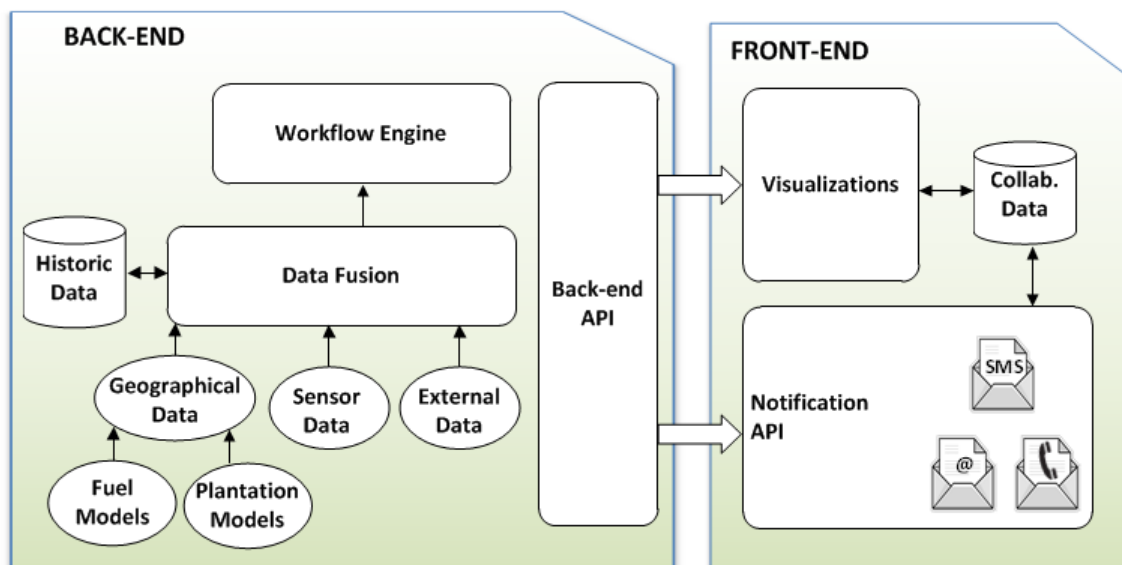
Κεφάλαιο 5

Αρχιτεκτονική Συστήματος

5.1	Γενική αρχιτεκτονική.....	35
5.1.1	Επίπεδο-1: Back-End	36
5.1.2	Επίπεδο-2: Front-End.....	37

5.1 Γενική αρχιτεκτονική

Σε αυτό το κεφάλαιο θα παρουσιαστεί η αρχιτεκτονική που ακολουθεί το σύστημα FireWatch. Στην Εικόνα 12 παρουσιάζεται η βασική αρχιτεκτονική που έχει υλοποιηθεί.



Εικόνα 12: Αρχιτεκτονική Συστήματος

Όπως φαίνεται το σύστημα αποτελείται από 2 επίπεδα – το back-end και το front-end. Το back-end επίπεδο είναι υπεύθυνο για την συλλογή των δεδομένων από τους αισθητήρες, τα γεωγραφικά δεδομένα και από εξωτερικές πηγές, την ανάλυση

τους και τα παραγωγή προειδοποιήσεων σε περίπτωση ανίχνευσης ή πρόβλεψης πυρκαγιάς. Το front-end επίπεδο είναι υπεύθυνο για την γραφική αναπαράσταση των δεδομένων, την διαπροσωπεία του back-end στους χρήστες και την παροχή των προειδοποιήσεων σε διάφορες μορφές στις αρμόδιες αρχές. Παραδείγματα τέτοιων προειδοποιήσεων είναι με SMS, Emails, Voice messages κτλ.

5.1.1 Επίπεδο-1: Back-End

Τώρα θα δοθεί μια λεπτομερής περιγραφή για κάθε component του επιπέδου back-end:

- **Δεδομένα αισθητήρων**

Σε αυτό το μέρος του συστήματος ανήκει το Ασύρματο Δίκτυο Αισθητήρων, που είναι και ο πυρήνας του FireWatch. Ένας αριθμός από ασύρματους κόμβους αισθητήρων θα εξαπλωθεί στο δάσος με σκοπό την καταγραφή περιβαλλοντικών μετρήσεων σε τακτά χρονικά διαστήματα.

Τα δεδομένα θα μαζεύονται από ένα κεντρικό κόμβο (sink node/base station), και θα αποθηκεύονται σε μια βάση δεδομένων αποκλειστικά προσαρμοσμένη για το δίκτυο αισθητήρων. Οι κόμβοι χρησιμοποιούν αποδοτικούς αλγόριθμους για την λειτουργία, μεταφορά δεδομένων και αναδιοργάνωση τους ώστε να αυξάνουν τον χρόνο ζωής του δικτύου και να μειώνουν τα κόστη συντήρησης.

- **Γεωγραφικά δεδομένα**

Τα Γεωγραφικά Δεδομένα προέρχονται από το γεωγραφικό σύστημα δεδομένων. Το μέρος αυτό αποτελεί ένα επίσης σημαντικό μέρος της ολικής αρχιτεκτονικής. Τα γεωγραφικά δεδομένα, θα παρέχονται από την εταιρεία T.C. Geomatic [5] και θα είναι προσβάσιμα επιτρέποντας να λαμβάνουμε χρήσιμα δεδομένα για την επιλεγμένη δασική περιοχή. Η εδαφική ανάλυση και η ψηφιακή μοντελοποίηση του εδάφους επιτρέπουν την προστασία της δασικής περιοχής αφού θα καθορίζουν παραμέτρους κατά την τοποθέτηση του ασύρματου δικτύου αισθητήρων και την εκτίμηση της συμπεριφοράς της πυρκαγιάς.

Επιπρόσθετα παρέχονται δεδομένα για δρόμους αλλά και δεδομένα εκτός δρόμων που χρησιμοποιούνται για να ανευρεθούν εναλλακτικά μονοπάτια. Με τα δεδομένα αυτά οι πυροσβέστες θα μπορούν να προσεγγίσουν πιο εύκολα δύσβατες περιοχές.

- **Συγχώνευση δεδομένων**

Στο σημείο αυτό γίνεται η συγχώνευση των δεδομένων. Τα γεωγραφικά δεδομένα μαζί με τις μετρήσεις των αισθητήρων (πρόσφατες αλλά και πιο παλιές), συνδυάζονται για να παράξουν χρήσιμες πληροφορίες προς τους χρήστες σχετικά με την κατάσταση του δικτύου.

Η συγχώνευση των δεδομένων από πολλές πηγές (δίκτυο αισθητήρων, γεωγραφικά δεδομένα, εξωτερικές πηγές) επιτρέπει στο σύστημά μας να χρησιμοποιεί αλγορίθμους για ανίχνευση πυρκαγιάς με μεγάλη ακρίβεια και οι πιθανές πυρκαγιές εντοπίζονται με μεγαλύτερη ακρίβεια. Επίσης με τη χρησιμοποίηση των εξωτερικών πηγών είναι εφικτή η χρήση εξελιγμένων αλγορίθμων για να προβλέψουμε την κατεύθυνση διάδοσης της πυρκαγιάς.

- **Παραγωγή ροής εργασιών**

Το Workflow Engine είναι υπεύθυνο για την ανάλυση τόσο των δεδομένων που λαμβάνονται σε πραγματικό χρόνο, όσο και των ιστορικών δεδομένων, έτσι ώστε να ανιχνεύει και να προβλέπει ανωμαλίες στις δασικές περιοχές.

Σε περίπτωση που γίνει ανίχνευση φωτιάς το τμήμα δασών ακολουθεί προκαθορισμένες ενέργειες και διαδικασίες για την κατάσβεση της. Με την παραγωγή ροής εργασιών οι διάφοροι εμπλεκόμενοι που θα καλούνται να αντιμετωπίσουν τη πυρκαγιά θα υποβοηθούνται από το σύστημα ώστε να γίνεται πιο αποδοτικά η άμεση ενημέρωση τους και σχετικές πληροφορίες για την πυρκαγιά. Με αυτό τον τρόπο θα επιταχύνουμε τις διαδικασίες συνεργασίας τους μαζί με όλους τους εμπλεκόμενους φορείς.

5.1.2 Επίπεδο-2: Front-End

Το δεύτερο επίπεδο – front-end – είναι αυτό που θα είναι «κοντά» και ορατό στους χρήστες. Τα κύρια συστατικά του θα αναλυθούν πιο κάτω:

- **Γραφικές απεικονίσεις**

Τα δεδομένα που προέρχονται από το επίπεδο back-end θα τροφοδοτούνται σε αυτό το επίπεδο και θα προβάλλονται με γραφικό τρόπο προς τον χρήστη.

Οι γραφικές απεικονίσεις θα είναι μέρος του web-application όπου εξουσιοδοτημένοι χρήστες θα μπορούν να δουν και να επιτηρούν το σύστημα. Παραδείγματα γραφικών απεικονίσεων είναι οι χάρτες, πίνακες, γραφικές παραστάσεις κτλ.

Βασική λειτουργία είναι η επιβολή επερωτήσεων προς το σύστημα και ανάκτηση δεδομένων από τα επιμέρους συστατικά του επιπέδου back-end.

- **API ειδοποιήσεων**

Σε περίπτωση ανίχνευσης πυρκαγιάς το API ειδοποιήσεων παρέχει τους μηχανισμούς ενημέρωσης των αρμόδιων αρχών για την λήψη μέτρων.

Τα διάφορα είδη ενημέρωσης που παρέχονται περιλαμβάνουν προειδοποιήσεις με την χρήση sms, voice mail, email, alarm και screen notifications.

Κεφάλαιο 6

Σχεδίαση Συστήματος

6.1	Εισαγωγή	39
6.2	Σχεδίαση ασύρματου δικτύου αισθητήρων	40
6.2.1	Συστατικά ασύρματου δικτύου αισθητήρων.....	40
6.2.2	Τοπολογία δικτύου	49
6.2.3	Πλαίσιο λογισμικού για το ασύρματο δίκτυο αισθητήρων.....	52
6.2.4	Δομή πακέτου αισθητήρων	58
6.2.5	Μετασχηματισμός ακατέργαστων μετρήσεων αισθητήρων σε παγκόσμιες μονάδες μέτρησης	61
6.2.6	Βάση Δεδομένων Ασύρματου Δικτύου Αισθητήρων	67
6.3	Σχεδίαση συστήματος γεωγραφικών δεδομένων.....	70
6.3.1	Λειτουργίες γεωγραφικού συστήματος δεδομένων	70
6.3.2	Επιμέρους συστατικά γεωγραφικού συστήματος δεδομένων.....	71
6.4	Σχεδίαση συστήματος συνεργασίας.....	71
6.4.1	Βάση δεδομένων συστήματος συνεργασίας	72
6.4.2	Συγχρονισμός βάσης δεδομένων αισθητήρων με βάση δεδομένων συστήματος συνεργασίας.....	78
6.4.3	Διαπροσωπεία FireWatch	79
6.4.4	Υπηρεσία επικοινωνίας	80
6.4.5	Παραγωγή ροής εργασιών.....	83

6.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα επεξηγηθεί η σχεδίαση του συστήματος. Έχοντας καθορίσει την αρχιτεκτονική του συστήματος και τις απαιτήσεις που διέπουν το σύστημα θα προχωρήσουμε στην σχεδίαση του.

Πιο αναλυτικά, θα γίνει η ανάλυση της σχεδίασης κάθε επιμέρους κομματιού της αρχιτεκτονικής: του ασύρματου δικτύου αισθητήρων, συστήματος γεωγραφικών δεδομένων και συστήματος συνεργασίας.

Η σχεδίαση του συστήματος θέτει τους βασικούς άξονες πάνω στους οποίους θα στηριχθεί η υλοποίηση. Οι προδιαγραφές και οι απαιτήσεις που καθορίστηκαν στο προηγούμενο κεφάλαιο θα συντελέσουν στην επακριβή σχεδίαση του συστήματος.

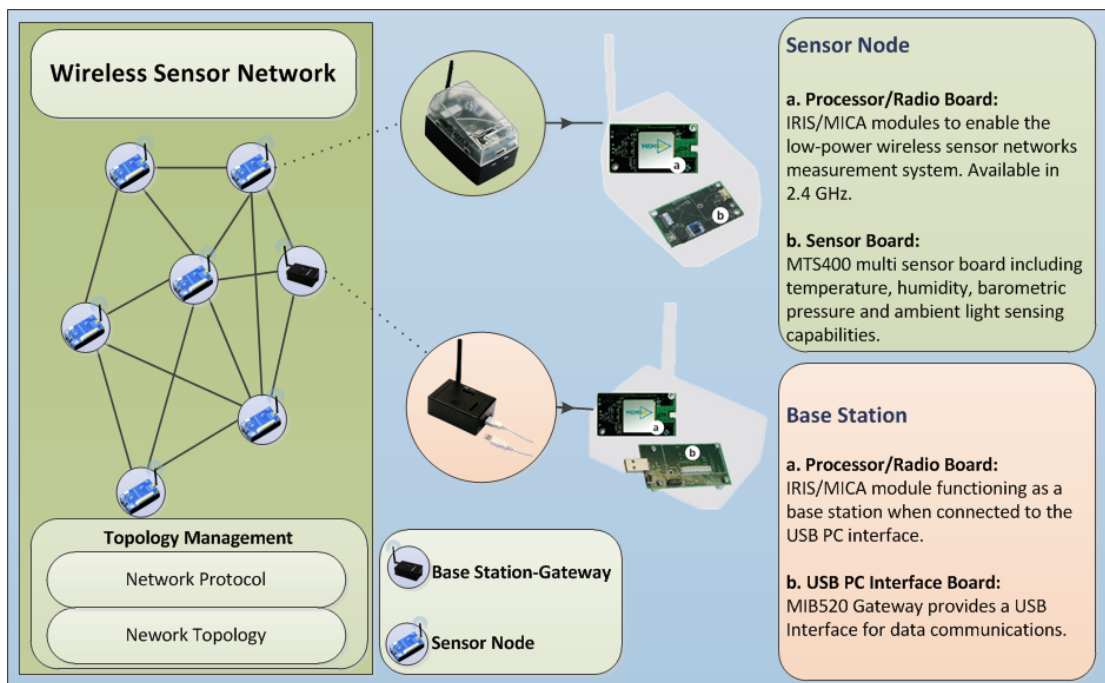
6.2 Σχεδίαση ασύρματου δικτύου αισθητήρων

Αρχίζοντας από το ασύρματο δίκτυο αισθητήρων θα δούμε ποια είναι τα επιμέρους συστατικά που το απαρτίζουν.

Συγκεκριμένα αυτή η ενότητα παρέχει μια επισκόπηση υψηλού επιπέδου των συνιστωσών και τις αλληλεπιδράσεων του Ασύρματου Δικτύου Αισθητήρων. Στη συνέχεια παρέχουμε μια πιο λεπτομερή περιγραφή του κάθε συστατικού.

6.2.1 Συστατικά ασύρματου δικτύου αισθητήρων

Σύμφωνα με την Εικόνα 13, το ασύρματο δίκτυο αισθητήρων αποτελείται από έναν αριθμό κόμβων αισθητήρων με δυνατότητες ασύρματης επικοινωνίας. Υπάρχει επίσης ο βασικό σταθμό (base station), ο οποίος συνδέεται σε ένα τοπικό υπολογιστή και συλλέγει τα δεδομένα από τους αισθητήρες.



Εικόνα 13: Κύρια συστατικά Ασύρματου Δικτύου Αισθητήρων

Οι κόμβοι αισθητήρων είναι διεσπαρμένοι στο χώρο, σχηματίζοντας το δίκτυο. Περιοδικά οι μετρήσεις που καταγράφουν για το περιβάλλον συγκεντρώνονται στο βασικό σταθμό όπου και καταγράφονται μόνιμα σε μια βάση δεδομένων.

Η οργάνωση του ασύρματου δικτύου αισθητήρων επιτυγχάνεται μέσω κάποιου δικτυακού πρωτοκόλλου που θα επεξηγηθεί αργότερα σε αυτό το κεφάλαιο. Το πρωτόκολλο αυτό είναι υπεύθυνο και για την αναδιοργάνωση του δικτύου κάθε φορά που τυχαίνει οποιαδήποτε αλλαγή σε αυτό (βλάβη, καταστροφή, έλλειψη ενέργειας κτλ).

Κόμβοι αισθητήρων:

Για τον σχεδιασμό του συστήματος του FireWatch η επιλογή κατάλληλων αισθητήρων ήταν αναγκαία. Οι κόμβοι των αισθητήρων όπως καθορίστηκε στις απαιτήσεις πρέπει να έχουν την δυνατότητα να καταγράφουν περιβαλλοντικά φαινόμενα που να έχουν άμεση σχέση με την ανίχνευση φωτιάς.

Τέτοιοι αισθητήρες υπάρχουν πολλοί στην αγορά. Η επιλογή μας ήταν στηριγμένη στο να βρούμε αισθητήρες που είναι κατασκευασμένοι από εταιρεία με κύρος και ιστορία γύρω από το χώρο των ασύρματων αισθητήρων και κατ' επέκταση να είναι ποιοτικοί και αποδοτικοί στη λειτουργία τους.

Οι αισθητήρες που χρησιμοποιήθηκαν έχουν τα εξής χαρακτηριστικά:

- **Επεξεργαστής IRIS M2110:**



Εικόνα 14: Επεξεργαστής IRIS M2110

Η τελική επιλογή για τον επεξεργαστή των αισθητήρων του συστήματος είναι το Iris Wireless Measurement System M2110 [6] από την εταιρεία MEMSIC [7] που φαίνεται στην Εικόνα 14. Χρησιμοποιείται για χαμηλής ισχύος ασύρματα δίκτυα

αισθητήρων. Τα IRIS Motes έχουν αρκετές νέες δυνατότητες που βελτιώνουν τη συνολική λειτουργία των ασύρματων προϊόντων δικτύωσης MEMSIC.

Ο συγκεκριμένος επεξεργαστής διαθέτει connector (Εικόνα 15) που επιτρέπει την εγκατάσταση sensor board πάνω από αυτόν, δίνοντας τη δυνατότητα να παίρνουμε μετρήσεις για θερμοκρασία, φωτεινότητα, βαρομετρική πίεση/θερμοκρασία, υγρασία κτλ.

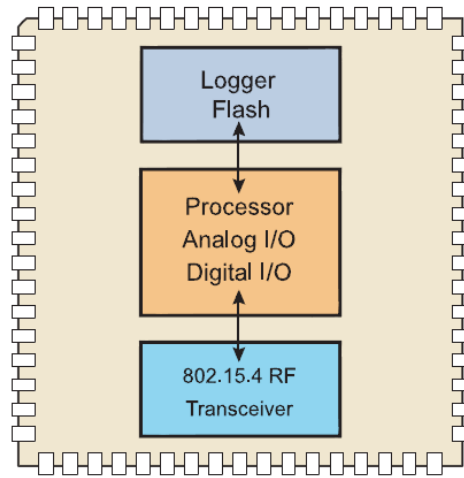


Εικόνα 15: Connector (με άσπρο χρώμα) του επεξεργαστή IRIS M2110 με μπαταριοθήκη

Σε σύγκριση με άλλα motes προσφέρει τα εξής πλεονεκτήματα:

- Έχουν 3 φορές πιο βελτιωμένη εμβέλεια και 2 φορές μεγαλύτερη μνήμη προγράμματος σε σύγκριση με τα προηγούμενα MICA motes.
- Έχουν δοκιμαστεί σε ευθεία και μπορούν να επικοινωνούν σε απόσταση 500 μέτρων χωρίς ενίσχυση.
- Χρησιμοποιεί IEEE 802.15.4/ZigBee συμβατό πομποδέκτη RF
- Λειτουργούν στην παγκόσμια συμβατή μπάντα συχνοτήτων 2.4-2.48 GHz
- Άμεση εξάπλωση του ραδιοφάσματος που είναι ανθεκτικό στις παρεμβολές RF και παρέχει ασφάλεια δεδομένων.
- Έχει μέχρι 250kbps data rate
- Υποστηρίζεται από την πλατφόρμα ασύρματης δικτύωσης αισθητήρων MoteWorks™ για αξιόπιστο ad-hoc πλέγμα δικτύωσης. Το MoteWorks™ επιτρέπει την ανάπτυξη προσαρμοσμένων εφαρμογών αισθητήρων και είναι βελτιστοποιημένο για ελάχιστη κατανάλωση μπαταρίας. Είναι επίσης στηριγμένο στο open-source TinyOS λειτουργικό σύστημα και προσδίδει αξιόπιστη ad-hoc mesh δικτύωση, δυνατότητες προγραμματισμού αισθητήρων ασύρματα, εργαλεία ανάπτυξης ανεξαρτήτως πλατφόρμας, server middleware

για ενσωμάτωση δικτύου επιχειρήσεων και διαπροσωπεία χρήστη για την ανάλυση και ρύθμιση.



Εικόνα 16: Διάγραμμα Chip XM2110CA

Ο επεξεργαστής XM2110CB (Εικόνα 16) είναι βασισμένος στον Atmel ATmega1281. Ο ATmega1281 είναι ένας χαμηλής ισχύος microcontroller που τρέχει MoteWorks από την εσωτερική του μνήμη flash.

Μια πλακέτα επεξεργαστή XM2110 μπορεί να ρυθμιστεί ώστε να τρέχει την εφαρμογή των αισθητήρων και την επεξεργασία τους μαζί με την επικοινωνία του δικτύου ταυτόχρονα.

Ο connector του IRIS με τα 51 pins υποστηρίζει αναλογικά inputs, ψηφιακά I/O, δίαυλο Inter-Integrated Circuit (I²C), δίαυλο Serial Peripheral Interface (SPI) και Universal Asynchronous Receiver/Transmitter (UART) διαπροσωπείες. Με αυτές τις διαπροσωπείες είναι εύκολο να προστεθούν εξωτερικές περιφερειακές συσκευές.

Ο Πίνακας 2 προβάλλει συνοπτικά τα τεχνικά χαρακτηριστικά του επεξεργαστή XM2110CA.

Processor/Radio Board	XM2110CB	Remarks
Processor Performance		
Program Flash Memory	128K bytes	
Measurement (Serial) Flash	512K bytes	> 100,000 Measurements
RAM	8K bytes	
Configuration EEPROM	4K bytes	
Serial Communications	UART	0-3V transmission levels
Analog to Digital Converter	10 bit ADC	8 channel, 0-3V input
Other Interfaces	Digital I/O,I2C,SPI	
Current Draw	8 mA	Active mode
	8 μ A	Sleep mode (total)
RF Transceiver		
Frequency band	2405 MHz to 2480 MHz	ISM band, programmable in 1 MHz steps
Transmit (TX) data rate	250 kbps	
RF power	3 dBm (typ)	
Receive Sensitivity	-101 dBm (typ)	
Adjacent channel rejection	36 dB	+ 5 MHz channel spacing
	34 dB	- 5 MHz channel spacing
Outdoor Range	> 300 m	1/4 wave dipole antenna, LOS
Indoor Range	> 50 m	1/4 wave dipole antenna, LOS
Current Draw	16 mA	Receive mode
	10 mA	TX, -17 dBm
	13 mA	TX, -3 dBm
	17 mA	TX, 3 dBm
Electromechanical		
Battery	2X AA batteries	Attached pack
External Power	2.7 V - 3.3 V	Molex connector provided
User Interface	3 LEDs	Red, green and yellow
Size (in)	2.25 x 1.25 x 0.25	Excluding battery pack
(mm)	58 x 32 x 7	Excluding battery pack
Weight (oz)	0.7	Excluding batteries
(grams)	18	Excluding batteries
Expansion Connector	51-pin	All major I/O signals

Πίνακας 2: Τεχνικά Χαρακτηριστικά επεξεργαστή XM2110CA

- **MTS400 Βασική Πλακέτα Αισθητήρα Περιβάλλοντος:**

Για την επιλογή του sensor board λήφθηκε υπόψη το γεγονός ότι θα έπρεπε να συμβαδίζει με τις προδιαγραφές του συστήματος μας, όπως έχουμε προαναφέρει για την εφαρμογή.

Η MEMSIC προσφέρει μια ποικιλία από sensor και data acquisition boards για το IRIS Mote. Όλες αυτές οι πλακέτες ενώνονται με το IRIS μέσω του 51-pin connector. Για το σύστημα μας, χρησιμοποιήσαμε την πλακέτα Αισθητήρα Περιβάλλοντος MTS400 [8] η οποία φαίνεται στην Εικόνα 17.

Οι δυνατότητες του συγκεκριμένου μοντέλου είναι ιδανικές για την καταγραφή ισχυρών περιβαλλοντικών φαινομένων. Ενώνοντας το με τον IRIS μπορούμε να καταγράφουμε μετρήσεις όπως θερμοκρασία, υγρασία, βαρομετρική πίεση, φωτισμό και accelerometer στους άξονες x,y.

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



Εικόνα 17: MTS400 Environmental Sensor Board

Η πλακέτα MTS400 χρησιμοποιεί διαφορετικούς μικροεπεξεργαστές που είναι σχεδιασμένοι για να μετρούν συγκεκριμένα περιβαλλοντικά φαινόμενα.

Στη συνέχεια θα δούμε ποιοι είναι αυτοί λεπτομερώς, τα τεχνικά τους χαρακτηριστικά και την λειτουργικότητα τους.

Αισθητήρας Υγρασίας και Θερμοκρασίας:

Ο Sensirion® SHT11 [9] είναι ένας single-chip αισθητήρας υγρασίας και θερμοκρασίας με βαθμονομημένη ψηφιακή έξοδο. Διαθέτει εσωτερικό analog-to-digital μετατροπέα 14-bit και σειριακή θύρα.

Η τροφοδοσία του αισθητήρα ενεργοποιείται μέσω ενός προγραμματιζόμενου διακόπτη, όπως επίσης και τα σήματα της θύρας ελέγχου. Τα αναλογικά σήματα που λαμβάνονται για την θερμοκρασία και υγρασία, μετατρέπονται σε ψηφιακές μονάδες με τον analog-to-digital μετατροπέα. Ο πιο κάτω πίνακας παρουσιάζει συνοπτικά τα τεχνικά χαρακτηριστικά του συγκεκριμένου αισθητήρα.

Τύπος Αισθητήρα	Sensirion SHT11	
Κανάλια	Υγρασία	Θερμοκρασία
Διακύμανση	0 μέχρι 100%	-40°C μέχρι 80°C
Ακρίβεια	± 3.5% RH (τυπικό)	± 2°C
Εμβέλεια Λειτουργίας	3.6 μέχρι 2.4 volts	
Θύρα	Ψηφιακή Θύρα	

Πίνακας 3: Περίληψη των προδιαγραφών του Sensirion® SHT11

Αισθητήρας Βαρομετρικής Πίεσης και Θερμοκρασίας:

Ο Intersema® MS55ER [10] είναι μια SMD-υβριδική συσκευή που περιλαμβάνει ένα πιεσοαντιστατικό αισθητήρα πίεσης και ολοκληρωμένο κύκλωμα analog-to-digital μετατροπέα. Δίνει μετρήσεις για βαρομετρική πίεση και θερμοκρασία υπό μορφή 16-bit δεδομένων.

Η τροφοδοσία του αισθητήρα ενεργοποιείται μέσω ενός προγραμματιζόμενου διακόπτη, όπως επίσης και τα σήματα της θύρας ελέγχου. Τα αναλογικά σήματα που λαμβάνονται για την βαρομετρική πίεση και θερμοκρασία, μετατρέπονται σε ψηφιακές μονάδες με τον analog-to-digital μετατροπέα. Ο πιο κάτω πίνακας παρουσιάζει συνοπτικά τα τεχνικά χαρακτηριστικά του συγκεκριμένου αισθητήρα.

Τύπος Αισθητήρα	Intersema MS55ER	
Κανάλια	Πίεση	Θερμοκρασία
Διακύμανση	300 μέχρι 110 mbar	-10°C μέχρι 60°C
Ακρίβεια	± 3.5%	± 2°C
Εμβέλεια Λειτουργίας	3.6 μέχρι 2.4 volts	
Θύρα	Ψηφιακή Θύρα	

Πίνακας 4: Περίληψη των προδιαγραφών του Intersema® MS55ER

Αισθητήρας Φωτός:

Ο TLS2550 είναι ένας ψηφιακός αισθητήρας φωτός με δύο σύρματα, SMBus σειριακής θύρας. Κατασκευάζεται από την TAOS, Inc. [11]. Συνδυάζει δύο φωτοδιόδους και analog-to-digital μετατροπέα σε ένα ενιαίο ολοκληρωμένο κύκλωμα CMOS για να παρέχει μετρήσεις φωτός σε ένα αποτελεσματικό 12-bit δυναμικό εύρος.

Η τροφοδοσία του αισθητήρα ενεργοποιείται μέσω ενός προγραμματιζόμενου διακόπτη, όπως επίσης και τα σήματα της θύρας ελέγχου. Τα αναλογικά σήματα που λαμβάνονται για το φως, μετατρέπονται σε ψηφιακές μονάδες με τον analog-to-digital μετατροπέα. Ο πιο κάτω πίνακας παρουσιάζει συνοπτικά τα τεχνικά χαρακτηριστικά του συγκεκριμένου αισθητήρα.

Τύπος Αισθητήρα	Taos TSL2550
Κανάλια	Φως
Διακύμανση	400 μέχρι 1000 nm
Εμβέλεια Λειτουργίας	3.6 μέχρι 2.7 volts
Θύρα	Ψηφιακή Θύρα

Πίνακας 5: Περίληψη των προδιαγραφών του TAOS TSL2550

Επιταχυνσιόμετρο 2 αξόνων:

Το επιταχυνσιόμετρο ADXL202JE είναι μια επιφάνεια μικροηλεκτρομηχανικού συστήματος 2 αξόνων με ακρίβεια ± 2 g. Μπορεί να χρησιμοποιηθεί για την ανίχνευση κλίσης, κίνησης, δόνησης ή σεισμού. Η έξοδος του αισθητήρα είναι συνδεδεμένη με ADC κανάλια πάνω στα ADC1 και ADC2 κανάλια του IRIS mote.

Ο πιο κάτω πίνακας παρουσιάζει συνοπτικά τα τεχνικά χαρακτηριστικά του συγκεκριμένου αισθητήρα.

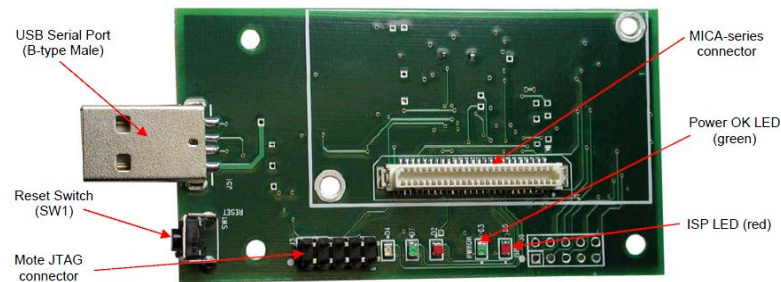
Τύπος Αισθητήρα	Αναλογικές Συσκευές ADXL202JE
Κανάλια	X (ADC1), Y (ADC2)
Διακύμανση	± 2 G (1 G = 9.81 m/s ²)
Ευαισθησία	167 mV/G, ± 17 %
Ανάλυση	2 mG (0.002 G) RMS
Offset	VATTERY/2 ± 0.4 V
Εμβέλεια Λειτουργίας	3.6 μέχρι 3.0 volts
Θύρα	Αναλογική Θύρα

Πίνακας 6: Περίληψη των προδιαγραφών του ADXL202JE

MIB520 USB Programming Board:

Το MIB520 (Εικόνα 18) παρέχει συνδεσιμότητα USB με την οικογένεια του IRIS motes για την επικοινωνία και το σύστημα προγραμματισμού. Παρέχει ισχύ στις συσκευές μέσω διαύλου USB.

Το MIB520 έχει ενσωματωμένο επεξεργαστή (in-system processor) που δίνει τη δυνατότητα προγραμματισμού των sensor motes. Ο κώδικας κατεβαίνει στο ISP μέσω της θύρας USB και στη συνέχεια το ISP εγκαθιστά τον κώδικα στο Mote.



Εικόνα 18: Άνω όψη MIB520CB

Βασικός σταθμός (Base Station):

Ο βασικός σταθμός (Εικόνα 19) παίζει το ρόλο της πύλης. Αποτελείται από ένα επεξεργαστή IRIS και ένα MIB520 USB Programming Board όπως είδαμε πιο πάνω. Ο σταθμός αυτός επιτρέπει τη συγκέντρωση των δεδομένων όλου του ασύρματου δικτύου αισθητήρων σε ένα προσωπικό υπολογιστή. Ο κάθε IRIS mote μπορεί να παίζει το ρόλο του σταθμού βάσης, αρκεί να είναι συνδεδεμένο με μια θύρα σε ένα υπολογιστή. Το MIB520 όπως είδαμε παρέχει μια σειριακή θύρα USB που επιτρέπει τόσο τον προγραμματισμό των αισθητήρων όσο και για την επικοινωνία τους.



Εικόνα 19: Βασικός σταθμός (Base Station)

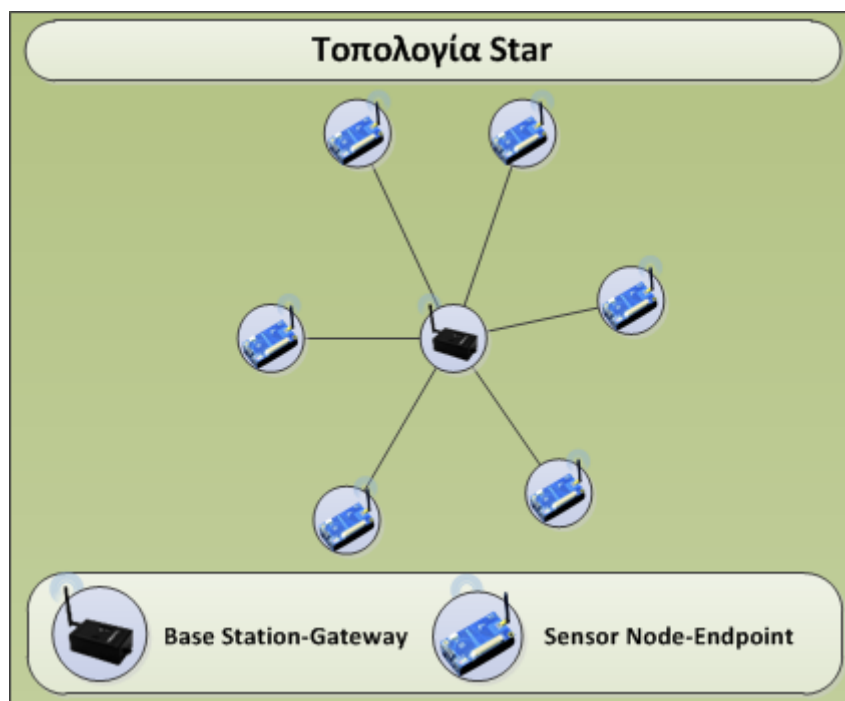
6.2.2 Τοπολογία δικτύου

Για την υλοποίηση ασύρματων δικτύων αισθητήρων υπάρχουν διάφορες αρχιτεκτονικές όπως star, mesh και star-mesh hybrid. Κάθε μια από αυτές τις τοπολογίες παρουσιάζει τις δικές τις προκλήσεις, πλεονεκτήματα και μειονεκτήματα.

Πιο κάτω θα εξηγηθούν οι τοπολογίες που προαναφέρθηκαν για καλύτερη κατανόηση της τελικής μας επιλογής.

Τοπολογία star (αστεριού):

Η τοπολογία star είναι ένα single-hop σύστημα στο οποίο όλοι οι αισθητήρες βρίσκονται σε απευθείας επικοινωνία – συνήθως 30 με 100 μέτρα - με τον gateway. Όλοι οι αισθητήρες είναι πανομοιότυποι και ο gateway εξυπηρετεί απευθείας όλους τους κόμβους αισθητήρων είτε για να στείλουν δεδομένα είτε για να στέλλει εντολές στους κόμβους αισθητήρων. Ο ρόλος του gateway είναι επίσης να μεταδίδει τα δεδομένα σε ένα υψηλού επιπέδου σύστημα ελέγχου. Οι κόμβοι/endpoints δεν ανταλλάζουν μεταξύ τους δεδομένα αλλά χρησιμοποιούν τον gateway για σημείο συντονισμού όπως φαίνεται στην Εικόνα 20.



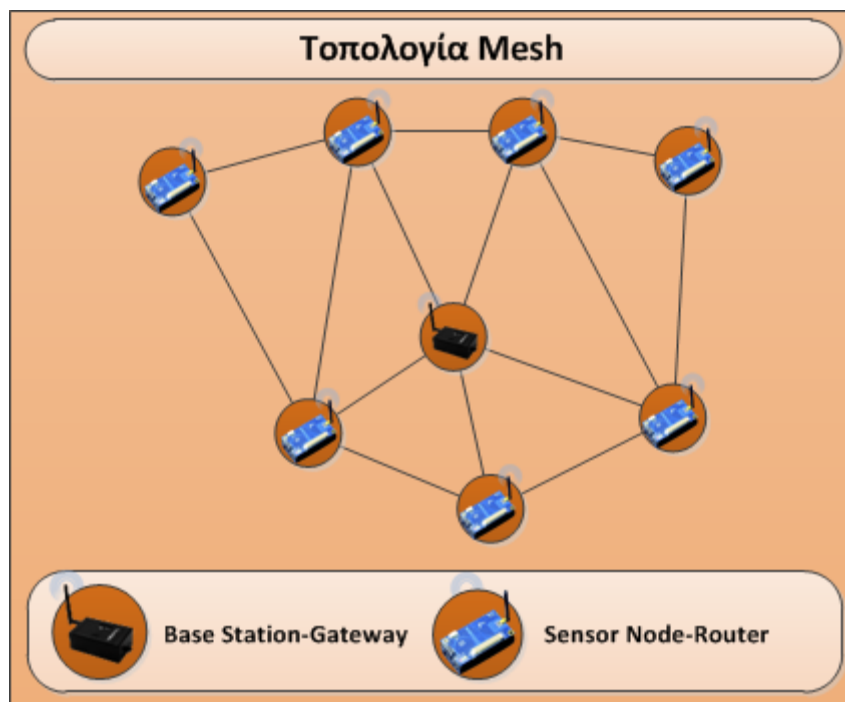
Εικόνα 20: Τοπολογία Star

Η τοπολογία Star δίνει την χαμηλότερη ολική κατανάλωση ενέργειας αλλά είναι περιορισμένη από την απόσταση μετάδοσης μεταξύ gateway και endpoint. Επίσης δεν υπάρχουν εναλλακτικά μονοπάτια επικοινωνίας στους κόμβους/endpoints, που σημαίνει ότι σε περίπτωση βλάβης, αν κάποιο μονοπάτι καταστραφεί τότε η επικοινωνία θα χαθεί προς αυτούς τους κόμβους.

Όπως προκύπτει από τα πιο πάνω μειονεκτήματα η τοπολογία star είναι ακατάλληλη για το FireWatch.

Τοπολογία mesh (πλέγματος):

Η τοπολογία mesh είναι ένα multi-hop σύστημα στο οποίο όλοι οι αισθητήρες είναι πανομοιότυποι – λειτουργούν ως δρομολογητές – και επικοινωνούν μεταξύ τους για να μεταφέρουν δεδομένα από και προς άλλους κόμβους αισθητήρων και τον gateway. Αντίθετα με την τοπολογία star, που οι κόμβοι επικοινωνούσαν μόνο με τον gateway, οι κόμβοι στην τοπολογία mesh μπορούν να στέλλουν μηνύματα μεταξύ άλλων κόμβων αισθητήρων όπως φαίνεται στην Εικόνα 21.



Εικόνα 21: Τοπολογία Mesh

Η μετάδοση δεδομένων από τους αισθητήρες μέσα από το πλέγμα επιτρέπει στο δίκτυο να επεκταθεί θεωρητικά σε μια απεριόριστη απόσταση. Το δίκτυο πλέγματος

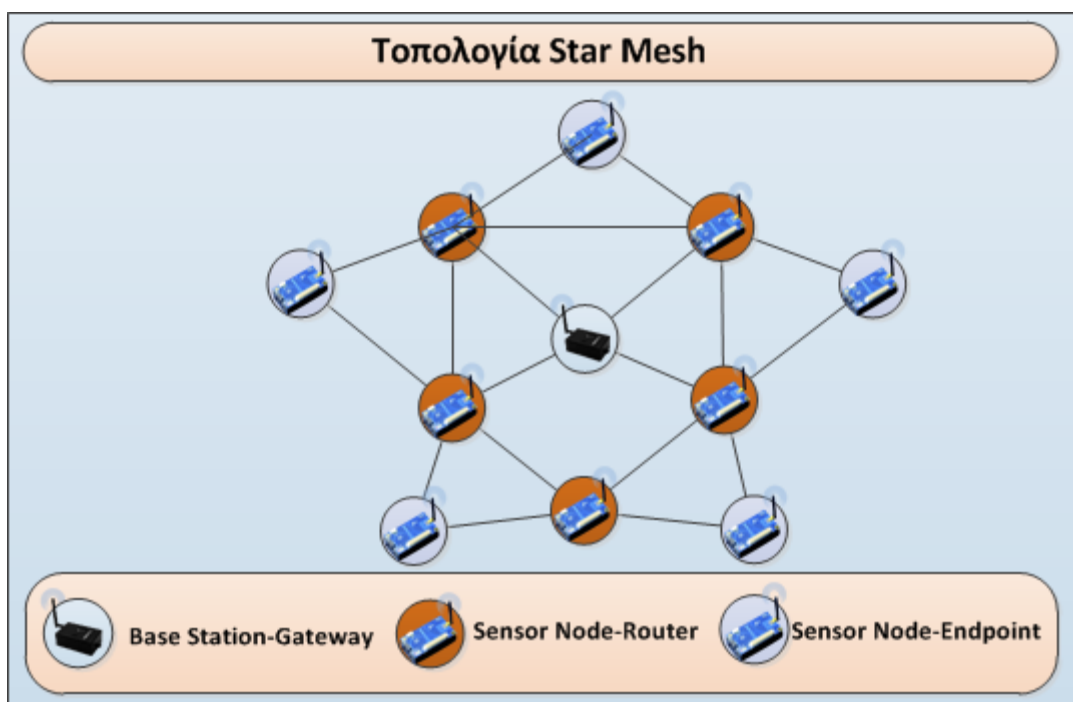
είναι επίσης πολύ ανεκτικό σε περίπτωση βλάβης αφού κάθε κόμβος αισθητήρα έχει πολλά εναλλακτικά μονοπάτια τώρα για να φτάσουν τα δεδομένα μέχρι το gateway και στους άλλους κόμβους. Αν κάποιος κόμβος καταστραφεί το δίκτυο θα μπορεί να αναδιοργανωθεί από μόνο του αυτόματα.

Μειονέκτημα της τοπολογίας αυτής είναι ότι καθώς ο αριθμός των κόμβων αυξηθεί μεταξύ τους, θα παρατηρηθεί στο δίκτυο πιο μεγάλη καθυστέρηση καθώς τα δεδομένα των αισθητήρων θα μεταδίδονται από τους κόμβους προς τον gateway.

Τοπολογία star mesh (αστεριού με πλέγμα):

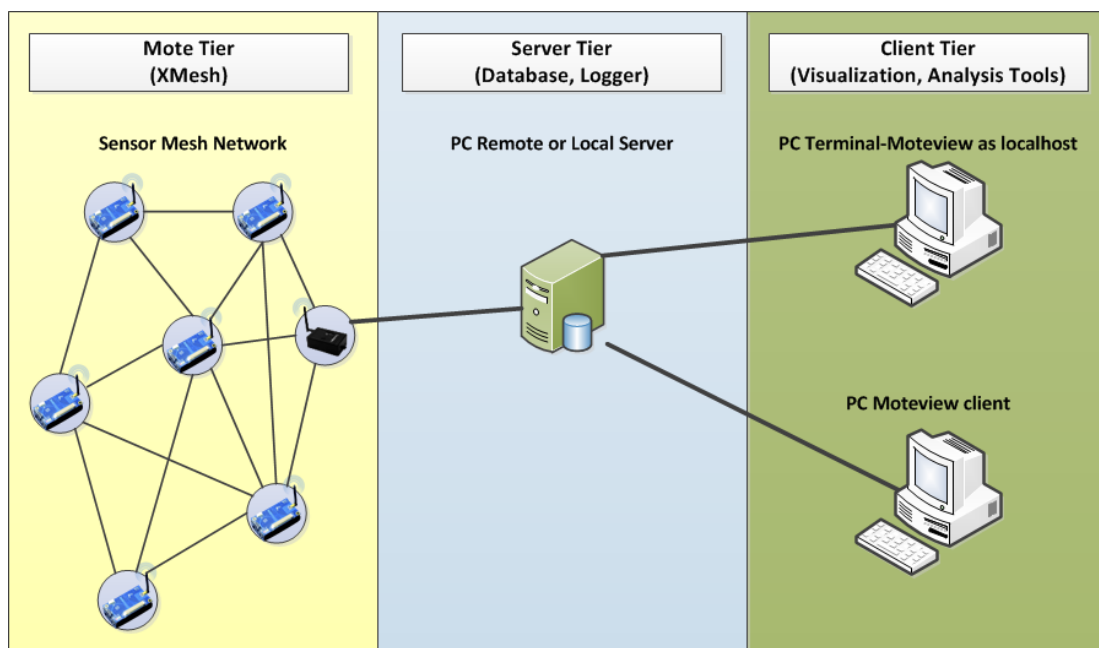
Η υβριδική τοπολογία star-mesh εκμεταλλεύεται την μικρή κατανάλωση ενέργειας και την απλότητα της τοπολογίας star, καθώς και τη διευρυμένη εμβέλεια που έχει μια τοπολογία mesh και την ικανότητα του να αυτοδιοργανώνεται. Συγκεκριμένα οι κόμβοι αισθητήρων οργανώνονται γύρω από τους δρομολογητές σε τοπολογία star, οι οποίοι στη συνέχεια σχηματίζονται σε δίκτυο mesh.

Οι δρομολογητές στοχεύουν να διευρύνουν το εύρος που καλύπτει το δίκτυο καθώς και να προσφέρουν ανθεκτικότητα σε αποτυχίες. Κατ' επέκταση αφού οι αισθητήρες μπορούν να επικοινωνήσουν με πολλαπλούς δρομολογητές το δίκτυο αυτοδιοργανώνεται αυτόματα σε περίπτωση αποτυχίας είτε κόμβου αισθητήρα είτε μονοπατιού όπως φαίνεται στην Εικόνα 22.



6.2.3 Πλαίσιο λογισμικού για το ασύρματο δίκτυο αισθητήρων

Σε αυτή την ενότητα θα δούμε αναλυτικά το framework που χρησιμοποιείται από το ασύρματο δίκτυο αισθητήρων. Η εγκατάσταση του ασύρματου δικτύου αισθητήρων αποτελείται από 3 ξεχωριστά επίπεδα λογισμικού όπως φαίνεται στην Εικόνα 23.



Εικόνα 23: Software framework για το ασύρματο δίκτυο αισθητήρων

Mote Tier/XMesh:

Το επίπεδο Mote, όπου τρέχει το XMesh [12], είναι ένα λογισμικό που τρέχει πάνω από το ασύρματο δίκτυο αισθητήρων σχηματίζοντας ένα mesh δίκτυο. Το λογισμικό XMesh παρέχει τους δικτυακούς αλγορίθμους που απαιτούνται για να σχηματιστεί μια αξιόπιστη επικοινωνία που θα συνδέει όλους τους κόμβους αισθητήρων με τον εξυπηρετητή. Έχει αναπτυχθεί από την εταιρεία Crossbow και τρέχει πάνω από το λειτουργικό σύστημα του TinyOS που είναι εγκατεστημένο στους αισθητήρες Iris.

Το XMesh είναι ένα πλήρες multi-hop, ad-hoc δικτυακό πρωτόκολλο mesh που όπως αναφέραμε πιο πάνω αναπτύχθηκε από την Crossbow, ειδικά για ασύρματα δίκτυα αισθητήρων.

Ένα XMesh δίκτυο αποτελείται από κόμβους (Motes) που επικοινωνούν ασύρματα μεταξύ τους και είναι ικανά να μεταπηδούν ράδιο-μηνύματα σε ένα κεντρικό σταθμό (base station) όπου και μεταφέρονται σε ένα ηλεκτρονικό υπολογιστή. Η μεταπήδηση επεκτείνει την απόσταση της ασύρματης επικοινωνίας αποδοτικά και μειώνει την ενέργεια που απαιτείται για να αποστέλλονται τα δεδομένα των αισθητήρων. Με την μεταπήδηση των δεδομένων με αυτό τον τρόπο το XMesh προσφέρει 2 σημαντικά οφέλη: βελτιώνει την κάλυψη και την αξιοπιστία. Ένα μήνυμα μπορεί να παραδοθεί σε 1 ή περισσότερους κόμβους ενδιάμεσα που θα δρομολογήσουν τα δεδομένα, οπότε 2 κόμβοι δεν χρειάζεται να είναι σε κατευθείαν επικοινωνία μεταξύ τους για να παραλάβουν ένα πακέτο. Με αυτό τον τρόπο, σε περίπτωση βλάβης κάποιου ενδιάμεσου αισθητήρα ή αδυναμία επικοινωνίας λόγω παρεμβολής ή άλλων εμποδίων, μπορούν να αντιμετωπιστούν με το να γίνει αναδρομολόγηση των πακέτων γύρω από την περιοχή με την κακή σύνδεση. Επίσης οι αισθητήρες είναι συνήθως σε sleep mode (95% του χρόνου) ώστε να έχουν μεγαλύτερη διάρκεια ζωής.

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

- Τους κόμβους αισθητήρων
- Τον βασικό σταθμό. Όπως είδαμε στην ενότητα 0 ο αισθητήρας αυτός είναι ακόμη ένας αισθητήρας όπως τους υπόλοιπους με τη διαφορά ότι ενώνεται με ένα Crossbow MIB520 interface board που καταλήγει ένα ηλεκτρονικό υπολογιστή. Επίσης ο αισθητήρας αυτός προγραμματίζεται έτσι ώστε να διαχειρίζεται το δίκτυο προωθώντας και λαμβάνοντας πακέτα από αυτό.
- Ένα υπολογιστή που λαμβάνει τα δεδομένα ή αποστέλλει εντολές προς το δίκτυο.

Το XMesh διαθέτει πολλές λειτουργίες και πλεονεκτήματα όπως:

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

κόμβοι μέσα στο XMesh δίκτυο είναι αυτό διοργανωμένοι και αυτό ρυθμισμένοι σε περίπτωση βλάβης.

- *Πολλαπλές Υπηρεσίες Μεταφοράς*: Το XMesh προσφέρει πολλαπλές υπηρεσίες επικοινωνίας μεταξύ των κόμβων αισθητήρων οι οποίες είναι:
 - ο Upstream: Προώθηση πακέτων από ένα κόμβο αισθητήρα στο base station.
 - ο Downstream: Προώθηση πακέτων από το base station σε ένα ή περισσότερους κόμβους αισθητήρων.
 - ο Single Hop: Προώθηση πακέτων σε γειτονικούς κόμβους μόνο.
- *Πολλαπλά είδη Ποιότητας Υπηρεσιών (QoS)*: Το XMesh διαθέτει πολλά είδη ποιότητας υπηρεσιών τα οποία είναι:
 - ο Best Effort: Link level acknowledgement όπου οι κόμβοι θα δοκιμάσουν πολλαπλές φορές να στείλουν ένα μήνυμα στον γειτονικό τους κόμβο.
 - ο Guaranteed Delivery: Επιτρέπει end-to-end acknowledgement όταν το μήνυμα αποστέλλεται μέσω του πλέγματος στον βασικό σταθμό ή αντίστροφα και ένα acknowledge message τότε στέλλεται πίσω στον αποστολέα.
- *Πολλαπλά είδη Ενέργειας*: Το XMesh μπορεί να ρυθμιστεί για να τρέχει ένα από τα 2 είδη power mode τα οποία είναι:
 - ο High Power (HP): Το HP mode έχει τα εξής χαρακτηριστικά:
 - Δυνατότητα TrueMesh.
 - Κάθε κόμβος στο δίκτυο μπορεί να δρομολογήσει δεδομένα.
 - Ψηλό Bandwidth, χαμηλή καθυστέρηση (πλήρης χρήση καναλιού).
 - Οι αντένες των κόμβων είναι πάντοτε ενεργοποιημένες.
 - ο Low Power (LP): Το LP mode έχει τα εξής χαρακτηριστικά:
 - Δυνατότητα TrueMesh.
 - Κάθε κόμβος στο δίκτυο μπορεί να δρομολογήσει δεδομένα.
 - Χαμηλό Bandwidth, ψηλή καθυστέρηση (ιδανικό για εφαρμογές με χαμηλό data rate).
 - Οι αντένες των κόμβων είναι κανονικά σε sleep mode με μικρή κατανάλωση ενέργειας και ξυπνούν περιοδικά για να ελέγξουν για διακίνηση δεδομένων.

ο Extended Low Power (ELP): Το ELP mode έχει τα εξής χαρακτηριστικά:

- Χρησιμοποιείται μόνο για τους τελικούς κόμβους του δικτύου.
 - Οι κόμβοι αυτοί δεν δρομολογούν δεδομένα.
 - Χρησιμοποιεί υβριδική star mesh ρύθμιση.
- *Διαγνωστικά Υγείας:* Μέσα στο δίκτυο XMesh, οι κόμβοι μπορούν αυτόματα να αποστέλλουν πληροφορίες σχετικά με την υγεία τους στο base station. Τα δεδομένα υγείας αφορούν στο πόσο καλά ένας κόμβος λειτουργεί στο δίκτυο όσο αφορά την αποστολή δεδομένων, τα αποθέματα μπαταρίας και το radio signal strength indicator (RSSI) του πατέρα τους.
 - *Συγχρονισμός Χρόνου:* Στο Low Power (LP) mode υποστηρίζεται ένας ενιαίος συγχρονισμός χρόνου στο ± 1 msec. Η χρονοσήμανση χρησιμοποιείται για να συγχρονίσει τα μηνύματα αλλά είναι διαθέσιμο και στους χρήστες για να συγχρονίσουν τις μετρήσεις των αισθητήρων.
 - *Προγραμματισμός εξ αέρος (ασύρματα):* Το XMesh υποστηρίζει προγραμματισμό εξ αέρος (ασύρματα), επιτρέποντας στους χρήστες να επαναπρογραμματίζουν όλους τους κόμβους στο πλέγμα με νέο κώδικα.

XSensor:

Οι εφαρμογές XSensor είναι εφαρμογές δοκιμής των αισθητήρων και των boards παραλαβής δεδομένων της Crossbow. Επιτρέπουν στο χρήστη να δοκιμάζει εύκολα και γρήγορα τους αισθητήρες όταν εγκατασταθούν στο IRIS mote.

Server Tier/XServe:

Το Server Tier είναι μια συνεχής διεργασία που χειρίζεται την μετάφραση και την εναποθήκευση των δεδομένων που προέρχονται από το ασύρματο δίκτυο αισθητήρων. Παρέχει την γεφύρωση μεταξύ των ασύρματων κόμβων και των διαδικτυακών πελατών. Το XServe [13] είναι εφαρμογή στο Server Tier που τρέχει σε υπολογιστή και παίζει τον ρόλο του gateway μεταξύ του πλέγματος του ασύρματου δικτύου και των εφαρμογών που αλληλεπιδρούν με το πλέγμα.

Στον πυρήνα του το XServe παρέχει υπηρεσίες που δρομολογούν πακέτο από και προς το mesh δίκτυο με υψηλού επιπέδου υπηρεσίες για να αναλύσει, μετατρέψει και να επεξεργαστεί τα δεδομένα καθώς μεταφέρονται ανάμεσα στο πλέγμα και τις

εφαρμογές. Οι υψηλές επιπέδου υπηρεσίες μπορούν να ρυθμιστούν χρησιμοποιώντας XML σύνταξη.

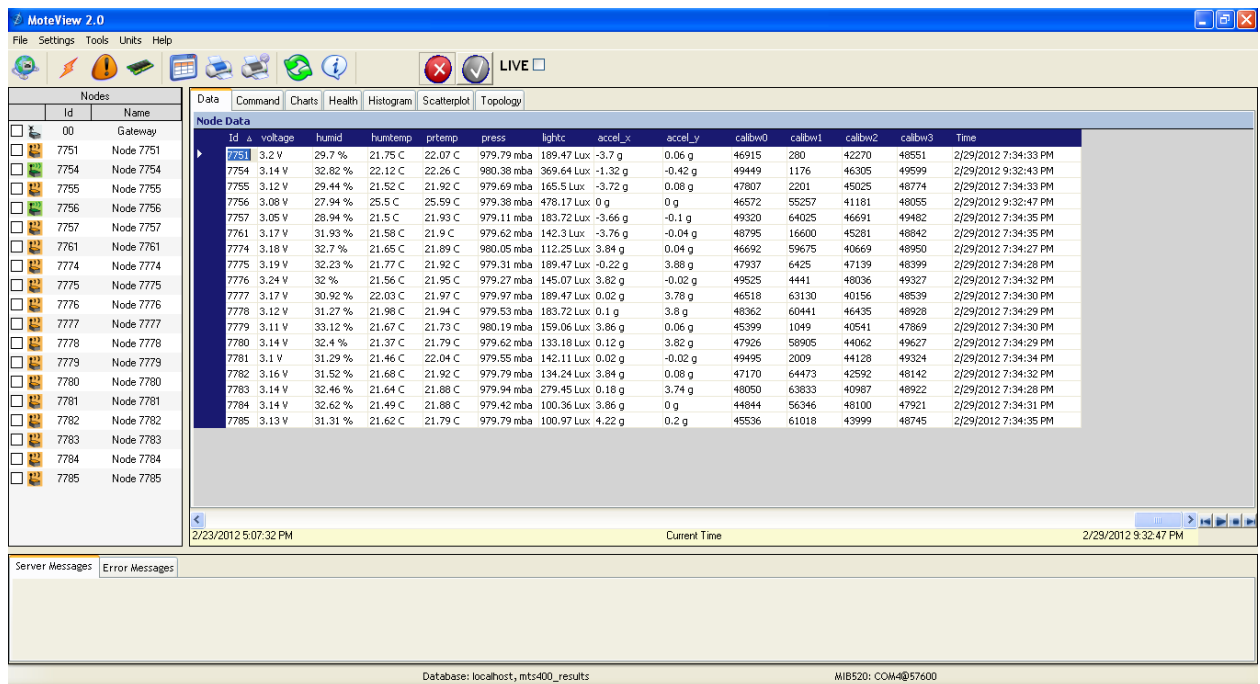
Το XServe μπορεί να ενωθεί με το ασύρματο δίκτυο αισθητήρων που χρησιμοποιούν το πρωτόκολλο XMesh μέσω του base station στο οποίο τρέχει το XMeshBase που προσφέρει η Crossbow. Παρέχει διαφορετικά επίπεδα υπηρεσιών ανάλογα με το πώς έχει ρυθμιστεί και εγκατασταθεί:

- *Serial Forwarder*: Στον πυρήνα του το XServe παρέχει τις βασικές υπηρεσίες δρομολόγησης στο δίκτυο αισθητήρων. Σαν *Serial Forwarder* το XServe επιτρέπει στις εφαρμογές να επικοινωνούν κατευθείαν με την εφαρμογή. Οι εφαρμογές στέλλουν και λαμβάνουν ακατέργαστα δεδομένα κατευθείαν από το δίκτυο αισθητήρων χωρίς καμία υψηλού επιπέδου μετατροπή ή επεξεργασία.
- *Application Server*: Το XServe μπορεί να λειτουργήσει σαν ένας application server παρέχοντας υψηλού επιπέδου υπηρεσίες για ανάλυση και επεξεργασία των δεδομένων κατά τον χρόνο εκτέλεσης. Οι χρήστες μπορούν να αναλύσουν και να μετατρέψουν τα δεδομένα του Mote tier που είναι σε ακατέργαστη μορφή σε παγκόσμιες μονάδες μέτρησης. Το XServe παρέχει προ-εγκατεστημένες λειτουργίες επεξεργασίας για να τυπώνει στην οθόνη τα δεδομένα, να τα γράφει σε αρχείο, να τα αποθηκεύει σε βάση δεδομένων ή να τα εξάγει σαν XML stream. Εμείς έχουμε χρησιμοποιήσει το XServe για την καταγραφή των δεδομένων σε βάση.

Client Tier:

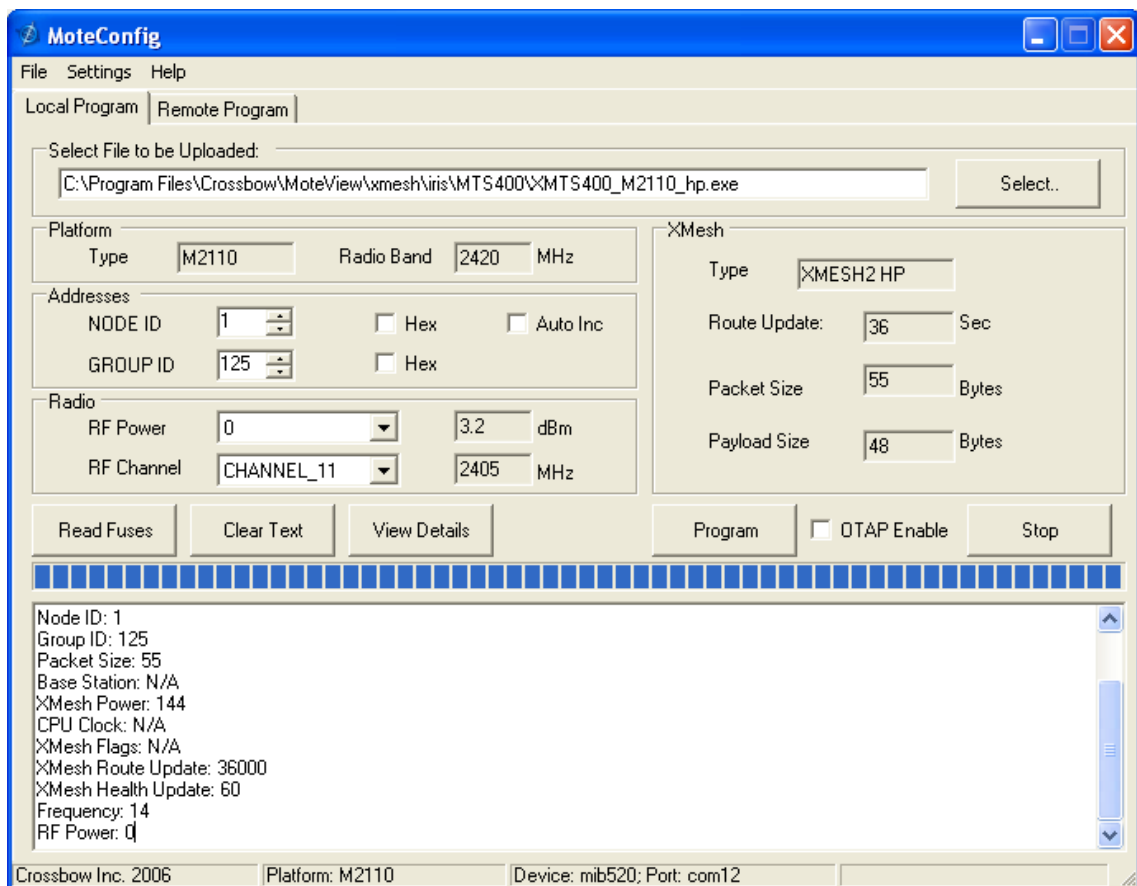
Το Client Tier παρέχει το λογισμικό της διεπιφάνειας του χρήστη και την γραφική διαπροσωπεία για τον έλεγχο του δικτύου. Η Crossbow παρέχει το λογισμικό MoteView [14] που δένει τα λογισμικά όλων των επιπέδων. Μέσω του λογισμικού MoteView γίνεται εύκολα η παρακολούθηση και ο έλεγχος των sensor, η σύνδεση στη βάση δεδομένων, η ανάλυση και η παρουσίαση των μετρήσεων των αισθητήρων. Η διαπροσωπεία του MoteView φαίνεται στην Εικόνα 24.

Έχουμε χρησιμοποιήσει το MoteView ως μια αναφορά και ένα εργαλείο μάθησης για το πώς θα μπορούσαμε να υλοποιήσουμε κάποια κομμάτια του συστήματος FireWatch.



Εικόνα 24: Διαπροσωπεία MoteView

Επιπλέον το MoteView έχει εγκατεστημένο ακόμα ένα λογισμικό – το MoteConfig. Το MoteConfig είναι ένα εργαλείο που τρέχει πάνω σε Windows OS και επιτρέπει τον προγραμματισμό των IRIS motes. Παρέχει την διαπροσωπεία που φαίνεται στην Εικόνα 25 για την ρύθμιση και την εγκατάσταση pre-compiled XMesh/TinyOS λογισμικά πάνω στους αισθητήρες. Το MoteConfig επιτρέπει στους χρήστες να καθορίσουν το Mote ID, το Group ID, το RF κανάλι και τη RF ισχύ.



Εικόνα 25: Διαπροσωπεία MoteConfig

6.2.4 Δομή πακέτου αισθητήρων

Σε αυτή την ενότητα θα δούμε αναλυτικά την όλη δομή του πακέτου που αποστέλλεται από κάθε αισθητήρα μέχρι να καταλήξει στο βασικό σταθμό.

Τα πακέτα που παράγονται από το IRIS mote με χρήση του MTS400 sensor board έχει την εξής δομή που φαίνεται στον πιο κάτω πίνακα.

Bytes: 5	0/7	4	2	2	2	20	2
TinyOS Header	XMesh Header	XSensor Header	Voltage	Humidity	Temperature	...	CRC
MTS400 Payload							

Πίνακας 7: Δομή Πακέτου MTS400

Στη συνέχεια θα δούμε την κάθε επικεφαλίδα του πακέτου (header), καθώς και το payload του MTS400.

Επικεφαλίδες πακέτου TinyOS:

Όπως έχουμε προαναφέρει οι αισθητήρες τρέχουν το λειτουργικό σύστημα TinyOS. Κατά την αποστολή δεδομένων υπάρχουν headers που καθορίζονται από το TinyOS. Η δομή της επικεφαλίδας έχει την δομή που φαίνεται στον πιο κάτω πίνακα.

Bytes: 2	1	1	1	Variable
Destination Address	AM Type	AM Group	Length	Data Payload
TinyOS Header				

Πίνακας 8: Δομή Επικεφαλίδας TinyOS

Τα πεδία του TinyOS πακέτου περιγράφονται στον πιο κάτω πίνακα.

Τύπος	Όνομα Πεδίου	Περιγραφή
uint16_t	addr	Single hop destination address
uint8_t	type	Active message type
uint8_t	group	Active message group ID
uint8_t	length	Length of entire message

Πίνακας 9: Πεδία πακέτου TinyOS

Επικεφαλίδες πακέτου XMesh:

Όπως έχουμε προαναφέρει ελέγχονται μέσω του πρωτοκόλλου XMesh που αναλύθηκε στην ενότητα 0. Κατά την αποστολή δεδομένων υπάρχουν headers που καθορίζονται από το XMesh. Η δομή της επικεφαλίδας έχει την δομή που φαίνεται στον πιο κάτω πίνακα.

Bytes: 5	2	2	2	1	Variable
TinyOS Header	Source Address	Origin Address	Sequence Number	Application ID	Data Payload
	XMesh Header				

Πίνακας 10: Δομή Επικεφαλίδας XMesh

Τα πεδία του XMesh πακέτου περιγράφονται στον πιο κάτω πίνακα.

Τύπος	Όνομα Πεδίου	Περιγραφή
uint16_t	sourceaddr	Single hop sender address
uint16_t	originaddr	Node ID of originator of message
int16_t	seqno	Sequence number for link estimation
uint8_t	socket	Application ID

Πίνακας 11: Πεδία πακέτου XMesh

Επικεφαλίδες πακέτου XSensor:

Κατά την αποστολή δεδομένων υπάρχουν headers που καθορίζονται από την εφαρμογή XSensor. Η δομή της επικεφαλίδας έχει την δομή που φαίνεται στον πιο κάτω πίνακα.

Bytes: 5	0/7	1	1	2	Variable
TinyOS Header	XMesh Header	Sensor Board ID	Sensor Packet ID	Parent	Data Payload
		XSensor Header			

Πίνακας 12: Δομή Επικεφαλίδας XSensor

Τα πεδία του XSensor πακέτου περιγράφονται στον πιο κάτω πίνακα.

Τύπος	Όνομα Πεδίου	Περιγραφή
uint16_t	boardid	0x85 (hex), 133 (dec) σταθερό για τον MTS400
uint16_t	packetid	ID of packet
uint16_t	parent	ID of parent sensor

Πίνακας 13: Πεδία πακέτου XSensor

Φορτίο πακέτου MTS400:

Όπως έχουμε προαναφέρει οι αισθητήρες έχουν εγκατεστημένη το MTS400 sensor board για την καταγραφή των περιβαλλοντικών μετρήσεων. Κατά την αποστολή δεδομένων το MTS400 θέτει τις πληροφορίες που είναι σχετικές με τις μετρήσεις αυτές. Η δομή του φορτίου έχει την δομή που φαίνεται στον πιο κάτω πίνακα.

Τύπος	Όνομα Πεδίου	Περιγραφή
uint16_t	voltage	Τιμή Μπαταρίας (γνωστή ως vref ή voltage)
uint16_t	humidity	Τιμή υγρασίας
uint16_t	temp	Τιμή θερμοκρασίας
uint16_t	cal_word1	Calibration word 1 πίεσης
uint16_t	cal_word2	Calibration word 2 πίεσης
uint16_t	cal_word3	Calibration word 3 πίεσης
uint16_t	cal_word4	Calibration word 4 πίεσης
uint16_t	intersemaptemp	Τιμή βαρομετρικής πίεσης
uint16_t	intersemapressure	Τιμή βαρομετρικής θερμοκρασίας
uint16_t	taosch0	Τιμή φωτισμού (κανάλι 1)
uint16_t	taosch1	Τιμή φωτισμού (κανάλι 2)
uint16_t	accel-x	Τιμή επιταχυνσιόμετρου στον άξονα x
uint16_t	accel-y	Τιμή επιταχυνσιόμετρου στον άξονα y

Πίνακας 14: Δομή Φορτίου MTS400

CRC:

Το πακέτο τελειώνει με το πεδίο CRC που υπολογίζεται με βάση ολόκληρο το πακέτο, εκτός από το header του και το πεδίο CRC. Ο υπολογισμός του γίνεται κάνοντας XOR όλα τα παρόν bytes και μετά shifted με τον CRC accumulator. Το CRC αποτελείται πάντα από 2 bytes. Στο Παράρτημα Α παρουσιάζεται ο τρόπος που υπολογίζεται το CRC.

6.2.5 Μετασχηματισμός ακατέργαστων μετρήσεων αισθητήρων σε παγκόσμιες μονάδες μέτρησης

Τα πακέτα μετρήσεων που λαμβάνονται από το base station δεν είναι στη μορφή που απαιτεί το σύστημα του FireWatch, αλλά σε μια ακατέργαστη μορφή. Αυτό που πρέπει να γίνει για να έχουν αξία τα δεδομένα είναι να χρησιμοποιηθούν μέθοδοι οι οποίοι θα μετατρέπουν τα ακατέργαστα δεδομένα σε πραγματικές τιμές.

Όπως προδιαγράφηκε στο τμήμα των απαιτήσεων οι μετρήσεις πρέπει να μετασχηματίζονται προτού παρουσιάζονται στο χρήστη έτσι ώστε να παίρνουν μορφές που να μπορούν να χρησιμοποιηθούν αργότερα όχι μόνο για την ανίχνευση και πρόβλεψη πυρκαγιών αλλά και για να είναι κατανοητές προς τους χρήστες.

Κάθε μικροαισθητήρας στο sensor board του IRIS Mote, που περιγράφηκε στην ενότητα 0, έχει την δική του μαθηματική εξίσωση για την μετατροπή των ακατέργαστων δεδομένων που λαμβάνει, σε υψηλής ποιότητας μονάδες μέτρησης. Εν συνεχεία, θα δούμε αναλυτικά ποιες είναι αυτές οι μετατροπές.

Μετατροπή ακατέργαστης μέτρησης μπαταρίας σε milliVolts (mV):

Η πιο κάτω εξίσωση μετατρέπει από την ακατέργαστη μέτρηση της μπαταρίας ADC σε μονάδες μέτρησης mV:

Εξίσωση:

$$voltage = RV \times ADC_FS/ADCVref$$

Όπου:

voltage = Voltage μπαταρίας

ADC_FS = 1023

RV = 1.223 volts - Voltage Reference από τον IRIS/mica2

ADCVref = ακατέργαστη τιμή δεδομένων adc από το κανάλι 1

Τελική Εξίσωση:

$$voltage (volts) = 1252.352/data \text{ voltage (V)} = 1252352/data (mV)$$

Μετατροπή ακατέργαστης μέτρησης θερμοκρασίας από το μικροαισθητήρα Sensirion SHT11 σε βαθμούς κελσίου (°C):

Η πιο κάτω εξίσωση μετατρέπει από την ακατέργαστη μέτρηση της θερμοκρασίας ADC σε μονάδες μέτρησης °C:

Εξίσωση:

$$humtemp = -38.4 + 0.0098 * TempData \text{ } ^\circ C$$

Όπου:

humtemp = Θερμοκρασία σε °C

TempData = ακατέργαστη τιμή δεδομένων θερμοκρασίας όπως λαμβάνεται από τον Sensirion SHT11 (14 bit)

Μετατροπή ακατέργαστης μέτρησης σχετικής υγρασίας από το μικροαισθητήρα Sensirion SHT11 σε ποσοστό (%):

Η πιο κάτω εξίσωση μετατρέπει από την ακατέργαστη μέτρηση της υγρασίας ADC σε μονάδες μέτρησης %:

Εξίσωση χωρίς διόρθωση σε σχέση με την θερμοκρασία:

$$humid1 = -4.0 + 0.0405 * HumData - 0.0000028 * HumData * HumData$$

Εξίσωση με διόρθωση σε σχέση με την θερμοκρασία:

$$fTemp = -38.4 + 0.0098 * (float)TempData$$

$$humid = (fTemp - 25.0) * (0.01 + 0.00008 * HumData) + humid1$$

Όπου:

humid1 = Υγρασία σε % χωρίς διόρθωση σε σχέση με την θερμοκρασία

HumData = ακατέργαστη τιμή δεδομένων υγρασίας όπως λαμβάνεται από τον Sensirion SHT11 (12 bits)

fTemp = Θερμοκρασία σε °C

TempData = ακατέργαστη τιμή δεδομένων θερμοκρασίας όπως λαμβάνεται από τον Sensirion SHT11 (14 bits)

humid = Υγρασία σε % με διόρθωση σε σχέση με την θερμοκρασία

Το σύστημα του FireWatch χρησιμοποιεί τον 2^ο τύπο που υπολογίζει την σχετική υγρασία σε σχέση με την θερμοκρασία.

Μετατροπή ακατέργαστης μέτρησης βαρομετρικής πίεσης από το μικροαισθητήρα Intersema MS5534A σε millibar (mbar):

Η πιο κάτω εξίσωση μετατρέπει από την ακατέργαστη μέτρηση της βαρομετρικής πίεσης ADC σε μονάδες μέτρησης mbar:

Εξίσωση:

```

calibration[0] = cal_word1
calibration[1] = cal_word2
calibration[2] = cal_word3
calibration[3] = cal_word4
C1 = calibration [0] >>1
C2 = ((calibration[2] & 0x3f) <<6) | (calibration & 0x3f)
C3 = calibration[3] >>6
C4 = calibration[2] >>6
C5 = ((calibration[0]&1) <<10) | (calibration[1] >>6)
//C6 = calibration[1] & 0x3f
UT1=8* (float) C5+20224
dT = (float)TempData-UT1
OFF = (float)C2*4+((float)C3*dT)/1024 + 24576
SENS = (SENS*((float)PressureData-7168.0))/16384 -OFF
press = X/32.0 + 250.0

```

Όπου:

C1 - C6 = Calibration Coefficients τα οποία εξάγονται από τις εξής τέσσερις 16 bit words cal_word1, cal_word2, cal_word3, cal_word4 από το πακέτο που λαμβάνεται από τον αισθητήρα

calibration[0] - calibration[3] = words cal_word1, cal_word2, cal_word3, cal_word4 από το πακέτο που λαμβάνεται από τον αισθητήρα αντίστοιχα.

TempData = ακατέργαστη τιμή δεδομένων βαρομετρικής θερμοκρασίας όπως λαμβάνεται από τον Intersema MS5534A

PressureData = ακατέργαστη τιμή δεδομένων βαρομετρικής πίεσης όπως λαμβάνεται από τον Intersema MS5534A

press = Βαρομετρική πίεση σε mbar

Μετατροπή ακατέργαστης μέτρησης βαρομετρικής θερμοκρασίας από το μικροαισθητήρα Intersema MS5534A σε βαθμούς κελσίου (°C):

Η πιο κάτω εξίσωση μετατρέπει από την ακατέργαστη μέτρηση της βαρομετρικής θερμοκρασίας ADC σε μονάδες μέτρησης °C:

Εξίσωση:

calibration[0] = cal_word1

calibration[1] = cal_word2

C5 = ((calibration[0] & 1) << 10) | (calibration[1] >> 6);

C6 = calibration[1] & 0x3f

UT1=8(float)C5+20224*

dT = (float)TempData-UT1

prtemp = 200.0 + dT((float)C6+50.0)/1024.0*

prtemp /=10.0

Όπου:

C1 - C6 = Calibration Coefficients τα οποία εξάγονται από τις εξής τέσσερις 16 bit words *cal_word1*, *cal_word2*, *cal_word3*, *cal_word4* από το πακέτο που λαμβάνεται από τον αισθητήρα

calibration[0] - calibration[3] = words *cal_word1*, *cal_word2*, *cal_word3*, *cal_word4* από το πακέτο που λαμβάνεται από τον αισθητήρα αντίστοιχα.

TempData = ακατέργαστη τιμή δεδομένων βαρομετρικής θερμοκρασίας όπως λαμβάνεται από τον Intersema MS5534A

prtemp = Βαρομετρική Θερμοκρασία σε °C

Μετατροπή ακατέργαστης μέτρησης φωτός από το μικροαισθητήρα Taos TSL2550 σε φωτεινότητα (lux):

Η πιο κάτω εξισώσεις μετατρέπουν από την ακατέργαστη μέτρηση του φωτός ADC σε μονάδες μέτρησης lux:

Εξίσωση:

ChData = taosch0 & 0x00ff

if (ChData == 0xff) Invalid Value // Taos Ch0 data: OVERFLOW

ST1 = ChData & 0xf

CH1 = (ChData & 0x70) >> 4

$CV1 = 1 \ll CH1$

$CNT1 = (int)(16.5*(CV1-1)) + ST1*CV1$

$ACNT0 = (int)CNT1$

$ChData = taosch1 \& 0xff$

if (ChData == 0xff) Invalid Value // Taos Ch1 data: OVERFLOW

$ST1 = ChData \& 0xf$

$CH1 = (ChData \& 0x70) \gg 4$

$CV1 = 1 \ll CH1$

$CNT1 = (int)(16.5*(CV1-1)) + ST1*CV1$

$ACNT1 = (int)CNT1$

$R = ((float)ACNT1)/((float)ACNT0)$

$light = (float)ACNT0*0.46/exp(3.13*R)$

Όπου:

taosch0, taosch1 = δεδομένα που εξάγονται από το πακέτο που λαμβάνεται από τον αισθητήρα

light = Επίπεδα φωτεινότητας σε Lux

Μετατροπή ακατέργαστης μέτρησης επιταχυνσιόμετρου από το μικροαισθητήρα ADXL202JE σε milli gram (mg):

Η πιο κάτω εξίσωση μετατρέπει από την ακατέργαστη μέτρηση του επιταχυνσιόμετρου ADC σε μονάδες μέτρησης mg:

Εξίσωση:

$minus_one_calibration = 400;$

$plus_one_calibration = 500;$

$scale_factor = (plus_one_calibration - minus_one_calibration) / 2;$

$accel = 1.0 - (plus_one_calibration - AccelData) / scale_factor;$

$accel = accel * 1000.0;$

Όπου:

AccelData = δεδομένα που εξάγονται από το πακέτο που λαμβάνεται από τον αισθητήρα

accel = Τιμή Επιταχυνσιόμετρου σε mg

6.2.6 Βάση Δεδομένων Ασύρματου Δικτύου Αισθητήρων

Τα δεδομένα που καταγράφουν οι αισθητήρες και συγκεντρώνονται στο βασικό σταθμό (Base station) θα πρέπει να αποθηκεύονται κάπου για να είναι προσβάσιμα από τα άλλα υποσυστήματα του FireWatch.

Οι αισθητήρες της MEMSIC έρχονται με προκαθορισμένη Βάση Δεδομένων με σκοπό να αποθηκεύονται μόνιμα τα δεδομένα με τη χρήση του XServe όπως περιγράψαμε στην ενότητα 0.

Κατά το installation των λογισμικών της Crossbow, γίνεται εγκατάσταση και της βάσης δεδομένων PostgreSQL. Η PostgreSQL είναι μια προχωρημένη σχεσιακή βάση δεδομένων που δίνει την δυνατότητα στους χρήστες να επικοινωνούν κατευθείαν με τους πίνακες.

Στη βάση αυτή θα αποθηκεύονται οι μετρήσεις των αισθητήρων και μόνο, ώστε να κρατήσουμε όσο πιο απλό γίνεται το συγκεκριμένο υποσύστημα. Συγκεκριμένα θα πρέπει να αποθηκεύει τις περιβαλλοντικές μετρήσεις και τις μετρήσεις που αφορούν την υγεία τους.

Πίνακες Βάσης Δεδομένων:

Με την εγκατάσταση του λογισμικού της Crossbow, γίνεται όπως είπαμε αυτόματη εγκατάσταση και της σχεσιακής ΒΔ PostgreSQL. Επίσης αυτόματα κατασκευάζονται και οι πίνακες που χρειάζονται για την αποθήκευση των μετρήσεων των αισθητήρων.

Οι πίνακες που αφορούν τους αισθητήρες έχουν αντίστοιχο όνομα με το είδος του sensor board που χρησιμοποιούμε, δηλαδή MTS400. Το όνομα του πίνακα στη ΒΔ έχει το όνομα *mts400_results*. Ο πιο κάτω πίνακας παρουσιάζει σχήμα του συγκεκριμένου πίνακα.

Column	Type	Bytes	Description
result_time	timestamp without time zone	8	Χρόνος που λήφθηκε το πακέτο
nodeid	integer	4	Μοναδικός αριθμός αισθητήρα
parent	integer	4	Πατρικός μοναδικός αριθμός κόμβου αισθητήρα
voltage	integer	4	Τιμή τάσης
humid	integer	4	Τιμή υγρασίας (Sensirion SHT11)
humtemp	integer	4	Τιμή θερμοκρασίας (Sensirion SHT11)
prtemp	integer	4	Τιμή βαρομετρικής θερμοκρασίας (Intersema MS55ER)
press	integer	4	Τιμή βαρομετρικής πίεσης (Intersema MS55ER)
taosch0	integer	4	Τιμή έντασης φωτός (Taos TSL2550)
taosch1	integer	4	Τιμή έντασης φωτός (Taos TSL2550)
accel_x	integer	4	Τιμή επιταχυνσιόμετρου στον άξονα x (ADXL202JE)
accel_y	integer	4	Τιμή επιταχυνσιόμετρου στον άξονα y (ADXL202JE)
cal_word1	integer	4	Calibration word 1 πίεσης (Intersema MS55ER)
cal_word2	integer	4	Calibration word 2 πίεσης (Intersema MS55ER)
cal_word3	integer	4	Calibration word 3 πίεσης (Intersema MS55ER)
cal_word4	integer	4	Calibration word 4 πίεσης (Intersema MS55ER)

Πίνακας 15: Πίνακας mts400_results

Ο πίνακας που αφορά την υγεία του δικτύου ονομάζεται *node_health* και το σχήμα του φαίνεται στον πιο κάτω πίνακα .

Column	Type	Bytes	Description
result_time	timestamp without time zone	8	Χρόνος που λήφθηκε το πακέτο
nodeid	integer	4	Μοναδικός αριθμός αισθητήρα
parent	integer	4	Πατρικός μοναδικός αριθμός κόμβου αισθητήρα
health_pkts	integer	4	Συνολικά πακέτα υγείας
node_pkts	integer	4	
forwarded	integer	4	Προωθημένα πακέτα
dropped	integer	4	Χαμένα πακέτα
retries	integer	4	Προσπάθειες αποστολής πακέτου
battery	integer	4	Υπόλοιπο μπαταρίας
power_sum	integer	4	Πάντα 0. Για μελλοντική χρήση
board	integer	4	Αριθμός board
quality_tx	integer	4	Ποιότητα κόμβου-πατέρα, τιμές 0-15 αντιστοιχούν σε 0-100%
quality_rx	integer	4	Ποιότητα πατέρα- κόμβου, ίδια αντιστοιχία με πιο πάνω
path_cost	integer	4	Εκτίμηση του αριθμού των αναγκών μεταδόσεων για αποστολή πακέτου από τον κόμβο στο base κόμβο. $\text{Path Cost} = 4 * \text{Estimated Transmission Number (ETX)}$ $\text{Path Cost} = 2^{10} / (\text{quality_tx} * \text{quality_rx})$ Η πιο χαμηλή πιθανή τιμή είναι 4: απαιτείται τουλάχιστον μία μετάδοση για αποστολή πακέτου από το node στο base station.
parent_rssi	integer	4	Received Signal Strength Indicator πατρικού κόμβου

Πίνακας 16: Πίνακας node_health

Όπως φαίνεται πιο πάνω δεν μας παρέχεται η γεωγραφική τοποθεσία των αισθητήρων. Κατά την τοποθέτηση τους στην δασική περιοχή θα καταγράφεται με GPS ψηλής ανάλυσης η τοποθεσία που έχουν εγκατασταθεί. Οι τιμές αυτές θα καταγράφονται στη Βάση Δεδομένων του Συστήματος Συνεργασίας όπως θα δούμε πιο κάτω.

Λειτουργίες βάσης δεδομένων:

Οι βασικές λειτουργίες που αφορούν το σύστημα της βάσης δεδομένων του ασύρματου δικτύου αισθητήρων είναι η κατασκευή των stored procedures που θα επιτρέπουν στη βάση δεδομένων συνεργασίας να ανακτά τα δεδομένα που αποθηκεύονται.

Επίσης με βάση τους μετασχηματισμούς των ακατέργαστων δεδομένων σε παγκόσμιες μονάδες μέτρησης πρέπει να υλοποιηθούν τα functions τα οποία κάνουν τις συγκεκριμένες μετατροπές.

6.3 Σχεδίαση συστήματος γεωγραφικών δεδομένων

Στα πλαίσια του FireWatch συστήματος είναι εμπλεκόμενος φορέας η εταιρεία T.C. Geomatic η οποία ειδικεύεται στην παροχή υπηρεσιών γεωγραφικών δεδομένων.

Η ΒΔ της Geomatic's είναι προσβάσιμη από το σύστημα Συνεργασίας. Συγκεκριμένα η T.C. Geomatic παρέχει Διεπαφή Προγραμματισμού Εφαρμογών και επιτρέπει στους χρήστες να αλληλεπιδρούν με τα γεωγραφικά δεδομένα μέσω JavaScript. Το συγκεκριμένο API [15] που παρέχεται από την Geomatic βρίσκεται στο Παράρτημα Β. Η JavaScript είναι μια δυναμική γλώσσα προγραμματισμού και η πιο ευρέως χρησιμοποιούμενη στο χώρο των διαδικτυακών εφαρμογών. Επίσης υποστηρίζεται από όλους τους πλοηγούς όπως Internet Explorer, Firefox, Chrome, Opera, και Safari.

6.3.1 Λειτουργίες γεωγραφικού συστήματος δεδομένων

Το Γεωγραφικό Σύστημα Δεδομένων θα μας δίνει σημαντικά δεδομένα. Τα γεωγραφικά δεδομένα, που είναι αποθηκευμένα σε εξειδικευμένη βάση δεδομένων της T.C. Geomatic, παρέχουν σημαντικές πληροφορίες για την επιλεγμένη δασική περιοχή. Η ανάλυση του εδάφους και η μοντελοποίηση της βλάστησης σε συγκεκριμένες δασικές περιοχές θα δίνουν στο σύστημα δεδομένα τα οποία θα βοηθούν τους πυροσβέστες για την κατάσβεσή της.

Επίσης θα παρέχει στο Τμήμα Δασών πληροφορίες σχετικά με εναλλακτικά μονοπάτια που θα μπορούν να ακολουθηθούν από τους πυροσβέστες ώστε να μπορούν να προσεγγίσουν ακόμα και τις πιο δύσβατες περιοχές.

6.3.2 Επιμέρους συστατικά γεωγραφικού συστήματος δεδομένων

Το Σύστημα Γεωγραφικών Δεδομένων αποτελείται από 3 κύρια συστατικά. Αυτά είναι:

- Αντικείμενο Map (Χάρτη): Είναι το κύριο αντικείμενο του συστήματος γεωγραφικών δεδομένων. Χρησιμοποιείται σαν container επιτρέποντας την πρόσθεση επιπρόσθετων αντικειμένων που υποστηρίζονται από το Σύστημα Γεωγραφικών Δεδομένων όπως είναι τα 2 πιο κάτω.
- Αντικείμενο Layer (Επίπεδο): Τα επίπεδα επιτρέπουν την κατασκευή και παρουσίαση γεωγραφικών σχημάτων σε συγκεκριμένα σημεία του χάρτη που εμείς θέτουμε. Παράδειγμα των επιπέδων είναι η εναλλαγή παρουσίασης μιας περιοχής από δορυφορική όψη ή όψη οδικού δικτύου.

Στα πλαίσια του FireWatch θα προστεθούν 3 επιπλέον επίπεδα:

- ο Επίπεδο Off-road: Το επίπεδο αυτό θα παρουσιάζει δρομολόγια εκτός δρόμων που όπως είδαμε θα μπορούν να χρησιμοποιηθούν από τους πυροσβέστες για να μπορούν να φτάσουν γρήγορα και με ασφάλεια στο χώρο που επιθυμούν.
 - ο Επίπεδο Land-use: Το επίπεδο αυτό θα παρουσιάζει τον τύπο βλάστησης στην προστατευόμενη δασική περιοχή έτσι ώστε να ενημερώνει τους πυροσβέστες με επιπλέον περιβαλλοντικές παραμέτρους για να τις έχουν υπόψη τους π.χ. ξηρή βλάστηση.
 - ο Επίπεδο Terrain: Το επίπεδο αυτό θα παρουσιάζει την ανάλυση του εδάφους.
- Αντικείμενο POI: Τα αντικείμενα αυτού του τύπου επιτρέπουν την παρουσίαση διαφόρων σημείων πάνω σε συγκεκριμένες περιοχές του χάρτη. Το FireWatch θα χρησιμοποιήσει αυτά τα αντικείμενα για να τοποθετήσει τα σημεία όπου είναι εγκατεστημένοι οι αισθητήρες καθώς και άλλα σημεία που θα καθοριστούν από το Τμήμα Δασών, π.χ. Δασικοί Σταθμοί.

6.4 Σχεδίαση συστήματος συνεργασίας

Σε αυτή την ενότητα θα δούμε τη σχεδίαση του Συστήματος Συνεργασίας. Οι πληροφορίες που παρέχονται από τα πιο πάνω συστήματα (Σύστημα Ασύρματου

Δικτύου Αισθητήρων και Σύστημα Γεωγραφικών Δεδομένων), θα συνδυάζονται για να παρέχονται χρήσιμες πληροφορίες στους τελικούς χρήστες ώστε να μπορούν να αντιμετωπίσουν τις πυρκαγιές.

Σε περίπτωση ανίχνευσης ή πρόβλεψης πυρκαγιάς θα επιταχύνει τις διαδικασίες αντιμετώπισης της από τους διάφορους κυβερνητικούς φορείς με το να τους προειδοποιεί για τον πιθανό κίνδυνο. Με την παροχή των πληροφοριών από τα προηγούμενα 2 επίπεδα οι χρήστες θα μπορούν να επιβλέπουν τη προστατευμένη δασική περιοχή πιο εύκολα μέσω της διαδικτυακής εφαρμογής του FireWatch.

Στη συνέχεια παραθέτουμε τα συστατικά που αποτελούν το Σύστημα Συνεργασίας.

6.4.1 Βάση δεδομένων συστήματος συνεργασίας

Τα δεδομένα που καταγράφουν οι αισθητήρες και αποθηκεύονται στη ΒΔ των αισθητήρων πρέπει να είναι προσβάσιμα και στη Βάση Δεδομένων του Συστήματος Συνεργασίας. Επομένως χρειάζονται να κατασκευαστούν πίνακες που να κρατούν τις μετρήσεις των αισθητήρων με σχήμα όπως αυτό που παρουσιάστηκε στην ενότητα 6.2.6.

Στην ΒΔ του Συστήματος Συνεργασίας θα αποθηκεύονται επίσης οι επαφές του FireWatch καθώς και οι χρήστες που είναι εξουσιοδοτημένοι να χρησιμοποιούν το σύστημα. Ακόμη, τα παραγόμενα alerts του συστήματος μετά την ανίχνευση πιθανής πυρκαγιάς θα αποθηκεύονται στη ΒΔ.

Για τη ΒΔ μπορούμε να χρησιμοποιήσουμε οποιαδήποτε θέλουμε αρκεί να επιτρέπει τις λειτουργίες που περιγράφουμε πιο κάτω. Στην περίπτωση μας χρησιμοποιήσαμε SQL Server 2005.

Πίνακες βάσης δεδομένων:

Η ΒΔ σε SQL Server 2005 αφορά και μόνο το σύστημα συνεργασίας. Οι πίνακες κρατούν πληροφορίες για τους χρήστες του συστήματος, τις διάφορες επαφές επικοινωνίας, τα μηνύματα που αποστέλλονται, το κίνδυνο πυρκαγιάς και τις μετρήσεις των αισθητήρων σε συνεργασία με την ΒΔ του Ασύρματου Δικτύου Αισθητήρων. Πιο κάτω παρουσιάζονται αναλυτικά οι πιο σημαντικοί πίνακες.

Πίνακες που αφορούν τις επαφές του συστήματος:

Ο πιο κάτω πίνακας παρουσιάζει το σχήμα του πίνακα *CONTACTS* ο οποίος κρατάει τα στοιχεία κάθε εμπλεκόμενου (υπηρεσίες, πυροσβεστικοί σταθμοί κτλ).

Column	Type	Bytes	Description
CONTACT_ID	int	4	Μοναδικό ID επαφής
CONTACT_NAME	nvarchar(100)	202 max	Όνομα επαφής
TEL_WORK1	varchar(15)	32 max	Τηλέφωνο εργασίας 1
TEL_WORK2	varchar(15)	32 max	Τηλέφωνο εργασίας 2
TEL_HOUSE1	varchar(15)	32 max	Τηλέφωνο οικίας 1
TEL_HOUSE2	varchar(15)	32 max	Τηλέφωνο οικίας 2
TEL_MOBILE1	varchar(15)	32 max	Κινητό τηλέφωνο 1
TEL_MOBILE2	varchar(15)	32 max	Κινητό τηλέφωνο 2
FAX	varchar(15)	32 max	Φαξ
EMAIL	varchar(50)	62 max	Ηλεκτρονικό Ταχυδρομείο επαφής
CITY_ID	smallint	2	Πόλη Διαμονής (Foreign Key)
DISTRICT_ID	tinyint	1	Επαρχία Διαμονής (Foreign Key)
INACTIVE	bit	1-4	Δηλώνει αν η επαφή είναι ενεργή ή όχι
CONTACT_TYPE_ID	smallint	2	Τύπος Επαφής (Foreign Key)
SPECIAL_INFO	nvarchar(100)	202 max	Ειδικές πληροφορίες για την επαφή
TITLE	nvarchar(100)	202 max	Τίτλος Συγκεκριμένος για την επαφή

Πίνακας 17: Πίνακας **CONTACTS**

Ο πίνακας που αφορά τις πόλεις της Κύπρου, ονομάζεται *CITIES* κα το σχήμα του φαίνεται στον πιο κάτω πίνακα .

Column	Type	Bytes	Description
CITY_ID	smallint	2	Μοναδικό ID πόλης
DISTRICT_ID	tinyint	1	Επαρχία που ανήκει η πόλη (Foreign Key)
GREEK	nvarchar(50)	102 max	Όνομα πόλης στα ελληνικά
ENGLISH	nvarchar(50)	102 max	Όνομα πόλης στα αγγλικά

Πίνακας 18: Πίνακας **CITIES**

Ο πίνακας που αφορά τις επαρχίες της Κύπρου, ονομάζεται *DISTRICTS* και το σχήμα του φαίνεται στον πιο κάτω πίνακα.

Column	Type	Bytes	Description
DISTRICT_ID	tinyint	1	Μοναδικό ID επαρχίας
DISTRICT_NAME_GR	nvarchar(50)	102 max	Όνομα επαρχίας στα ελληνικά
DISTRICT_NAME_EN	nvarchar(50)	102 max	Όνομα επαρχίας στα αγγλικά

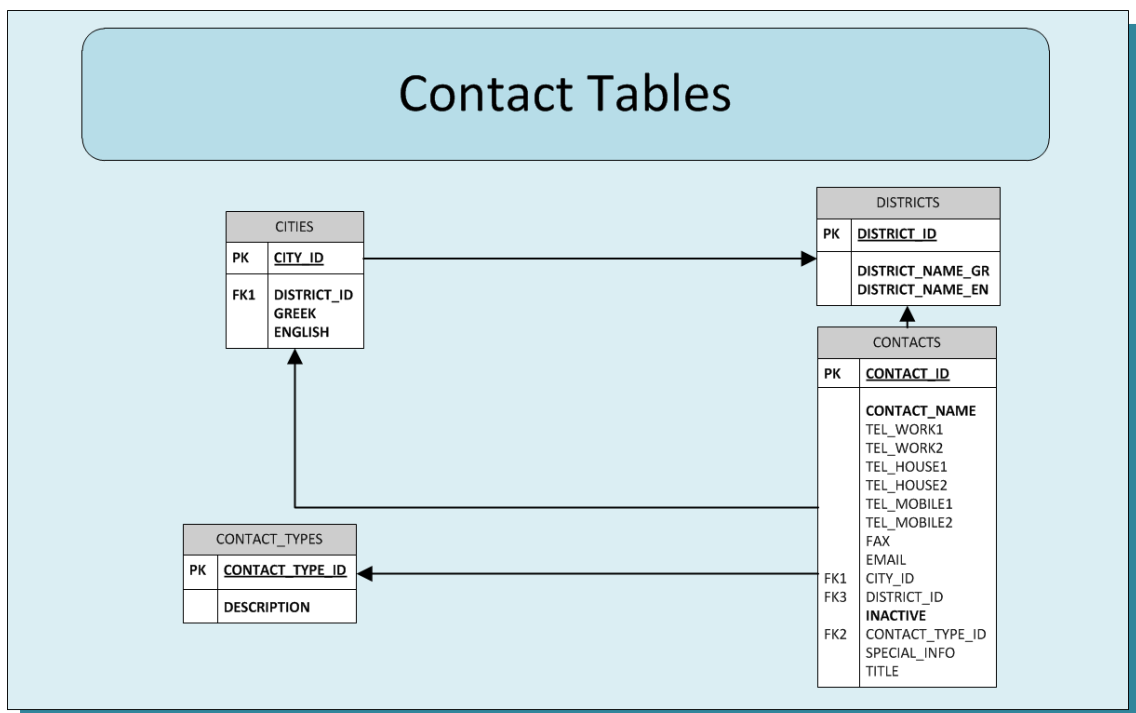
Πίνακας 19: Πίνακας DISTRICTS

Ο πίνακας που αφορά τους τύπους των επαφών, ονομάζεται *CONTACT_TYPES* και το σχήμα του φαίνεται στον πιο κάτω πίνακα.

Column	Type	Bytes	Description
CONTACT_TYPE_ID	smallint	2	Μοναδικό ID τύπου επαφής
DESCRIPTION	nvarchar(100)	202 max	Περιγραφή τύπου

Πίνακας 20: Πίνακας CONTACT_TYPES

Στην Εικόνα 26 παρουσιάζεται συνοπτικά μια σχηματική αναπαράσταση των πινάκων που αφορά τις επαφές του συστήματος.



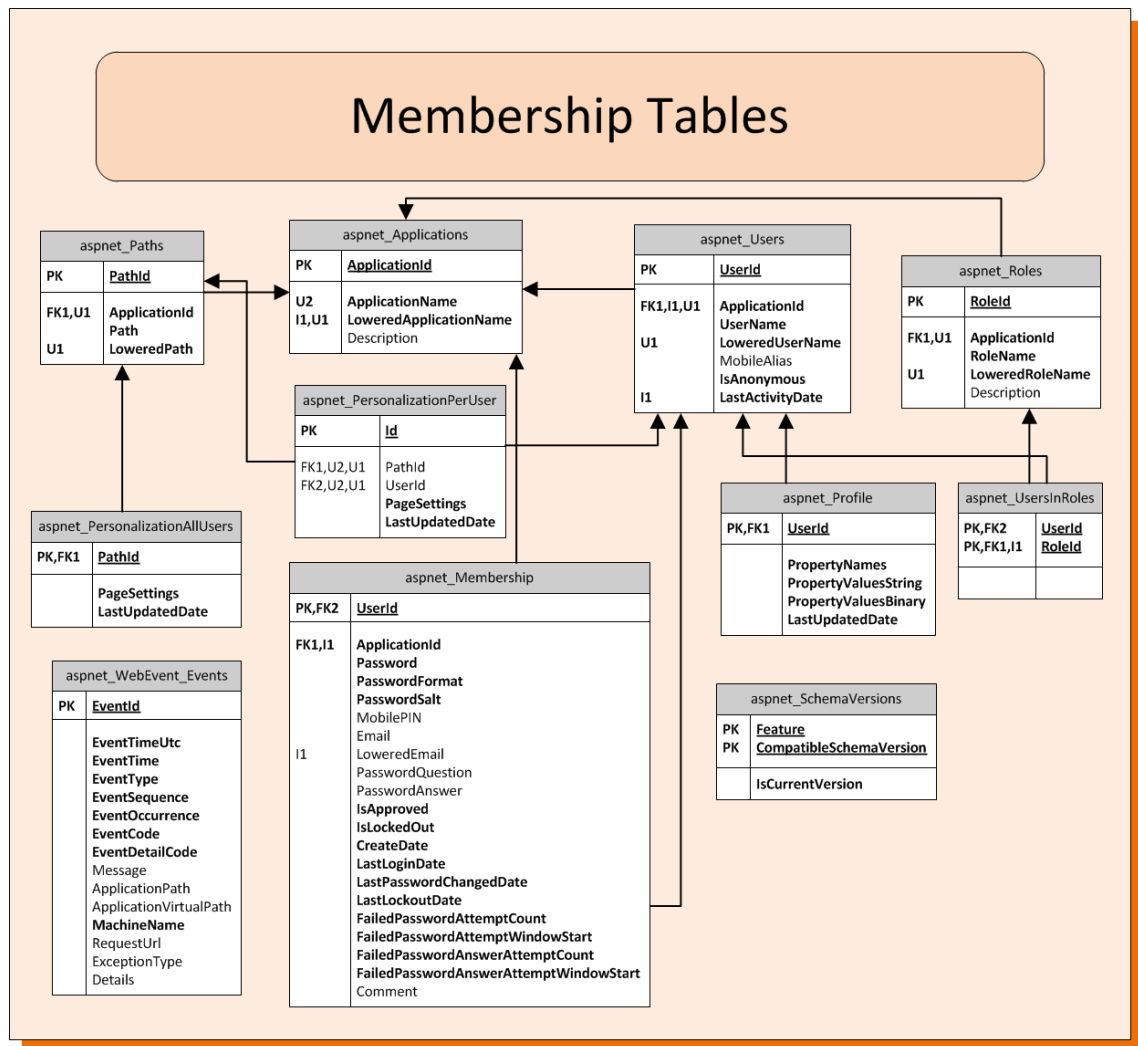
Εικόνα 26: Πίνακες επαφών

Πίνακες που αφορούν τους χρήστες του συστήματος:

Για την πρόσβαση στο σύστημα δίνεται η δυνατότητα μόνο σε εξουσιοδοτημένους χρήστες. Για το παρόν στάδιο οι πίνακες αυτοί δεν έχουν ανάμιξη με τους πίνακες των επαφών (αν και μελλοντικά πιθανόν να σμίξουν).

Για την σχεδίαση των user accounts λάβαμε υπόψη μας που αναπτύχθηκε από την Microsoft για ASP.NET 2.0 και μετά. Παρέχει το Membership framework που δίνουν προγραμματιστικές δυνατότητες για να ελέγχονται οι διαδικασίες που αφορούν τα user accounts. Αυτό το framework προσφέρει δύο Membership providers: ActiveDirectoryMembershipProvider and SqlMembershipProvider. Ο τελευταίος provider χρησιμοποιεί SQL Server και προϋποθέτει ότι υπάρχουν οι απαραίτητοι πίνακες, views, stored procedures κτλ. Για να προσθέσουμε το σχήμα της βάσης που απαιτεί ο SqlMembershipProvider η Microsoft προσφέρει ένα wizard, το aspnet_regsql.exe [16] που βρίσκεται στον κατάλογο %WINDIR%\Microsoft.Net\Framework\v4.0.30319\. Για περισσότερες πληροφορίες σχετικά με το σχήμα των πινάκων ανατρέξτε στο [16].

Στην Εικόνα 27 παρουσιάζεται συνοπτικά μια σχηματική αναπαράσταση των πινάκων που αφορά τους χρήστες του συστήματος.



Εικόνα 27: Πίνακες membership

Πίνακες που αφορούν τις μετρήσεις των αισθητήρων:

Ο πιο κάτω πίνακας παρουσιάζει το σχήμα του πίνακα *SENSORS* ο οποίος κρατάει τη γεωγραφική τοποθεσία κάθε sensor, αφού το sensor board που επιλέξαμε δεν υποστηρίζει GPS.

Column	Type	Bytes	Description
SENSOR_ID	int	4	Μοναδικό ID Αισθητήρα
LAT	decimal (9, 6)	202 max	Γεωγραφικό πλάτος θέσης αισθητήρα
LON	decimal (9, 6)	32 max	Γεωγραφικό μήκος θέσης αισθητήρα

Πίνακας 21: Πίνακας SENSORS

Ο πιο κάτω πίνακας παρουσιάζει το σχήμα του πίνακα *SENSOR_RESULTS* ο οποίος αποθηκεύει τις περιβαλλοντικές μετρήσεις των sensor σε μετασχηματισμένη μορφή (παγκόσμιες μονάδες μέτρησης). Ο πίνακας αυτός είναι συγχρονισμένος με τον πίνακα *mts400_results* της ενότητας 0.

Column	Type	Bytes	Description
result_time	datetime	8	Χρόνος που λήφθηκε το πακέτο
nodeid	int	4	Μοναδικός αριθμός αισθητήρα
parent	int	4	Πατρικός μοναδικός αριθμός κόμβου αισθητήρα
voltage	int	4	Τιμή τάσης σε mV
humid	real	4	Τιμή υγρασίας σε %
humtemp	real	4	Τιμή θερμοκρασίας σε °C
prtemp	real	4	Τιμή βαρομετρικής θερμοκρασίας σε °C
press	real	4	Τιμή βαρομετρικής πίεσης σε mbar
light	real	4	Τιμή έντασης φωτός σε Lux
accel_x	real	4	Τιμή επιταχυνσιόμετρου στον άξονα x σε mg
accel_y	real	4	Τιμή επιταχυνσιόμετρου στον άξονα y σε mg
epoch	int	4	

Πίνακας 22: Πίνακας SENSOR_RESULTS

Ο πιο κάτω πίνακας παρουσιάζει το σχήμα του πίνακα *SENSOR_HEALTH* ο οποίος αποθηκεύει την κατάσταση υγείας των sensor. Ο πίνακας αυτός είναι συγχρονισμένος με τον πίνακα *node_health* της ενότητας 0.

Column	Type	Bytes	Description
result_time	datetime	8	Χρόνος που λήφθηκε το πακέτο
nodeid	int	4	Μοναδικός αριθμός αισθητήρα
parent	int	4	Πατρικός μοναδικός αριθμός κόμβου αισθητήρα
health_pkts	int	4	Συνολικά πακέτα υγείας
node_pkts	int	4	
forwarded	int	4	Προωθημένα πακέτα
dropped	int	4	Χαμένα πακέτα
retries	int	4	Προσπάθειες αποστολής πακέτου
battery	int	4	Υπόλοιπο μπαταρίας
power_sum	int	4	Πάντα 0. Για μελλοντική χρήση

board_id	int	4	Αριθμός board
quality_tx	int	4	Ποιότητα κόμβου-πατέρα, τιμές 0-15 αντιστοιχούν σε 0-100%
quality_rx	int	4	Ποιότητα πατέρα- κόμβου, ίδια αντιστοιχία με πιο πάνω
path_cost	int	4	Εκτίμηση του αριθμού των αναγκών μεταδόσεων για αποστολή πακέτου από τον κόμβο στο base κόμβο. Path Cost = 4 * Estimated Transmission Number (ETX) $\text{Path Cost} = 2^{10} / (\text{quality_tx} * \text{quality_rx})$ Η πιο χαμηλή πιθανή τιμή είναι 4: απαιτείται τουλάχιστον μία μετάδοση για αποστολή πακέτου από το node στο base station.
parent_rssi	int	4	Received Signal Strength Indicator πατρικού κόμβου
epoch	int	4	

Πίνακας 23: Πίνακας SENSOR_HEALTH

Λειτουργίες βάσης δεδομένων:

Πέραν από την αποθήκευση και συντήρηση των πιο πάνω πινάκων, η ΒΔ του συστήματος συνεργασίας είναι υπεύθυνη να παρέχει τα stored procedures για την εύκολη πρόσβαση στις διάφορες πληροφορίες, την εισαγωγή, την διαγραφή ή την τροποποίηση τους. Για παράδειγμα η σελιδοποίηση των δεδομένων της ενότητας 0 προϋποθέτει ομαδοποίηση των δεδομένων. Επίσης πρέπει να παρέχει search functionality σε δεδομένα με αρκετά αποτελέσματα για την γρήγορη πρόσβαση.

Ακόμα, καθώς γίνεται η εισαγωγή δεδομένων από τους αισθητήρες (συγχρονισμός ΒΔ), on the fly αναλύονται τα δεδομένα για να γίνει τυχόν ανίχνευση φωτιάς με τη χρήση ενός αλγορίθμου. Τότε θα πρέπει να αρχίσει η διαδικασία έγερσης προειδοποιήσεων.

6.4.2 Συγχρονισμός βάσης δεδομένων αισθητήρων με βάση δεδομένων συστήματος συνεργασίας

Για την παραγωγή προειδοποιήσεων είναι υπεύθυνο το Σύστημα Συνεργασίας. Όμως τα δεδομένα των αισθητήρων είναι αποθηκευμένα σε άλλη ΒΔ, που ίσως να είναι απομακρυσμένη από τον εξυπηρετητή που κάνει host την Βάση Δεδομένων του Συστήματος Συνεργασίας.

Για την ανάλυση των δεδομένων των αισθητήρων και εξαγωγή συμπερασμάτων σε σχέση με το κατά πόσο υπάρχει κίνδυνος πυρκαγιάς, τίθεται η

ανάγκη για συγχρονισμό των 2 ΒΔ. Αυτό θα έχει το πλεονέκτημα όπως θα δούμε της χρονικής και χωρικής τοπικότητας.

Χρονική και χωρική τοπικότητα:

Έχοντας συγχρονίσει τις δύο βάσεις δεδομένων έχουμε άμεσα το πλεονέκτημα της τοπικής και χωρικής τοπικότητας (temporal/spatial locality). Την χρονική και χωρική τοπικότητα την έχουμε λόγω του ότι ένας χρήστης που θα ζητήσει πληροφορίες για ένα αισθητήρα μέσω της διαδικτυακής εφαρμογής, τα αποτελέσματα θα βρίσκονται τοπικά στον εξυπηρετητή. Δεν θα χρειάζεται κάθε φορά να ανατρέχει στην απομακρυσμένη ΒΔ των αισθητήρων για να επιστρέψει τα αποτελέσματα που ζητήθηκαν. Επίσης κάτι τέτοιο θα αύξανε την πολυπλοκότητα των λειτουργιών που θα έπρεπε να υποστηρίζει η απομακρυσμένη ΒΔ. Επιπλέον μελλοντικά επερωτήματα για τις ίδιες πληροφορίες θα εξυπηρετούνται πάντα με τον ίδιο τρόπο.

6.4.3 Διαπροσωπεία FireWatch

Το Σύστημα Συνεργασίας είναι αυτό που θα είναι ορατό στο χρήστη για να μπορεί να χειρίζεται τα δεδομένα που λαμβάνονται από τα υπόλοιπα συστήματα. Επομένως η πρόσβαση στα δεδομένα αυτά θα πρέπει να γίνεται έχοντας πολύ μικρή καθυστέρηση ως αμελητέα, διαφορετικά χάνουμε το χαρακτηριστικό του πραγματικού χρόνου.

Έχοντας υπόψη τις απαιτήσεις που τέθηκαν για το συγκεκριμένο σύστημα στην Ενότητα 4.3.3 η διαδικτυακή εφαρμογή θα πρέπει να τις πληρεί. Κάποια κύρια σημεία που χρειάζονται προσοχή έχουν καθοριστεί στην παρόν φάση του σχεδιασμού.

Εύκολη πρόσβαση στις τελευταίες μετρήσεις:

Το σύστημα έχει εύκολη πρόσβαση σε μετρήσεις που λήφθηκαν σε παρόν χρόνο. Οι μετρήσεις αυτές έχουν και το πιο μεγάλο βάρος σε σχέση με τις πιο παλιές μετρήσεις.

Η αναπαράσταση των δεδομένων αυτών θα γίνεται με πολλαπλούς τρόπους, όπως γραφικές παραστάσεις, πίνακες κτλ.

Αναζήτηση δεδομένων με πολλαπλά φίλτρα:

Οι χρήστες έχουν την δυνατότητα να εκτελούν επερωτήσεις στο σύστημα και να ενημερώνονται σχετικά με όλες τις μετρήσεις που έχουν αποθηκευτεί στο σύστημα σε οποιαδήποτε χρονική περίοδο. Ιστορικές μετρήσεις είναι εξίσου σημαντικές εφόσον θα αποτελούν τεκμήρια για το μέλλον.

Με χρήση κατάλληλων φίλτρων οι χρήστες του FireWatch θα μπορούν να κάνουν εξειδικευμένες αναζητήσεις ανάλογα με τους αισθητήρες και τις μετρήσεις τους. Για παράδειγμα θα μπορούν να ζητούν όλες τις μετρήσεις που είχαν καταγραφισμένη θερμοκρασία πάνω από 35 βαθμούς κελσίου.

Όπως μπορεί κάποιος να διαπιστώσει τα αποτελέσματα κάποιας επερώτησης μπορεί να είναι πολλά σε πλήθος. Επομένως χρειάζεται ένας μηχανισμός που να οργανώνει τα αποτελέσματα κατά την παρουσίαση τους. Πιθανή υλοποίηση είναι με την χρήση σελιδοποίησης, συγκεκριμένη χρονική περίοδος και ούτω καθεξής.

Η αναπαράσταση των δεδομένων αυτών θα γίνεται με πολλαπλούς τρόπους, όπως γραφικές παραστάσεις, πίνακες κτλ.

Χειρισμός χάρτη με γεωγραφικά δεδομένα αισθητήρων:

Ο χρήστης μπορεί εύκολα να χειριστεί τον χάρτη με τους αισθητήρες. Διάφοροι μηχανισμοί πρόσβασης δίνουν γρήγορη πρόσβαση στις τελευταίες μετρήσεις και προβάλλουν τη θέση κάθε αισθητήρα.

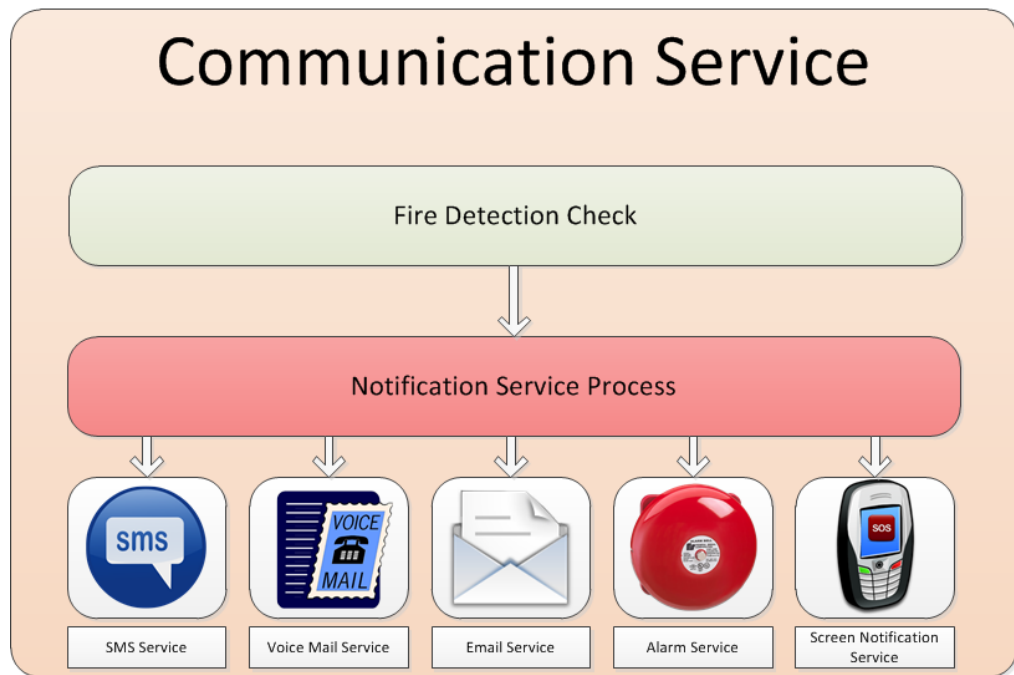
Επίσης παρέχονται δυνατότητες προβολής διαφόρων επιπέδων στο χάρτη μέσω ενός toolbar, όπως εναλλαγή δορυφορικής όψης με όψης οδικού δικτύου, εναλλαγή αγγλικών/ελληνικών οδών κτλ.

Χειρισμός επαφών του συστήματος FireWatch:

Ο χειρισμός των επαφών του συστήματος πρέπει να συμβαδίζει με την υπηρεσία επικοινωνίας. Οι χρήστες θα πρέπει να μπορούν εύκολα να διαλέξουν μια κατηγορία επαφών και να τους αποστείλουν ενημερώσεις.

6.4.4 Υπηρεσία επικοινωνίας

Τα κύρια συστατικά της υπηρεσίας επικοινωνίας και προειδοποιήσεων φαίνεται στην Εικόνα 28. Κάθε συστατικό της συγκεκριμένης υπηρεσίας θα επεξηγηθεί στη συνέχεια.



Εικόνα 28: Συστατικά Υπηρεσίας Επικοινωνίας

Συστατικό ανίχνευσης πυρκαγιάς:

Το συστατικό ανίχνευσης πυρκαγιάς είναι υπεύθυνο να ελέγχει συνεχώς τα δεδομένα των αισθητήρων και να τα αναλύει. Σε περίπτωση που υπάρχει υποψία για πυρκαγιά θα πυροδοτεί την υπηρεσία προειδοποιήσεων ώστε να ενημερώνει όλους τους εμπλεκόμενους φορείς.

Στην ενότητα 7.5.3 του επόμενου κεφαλαίου θα δούμε τον αλγόριθμο που χρησιμοποιείται για την ανίχνευση φωτιάς.

Υπηρεσία προειδοποιήσεων:

Η υπηρεσία προειδοποιήσεων είναι υπεύθυνη για τον συντονισμό όλων των μεθόδων επικοινωνίας. Ανάλογα με κάθε μέθοδο επικοινωνίας παρέχει τα απαραίτητα δεδομένα για να μπορούν να λειτουργήσουν. Επίσης κάνει τους απαραίτητους ελέγχους σχετικά με τα δεδομένα για την ορθότητα τους.

Μέθοδοι προειδοποιήσεων:

Πιο κάτω παραθέτουμε τις μεθόδους ενημέρωσης των αρχών που χρησιμοποιεί από το σύστημα FireWatch:

- Υπηρεσία SMS: Τα κινητά τηλέφωνα των χρηστών του FireWatch είναι γνωστά και αποθηκεύονται στη βάση δεδομένων του Συστήματος Συνεργασίας. Η συγκεκριμένη υπηρεσία μπορεί να τους ειδοποιήσει σε περίπτωση έκτακτης ανάγκης με άμεσο και διακριτικό τρόπο. Το σύστημα FireWatch μπορεί να αποστείλει μηνύματα με τη χρήση ενός παρόχου αποστολής πλήθους SMS.
- Υπηρεσία Email: Μια ομάδα από εσωτερικούς και εξωτερικούς φορείς / οργανισμούς θα ειδοποιηθούν μέσω email μηνύματα, τα οποία είναι η πιο οικονομικά αποδοτική λύση. Μηνύματα ηλεκτρονικού ταχυδρομείου μπορούν επίσης να διατηρηθούν ως ιστορικές αποδείξεις σχετικά με τα μέτρα που λήφθηκαν.
- Υπηρεσία Voice Email: Προεγγεγραμμένα φωνητικά μηνύματα θα μπορούν να σταλούν σε υψηλόβαθμους κυβερνητικούς πράκτορες, ανάλογα με την περίπτωση και τη σοβαρότητα κάποιου συναγερμού. Τα ηχογραφημένα μηνύματα θα περιέχουν σημαντικές πληροφορίες για περίπτωση έκτακτης ανάγκης, όπως είναι ο χρόνος που εντοπίστηκε, γεωγραφική τοποθεσία, την ακρίβεια του συναγερμού, το τμήμα που ενημερώθηκε κλπ.
- Υπηρεσία Alarm & Screen Notifications: Το σύστημα FireWatch θα έχει μηνύματα με επίκεντρο τους χρήστες του συστήματος. Αυτό σημαίνει ότι κάθε μήνυμα δεν θα είναι κοινό σε όλους αλλά μόνο τους χρήστες του συστήματος. Αυτά τα μηνύματα θα είναι διαθέσιμο για προβολή σε οποιαδήποτε στιγμή με τη μορφή pop-up παράθυρα κατά την εισαγωγή ενός χρήστη στο σύστημα. Οι χρήστες θα έχουν τις επιλογές της διαγραφής, προώθησης, αναστολής (για προκαθορισμένα χρονικά διαστήματα) και απόρριψης. Ομοίως, το σύστημα FireWatch θα είναι σε θέση να παίξει προειδοποιήσεις με ήχο ειδοποίησης, αν η περίπτωση κατατάσσεται ως υψηλής προτεραιότητας. Οι χρήστες θα έχουν τις επιλογές της αναστολής (για προκαθορισμένα χρονικά διαστήματα) και απόρριψης. Κατά τα τελευταία στάδια του project θα μελετηθεί η προσαρμογή των υπηρεσιών συναγερμού και σε smartphone συσκευές.

6.4.5 Παραγωγή ροής εργασιών

Ροή εργασίας σε περίπτωση ανίχνευσης πυρκαγιάς:

Σε περίπτωση που γίνει ανίχνευση πυρκαγιάς, θα γίνεται χρήση όλων των μεθόδων ενημέρωσης που προσφέρονται από το σύστημα. Το σύστημα θα χρησιμοποιήσει τα μηνύματα sms, screen notifications, sound alarms και voice μηνύματα.

Λόγω της σοβαρότητας της κατάστασης, τα sound alarms δεν θα σταματούν να χτυπούν μέχρι κάποιος να τα απενεργοποιήσει από το σύστημα. Επίσης οι προειδοποιήσεις που φαίνονται στην οθόνη θα βρίσκονται συνεχώς στην κορυφή όλων των παραθύρων των χρηστών μέχρι να πάρει κάποιος την πρωτοβουλία να το σβήσει. Θα παρέχονται πληροφορίες για το συμβάν όπως η ώρα ανίχνευσης, το σημείο που προκλήθηκε κτλ. Με αυτό τον τρόπο ο χρήστης έχει πάρει την ευθύνη για να ενημερώσει τους υπόλοιπους για μετέπειτα χειρισμούς της κατάστασης δίνοντας τους τις απαραίτητες πληροφορίες.

Οι επαφές του συστήματος θα λαμβάνουν μηνύματα ήχου ή SMS ανάλογα με τι διαθέτουν. Τα μηνύματα SMS περιέχουν τις βασικές πληροφορίες σχετικά με το συμβάν και θα αποστέλλονται στους υπεύθυνους. Τα ηχογραφημένα μηνύματα ήχου θα αποστέλλονται στους κοντινούς σταθμούς της πυρκαγιάς και σε άλλους φορείς. Για τις αποστολές των μηνυμάτων θα κρατείται αρχείο.

Ροή εργασίας σε περίπτωση πρόβλεψης πυρκαγιάς:

Σε περίπτωση που γίνει πρόβλεψη πυρκαγιάς, θα γίνεται και πάλι η διαδικασία ενημέρωσης των διαφόρων φορέων για τις αιτίες που μπορούν να οδηγήσουν σε φωτιά. Η δράση που θα λάβουν δε θα είναι τόσο αναγκαία όπως στην περίπτωση ανίχνευσης, γι' αυτό θα αποστέλλονται μόνο μέσω email και SMS, με όλες τις πληροφορίες, στους κατάλληλους φορείς. Το σύστημα θα χρησιμοποιήσει τα screen notifications για να προειδοποιήσει τους χρήστες του συστήματος.

Κεφάλαιο 7

Υλοποίηση Συστήματος

7.1	Εισαγωγή	84
7.2	Τεχνικά χαρακτηριστικά υπολογιστή ανάπτυξης του FireWatch.....	85
7.3	Υλοποίηση συστήματος ασύρματου δικτύου αισθητήρων.....	85
7.3.1	Εργαλεία ανάπτυξης.....	86
7.3.2	Εκκίνηση/Τερματισμός καταγραφής μετρήσεων από το δίκτυο αισθητήρων.....	87
7.3.3	Προγραμματισμός κόμβων αισθητήρων	88
7.3.4	Υλοποίηση λειτουργιών βάσης δεδομένων ασύρματου δικτύου αισθητήρων	88
7.4	Υλοποίηση συστήματος γεωγραφικών δεδομένων	90
7.4.1	Εργαλεία ανάπτυξης.....	90
7.4.2	Κατασκευή χάρτη.....	90
7.4.3	Πρόσθεση επιπέδων στον χάρτη	90
7.4.4	Πρόσθεση σημείων ενδιαφέροντος στον χάρτη.....	91
7.5	Υλοποίηση συστήματος συνεργασίας	91
7.5.1	Εργαλεία ανάπτυξης.....	91
7.5.2	Υλοποίηση λειτουργιών βάσης δεδομένων συστήματος συνεργασίας.....	93
7.5.3	Αλγόριθμος ανίχνευσης φωτιάς	95
7.5.4	Μεθοδολογία ανάπτυξης και παρουσίαση εφαρμογής	101

7.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα επεξηγηθεί η υλοποίηση του συστήματος. Έχοντας καθορίσει την αρχιτεκτονική του συστήματος, τις απαιτήσεις που διέπουν το σύστημα και τις βασικές αρχές σχεδίασης θα προχωρήσουμε στην υλοποίηση του.

Θα δούμε σε βάθος την υλοποίηση κάθε συστήματος ξεχωριστά. Ακόμη, στο κεφάλαιο αυτό θα γίνει αναφορά στα τεχνολογικά εργαλεία τα οποία χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Επίσης στο παρόν στάδιο συγγραφής κάποια κομμάτια της υλοποίησης είναι υπό ανάπτυξη και δοκιμή. Η παρόν υλοποίηση έχει σχεδόν προσεγγίσει όλες τις πτυχές που θέσαμε σε προηγούμενα κεφάλαια. Φυσικά είναι ανοικτή για οποιεσδήποτε αλλαγές, τροποποιήσεις, προσθήσεις, βελτιστοποιήσεις και εισηγήσεις.

Τελικός αποδέκτης του FireWatch είναι το Τμήμα Δασών. Οι πελάτες του συστήματος είναι αυτοί που καθορίζουν την πορεία υλοποίησης.

7.2 Τεχνικά χαρακτηριστικά υπολογιστή ανάπτυξης του FireWatch

Αξίζει να αναφέρουμε ότι η υλοποίηση του FireWatch είναι υλοποιημένη για σκοπούς demonstration και ανάπτυξης σε υπολογιστή με τα εξής χαρακτηριστικά:

- Επεξεργαστής: Intel Core 2 Duo, E8400 @ 3.00 GHz
- Μνήμη: 2 GB
- Αρχιτεκτονική: 64 Bit
- OS: Windows Server 2008 R2

Το σύστημα του FireWatch έχει αναπτυχθεί με τέτοιο τρόπο που να είναι επεκτάσιμο και συντηρήσιμο χρησιμοποιώντας Virtual Machines για κάθε υποσύστημα. Αυτό θα προσφέρει ευελιξία κατά την εφαρμογή του. Συγκεκριμένα κάθε υποσύστημα βρίσκεται σε ξεχωριστό VM στο συγκεκριμένο μηχάνημα που μπορεί να συντηρηθεί ξεχωριστά από τα υπόλοιπα κάτι που δίνει μεγάλη ευχέρεια. Το πρώτο VM χειρίζεται τους το ασύρματο δίκτυο αισθητήρων και την βάση PostgreSQL. Το δεύτερο VM αποτελεί το Σύστημα Συνεργασίας με την δική του βάση δεδομένων και την παραγωγή δεδομένων ροής εργασίας. Το τρίτο VM αφορά τον IIS Webserver στον οποίο είναι εγκατεστημένος η διαδικτυακή μας εφαρμογή. Υπάρχει και 4^ο VM το οποίο συντηρείται από την Geomatic και παρέχει τα γεωγραφικά δεδομένα προς της διαδικτυακή μας εφαρμογή.

7.3 Υλοποίηση συστήματος ασύρματου δικτύου αισθητήρων

Σε αυτό το σημείο αρκεί να αναφέρουμε ότι υπάρχει ο περιορισμός χρήσης των λογισμικών της Crossbow σε λειτουργικό σύστημα Windows XP που είναι fully compatible.

7.3.1 Εργαλεία ανάπτυξης

Οι αισθητήρες της εταιρείας MEMSIC έρχονται με προκαθορισμένα λογισμικά που διευκολύνουν την ανάπτυξη εφαρμογών που έχουν να κάνουν με το ασύρματο δίκτυο αισθητήρων.

- *MoteView*: Μέσω του λογισμικού MoteView γίνεται εύκολα η παρακολούθηση και ο έλεγχος των sensor, η σύνδεση στη βάση δεδομένων, η ανάλυση και η παρουσίαση των μετρήσεων των αισθητήρων. Η διαπροσωπεία του MoteView φαίνεται στην Εικόνα 24. Όπως προαναφέρθηκε έχουμε χρησιμοποιήσει το MoteView ως μια αναφορά και ένα εργαλείο μάθησης για το πώς θα μπορούσαμε να υλοποιήσουμε κάποια κομμάτια του συστήματος FireWatch.
- *MoteConfig*: Το MoteConfig που εγκαθίσταται μαζί με τα υπόλοιπα λογισμικά που παρέχει η Crossbow, είναι ένα εργαλείο που τρέχει πάνω σε Windows OS και επιτρέπει τον προγραμματισμό των IRIS motes. Παρέχει την διαπροσωπεία που φαίνεται στην Εικόνα 25 για την ρύθμιση και την εγκατάσταση pre-compiled XMesh/TinyOS λογισμικά πάνω στους αισθητήρες. Το MoteConfig επιτρέπει στους χρήστες να καθορίσουν το Mote ID, το Group ID, το RF κανάλι και τη RF ισχύ.
- *MoteWorks*: Το λογισμικό επιτρέπει την ανάπτυξη εφαρμογών με κόμβους αισθητήρων και είναι ειδικά βελτιστοποιημένο για δίκτυα τα οποία λειτουργούν με χαμηλή κατανάλωση ενέργειας κάνοντας χρήση μπαταριών. Είναι επίσης στηριγμένο στο open-source TinyOS λειτουργικό σύστημα και προσδίδει αξιόπιστη ad-hoc mesh δικτύωση, δυνατότητες προγραμματισμού αισθητήρων ασύρματα, εργαλεία ανάπτυξης ανεξαρτήτως πλατφόρμας, server middleware για ενσωμάτωση δικτύου επιχειρήσεων και διαπροσωπεία χρήστη για την ανάλυση και ρύθμιση.
- *PostgreSQL*: Όσο αφορά τη βάση δεδομένων και τη διαχείριση των δεδομένων χρησιμοποιήσαμε τη PostgreSQL για βάση και για εργαλείο διαχείρισης το Postgre Studio. Η PostgreSQL είναι μια open source, σχεσιακή βάση δεδομένων ανεπτυγμένη σε C. Μετά σε μια 20ετία σχεδόν απέκτησε ισχυρή φήμη για την αξιοπιστία, την ακεραιότητα των δεδομένων, και την ορθότητα της. Τρέχει στα πλείστα γνωστά λειτουργικά συστήματα, μεταξύ των οποίων τα Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), και τα Windows.

- *PL/pgSQL*: είναι η Procedural language που χρησιμοποιήθηκε για την κατασκευή των functions και stored procedures. Συνδυάζει απλές εντολές SQL με πιο εξειδικευμένες λειτουργίες.

7.3.2 Εκκίνηση/Τερματισμός καταγραφής μετρήσεων από το δίκτυο αισθητήρων

Το MoteView δίνει την δυνατότητα να εκκινήσουμε ή να τερματίσουμε τη διεργασία xserver.exe που είναι υπεύθυνη για την καταγραφή των δεδομένων που λαμβάνει ο base station (0

Server Tier/XServe). Η δική μας εφαρμογή θέλουμε να είναι ανεξάρτητη από το MoteView.

Για την εκκίνηση ή τον τερματισμό της συγκεκριμένης διεργασίας χρησιμοποιήσαμε command line εντολές και το documentation του Xserve που προσφέρει η Crossbow. Στο δίνεται η πλήρης περιγραφή των παραμέτρων που λαμβάνει.

Συγκεκριμένα για να ξεκινήσει η καταγραφή δεδομένων μπορούμε να χρησιμοποιήσουμε σε command line του Cygwin την πιο κάτω εντολή:

```
xserve -s=COM4 -b=57600 -l -xml file=XmlStream.xml -xmlp -xmlport=9005
```

Η πιο πάνω εντολή προϋποθέτει ότι έχει γίνει ανάθεση στην environmental variable Path το μονοπάτι C:\Program Files\Crossbow\MoteView\xserve2\bin ή κάτι αντίστοιχο για να μπορεί να βρεθεί το executable file xserve.exe και ότι το base station είναι εγκατεστημένο στη USB θύρα COM4.

Φυσικά η εκκίνηση της πιο πάνω διεργασίας δεν θα πρέπει να απαιτεί την παρουσία μας στον απομακρυσμένο υπολογιστή που καταγράφει μετρήσεις των αισθητήρων. Οπότε κατασκευάστηκε ένα service με το οποίο μπορεί να επικοινωνεί το σύστημα συνεργασίας και να εκτελεί τις εντολές αυτές.

Για την εκκίνηση μιας διεργασίας από κώδικα C# μπορεί να γίνει χρήση της εντολής:

```
process = System.Diagnostics.Process.Start("xserve.exe", "-s=COM4 -b=57600 -l -xml file=XmlStream.xml -xmlp -xmlport=9005");
```

Για τον τερματισμό μιας διεργασίας από κώδικα C# μπορεί να γίνει χρήση της εντολής:

```
process.Kill();
```

7.3.3 Προγραμματισμός κόμβων αισθητήρων

Με την παραλαβή των αισθητήρων έπρεπε ο καθένας να ρυθμιστεί. Με τη χρήση του MoteConfig η διαδικασία ήταν εύκολη. Προγραμματίσαμε τους αισθητήρες του ασύρματου δικτύου αισθητήρων με ένα από τα pre-compiled IRIS XMesh Applications της Crossbow και συγκεκριμένα το *XMTS400_M2110_<mode>*, το οποίο αντιστοιχεί στο sensor board MTS400 και το IRIS processor M2110 και βρίσκεται στον κατάλογο εγκατάστασης που είναι παρόμοιος με αυτόν C:\Program Files\Crossbow\MoteView\xmesh\iris\MTS400.

Στη συνέχεια, προγραμματίσαμε τον base node με επίσης ένα pre-compiled IRIS XMesh Applications και συγκεκριμένα το *XMeshBase_M2110_<mode>*.

Το <mode> που παρουσιάζεται πιο πάνω αντιστοιχεί σε hp ή lp και καθορίζει αν οι αισθητήρες θα λειτουργούν σε high ή low power αντίστοιχα. Οι περιγραφές των δύο βρίσκονται στην ενότητα 0. Επίσης ο πιο κάτω πίνακας συνοψίζει τις βασικές διαφορές τους.

Εφόσον το σύστημα ήταν σε θέση να καταγράφει τις μετρήσεις προχωρήσαμε στην εξειδίκευση της ΒΔ.

Παράμετρος	XMesh-HP	XMesh-LP
Route Update Interval	36 sec.	360 sec.
Data Message Rate	10 sec, typ.	180 sec., typ.
Mesh formation time	2-3 times Route Update Interval for mesh with average of 2.5 hops	
Average current usage	20-30 mA	<250 μ A [1]< 400 μ A [2]

Πίνακας 24: XMesh Performance Summary

7.3.4 Υλοποίηση λειτουργιών βάσης δεδομένων ασύρματου δικτύου αισθητήρων

Για την καταγραφή μετρήσεων από τους αισθητήρες υπήρχε το πρόβλημα ότι έλειπαν κάποια δεδομένα από την εγγραφή των πακέτων. Συγκεκριμένα τα Calibration words της βαρομετρικής πίεσης δεν καταγράφονταν στη βάση.

Έτσι αρχικά θα έπρεπε να τροποποιηθούν οι πίνακες ώστε να καταγράφουν τις συγκεκριμένες τιμές, διαφορετικά θα αδυνατούσαμε να μετατρέψουμε τα ακατέργαστα δεδομένα σε παγκόσμιες μονάδες μέτρησης όπως περιγράψαμε στην ενότητα 6.2.5.

Για να επιτευχθεί η καταγραφή αυτή αλλάξαμε τα προκαθορισμένα αρχεία που εγκαθίστανται κατά το installation με αυτά που υπάρχουν στο Παράρτημα Δ. Συγκεκριμένα τροποποιήθηκαν τα αρχεία db_xsensor.sql, mts400_mesh_config.xml και mts400_sensor_config.xml κάτω από το directory C:\Program Files\Crossbow\MoteView.

Μετασχηματισμός ακατέργαστων δεδομένων:

Για τον μετασχηματισμό των ακατέργαστων δεδομένων που λαμβάνονται από το δίκτυο υλοποιήσαμε τις απαραίτητες συναρτήσεις στη σχεσιακή βάση δεδομένων PostgreSQL που μετατρέπει κάθε πεδίο σε αντίστοιχες μονάδες μέτρησης όπως αναφέραμε στην ενότητα 6.2.5.

Συγκεκριμένα έχουν υλοποιηθεί οι εξής συναρτήσεις:

- FN_CONVERT_ACCEL(AccelData): Μετατρέπει τις τιμές του επιταχυνσιόμετρου.
- FN_CONVERT_BATTERY(ADCVref): Μετατρέπει τις τιμές της μπαταρίας.
- FN_CONVERT_HUMIDITY(HumData, TempData): Μετατρέπει τις τιμές της υγρασίας.
- FN_CONVERT_INTERSEMAPRESSURE(cal_word1, cal_word2, cal_word3, cal_word4, TempData, PressureData): Μετατρέπει τις τιμές της βαρομετρικής πίεσης.
- FN_CONVERT_INTERSEMAPTEMP(cal_word1, cal_word2, TempData): Μετατρέπει τις τιμές της βαρομετρικής θερμοκρασίας..
- FN_CONVERT_LIGHT(taosch0, taosch1): Μετατρέπει τις τιμές του αισθητήρα φωτός.
- FN_CONVERT_TEMP(TempData): Μετατρέπει τις τιμές της θερμοκρασίας.

Επίσης έχουν υλοποιηθεί κάποιες άλλες βοηθητικές συναρτήσεις για την ανάκτηση των δεδομένων από την βάση δεδομένων του Συστήματος Συνεργασίας.

- FN_GET_FRESH_SENSOR_DATA(date_from, date_to): Επιστρέφει όλες τις μετρήσεις σχετικές με τις περιβαλλοντικές μετρήσεις που λήφθηκαν μεταξύ date_from και date_to.
- FN_GET_FRESH_SENSOR_HEALTH(date_from, date_to): Επιστρέφει όλες τις μετρήσεις σχετικές με την υγεία των αισθητήρων που λήφθηκαν μεταξύ date_from και date_to.

7.4 Υλοποίηση συστήματος γεωγραφικών δεδομένων

Το σύστημα Γεωγραφικών Δεδομένων έχει αναπτυχθεί από την εταιρεία Geomatic και έχει ενσωματωθεί στο web application μας.

7.4.1 Εργαλεία ανάπτυξης

JavaScript: Όπως προαναφέρθηκε η πρόσβαση στα γεωγραφικά δεδομένα γίνεται με την χρήση JavaScript. Όπως θα περιγράψουμε στην ενότητα 7.5, έχουμε χρησιμοποιήσει ASPX.NET για την υλοποίηση της διαδικτυακής εφαρμογής που είναι 100% compatible με την χρήση JavaScript.

Πιο κάτω παραθέτω κάποια κομμάτια κώδικα, σημαντικά για την κατανόηση της υλοποίησης του συστήματος γεωγραφικών δεδομένων. Πλήρης περιγραφή του API της Geomatic δίνεται στο Παράρτημα Β.

7.4.2 Κατασκευή χάρτη

```
map = new geomatic.maps.Map('maplargediv', /* The div id of the map */
    {
        base: 'map',
        lang: 'EN',
        zoom: 14,
        center: [33.253355, 34.949911],
        zoombar: 'full'
    });
```

Το πιο πάνω κομμάτι JavaScript αρχικοποιεί τον χάρτη, θέτει διευθύνσεις στην αγγλική γλώσσα, καθορίζεται το zoom level, η συντεταγμένες εστίασης και εμφανίζεται το μενού για μεγέθυνση.

7.4.3 Πρόσθεση επιπέδων στον χάρτη

```
landuse_layer = map.addLayer({
    id: "landuse",
```

```

type: "wms",
url: "http://geomatic.com.cy:8054/geoserver/gwc/service/wms",
transparent: true,
layerName: "geomaps:Landuse_FW",
visibility: true,
opacity: 1
});

```

Το πιο πάνω κομμάτι JavaScript προσθέτει στον χάρτη το layer που είναι υπεύθυνο για τη βλάστηση μιας συγκεκριμένης περιοχής.

Με παρόμοιο τρόπο έχουν προστεθεί επίπεδα για να αλλάζει η εμφάνιση του χάρτη από οδικό δίκτυο σε όψη από δορυφόρο, εμφάνιση off-road δρομολογίων κτλ.

7.4.4 Πρόσθεση σημείων ενδιαφέροντος στον χάρτη

```

var point = map.addPoint(sensor_data[0], sensor_data[1], sensor_data[2],
'http://www.cs.ucy.ac.cy/~gconst02/sensor_img.png', "Sensor " + sensor_data[0],
"<a href=\"WSNMonitoring.aspx?SENSOR_ID=\" + sensor_data[0] + \"\">Show
Data</a><br/>Latitude: " + sensor_data[1] + " <br/>Longitude: " + sensor_data[2]
+ "<br /><br />");

```

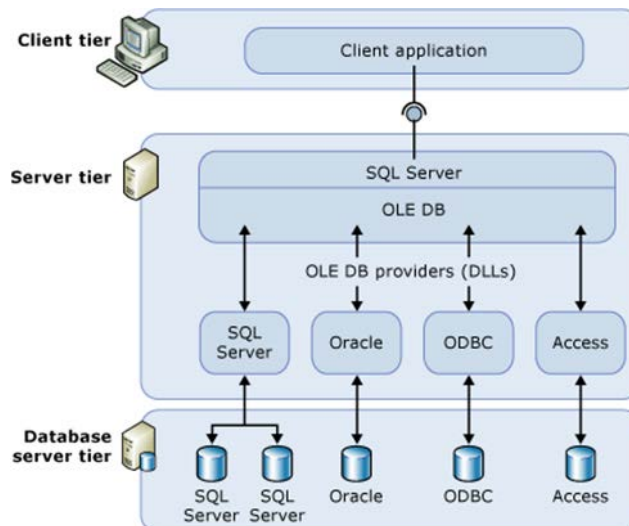
Το πιο πάνω κομμάτι JavaScript προσθέτει στον χάρτη σημείο ενδιαφέροντος μαζί με κάποιες επιπρόσθετες πληροφορίες που μπορούν να ενσωματωθούν σαν κώδικας HTML.

Με τον ίδιο τρόπο μπορούν να προστεθούν επιπλέον σημεία στον χάρτη, όπως δασικοί σταθμοί και ότι άλλη ανάγκη προκύψει περαιτέρω.

7.5 Υλοποίηση συστήματος συνεργασίας

7.5.1 Εργαλεία ανάπτυξης

Microsoft SQL Server 2005: Η ΒΔ του Συστήματος Συνεργασίας θα είναι εγκατεστημένοι σε Microsoft SQL Server 2005. Είναι μια σχεσιακή ΒΔ που προσφέρει πολλές δυνατότητες ανάπτυξης, ελέγχου και διαφύλαξης δεδομένων. Επίσης διαθέτει ρυθμίσεις για ένωση της με εξωτερικούς εξυπηρετητές που επιτρέπει στον SQL Server να εκτελεί επερωτήσεις σε εξωτερικές OLE DB πηγές δεδομένων λαμβάνοντας δεδομένα από απομακρυσμένους εξυπηρετητές. Αυτό μας δίνει την δυνατότητα να επικοινωνούμε με την PostgreSQL για να λαμβάνουμε τα δεδομένα που χρειαζόμαστε από το δίκτυο αισθητήρων όπως φαίνεται στην Εικόνα 29.



Εικόνα 29: Linked Servers

Οι ένωση των 2 server έχει τα εξής πλεονεκτήματα:

- Απομακρυσμένη πρόσβαση στον εξυπηρετητή
- Δυνατότητα να πραγματοποιούμε κατανεμημένες επερωτήσεις, ανανεώσεις, εντολές και δοσοληψίες σε ετερογενείς πηγές δεδομένων.
- Να χειριζόμαστε διεσπαρμένες πηγές δεδομένων με παρόμοιο τρόπο.

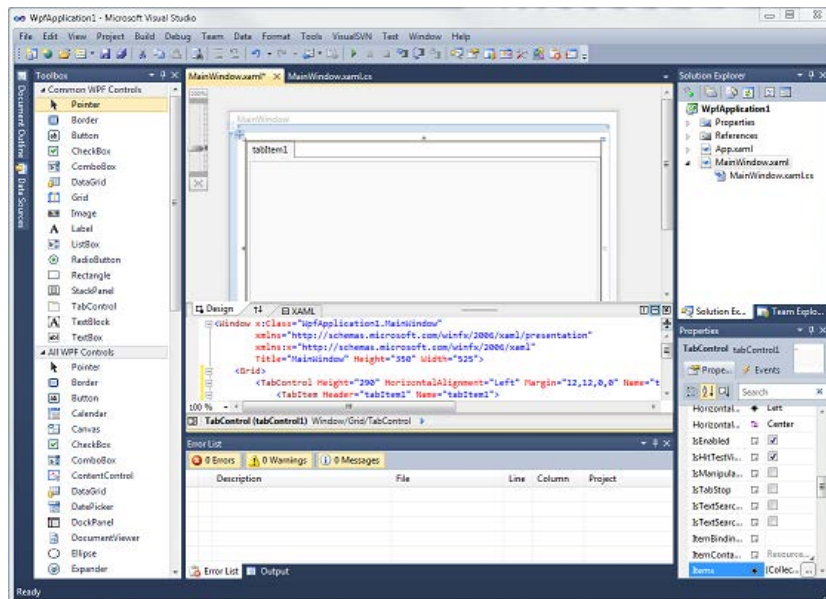
Για την ένωση των 2 server ανατρέξτε στο Παράρτημα Ε για ένα Tutorial.

SQL Server Agent: Έχει γίνει η χρησιμοποίηση του SQL Server Agent ο οποίος επιτρέπει την εκτέλεση stored procedures σε συγκεκριμένη χρονική περίοδο και κατ' επανάληψη. Με αυτό τον τρόπο σε συνδυασμό με την ένωση των 2 server έχουμε καταφέρει να αποθηκεύουμε στον SQL Server τα δεδομένα που υπάρχουν στην απομακρυσμένη βάση των αισθητήρων. Για την εκτέλεση ενός επερωτήματος στον απομακρυσμένο εξυπηρετητή χρησιμοποιούμε την εξής σύνταξη σε T-SQL (OPENQUERY):

```
SET @TSQL = '
SELECT LM.*
FROM OPENQUERY(LINKED_MOTES, 'SELECT * FROM
"FN_GET_FRESH_SENSOR_DATA"(' + CONVERT(VARCHAR, @DATE_FROM,
21) + ''', '' + CONVERT(VARCHAR, @DATE_TO, 21) + ''')') LM'
INSERT INTO SENSOR_RESULTS
EXEC (@TSQL)
```

Visual Studio 2010: Για την ανάπτυξη της ιστοσελίδας χρησιμοποιήθηκε το Microsoft Visual Studio 2010 (Εικόνα 30). Είναι ένα IDE κατάλληλο για διαδικτυακές εφαρμογές που υποστηρίζει πολλαπλές πλατφόρμες όπως .NET. Επιπλέον υποστηρίζει

χρήσιμες γλώσσες προγραμματισμού διαδικτυακών εφαρμογών JavaScript, Ajax, CSS και HTML.



Εικόνα 30: Microsoft Visual Studio 2010 Screenshot

7.5.2 Υλοποίηση λειτουργιών βάσης δεδομένων συστήματος συνεργασίας

Όπως αναφέραμε στο προηγούμενο κεφάλαιο υπήρχε η ανάγκη για υλοποίηση διαφόρων λειτουργιών που πρέπει να υποστηρίζει το σύστημα. Συγκεκριμένα έχουν υλοποιηθεί τα εξής stored procedures:

- *SP_ADD_NEW_CONTACT/SP_DELETE_CONTACT/SP_EDIT_CONTACT*: Δίνουν τη δυνατότητα να εισάγουμε, να διαγράψουμε ή να σβήσουμε μια επαφή.
- *SP_GET_CITIES*: Επιστρέφει τις περιοχές.
- *SP_GET_CONTACT_TYPES*: Επιστρέφει τους τύπους των επαφών.
- *SP_GET_DISTRICTS*: Επιστρέφει τις επαρχίες.
- *SP_GET_FRESH_SENSOR_DATA*: Εισάγει νέα δεδομένα μετρήσεων από τη βάση των αισθητήρων.
- *SP_GET_FRESH_SENSOR_HEALTH*: Εισάγει νέα δεδομένα κατάστασης υγείας από τη βάση των αισθητήρων.
- *SP_GET_SENSOR_DATA_LOCAL_COUNT*: Επιστέφει τον αριθμό των records που θα επιστραφούν για ένα συγκεκριμένο query πάνω στα

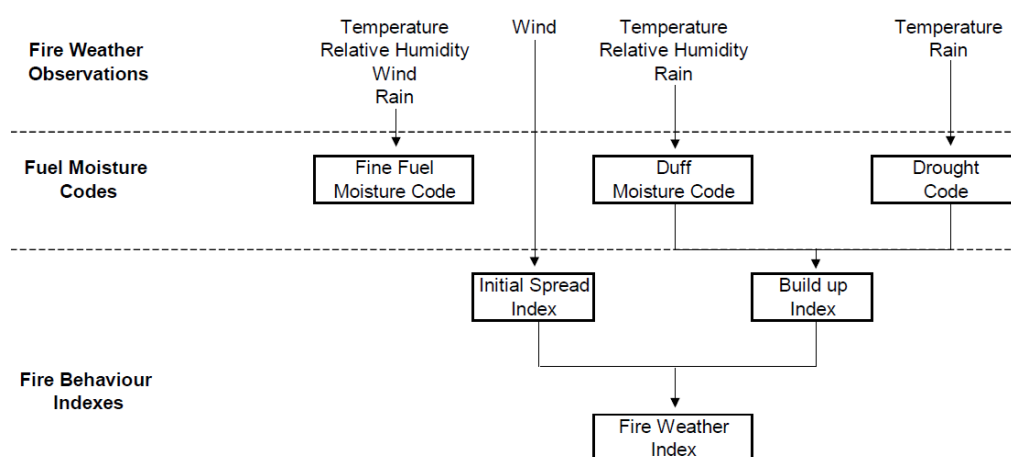
αποτελέσματα των μετρήσεων των αισθητήρων (χρησιμοποιείται για το paging των δεδομένων).

- **SP_GET_SENSOR_DATA_LOCAL_PAGED:** Επιστρέφει τις μετρήσεις των αισθητήρων σελιδοποιημένα, για ένα συγκεκριμένο query.
- **SP_GET_SENSOR_HEALTH_LOCAL_COUNT:** Επιστρέφει τον αριθμό των records που θα επιστραφούν για ένα συγκεκριμένο query πάνω στα αποτελέσματα της κατάστασης υγείας των αισθητήρων (χρησιμοποιείται για το paging των δεδομένων).
- **SP_GET_SENSOR_HEALTH_LOCAL_PAGED:** Επιστρέφει τις μετρήσεις της κατάστασης υγείας των αισθητήρων σελιδοποιημένα, για ένα συγκεκριμένο query.
- **SP_SEARCH_CONTACT:** Παρέχει τον μηχανισμό αναζήτησης μιας συγκεκριμένης επαφής.
- **SP_GET_SENSOR_QUERIES:** Επιστρέφει τα πιθανά queries που υποστηρίζει το σύστημα. Αυτά είναι:
 - ο **LAST 10:** Τελευταίες 10 μετρήσεις
 - ο **LAST 25:** Τελευταίες 25 μετρήσεις
 - ο **LAST 50:** Τελευταίες 50 μετρήσεις
 - ο **CURRENT DAY:** Μετρήσεις της ημέρας
 - ο **LAST 5 DAYS:** Μετρήσεις τελευταίων 5 ημερών
 - ο **LAST 10 DAYS:** Μετρήσεις τελευταίων 10 ημερών
 - ο **LAST 20 DAYS:** Μετρήσεις τελευταίων 20 ημερών
 - ο **LAST 30 DAYS:** Μετρήσεις τελευταίων 30 ημερών
 - ο **CURRENT MONTH:** Μετρήσεις τρέχον μήνα
 - ο **PREVIOUS MONTH:** Μετρήσεις περασμένου μήνα
 - ο **LAST 2 MONTHS:** Μετρήσεις περασμένων 2 μηνών
 - ο **LAST 3 MONTHS:** Μετρήσεις περασμένων 3 μηνών
 - ο **LAST 6 MONTHS:** Μετρήσεις περασμένων 6 μηνών
 - ο **LAST 12 MONTHS:** Μετρήσεις περασμένων 12 μηνών
 - ο **USER DEFINED:** Query που καθορίζεται από τον χρήστη. Μπορεί να καθορίζει ένα περιθώριο ημερομηνίας (from, to) ή να ορίζει ένα LIMIT (π.χ. τα τελευταία 300 records).

- SP_GET_SENSORS: Επιστρέφονται οι sensors με τις συντεταγμένες τους.
- SP_GET_SENSORS_IDS: Επιστρέφονται μόνο τα ids των sensors.
- SP_GET_SENSOR_[Measurement]: Το [Measurement] είναι οι μετρήσεις που διαθέτουν οι αισθητήρες όμως HUMTEMP για θερμοκρασία, PRESS για βαρομετρική πίεση κοκ. Επιστρέφει ένα συγκεκριμένο υποσύνολο των μετρήσεων που καθορίζεται στο [Measurement]. Το υποσύνολο καθορίζεται από το query.

7.5.3 Αλγόριθμος ανίχνευσης φωτιάς

Για την ανίχνευση πυρκαγιών έχουμε σε αυτό το στάδιο υιοθετήσει τον αλγόριθμο Canadian Fire Weather Index (FWI) [17] που είναι καθιερωμένο και αποδεκτό μοντέλο.



Εικόνα 31: Σχηματική Αναπαράσταση Αλγόριθμου Ανίχνευσης Φωτιάς FWI

Όπως φαίνεται στην Εικόνα 31, ο δείκτης επικινδυνότητας πυρκαγιάς αποτελείται από 6 επιμέρους συστατικά-δείκτες. Αυτοί αφορούν την ημερήσια διακύμανση της περιεκτικότητας σε νερό για καύσιμα με διαφορετικές αλλαγές στο χρόνο απόκρισης σε καιρικές συνθήκες, ο αρχικός ρυθμός εξάπλωσης για την διάδοση της, την ποσότητα του καυσίμου και την αναμενόμενη ένταση του μετώπου της πυρκαγιάς.

Η τιμή που δίνει το FWI αντιπροσωπεύει την ένταση της διαδιδόμενης πυρκαγιάς, ανάλογα με την ποσότητα της ενέργειας που απελευθερώνεται από τη

γραμμική μονάδα του μετώπου της πυρκαγιάς. Οι δείκτες που απαρτίζουν FWI θα περιγραφούν στα ακόλουθα υποκεφάλαια.

Απαραίτητα δεδομένα εισόδου:

Τα απαραίτητα δεδομένα εισόδου για τον FWI είναι:

- T: Θερμοκρασία το μεσημέρι, σε βαθμούς κελσίου.
- H: Σχετική υγρασία το μεσημέρι, σε ποσοστό (%).
- W: Ταχύτητα ανέμου το μεσημέρι, σε χιλιόμετρα ανά ώρα.
- r_0 : Ημερήσια βροχόπτωση σε μιλίμετρα.

Στις βασικές εξισώσεις χρησιμοποιούμε τα ακόλουθα σύμβολα:

- f(D): duff humidity factor
- R: initial spread index (ISI)
- U: build up index (BUI)
- B: FWI (intermediate form)
- S: FWI (form final)

Το f(D) υπολογίζεται ως εξής:

$$\text{Αν } U \leq 80 \text{ τότε } f(D) = 0.626 U^{0.809+2}$$

$$\text{Αν } U > 80 \text{ τότε } f(D) = 1000/(25+108.64e^{-0.023U})$$

Το B υπολογίζεται ως εξής:

$$B = 0.1 R f(D)$$

Το S υπολογίζεται ως εξής:

$$\text{Αν } B > 1 \text{ τότε } S = \text{EXP}(2.72(0.434 \ln(B))^{0.647})$$

$$\text{Αν } B \leq 1 \text{ τότε } S = B$$

Fine Fuel Moisture Code (FFMC):

Το Fine Fuel Moisture Code (FFMC) είναι μέρος του Καναδικού FWI. Ο κώδικας είναι η έκφραση της περιεκτικότητας σε νερό των litter και fine dead fuels. Δείχνει τη σχετική ευκολία ανάφλεξης και την ευφλεκτότητα των fine dead fuels. Οι απαιτούμενες είσοδοι είναι οι ίδιες με το FWI.

Στις βασικές εξισώσεις χρησιμοποιούμε τα ακόλουθα σύμβολα:

m_0	water content in fine fuel of the previous day
m_r	water content in fine fuel after the rain
m	water content in fine fuel after the drainage
r_f	real rain, FFMC
E_d	TEE (equilibrium water content) of fine fuel after the drainage
E_w	TEE (equilibrium water content) of fine fuel after moistening
k_0	intermediate value of k_d
k_d	intermediate drainage speed, FFMC, $\text{Log}_{10}m/\text{day}$
k_1	intermediate value of k_w
k_w	logarithmic moisture speed, $\text{Log}_{10}m/\text{day}$
F	FFMC
F_0	FFMC of the previous day

Οι τιμές υπολογίζονται ως εξής:

1. F of the previous day is F_0
2. $m_0 = 147.2(101-F_0)/(59.5 + F_0)$
3.
 - a. $\text{Av } r_0 \leq 0.5\text{mm}$ τότε $r_0 = r_f$
 $\text{Av } r_0 > 0.5\text{mm}$ τότε $r_f = r_0 - 0.5$
 - b. $\text{Av } m_0 \leq 150$ τότε $m_r = m_0 + 42.5 r_f(e^{-100(251-m_0)})(1-e^{-6.93/r_f})$
 $\text{Av } m_0 > 150$ τότε $m_r = m_0 + 42.5 r_f(e^{-100(251-m_0)})(1-e^{-6.93/r_f}) + 0.0015(m_0 - 150)^2 r_f^{0.5}$
 $\text{Av } m_r > 250$ τότε $m_r = 250$
 - c. $m_r = m_0$
4. $E_d = 0.942 H^{0.679} + 11e^{(H-100)/10} + 0.18(21.1 - T)(1-e^{-0.115H})$
5.
 - a. $\text{Av } m_0 > E_d$ τότε K_d
 $k_0 = 0.424[1-(H/100)^{1.7}] + 0.0694W^{0.5}[1-(H/100)^8]$
 $k_d = k_0 * 0.581 e^{0.0365T}$
 - b. $m = E_d + (m_0 - E_d) * 10^{-k_d}$
6. $\text{Av } m_0 < E_d$ τότε E_w

$$E_w = 0.618H^{0.753} + 10e^{(H-100)/10} + 0.18 (21.1 - T)(1 - e^{-0.115H})$$

7.

a. $Av m_0 < E_w$ τότε k_w

$$k_1 = 0.424 \{1 - [(100-H)/100]^{1.7}\} + 0.0694W^{0.5} \{1 - [(100-H)/100]^8\}$$

$$k_w = k_1 * 0.581 e^{0.0365T}$$

b. $m = E_w - (E_w - m_0) * 10^{-k_d}$

8. $Av E_d \geq m_0 \geq E_w$ τότε $m = m_0$

9. $F = 59.5 (250 - m)/(147.2 + m)$

Duff Moisture Code (DMC):

Το Duff Moisture Code (DMC) είναι μέρος του Καναδικού FWI. Ο κώδικας αντιπροσωπεύει την περιεκτικότητα σε νερό ενός μεσαίου πάχους οργανικού στρώματος. Παρέχει επίσης μια εκτίμηση της ποσότητας μεσαίου μεγέθους καυσίμου για την καύση. Οι απαιτούμενες εισοδοί είναι οι ίδιες με το FWI.

Στις βασικές εξισώσεις χρησιμοποιούμε τα ακόλουθα σύμβολα:

M_0	water content in duff of the previous day
M_r	water content in duff after the rain
M	water content in duff after the drainage
K	logarithmic drainage speed, DMC, $\text{Log}_{10}\text{m/day}$
r_e	real rain, DMC
L_e	effective day length DMC, hours
b	slope factor in DMC
P_0	DMC after the rain
P_r	DMC after the rain
P	DMC

Οι τιμές υπολογίζονται ως εξής:

a. $Av r_0 \leq 1.5$ τότε $r_0 = r_e$

$$Av r_0 > 1.5 \text{ τότε } r_e = 0.92r_0 - 1.27$$

b. $M_0 = 20 + e^{(5.6348 - P_0/43.43)}$

c. $Av P_0 \leq 33 \quad b = 100/(0.5 + 0.3 P_0)$

$$\begin{aligned} \text{Av } 33 < P_0 \leq 65 & \quad b = 14 - 1.3 \ln P_0 \\ \text{Av } P_0 > 65 & \quad b = 6.2 \ln P_0 - 17.2 \end{aligned}$$

$$\text{d. } M_r = M_0 + 1000 r_e / (48.77 + b r_e)$$

$$\text{e. } P_r = 244.72 - 43.43 \ln(M_r - 20)$$

$$\text{Av } P_r < 0 \text{ τότε } P_r = 0$$

$$P_r = P_0$$

2. Search L_e in the following table

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
L_e	6.5	7.5	9.0	12.8	13.9	13.9	12.4	10.9	9.4	8.0	7.0	6.0

$$3. \quad K = 1.894 (T + 1.1)(100 - H)L_e * 10^{-6}$$

$$\text{Av } T < -1.1 \text{ τότε } T = -1.1$$

$$4. \quad P = P_0 + 100K$$

Drought Code (DC):

Το Drought Code (DC) είναι μέρος του καναδικού FWI. Αντιπροσωπεύει μια βαθμολογία της περιεκτικότητας του νερού ενός βαθιού, συμπαγές οργανικού στρώματος στο έδαφος. Είναι ένας καλός δείκτης της εποχικής ξηρασίας σε μεγάλου μεγέθους καύσιμα. Οι απαιτούμενες εισοδοί είναι οι ίδιες με το FWI.

Στις βασικές εξισώσεις χρησιμοποιούμε τα ακόλουθα σύμβολα:

Q	equivalent humidity of DC, multiple to 0.254mm
Q_0	equivalent humidity of DC of the previous day
Q_r	equivalent humidity after the rain
r_d	real rain, DC
V	Potential evapotranspiration, multiple of 0.254mm of water/day
L_f	effective day length DC, hours
D_0	DC of the previous day
D_r	DC after the rain
D	DC

Οι τιμές υπολογίζονται ως εξής:

1.

- a. $\text{Av } r_0 \leq 2.8 \text{ τότε } r_0 = r_d$
 $\text{Av } r_0 > 2.8 \text{ τότε } r_d = 0.83r_0 - 1.27$
- b. $Q_0 = 800e^{-D_0/400}$
- c. $Q_r = Q_0 + 3.937r_d$
- d. $D_r = 400 \ln(800/Q_r)$
 $\text{if } D_r < 0 \text{ then } D_r = 0$
 $D_r = D_0$

2. Search L_f in the following table

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
L_f	-1.6	-1.6	-1.6	0.9	3.8	5.8	6.4	5.0	2.4	0.4	-1.6	-1.6

3. $V = 0.36(T + 2.8) + L_f$

$\text{Av } T < -2.8 \text{ τότε } T = -2.8$

$\text{Av } V < 0 \text{ τότε } V = 0$

4. $D = D_0 + 0.5V$

Initial Spread Index (ISI):

Το Initial Spread Index (ISI) είναι μέρος του καναδικού FWI. Παρέχει μια εκτίμηση της αναμενόμενης εξάπλωσης του μετώπου της πυρκαγιάς, χωρίς να λαμβάνεται υπόψη η ποικιλομορφία των καυσίμων. Συνδυάζει το αποτέλεσμα του FFMC και της αιολικής ενέργειας. Οι απαιτούμενες είσοδοι είναι οι ίδιες με το FWI.

Στις βασικές εξισώσεις χρησιμοποιούμε τα ακόλουθα σύμβολα:

$f(W)$ wind factor

$f(F)$ fine fuel humidity factor

m water content in fine fuel after the drainage

R ISI

Οι τιμές υπολογίζονται ως εξής:

1. $f(W) = e^{0.05039W}$
2. $f(F) = 91.9e^{-0.1386m} [1 + m^{5.31} / (4.93 * 10^7)]$
3. $R = 0.208 f(W) f(F)$

Build Up Index (BUI):

Ο Build Up Index (BUI) αποτελεί τμήμα του καναδικού FWI. Αντιπροσωπεύει μια βαθμολογία των συνολικών καυσίμων που διατίθενται για καύση. Συνδυάζει τους δύο κωδικούς DMC και DC. Οι απαιτούμενες εισοδοί είναι οι ίδιες με το FWI.

Στις βασικές εξισώσεις χρησιμοποιούμε τα ακόλουθα σύμβολα:

U	build up index (BUI)
P	DMC
D	DC

Οι τιμές υπολογίζονται ως εξής:

$$\text{Αν } P \leq 0.4D \text{ τότε } U = 0.8 PD / (P + 0.4D)$$

$$\text{Αν } P > 0.4D \text{ τότε } U = P - [1 - 0.8D / (P + 0.4D)][0.92 + (0.0114P)^{1.7}]$$

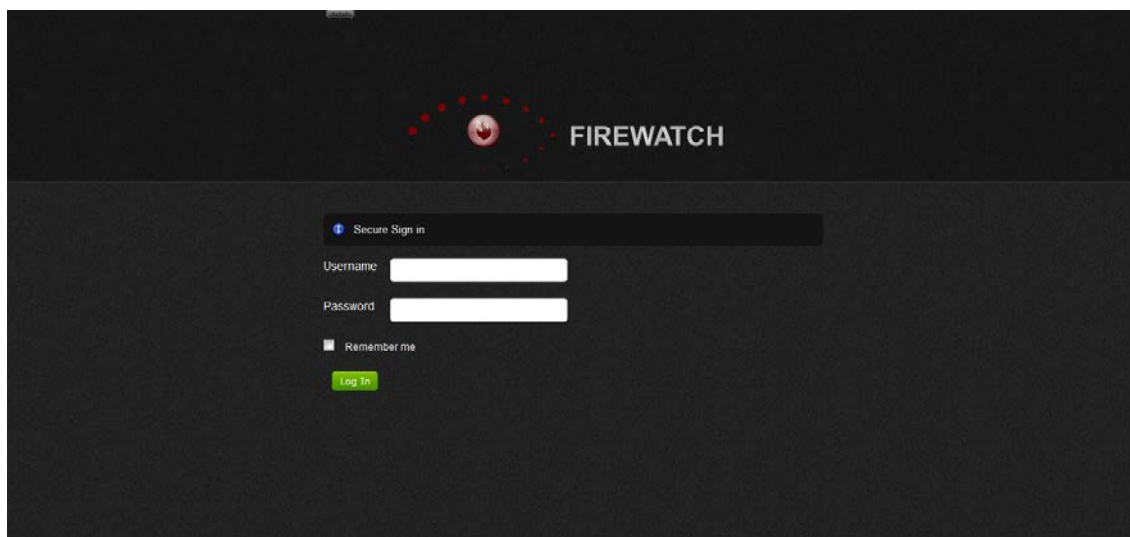
7.5.4 Μεθοδολογία ανάπτυξης και παρουσίαση εφαρμογής

Πιο κάτω θα παρουσιαστεί το web application. Μέσα από τα screenshots θα περιγράφεται η λειτουργία του κάθε υποσυστήματος όπως περιγράφηκε πιο πάνω.

Είσοδος στο σύστημα:

Η είσοδος στο σύστημα (Εικόνα 32) γίνεται με χρήση username και password. Μόνο εξουσιοδοτημένοι χρήστες μπορούν να εισέρχονται στο σύστημα.

Υποστηρίζονται persistent cookies με μέγιστη διάρκεια 1 μέρα για λόγους ασφαλείας. Ακολούθως γίνεται expire αν κάποιος χρήστης δεν εισέλθει ξανά για να ανανεωθεί το cookie.



Εικόνα 32: Screenshot Εισαγωγής στο Σύστημα FireWatch

Dashboard:

Το Dashboard όπως φαίνεται στην Εικόνα 33 παρέχει μια επισκόπηση του όλου συστήματος. Οι χρήστες μπορούν να δουν σύντομα και περιληπτικά τις τελευταίες μετρήσεις των αισθητήρων σε μορφή πίνακα και γραφικών παραστάσεων.

Οι αισθητήρες απεικονίζονται στον χάρτη, ενώ υπάρχουν επιλογές για την διαχείριση του χάρτη και των επιπέδων που προβάλλονται σ' αυτόν. (π.χ. το επίπεδο που παρουσιάζει τη βλάστηση).



Εικόνα 33: Screenshot Dashboard

Η εικόνα Εικόνα 34 δείχνει την αντιστοίχιση του είδους βλάστησης με κάθε χρώμα που φαίνεται στον χάρτη.



Εικόνα 34: Legend

WSN Monitoring:

Το WSN Monitoring όπως φαίνεται στην Εικόνα 35 και Εικόνα 36 παρέχει τη δυνατότητα στους χρήστες να χειριστούν τα δεδομένα των αισθητήρων. Συγκεκριμένα οι χρήστες μπορούν να ζητήσουν δεδομένα με διάφορα queries σχετικά με τις μετρήσεις των αισθητήρων και την κατάσταση του δικτύου.

Τα αποτελέσματα παρουσιάζονται σελιδοποιημένα, με δυνατότητα ταξινόμησης με οποιαδήποτε στήλη και σε οποιαδήποτε σειρά – φθίνουσα ή αύξουσα – κάτι που δίνει μεγαλύτερη ελευθερία χειρισμού των δεδομένων.

The screenshot shows the WSN Monitoring interface. On the left is a sidebar with navigation options: Dashboard, WSN Monitoring (selected), WSN Data, WSN Chart, Enhanced G.I.S., Collaboration, Account Settings, and Logout. The main area displays two tables. The top table is 'Sensor Data' and the bottom table is 'Sensor Health Data'. Both tables have a search bar and pagination controls.

Sensor Data Table:

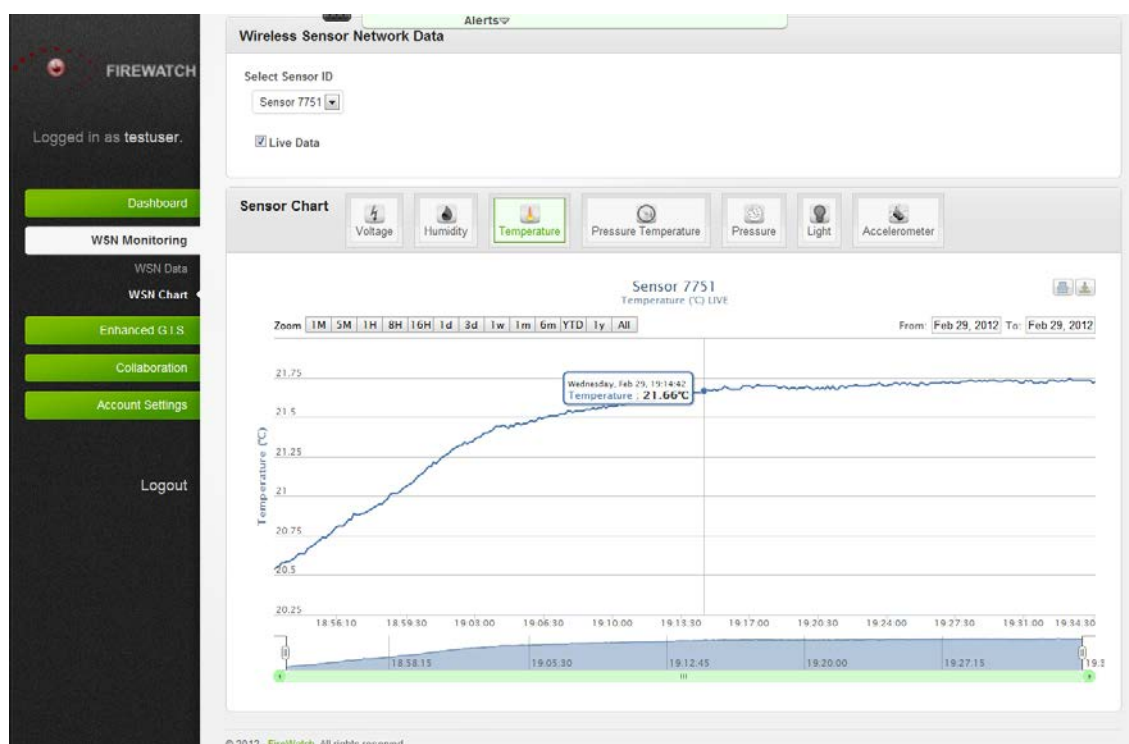
Sensor ID	Latitude	Longitude	Date/Time	Temperature	Humidity	Pressure	Light	Soil Moisture	Accelerometer X	Accelerometer Y	Accelerometer Z
7756	33.259864	34.966923	2/29/2012 9:32:47 PM	25.40 °C	27.93 %	980.50 mbar	369.64 Lux	3.14 V	-0.00 g	-0.42 g	0.00 g
7754	33.269653	34.961000	2/29/2012 9:32:43 PM	22.09 °C	32.81 %	979.37 mbar	478.17 Lux	3.09 V	-0.02 g	0.00 g	0.00 g
7756	33.259864	34.966923	2/29/2012 9:32:39 PM	25.46 °C	29.90 %	980.38 mbar	369.64 Lux	3.14 V	-1.30 g	-0.58 g	-0.40 g
7754	33.269653	34.961000	2/29/2012 9:32:35 PM	22.08 °C	32.92 %	979.45 mbar	478.17 Lux	3.09 V	0.00 g	0.02 g	0.40 g
7756	33.259864	34.966923	2/29/2012 9:32:31 PM	25.50 °C	30.22 %	980.44 mbar	369.64 Lux	3.14 V	1.30 g	-0.40 g	0.00 g
7754	33.269653	34.961000	2/29/2012 9:32:27 PM	22.05 °C	32.84 %	980.50 mbar	369.64 Lux	3.14 V	-1.30 g	-0.42 g	0.00 g
7756	33.259864	34.966923	2/29/2012 9:32:23 PM	25.51 °C	30.08 %	979.37 mbar	478.17 Lux	3.09 V	-0.02 g	0.00 g	0.00 g
7754	33.269653	34.961000	2/29/2012 9:32:19 PM	22.03 °C	32.88 %	980.38 mbar	369.64 Lux	3.14 V	-1.30 g	-0.58 g	-0.40 g
7756	33.259864	34.966923	2/29/2012 9:32:15 PM	25.54 °C	30.44 %	979.45 mbar	478.17 Lux	3.09 V	0.00 g	0.02 g	0.40 g
7754	33.269653	34.961000	2/29/2012 9:32:12 PM	22.01 °C	32.56 %	980.44 mbar	369.64 Lux	3.14 V	1.30 g	-0.40 g	0.00 g

Sensor Health Data Table:

Sensor ID	Latitude	Longitude	Date/Time	Health Packet	Node Packets	Forwarders	Dropped	Batteries	Battery	Power Sum	Board ID	Parent	Quality TX	Quality RX	Path Cost	Parent R
7756	33.259864	34.966923	2/29/2012 9:32:38	10.00 %	70.00 %	0.00 %	30.00 %	0.00 %	3.00 V	0.00 mAh	133	7754	100.00 %	100.00 %	24	63
7754	33.269653	34.961000	2/29/2012 9:31:30	6.87 %	93.33 %	0.00 %	6.87 %	0.00 %	3.10 V	0.00 mAh	133	0	100.00 %	100.00 %	0	38
7754	33.269653	34.961000	2/29/2012 9:29:30	7.14 %	89.29 %	0.00 %	10.71 %	0.00 %	3.10 V	0.00 mAh	133	0	100.00 %	100.00 %	0	37
7756	33.259864	34.966923	2/29/2012 9:28:01	7.41 %	88.89 %	0.00 %	11.11 %	0.00 %	3.00 V	0.00 mAh	133	0	100.00 %	96.67 %	0	32
7754	33.269653	34.961000	2/29/2012 9:27:30	10.00 %	70.00 %	0.00 %	30.00 %	0.00 %	3.10 V	0.00 mAh	133	0	100.00 %	100.00 %	0	42
7756	33.259864	34.966923	2/29/2012 9:26:01	10.00 %	70.00 %	0.00 %	30.00 %	0.00 %	3.00 V	0.00 mAh	133	0	100.00 %	100.00 %	0	46
7756	33.259864	34.966923	2/29/2012 9:22:34	6.35 %	75.00 %	0.00 %	25.00 %	3.13 %	3.00 V	0.00 mAh	133	0	100.00 %	100.00 %	0	47
7777	33.251211	34.941250	2/29/2012 7:34:07	5.88 %	89.15 %	0.00 %	0.85 %	18.52 %	3.10 V	0.00 mAh	133	0	86.67 %	93.33 %	0	30
7770	33.228546	34.948453	2/29/2012 7:34:04	6.00 %	99.14 %	0.00 %	0.86 %	12.57 %	3.10 V	0.00 mAh	133	0	86.67 %	100.00 %	0	34
7780	33.255355	34.962313	2/29/2012 7:34:02	5.63 %	93.03 %	5.90 %	1.07 %	16.35 %	3.10 V	0.00 mAh	133	0	66.67 %	93.33 %	0	21

Εικόνα 35: Screenshot WSN Monitoring – WSN Data

Επίσης εκτός από την προβολή των δεδομένων σε πινακοειδή μορφή οι χρήστες μπορούν να δουν γραφικά τα δεδομένα υπό την μορφή γραφικών. Στις γραφικές παραστάσεις υπάρχει η δυνατότητα να εκτελέσει συγκεκριμένα queries που όπως είδαμε υποστηρίζονται από τα stored procedures. Εναλλακτικά μπορεί να δει Live την ενημέρωση της γραφικής. Σε αυτή τη περίπτωση καθώς λαμβάνονται νέες μετρήσεις θα παρουσιάζονται αυτόματα στην υπάρχον γραφική παράσταση.

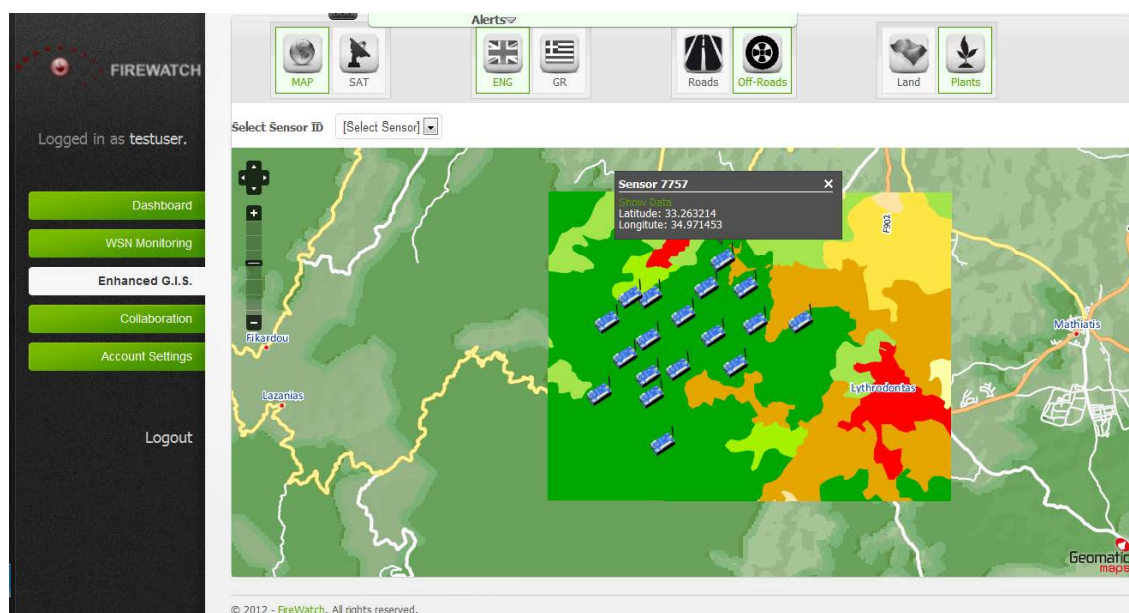


Εικόνα 36: Screenshot WSN Monitoring – WSN Chart

Enhanced G.I.S.:

Το Enhanced G.I.S. όπως φαίνεται στην Εικόνα 37 παρέχει τη δυνατότητα στους χρήστες να χειριστούν τα γεωγραφικά δεδομένα των αισθητήρων.

Ο χάρτης δίνει τις πληροφορίες όπως την θέση των αισθητήρων, τις τελευταίες μετρήσεις, τις πιθανές περιοχές πυρκαγιάς, διάφορα επίπεδα του χάρτη, τις τοποθεσίες δασικών πυροσβεστικών σταθμών κτλ.

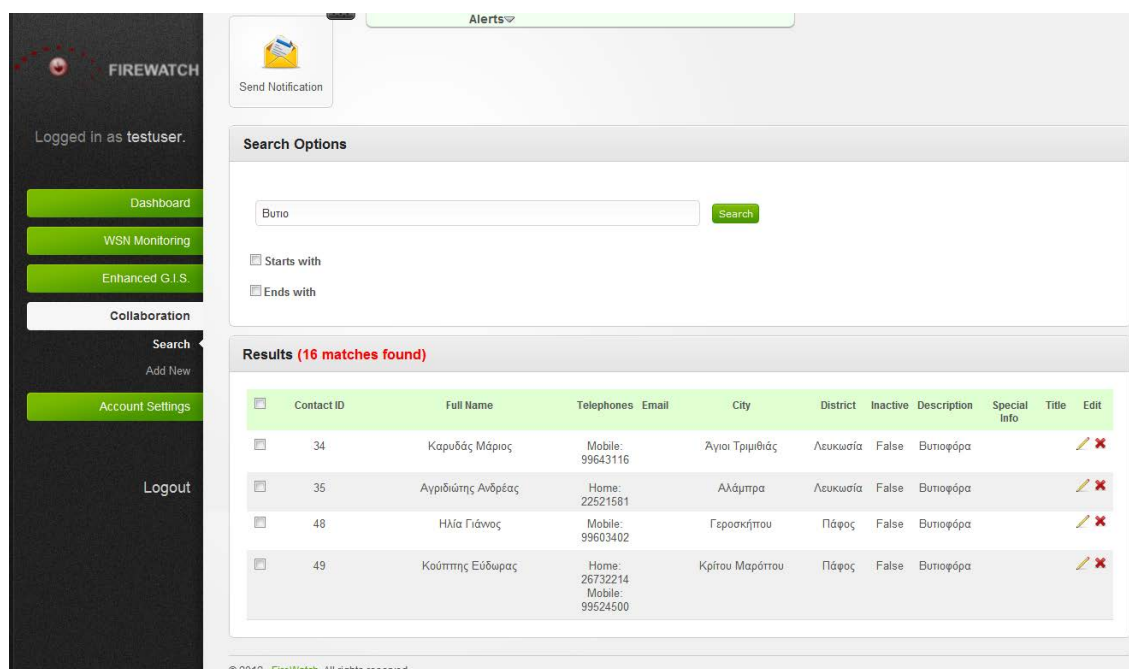


Εικόνα 37: Screenshot Enhanced G.I.S.

Collaboration:

Το Collaboration όπως φαίνεται στην Εικόνα 38 και Εικόνα 40 παρέχει τη δυνατότητα στους χρήστες να χειριστούν τις επαφές του συστήματος.

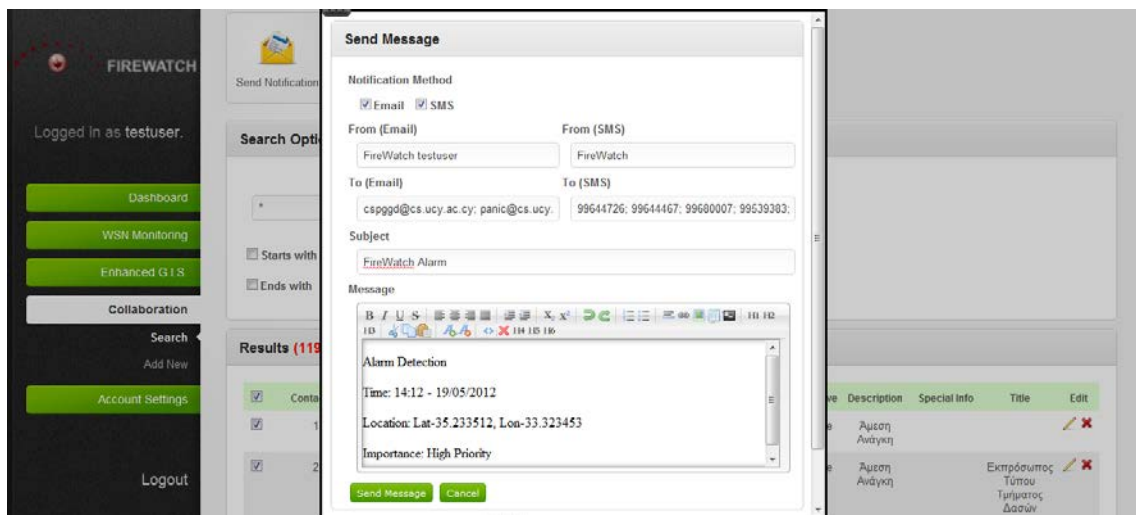
Ο χρήστης μπορεί να κάνει αναζήτηση μέσω ενός κουμπιού και να λάβει πληροφορίες για συγκεκριμένο υποσύνολο επαφών. Ακολούθως μπορεί να τροποποιήσει μιαν επαφή ή να την διαγράψει.



Εικόνα 38: Screenshot Collaboration-Search

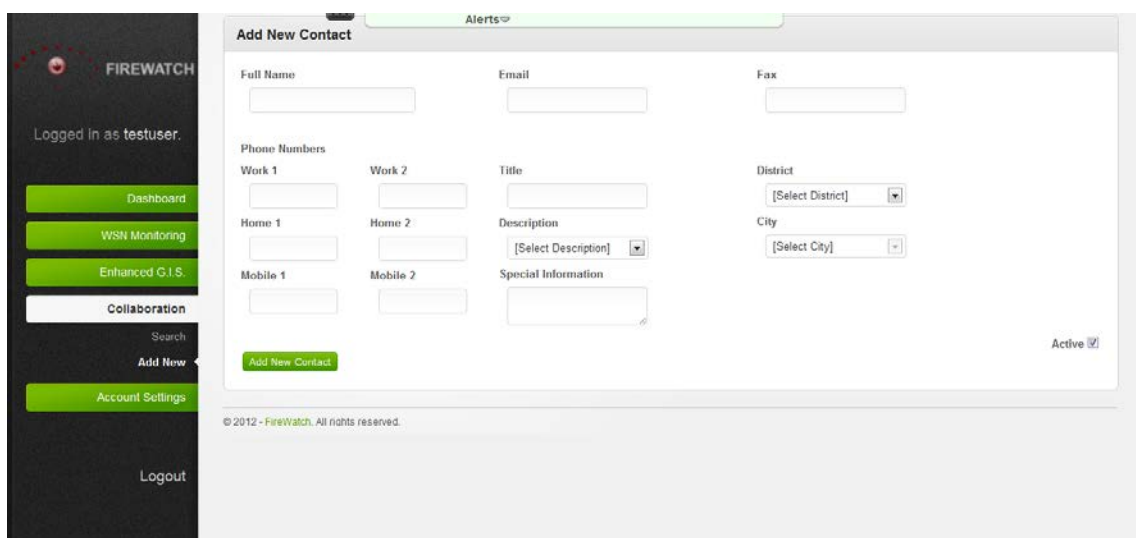
Ακόμη μπορεί να στείλει notifications όπως φαίνεται στην Εικόνα 39 με την χρήση του κουμπιού “Send Notification”. Στο παρόν στάδιο της συγγραφής υποστηρίζονται 2 ειδών notification-sms και μηνύματα emails.

Με την επιλογή από τις επαφές, τα ηλεκτρονικά ταχυδρομεία και τα κινητά τηλέφωνα τους χρησιμοποιούνται για να αποσταλούν μαζικά notifications. Για την αποστολή των μηνυμάτων sms χρησιμοποιούμε SMS Provider. Περισσότερες πληροφορίες για το API που παρέχεται βρίσκονται εδώ [18].



Εικόνα 39: Screenshot Collaboration-Send Notification

Δίνεται η δυνατότητα στους χρήστες να προσθέσουν νέες επαφές συμπληρώνοντας τα στοιχεία τους που είναι απαραίτητα.

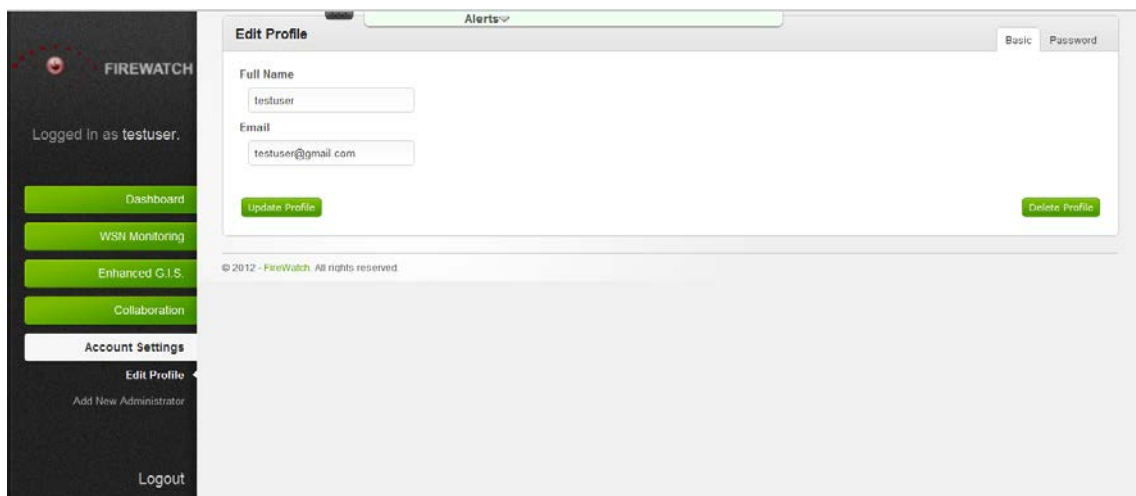


Εικόνα 40: Screenshot Collaboration-Add New

Account Settings:

Μέσα από τα Account Settings όπως φαίνεται στην Εικόνα 41 δίνεται η δυνατότητα στους χρήστες να χειριστούν το προφίλ τους, να αλλάξουν δεδομένα, κωδικό κτλ.

Επίσης από τα Account Settings, μπορούν να προστεθούν νέοι χρήστες στο σύστημα (νέοι admin). Στο παρόν στάδιο όλοι μπορούν να δημιουργήσουν νέους administrator. Σε μελλοντικό στάδιο, όταν έχουμε περισσότερα δεδομένα για το ποιος θα χειρίζεται το σύστημα στο Τμήμα Δασών θα έχει την συγκεκριμένη πρόσβαση.



Εικόνα 41: Screenshot Account Settings-Edit Profile

Κεφάλαιο 8

Συμπεράσματα

8.1	Ανασκόπηση συστήματος.....	108
8.2	Μελλοντική εργασία.....	109

8.1 Ανασκόπηση συστήματος

Αυτή η διπλωματική εργασία είχε ως κύριο στόχο την κατασκευή ενός καινοτόμου συστήματος που θα χρησιμοποιηθεί από το Τμήμα Δασών Κύπρου. Μελετώντας διάφορα άρθρα που αφορούν την ανίχνευση πυρκαγιών σε δάση παρατηρήσαμε ότι πολλά από αυτά υστερούν σε διάφορα σημεία. Στόχος του συστήματος μας είναι η έγκαιρη ανίχνευση και πρόβλεψη δασικών πυρκαγιών χρησιμοποιώντας ένα μίγμα τεχνολογιών όπως ασύρματα δίκτυα αισθητήρων, γεωγραφικά δεδομένα και σύστημα συνεργασίας με σκοπό την γρήγορη ανταπόκριση από τους διάφορους φορείς.

Ακολούθως καθορίσαμε τις απαιτήσεις του συστήματος, την αρχιτεκτονική που θα ακολουθήσουμε, τη σχεδίαση και τελικά την υλοποίηση του συστήματος FireWatch. Κάθε επιμέρους σύστημα έχει αναλυθεί ώστε να παραχθεί ένα ποιοτικό αποτέλεσμα.

Ειδικότερα σημαντική είναι η επεκτασιμότητα του συστήματος, χρησιμοποιώντας virtual machines, κάτι που επιτρέπει την διανομή κάθε υποσυστήματος σε διαφορετική τοποθεσία.

Η χρήση ενός καλά αποδεκτού αλγορίθμου ανίχνευσης όπως ο Καναδικός FWI επιτρέπει την πρόωρη ανίχνευση και πρόβλεψη κίνδυνου πυρκαγιάς.

8.2 Μελλοντική εργασία

Έχοντας υλοποιήσει επιτυχώς το σύστημα FireWatch, δεν σημαίνει ότι είναι το τελικό. Υπάρχουν αρκετά σημεία βελτίωσης του υπάρχον συστήματος για το κάθε ένα από τα επιμέρους συστατικά.

Συγκεκριμένα, πιθανές βελτιστοποιήσεις του συστήματος αφορούν:

- Το ασύρματο δίκτυο καταγράφει μετρήσεις σε τακτά χρονικά διαστήματα για όλους τους αισθητήρες. Αυτό σημαίνει ότι η βάση δεδομένων γεμίζει με δεδομένα που ίσως δεν χρειάζονται. Θα μπορούσαμε να εγκαταστήσουμε σε κάθε αισθητήρα το KSpot+ [19] [20]. Με το KSpot+ η κατανάλωση ενέργειας μειώνεται και οι μετρήσεις περιορίζονται μόνο στις πιο σημαντικές, αφού έχει τη δυνατότητα να:
 - ο Δημιουργεί ισορροπημένες τοπολογίες δικτύου
 - ο Αποτελεσματικότερη μετάδοση/λήψη δεδομένων
 - ο Μείωση αριθμού και μεγέθους μεταδιδόμενων πακέτων
 - ο Ρυθμίζει το παράθυρο που δεν είναι σε sleep mode οι αισθητήρες
 - ο Επιτρέπει δυνατότητα κατάταξης μιας πλειάδας μέσω Top-K query (να ζητώ τα TOP-K δεδομένα π.χ. βρες τους K αισθητήρες με την μεγαλύτερη θερμοκρασία)

Λεπτομέρειες για το λογισμικό ανοικτού πηγαίου κώδικα KSpot+ βρίσκονται εδώ: <http://www.cs.ucy.ac.cy/~panic/kspot/#overview>

- Οι προειδοποιήσεις voice messages για σταθερά τηλέφωνα χρειάζονται ξεχωριστό API για να λειτουργήσουν, δηλαδή για να μπορούμε να παίρνουμε τηλέφωνο και να ακούεται το ηχογραφημένο μήνυμα.
- Ειδική έκδοση FireWatch σε smartphones
 - ο Android, Windows Phone, iOS
 - ο Πρόσβαση από παντού και πάντα
 - ο Υποστήριξη Screen Notifications
 - ο Υποστήριξη Push Notifications για άμεση ενημέρωση
- Το layer για να βρίσκει εναλλακτικά μονοπάτια σε χωματόδρομους είναι υπό ανάπτυξη από την Geomatic.
- Επέκταση χάρτη ώστε να προβάλλονται γεωγραφικές τοποθεσίες χρηστών καθώς κινούνται στο χώρο της πυρκαγιάς.

- Ενσωμάτωση υφιστάμενων συστημάτων του Τμήματος Δασών με το FireWatch ώστε να παραχθεί ένα υβριδικό σύστημα που θα προσφέρει μεγαλύτερη ακρίβεια και έλεγχο.
- Ειδική έκδοση του FireWatch για χρήση από απλούς πολίτες θα μπορούσε να χρησιμοποιηθεί για έλεγχο δασικών πυρκαγιών με ένα crowdsourcing τρόπο. Έτσι θα συνέβαλλαν άνθρωποι στην έγκαιρη ανίχνευση πυρκαγιών.

Επίσης όσο αφορά την διαδικτυακή εφαρμογή, παίρνουμε feedback από τους χρήστες της (Τμήμα Δασών). Το σύστημα είναι ανοικτό για νέες εισηγήσεις, βελτιστοποιήσεις, τροποποιήσεις.

Βιβλιογραφία

- [1] «Δασικές Πυρκαγιές,» Τμήμα Δασών, [Ηλεκτρονικό]. Available: <http://www.moa.gov.cy/moa/fd/fd.nsf/a028aa7a82db918cc2256f1c0036a11c/28ab9b8c984e2c7fc2256f1c00375f64?OpenDocument>. [Πρόσβαση 17 Μάιος 2012].
- [2] «TinyOS,» TinyOS , [Ηλεκτρονικό]. Available: <http://www.tinyos.net/>. [Πρόσβαση 17 Μάιος 2012].
- [3] «FireBug: wireless wildfire monitoring,» University of California - Berkeley, [Ηλεκτρονικό]. Available: <http://sourceforge.net/projects/firebug/>. [Πρόσβαση 17 Μάιος 2012].
- [4] J. Solobera, «Detecting Forest Fires using Wireless Sensor Networks with Wasp mote,» Libelium, [Ηλεκτρονικό]. Available: http://www.libelium.com/wireless_sensor_networks_to_detec_forest_fires/.
- [5] «Geomatic Technologies,» T.C. Geomatic, [Ηλεκτρονικό]. Available: <http://www.geomatic.com.cy/>.
- [6] «IRIS,» [Ηλεκτρονικό]. Available: <http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=135%3Airis>.
- [7] «MEMSIC Homepage,» MEMSIC, [Ηλεκτρονικό]. Available: <http://www.memsic.com>.
- [8] «MTS400 Sensor Board,» [Ηλεκτρονικό]. Available: http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=174%3Amts400_420.
- [9] «Sensirion,» Sensirion-The sensor company, [Ηλεκτρονικό]. Available: <http://www.sensirion.com/>. [Πρόσβαση 5 May 2012].
- [10] «Intersema MS55ER,» Measurement Specialties, [Ηλεκτρονικό]. Available: <http://www.intersema.ch/>. [Πρόσβαση 12 Μάιος 2012].
- [11] «Taos TSL2550,» TAOS, [Ηλεκτρονικό]. Available: <http://www.taosinc.com/>. [Πρόσβαση 12 Μάιος 2012].
- [12] Crossbow, «XMesh User's Manual,» April 2007 . [Ηλεκτρονικό]. Available: http://bullseye.xbow.com:81/Support/Support_pdf_files/XMesh_Users_Manual.pdf.
- [13] Crossbow, «XServe User's Manual,» April 2007. [Ηλεκτρονικό]. Available: http://bullseye.xbow.com:81/Support/Support_pdf_files/XServe_Users_Manual.pdf.
- [14] Crossbow, «MoteView User's Manual,» May 2007. [Ηλεκτρονικό]. Available: http://bullseye.xbow.com:81/Support/Support_pdf_files/MoteView_Users_Manual.pdf.
- [15] «Geomatic's Map API,» [Ηλεκτρονικό]. Available: <http://www.geomatic.com.cy/geomaps2/api/doc/>.
- [16] «ASP.NET SQL Server Registration Tool (Aspnet_regsql.exe),» Microsoft, [Ηλεκτρονικό]. Available: [http://msdn.microsoft.com/en-us/library/ms229862\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/ms229862(v=vs.100).aspx).
- [17] Van Wagner CE, Pickett TL, «Equations and Fortran program for the Canadian

- Forest Fire Weather Index System,» Forestry Technical Report 33, Ottawa, 1987.
- [18] «HTTP SMS API - sendmessage method,» OZEKI NG SMS Gateway, [Ηλεκτρονικό]. Available: http://www.ozekisms.com/index.php?ow_page_number=413.
 - [19] P. Andreou, D. Zeinalipour-Yazti, M. Vassiliadou, P. K. Chrysanthis, G. Samaras, «KSpot: Effectively Monitoring the K Most Important Events in a Wireless Sensor Network,» σε *The 25th International Conference on Data Engineering March (ICDE'09)*, Shanghai, China, 2009.
 - [20] Panayiotis G. Andreou, Demetrios Zeinalipour-Yazti, George Samaras, Panos K. Chrysanthis, «Towards a Network-aware Middleware for Wireless Sensor Networks,» σε *8th International Workshop on Data Management for Sensor Networks*, Seattle, WA, USA, 2011.
 - [21] «MODIS,» NASA, [Ηλεκτρονικό]. Available: <http://modis.gsfc.nasa.gov/>. [Πρόσβαση 17 Μάιος 2012].
 - [22] «FireWatch: An Early Warning System for Forest Fires, successfully in the global use,» IQ wireless GmbH, [Ηλεκτρονικό]. Available: <http://www.fire-watch.de/>. [Πρόσβαση 17 Μάιος 2012].
 - [23] «The IR Radar,» [Ηλεκτρονικό]. Available: http://www.duramproducts.com.au/prod_firedetect.html. [Πρόσβαση 17 Μάιος 2012].
 - [24] Kostas Kalabokidis, Gavriil Xanthopoulos, Peter Moore, David Caballero, George Kallos, Juan Llorens, Olga Roussou and Christos Vasilakos, «Decision support system for forest fire protection,» σε *European Journal of Forest Research*, 2011.

Παράρτημα Α

Σε αυτό το παράρτημα παρατίθεται ο κώδικας για τον υπολογισμό του CRC πεδίου του πακέτου MTS400 όπως περιγράφηκε στην ενότητα 0:

```
uint16_t xcrc_byte(uint16_t crc, uint8_t b){
    uint8_t i;

    crc = crc ^ b << 8;
    i = 8;
    do if (crc & 0x8000)
        crc = crc << 1 ^ 0x1021;
    else
        crc = crc << 1;

    while (--i);

    return crc;
}

int xcrc_calc(char *packet, int index, int count) {
    int crc = 0;

    while (count > 0) {
        crc = xcrc_byte(crc, packet[index++]);
        count--;
    }

    return crc;
}
```

Παράρτημα Β

Πιο κάτω παρατίθεται το API της Geomatic. Οι μέθοδοι του μπορούν να χρησιμοποιηθούν μέσω JavaScript.

Class Details
geomatic.maps.Map (mapdiv, options, callback) Creates a new Geomatic map object var map = new geomatic.maps.Map('map', { base: 'map', satellite: true, lang: 'EN', zoombar: 'full'}, maploaded); Parameters: {string} mapdiv The id of the div in which the map will be rendered {object null} options A JSON object that holds the options Options: - lang {string} [EN GR] - base {string} [map sat] - satellite {boolean} [true false] - zoombar {string} [small full] If not defined no zoombar is displayed - zoom {number} the initial zoom level from 8 to 18 - center {array} the center lon lat in an array {string null} callback The function name to be called after the initialization of the object Returns: {object} The geomatic map object
Method Details
{object} addLayer (options) Add Layer to the map object Parameters: options {Object} A JSON object defining the parameters of the layer. Options: - id {string} the identifier of the layer - type {string} [wms vector] the layer type as defined by OpenLayers - visibility {boolean} [true false] specifies if the layer will be visible as soon as it is added - displayInLayerSwitcher {boolean} - transparent {boolean} - style {object} an OpenLayers style object - strategies {object} an OpenLayers strategies object Returns: {object} The OpenLayers layer object just added
{object} addPoint (id, lon, lon, icon, label, description) Adds a point feature on the map given the specific data Parameters: {number string} id The unique id of the point {string} lon The longitude value {string} lon The latitude value {string} icon The path of the icon of the point {string} label

The label of the point (will be displayed on hover) {string html} description The description of the point that will be displayed in the popup. Can be html Returns: {object} the OpenLayers feature object
clearNavigation() Clears the navigation layers
clearNavigationLine() Clears the navigation line layer. Used when changing the navigation while in navigation
{object} displayLonLat(lon, lat, zoom, displayPopup) Displays a specific lon lat on the map Parameters: {string} lon Longitude {string} lat Latitude {integer} zoom The zoom level to zoom in to {boolean} displayPopup Display feature popup (optional) Returns: {object} The OpenLayers feature object
displayPoiCategory(categoryId) Displays the features of a specific category Parameters: {string} categoryId The id of the category to display
displayPoint(point, displayPopup) Pans above the specific point Parameters: {object} point The point object {boolean} displayPopup Display feature popup
{object} featureFromLonLatString(lon, lat) Creates a feature object from the given lon lat Parameters: {string} lon Longitude {string} lat Latitude Returns: {object} The OpenLayers feature object
getCenter() Returns the center of the map
{string} getLanguage() Get the map language Returns: {string} [EN GR] the map language code

{array} getLayerArray() Gets the array of the layers added to the map Returns: {array} The layers array
{integer} getZoom() Returns the current map's zoom level Returns: {integer} The map zoom
{integer} getZoomOffset() Get the zoom offset Returns: {integer}
hideLayer(layer) Hides a specific layer from the map Parameters: {Object} layer the OpenLayers layer object
hideLayerById(layerId) Hides a specific layer from the map given the layer id Parameters: {string} layerId the OpenLayers layer object id
{object} lonlatFromPoint(xy) Gets the lonlat object from the specific xy point of the screen Parameters: {array} xy An array with the x and y pixels of the screen Returns: {object} The lonlat object
navigate(from, to, navMode, callback) Triggers the navigation calculation process Parameters: {object} from A js object containing the lon lat of the starting point {object} to A js object containing the lon lat of the ending point {string} navMode [var walk] The mode the navigation will be calculating the results {string} callback The function called when the navigation is completed
removePoiCategory(categoryId) Removes the features of a specific category Parameters: {string} categoryId The id of the category to display
rgeocAddressLonLat(lon, lat, zoom) Displays the address of a specific lon lat Parameters: {string} lon The longitude

{string} lat The latitude {integer null} zoom The zoom level the map should zoom in to
rgeocAddressXY(xy, zoom) Displays the address of a specific xy of the screen Parameters: {array} xy An array with x and y pixels {integer null} zoom The zoom level the map should zoom in to
rgeocPoiLonLat(lon, lat) Displays the nearest feature of a specific lon lat Parameters: {string} lon The longitude {string} lat The latitude
rgeocPoiXY(xy) Displays the nearest feature of a specific xy of the screen Parameters: {array} xy An array with x and y pixels
selectFeature(featureId, showPopup, layer) Selects a feature already added to the map Parameters: {string} featureId The unique identifier of the feature {boolean} showPopup Whether to display a popup or not {object} layer The OpenLayers layer object
setBaseLayer(base) Sets the base layer of the map. Parameters: {string} base [map sat] The base layer id
setCenter(lon, lat) Centers the map to the specific lon/lat Parameters: lon {Float} the longitude lat {Float} the latitude
setLanguage(lang) Sets the language of the map. Parameters:

{string} lang [EN GR] The language to set to
setLayerOpacity(layerId, opacity) Set a layer's opacity Parameters: {string} layerId The OpenLayers layer object id {double} opacity The opacity value from 0 to 1
setNaviFrom(fromPoint) Sets a specific point object as the starting point of the navigation Parameters: {object} fromPoint The from point object
setNaviTo(toPoint) Sets a specific point object as the ending point of the navigation Parameters: {object} toPoint The to point object
showInstruction(instrId) Selects the instruction feature produced by navigation Parameters: {string} instrId The instrauction unique identifier
showLayer(layer) Shows a specific layer from the map Parameters: {Object} layer The OpenLayers layer object
showLayerById(layerId) Shows a specific layer from the map given the layer id Parameters: {string} layerId The OpenLayers layer object id
zoomTo(zoom) Zooms map to a specific zoom level Parameters: {int} zoom The zoom level to zoom in
Event Details
locate() Triggers the locate control as defined in OpenLayers
onFeatureOut(feature) Parameters: feature
onFeatureOver(feature)

Parameters: feature
onFeatureSelect(feature) Parameters: feature
onFeatureUnselect(feature) Parameters: feature
onMouseDown(e) Parameters: e
onMouseUp(e) Parameters: e

Παράρτημα Γ

Παράμετροι που μπορεί να πάρει η εντολή xserve.exe

```
Usage: xserve <-?|r|a|p|c|xr|xp|xc|dbxmlr|xmlp|xmlc|v|alert|m>
<-l=tablename>
<-dbserver=servername> <-dbport=portnum>
<-dbname=database name> <-dbuser=username>
<-dbpasswd=password>
<-h=path,hostname,portnum,config_file>
<-m=com,baud,protocol,slaveaddress,defaultregistervaluesas>
<-xmlfile=filename> <-xmlport=portnum>
[<-sf=hostname:port> | <-fsf=hostname:port> | <-device=dev>]
<-port=num> <-baud=num> <-platform=plt>
<-debug=level>
<-configfiles=filename:filename:>
<-loadparsers=filename:filename:...>
<-loaddatasinks=filename:filename:...>
<-heartbeat=<num missed>
-? = display help [help]
-r = raw display of tos packets [raw]
-a = ascii display of tos packets [ascii]
-p = parsed display of tos packets [parsed]
-c = converted display of tos packets [conveted]
-xr = raw tos packets xported to file [export raw]
-xp = parsed tos packets exported to file [export parsed]
-xc = converted tos packets exported to file [export converted]
-db = parsed tos packets exported to db [database parsed]
-dbserver = database server name (default=localhost)
-dbport = database server port number (default=5432)
-dbdname = database name (default=MoteView db)
-dbuser = database user (default=MoteView user)
-dbpasswd = database user password (default=MoteView user
password)
-l = parsed tos packets exported to db
(deprecated) [database parsed]
-xmlr = raw tos packets exported to xml [xml raw]
-xmlp = parsed tos packets exported to xml [xml parsed]
-xmlc = converted tos packets exported to xml [xml converted]
-xmlfile = file name to store exported xml (default=screen)
-xmlport = port number to start the xml server
-v = show version of all modules
-h = display data through web server
```

-m = export data using modbus
-port = set server port <default = 9001>
-sf = connect to unframed serial forwarder
-fsf = connect to framed serial forwarder
-device = connect to serial device <default = /dev/ttyS0>
-baud = set serial baud rate <default = 57600>
-platform = set platform. <default = mica2>
values=mica2dot|mica2|mica|telos|micaz|m2100|m2110|m9100
-debug = set debug level. <default = DBG_WARNING>
-alert = alert when data values are above/below specified ranges
-daemon = run in daemon mode
-nomonitor = run without a system monitor
-configfiles = load xml configuration files.
-loadparsers = load only the listed parsers files.
(default=all files are loaded)
-loaddatasinks = load only the listed datasinks files.
(default=all files are loaded)
-heartbeat = turn on the heartbeat monitor and reset after <num missed>
-convZto2 = convert incoming network packets from micaZ headers to mica2 headers and vice versa
-conv2toZ = convert incoming network packets from mica2 headers to micaZ headers and vice versa

Πλήρες οδηγός υπάρχει στο Tutorial της Crossbow.

Παράρτημα Δ

Περιεχόμενα αρχείου db_xsensor.sql

```
-- $Id: db_xsensor.sql,v 1.31.2.3 2007/04/04 02:41:53 luckshman Exp $
-- Add MoteView client to db
-- Adding MoteView as client
DELETE FROM task_client_info where name='MoteView';
INSERT INTO task_client_info (name, type, clientinfo) VALUES
('MoteView', 'CONFIGURATION', '');
INSERT INTO task_mote_info (mote_id, x_coord, y_coord, z_coord, calib,
moteinfo, clientinfo_name) VALUES (0, 10, 10, 0, '', 'Gateway',
'MoteView');

-- Finalize db creation with xlisten tables

-- Creating xlisten tables for XSensorTutorial

-- Add MEP401/MEP500 enviro Tables:
-- Creating xlisten tables for MEP401 / MEP500
INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('inthum', 3, 1, 'Sensirion internal humidity reading') ;
INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('inttemp', 3, 1, 'Sensirion internal temperature reading') ;
INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('therm', 3, 1, 'Mica2dot thermistor reading') ;

-- Add MTS300/310 Tables:
-- Creating tables for MTS300/310
DROP TABLE mts300_results;
DROP TABLE mts300_results_L CASCADE;
DROP TABLE mts300_results_I CASCADE;
CREATE TABLE mts300_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer, temp
integer, light integer, mic integer );
CREATE TABLE mts300_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
temp integer, light integer, mic integer );
CREATE TABLE mts300_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);
CREATE RULE cache_mts300_results AS ON INSERT TO mts300_results DO (
DELETE FROM mts300_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mts300_results_L VALUES ( NEW.* ); );
CREATE INDEX mts300_time_idx ON mts300_results USING BTREE (
result_time ) ;
CREATE INDEX mts300_nodeid_idx ON mts300_results USING BTREE ( nodeid
) ;

DROP TABLE mts310_results;
DROP TABLE mts310_results_L CASCADE;
DROP TABLE mts310_results_I CASCADE;
CREATE TABLE mts310_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer, temp
integer, light integer, accel_x integer, accel_y integer, mag_x
integer, mag_y integer, mic integer );
```

```

CREATE TABLE mts310_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
temp integer, light integer, accel_x integer, accel_y integer, mag_x
integer, mag_y integer, mic integer );
CREATE TABLE mts310_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);
CREATE RULE cache_mts310_results AS ON INSERT TO mts310_results DO (
DELETE FROM mts310_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mts310_results_L VALUES ( NEW.* ); );
CREATE INDEX mts310_time_idx ON mts310_results USING BTREE (
result_time ) ;
CREATE INDEX mts310_nodeid_idx ON mts310_results USING BTREE ( nodeid
) ;

-- Add MTS400 Weather board tables:
-- Creating tables for MTS400
DROP TABLE mts400_results;
DROP TABLE mts400_results_L CASCADE;
DROP TABLE mts400_results_I CASCADE;
CREATE TABLE mts400_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer, humid
integer, humtemp integer, prtemp integer, press integer, taosch0
integer, taosch1 integer, accel_x integer, accel_y integer, calibW0
integer, calibW1 integer, calibW2 integer, calibW3 integer) ;
CREATE TABLE mts400_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
humid integer, humtemp integer, prtemp integer, press integer, taosch0
integer, taosch1 integer, accel_x integer, accel_y integer, calibW0
integer, calibW1 integer, calibW2 integer, calibW3 integer) ;
CREATE TABLE mts400_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);
CREATE RULE cache_mts400_results AS ON INSERT TO mts400_results DO (
DELETE FROM mts400_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mts400_results_L VALUES ( NEW.* ); );
CREATE INDEX mts400_time_idx ON mts400_results USING BTREE (
result_time ) ;
CREATE INDEX mts400_nodeid_idx ON mts400_results USING BTREE ( nodeid
) ;

-- Add MTS420 Weather board tables:
-- Creating tables for MTS420
DROP TABLE mts420_results;
DROP TABLE mts420_results_L CASCADE;
DROP TABLE mts420_results_I CASCADE;
CREATE TABLE mts420_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer, humid
integer, humtemp integer, prtemp integer, press integer, taosch0
integer, taosch1 integer, accel_x integer, accel_y integer, Hours
integer, Minutes integer, seconds bigint, Latitudedegree integer,
Latitudeminutes bigint, Longitudedegree integer, Longitudeminute
bigint, NSEwind integer, Fixed integer) ;
CREATE TABLE mts420_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
humid integer, humtemp integer, prtemp integer, press integer, taosch0
integer, taosch1 integer, accel_x integer, accel_y integer, Hours
integer, Minutes integer, seconds bigint, Latitudedegree integer,
Latitudeminutes bigint, Longitudedegree integer, Longitudeminute
bigint, NSEwind integer, Fixed integer) ;

```

```

CREATE TABLE mts420_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);
CREATE RULE cache_mts420_results AS ON INSERT TO mts420_results DO (
DELETE FROM mts420_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mts420_results_L VALUES ( NEW.* ); );
CREATE INDEX mts420_time_idx ON mts420_results USING BTREE (
result_time ) ;
CREATE INDEX mts420_nodeid_idx ON mts420_results USING BTREE ( nodeid
) ;

-- Add MTS450 Tables:
-- Creating tables for MTS450
DROP TABLE mts450_results;
DROP TABLE mts450_results_L CASCADE;
DROP TABLE mts450_results_I CASCADE;
CREATE TABLE mts450_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer, humtemp
integer, humid integer, gas integer);
CREATE TABLE mts450_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
humtemp integer, humid integer, gas integer);
CREATE TABLE mts450_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);
CREATE RULE cache_mts450_results AS ON INSERT TO mts450_results DO (
DELETE FROM mts450_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mts450_results_L VALUES ( NEW.* ); );
CREATE INDEX mts450_time_idx ON mts450_results USING BTREE (
result_time ) ;
CREATE INDEX mts450_nodeid_idx ON mts450_results USING BTREE ( nodeid
) ;

-- Add MTS101 Bread board:
-- Creating tables for MTS101
DROP TABLE mts101_results;
DROP TABLE mts101_results_L CASCADE;
DROP TABLE mts101_results_I CASCADE;
CREATE TABLE mts101_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer, temp
integer, light integer, adc2 integer, adc3 integer, adc4 integer ) ;
CREATE TABLE mts101_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
temp integer, light integer, adc2 integer, adc3 integer, adc4 integer)
;
CREATE TABLE mts101_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);
CREATE RULE cache_mts101_results AS ON INSERT TO mts101_results DO (
DELETE FROM mts101_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mts101_results_L VALUES ( NEW.* ); );
CREATE INDEX mts101_time_idx ON mts101_results USING BTREE (
result_time ) ;
CREATE INDEX mts101_nodeid_idx ON mts101_results USING BTREE ( nodeid
) ;

-- Add MDA100 Plant App Tables:
-- Creating tables for MDA100
DROP TABLE mda100_results;
DROP TABLE mda100_results_L CASCADE;

```

```

DROP TABLE mda100_results_I CASCADE;
CREATE TABLE mda100_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer, temp
integer, light integer, adc2 integer, adc3 integer, adc4 integer, adc5
integer, adc6 integer );
CREATE TABLE mda100_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
temp integer, light integer, adc2 integer, adc3 integer, adc4 integer,
adc5 integer, adc6 integer );
CREATE TABLE mda100_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);
CREATE RULE cache_mda100_results AS ON INSERT TO mda100_results DO (
DELETE FROM mda100_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mda100_results_L VALUES ( NEW.* ); ) ;
CREATE INDEX mda100_time_idx ON mda100_results USING BTREE (
result_time ) ;

```

```

-- Add xbw_da100 Plant App Tables:
-- Creating tables for xbw_da100 (old MDA100)
DROP TABLE xbw_da100_results;
DROP TABLE xbw_da100_results_L CASCADE;
DROP TABLE xbw_da100_results_I;
CREATE TABLE xbw_da100_results ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
temp integer, light integer, adc2 integer, adc3 integer, adc4 integer,
adc5 integer, adc6 integer );
CREATE TABLE xbw_da100_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
temp integer, light integer, adc2 integer, adc3 integer, adc4 integer,
adc5 integer, adc6 integer );
CREATE RULE cache_xbw_da100_results AS ON INSERT TO xbw_da100_results
DO ( DELETE FROM xbw_da100_results_L WHERE nodeid = NEW.nodeid; INSERT
INTO xbw_da100_results_L VALUES ( NEW.* ); ) ;
CREATE INDEX xbw_da100_time_idx ON xbw_da100_results USING BTREE (
result_time ) ;
CREATE INDEX xbw_da100_nodeid_idx ON xbw_da100_results USING BTREE (
nodeid ) ;
CREATE TABLE xbw_da100_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea);

```

```

-- Add xbw_da325 Plant App Tables:
-- Creating tables for xbw_da325 (old MDA320 with 5v reference )
DROP TABLE xbw_da325_results;
DROP TABLE xbw_da325_results_L CASCADE;
DROP TABLE xbw_da325_results_I;
CREATE TABLE xbw_da325_results ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
adc0 integer, adc1 integer, adc2 integer, adc3 integer, digi0 integer,
digi1 integer, digi2 integer, digi3 integer);
CREATE TABLE xbw_da325_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
adc0 integer, adc1 integer, adc2 integer, adc3 integer, digi0 integer,
digi1 integer, digi2 integer, digi3 integer);
CREATE RULE cache_xbw_da325_results AS ON INSERT TO xbw_da325_results
DO ( DELETE FROM xbw_da325_results_L WHERE nodeid = NEW.nodeid; INSERT
INTO xbw_da325_results_L VALUES ( NEW.* ); ) ;
CREATE INDEX xbw_da325_time_idx ON xbw_da325_results USING BTREE (
result_time ) ;

```

```

CREATE INDEX xbw_da325_nodeid_idx ON xbw_da325_results USING BTREE (
nodeid ) ;

CREATE TABLE xbw_da325_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea);

-- Add MDA300 Plant App Tables:
-- Creating tables for MDA300
DROP TABLE mda300_results;
DROP TABLE mda300_results_L CASCADE;
DROP TABLE mda300_results_I CASCADE;
CREATE TABLE mda300_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, adc0 integer, adc1
integer, adc2 integer, digi0 integer,digi1 integer,digi2
integer,voltage integer,humid integer, humtemp integer);
CREATE TABLE mda300_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, adc0 integer,
adc1 integer, adc2 integer, digi0 integer,digi1 integer,digi2
integer,voltage integer,humid integer, humtemp integer);
CREATE TABLE mda300_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);
CREATE RULE cache_mda300_results AS ON INSERT TO mda300_results DO (
DELETE FROM mda300_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mda300_results_L VALUES ( NEW.* ); ) ;
CREATE INDEX mda300_time_idx ON mda300_results USING BTREE (
result_time ) ;
CREATE INDEX mda300_nodeid_idx ON mda300_results USING BTREE ( nodeid
) ;

-- Add MDA300_D Plant App Tables:
-- Creating tables for MDA300_D
DROP TABLE mda300_precision_results;
DROP TABLE mda300_precision_results_L CASCADE;
DROP TABLE mda300_precision_results_I CASCADE;
CREATE TABLE mda300_precision_results ( result_time timestamp without
time zone,epoch integer, nodeid integer, parent integer,p_adc0
integer, p_adc1 integer, p_adc2 integer,p_adc3 integer,voltage
integer,humid integer, humtemp integer);
CREATE TABLE mda300_precision_results_l ( result_time timestamp
without time zone,epoch integer, nodeid integer, parent integer,p_adc0
integer, p_adc1 integer, p_adc2 integer,p_adc3 integer,voltage
integer,humid integer, humtemp integer);
CREATE TABLE mda300_precision_results_I ( nodeid integer, x_coord
integer, y_coord integer, z_coord integer, mote_name text, calib
bytea, board_id integer);
CREATE RULE cache_mda300_precision_results AS ON INSERT TO
mda300_precision_results DO ( DELETE FROM mda300_precision_results_L
WHERE nodeid = NEW.nodeid; INSERT INTO mda300_precision_results_L
VALUES ( NEW.* ); ) ;
CREATE INDEX mda300_precision_time_idx ON mda300_precision_results
USING BTREE ( result_time ) ;
CREATE INDEX mda300_precision_nodeid_idx ON mda300_precision_results
USING BTREE ( nodeid ) ;

INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('echo10', 3, 1, 'Echo10 soil moisture probe') ;

```

```

INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('echo20', 3, 1, 'Echo20 soil moisture probe') ;
INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('soiltemp', 3, 1, 'Davis soil temperature probe') ;

INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('adc0', 3, 1, 'MDA300 analog channel 0') ;
INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('adc1', 3, 1, 'MDA300 analog channel 1') ;
INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('adc2', 3, 1, 'MDA300 analog channel 2') ;

INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('digi0', 3, 1, 'MDA300 digital channel 0') ;
INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('digi1', 3, 1, 'MDA300 digital channel 1') ;
INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('digi2', 3, 1, 'MDA300 digital channel 2') ;

-- Add MDA320 Plant App Tables:
-- Creating tables for MDA320
DROP TABLE mda320_results;
DROP TABLE mda320_results_L CASCADE;
DROP TABLE mda320_results_I CASCADE;
CREATE TABLE mda320_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer, adc0
integer, adc1 integer, adc2 integer, adc3 integer, digi0 integer,
digi1 integer, digi2 integer, digi3 integer);
CREATE TABLE mda320_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
adc0 integer, adc1 integer, adc2 integer, adc3 integer, digi0 integer,
digi1 integer, digi2 integer, digi3 integer );
CREATE TABLE mda320_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);

CREATE RULE cache_mda320_results AS ON INSERT TO mda320_results DO (
DELETE FROM mda320_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mda320_results_L VALUES ( NEW.* ); ) ;
CREATE INDEX mda320_time_idx ON mda320_results USING BTREE (
result_time ) ;
CREATE INDEX mda320_nodeid_idx ON mda320_results USING BTREE ( nodeid
) ;

---- Dot boards
DROP TABLE mda500_results;
DROP TABLE mda500_results_L CASCADE;
DROP TABLE mda500_results_I CASCADE;
CREATE TABLE mda500_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer, temp
integer, adc2 integer, adc3 integer, adc4 integer, adc5 integer, adc6
integer, adc7 integer);
CREATE TABLE mda500_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
temp integer, adc2 integer, adc3 integer, adc4 integer, adc5 integer,
adc6 integer, adc7 integer);
CREATE TABLE mda500_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);

```

```

CREATE RULE cache_mda500_results AS ON INSERT TO mda500_results DO (
DELETE FROM mda500_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mda500_results_L VALUES (NEW.*)); );
CREATE INDEX mda500_time_idx ON mda500_results USING BTREE (
result_time ) ;
CREATE INDEX mda500_nodeid_idx ON mda500_results USING BTREE ( nodeid
) ;

--- MTS510
DROP TABLE mts510_results;
DROP TABLE mts510_results_L CASCADE;
DROP TABLE mts510_results_I CASCADE;
CREATE TABLE mts510_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer, temp
integer, light integer, accel_x integer, accel_y integer, mag_x
integer, mag_y integer, mic integer);
CREATE TABLE mts510_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
temp integer, light integer, accel_x integer, accel_y integer, mag_x
integer, mag_y integer, mic integer );
CREATE TABLE mts510_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);
CREATE RULE cache_mts510_results AS ON INSERT TO mts510_results DO (
DELETE FROM mts510_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mts510_results_L VALUES (NEW.*)); );
CREATE INDEX mts510_time_idx ON mts510_results USING BTREE (
result_time ) ;
CREATE INDEX mts510_nodeid_idx ON mts510_results USING BTREE ( nodeid
) ;

-- Finalize db creation with xlisten tables
-- Creating tables for Surge
--DROP TABLE surge_results;
--DROP TABLE surge_results_L CASCADE;
--DROP TABLE surge_results_I CASCADE;
--CREATE TABLE surge_results ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, light integer,
temp integer, voltage integer, mag_x integer, mag_y integer, accel_x
integer, accel_y integer);
--CREATE TABLE surge_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, light integer,
temp integer, voltage integer, mag_x integer, mag_y integer, accel_x
integer, accel_y integer);
--CREATE TABLE surge_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);
--CREATE RULE cache_surge_results AS ON INSERT TO surge_results DO (
DELETE FROM surge_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
surge_results_L VALUES ( NEW.* )); );
--CREATE INDEX surge_time_idx ON surge_results USING BTREE (
result_time ) ;
--CREATE INDEX surge_nodeid_idx ON surge_results USING BTREE ( nodeid
) ;

-- XSM600 tables

INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('quad', 3, 1, 'MSP410 quad detect');
```

```

INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('pir', 3, 1, 'MSP410 passive infrared detector');

DROP TABLE msp410_results;
DROP TABLE msp410_results_L CASCADE;
DROP TABLE msp410_results_I CASCADE;
CREATE TABLE msp410_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer, quad
integer, pir integer, audio integer, mag integer );
CREATE TABLE msp410_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage integer,
quad integer, pir integer, audio integer, mag integer );
CREATE TABLE msp410_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);

CREATE RULE cache_msp410_results AS ON INSERT TO msp410_results DO (
DELETE FROM msp410_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
msp410_results_L VALUES (NEW.*); );
CREATE INDEX msp410_time_idx ON msp410_results USING BTREE (
result_time ) ;
CREATE INDEX msp410_nodeid_idx ON msp410_results USING BTREE ( nodeid
) ;

-- Add MEP410/MEP510 mep Tables:
-- Creating tables for MEP410/MEP510
--INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('inthum', 3, 1, 'Sensirion internal humidity reading') ;
--INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('inttemp', 3, 1, 'Sensirion internal temperature reading') ;
--INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('therm', 3, 1, 'Mica2dot thermistor reading') ;
INSERT INTO task_attributes (name, typeid, power_cons, description)
VALUES ('board_id', 3, 1, 'Flag for MEP410/MEP510') ;

DROP TABLE mep_results;
DROP TABLE mep_results_L CASCADE;
DROP TABLE mep_results_I CASCADE;
CREATE TABLE mep_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer,
thermister integer, humidity integer, temp_hum integer, inthum
integer, inttemp integer, par_bottom integer, tsr_bottom integer,
par_top integer, tsr_top integer, temp_pr integer, pressure integer,
board_id integer);
CREATE TABLE mep_results_L ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer,
thermister integer, humidity integer, temp_hum integer, inthum
integer, inttemp integer, par_bottom integer, tsr_bottom integer,
par_top integer, tsr_top integer, temp_pr integer, pressure integer,
board_id integer);
CREATE TABLE mep_results_I ( nodeid integer, x_coord integer, y_coord
integer, z_coord integer, mote_name text, calib bytea, board_id
integer);

CREATE RULE cache_mep_results AS ON INSERT TO mep_results DO ( DELETE
FROM mep_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mep_results_L VALUES ( NEW.* ); );
CREATE INDEX mep_time_idx ON mep_results USING BTREE ( result_time ) ;
CREATE INDEX mep_nodeid_idx ON mep_results USING BTREE ( nodeid ) ;

```

```

--- MTP400
DROP TABLE mtp400_results;
DROP TABLE mtp400_results_L CASCADE;
DROP TABLE mtp400_results_I CASCADE;
CREATE TABLE mtp400_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, sensortype
integer,voltage integer, CH1_reading integer,CH2_reading integer,
CH3_reading integer);
CREATE TABLE mtp400_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer,sensortype
integer,voltage integer, CH1_reading integer,CH2_reading integer,
CH3_reading integer);
CREATE TABLE mtp400_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);
CREATE RULE cache_mtp400_results AS ON INSERT TO mtp400_results DO (
DELETE FROM mtp400_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mtp400_results_L VALUES (NEW.*); );
CREATE INDEX mtp400_time_idx ON mtp400_results USING BTREE (
result_time ) ;
CREATE INDEX mtp400_nodeid_idx ON mtp400_results USING BTREE ( nodeid
) ;

```

```

--- MTS410
DROP TABLE mts410_results;
DROP TABLE mts410_results_L CASCADE;
DROP TABLE mts410_results_I CASCADE;
CREATE TABLE mts410_results ( result_time timestamp without time zone,
epoch integer, nodeid integer, parent integer, voltage integer,pir
integer, humid integer,humtemp integer,accel_x integer, accel_y
integer, prtemp integer,press integer,taosch0 integer,taosch1
integer,mic integer,adc5 integer, adc6 integer, pir_max integer,
pir_sum integer);
CREATE TABLE mts410_results_L ( result_time timestamp without time
zone, epoch integer, nodeid integer, parent integer, voltage
integer,pir integer, humid integer,humtemp integer,accel_x integer,
accel_y integer, prtemp integer,press integer,taosch0 integer,taosch1
integer,mic integer,adc5 integer, adc6 integer, pir_max integer,
pir_sum integer);
CREATE TABLE mts410_results_I ( nodeid integer, x_coord integer,
y_coord integer, z_coord integer, mote_name text, calib bytea,
board_id integer);
CREATE RULE cache_mts410_results AS ON INSERT TO mts410_results DO (
DELETE FROM mts410_results_L WHERE nodeid = NEW.nodeid; INSERT INTO
mts410_results_L VALUES (NEW.*); );
CREATE INDEX mts410_time_idx ON mts410_results USING BTREE (
result_time ) ;
CREATE INDEX mts410_nodeid_idx ON mts410_results USING BTREE ( nodeid
) ;

```

```

--- add node_health table for MV146
DROP TABLE node_health;
DROP TABLE node_health_I CASCADE;
DROP TABLE node_health_L CASCADE;

```

```

CREATE TABLE node_health ( result_time timestamp without time
zone,epoch integer, nodeid integer, health_pkts integer, node_pkts

```

```

integer, forwarded integer, dropped integer, retries integer, battery
integer, power_sum integer, board_id integer, parent integer,
quality_tx integer, quality_rx integer, path_cost integer, parent_rssi
integer);
CREATE TABLE node_health_l ( result_time timestamp without time
zone,epoch integer, nodeid integer, health_pkts integer, node_pkts
integer, forwarded integer, dropped integer, retries integer, battery
integer, power_sum integer,board_id integer, parent integer,
quality_tx integer, quality_rx integer, path_cost integer, parent_rssi
integer);
CREATE TABLE node_health_I ( nodeid integer, x_coord integer, y_coord
integer, z_coord integer, mote_name text, calib bytea, board_id
integer);
CREATE RULE cache_node_health AS ON INSERT TO node_health DO ( DELETE
FROM node_health_l WHERE nodeid = NEW.nodeid;INSERT INTO node_health_l
VALUES (NEW.*); );
CREATE INDEX node_health_idx ON node_health USING BTREE ( result_time
) ;
CREATE INDEX node_health_nodeid_idx ON node_health USING BTREE (
nodeid ) ;

--- Neighbor_health table (delete any existing tables, MV 2.0E does
not support the info in this table)
DROP TABLE neighbor_health_results;
DROP TABLE neighbor_health_results_l;

--- Linkage -- query_id = xsensor board_id

DELETE FROM task_query_log where query_id=1;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(1, 'mda500_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=2;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(2, 'mts510_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=7;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(7, 'surge_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=129;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(129, 'mda300_results', 'MOTEVVIEW_1_0');

---MDA300_D sensor boardid uses 999,only for moteview dropdown menu
DELETE FROM task_query_log where query_id=999;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(999, 'mda300_precision_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=130;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(130, 'mts101_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=131;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(131, 'mts300_results', 'MOTEVVIEW_1_0');

```

```

DELETE FROM task_query_log where query_id=132;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(132, 'mts310_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=133;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(133, 'mts400_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=134;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(134, 'mts420_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=138;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(138, 'mep_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=144;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(144, 'mda320_results', 'MOTEVVIEW_1_0');

-- DELETE FROM task_query_log where query_id=145;
-- INSERT INTO task_query_log (query_id, table_name, query_type)
VALUES (145, 'mda100_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=145;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(145, 'xbw_da100_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=146;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(146, 'mts450_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=147;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(147, 'xbw_da325_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=160;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(160, 'msp410_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=161;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(161, 'mtp400_results', 'MOTEVVIEW_1_0');

DELETE FROM task_query_log where query_id=162;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(162, 'mts410_results', 'MOTEVVIEW_1_0');

--add for node_health
DELETE FROM task_query_log where query_id=3;
INSERT INTO task_query_log (query_id, table_name, query_type) VALUES
(3, 'node_health', 'MOTEVVIEW_1_0');

```

Περιεχόμενα αρχείου mts400_mesh_config.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE XServeConfig SYSTEM "../xserve_config.dtd">
<XServeConfig>
  <XFieldExtractor name="XMTS400 Multihop PK6 Config XML"
order="3">
    <XFields>
      <XField name="amtype" byteoffset="2" length="1"
type="uint8"/>
      <XField name="nodeid" byteoffset="7" length="2"
type="uint16" specialtype="nodeid"/>
      <XField name="parent" byteoffset="14" length="2"
type="uint16" specialtype="parentid"/>
      <XField name="group" byteoffset="3" length="1"
type="uint8"/>
      <XField name="socketid" byteoffset="11" length="1"
type="uint8"/>
      <XField name="board_id" byteoffset="12" length="1"
type="uint8" specialtype="boardid"/>
      <XField name="packet_id" byteoffset="13" length="1"
type="uint8"/>
      <XField name="voltage" byteoffset="16" length="2"
type="uint16">
        <XConversion function="(1252352/x)"
returntype="uint16">
          <XConvParam variablename="x"
fieldname="voltage" type="float"/>
        </XConversion>
      </XField>
      <XField name="humid" byteoffset="18" length="2"
type="uint16">
        <XConversion function="(-39.6 + 0.01*y -
25.0)*(0.01 + 0.00008 * x) - 4.0 + 0.0405 * x - 0.0000028 * x * x"
returntype="uint16">
          <XConvParam variablename="x"
fieldname="humid" type="float"/>
          <XConvParam variablename="y"
fieldname="humtemp" type="float"/>
        </XConversion>
      </XField>
      <XField name="humtemp" byteoffset="20" length="2"
type="uint16">
        <XConversion function="(-39.6 + 0.01*x)"
returntype="uint16">
          <XConvParam variablename="x"
fieldname="humtemp" type="float"/>
        </XConversion>
      </XField>
      <XField name="calibW0" byteoffset="22" length="2"
type="uint16"/>
      <XField name="calibW1" byteoffset="24" length="2"
type="uint16"/>
      <XField name="calibW2" byteoffset="26" length="2"
type="uint16"/>
      <XField name="calibW3" byteoffset="28" length="2"
type="uint16"/>
      <XField name="prtemp" byteoffset="30" length="2"
type="uint16">
```

```
<XConversion function="(200.0+(x-(8*((a & 1) && 10) | (b && 6))+20224))*((b & 63)+50.0)/1024.0)/10.0" returntype="float">
    <XConvParam variablename="x"
fieldname="prtemp" type="float"/>
    <XConvParam variablename="a"
fieldname="calibw0" type="uint16"/>
    <XConvParam variablename="b"
fieldname="calibw1" type="uint16"/>
</XConversion>
</XField>
<XField name="press" byteoffset="32" length="2"
type="uint16">
    <XConversion function="((((a && 1)+(d && 6)*(y-(8*((a & 1) && 10) | (b && 6))+20224))/1024.0+24576)*(x-7168.0))/16384 - (((c & 63) && 6) | (d & 63))*4 + ((c && 6)-512.0)*(y-(8*((a & 1) && 10) | (b && 6))+20224))/1024))/32.0 +250.0"
returntype="float">
    <XConvParam variablename="x"
fieldname="press" type="float"/>
    <XConvParam variablename="y"
fieldname="prtemp" type="float"/>
    <XConvParam variablename="a"
fieldname="calibw0" type="uint16"/>
    <XConvParam variablename="b"
fieldname="calibw1" type="uint16"/>
    <XConvParam variablename="c"
fieldname="calibw2" type="uint16"/>
    <XConvParam variablename="d"
fieldname="calibw3" type="uint16"/>
</XConversion>
</XField>
<XField name="taosch0" byteoffset="34" length="2"
type="uint16"/>
<XField name="taosch1" byteoffset="36" length="2"
type="uint16"/>
<XField name="accel_x" byteoffset="38" length="2"
type="uint16">
    <XConversion function="1000.0 * (1.0 - (500 - x)/((500 - 400)/2))" returntype="float">
        <XConvParam variablename="x"
fieldname="accel_x" type="uint16"/>
    </XConversion>
</XField>
<XField name="accel_y" byteoffset="40" length="2"
type="uint16">
    <XConversion function="1000.0 * (1.0 - (500 - x)/((500 - 400)/2))" returntype="float">
        <XConvParam variablename="x"
fieldname="accel_y" type="uint16"/>
    </XConversion>
</XField>
<XField name="taoch0" byteoffset="34" length="2"
type="uint16">
    <XConversion function="(16.5*((1 && ((x & 112) && 4))-1) + (x & 15)*((1 && ((x & 112) && 4))-1))*0.46 / exp(3.13*((16.5*((1 && ((y & 112) && 4))-1) + (y & 15)*((1 && ((y & 112) && 4))-1)))"
```

```

4)))-1)))/(16.5*((1 &lt;&lt; ((x &amp; 112) &gt;&gt; 4))-1) + (x &amp;
15)*((1 &lt;&lt; ((x &amp; 112) &gt;&gt; 4))-1))))"
returntype="float">
                                <XConvParam variablename="x"
fieldname="taosch0" type="float"/>
                                <XConvParam variablename="y"
fieldname="taosch1" type="float"/>
                                </XConversion>
                                </XField>
                                <XField name="calibB0" bytearray="22" length="1"
type="uint8"/>
                                <XField name="calibB1" bytearray="23" length="1"
type="uint8"/>
                                <XField name="calibB2" bytearray="24" length="1"
type="uint8"/>
                                <XField name="calibB3" bytearray="25" length="1"
type="uint8"/>
                                <XField name="calibB4" bytearray="26" length="1"
type="uint8"/>
                                <XField name="calibB5" bytearray="27" length="1"
type="uint8"/>
                                <XField name="calibB6" bytearray="28" length="1"
type="uint8"/>
                                <XField name="calibB7" bytearray="29" length="1"
type="uint8"/>
                                </XFields>
                                <XFilter>
                                <XCondAnd>
                                <!--
                                Check if this is a sensor type
                                packet...AMTYPE_MULTIHOP or AMTYPE_XDEBUG
                                The Byte offsets defined above assume
                                that there is a multihop header
                                /-->
                                <XCondOr>
                                <XCond name="IsEqual">
                                <XFilterParam name="fieldname"
value="amtype"/>
                                <XFilterParam name="fieldvalue"
value="0x31"/>
                                </XCond>
                                <XCond name="IsEqual">
                                <XFilterParam name="fieldname"
value="amtype"/>
                                <XFilterParam name="fieldvalue"
value="0x33"/>
                                </XCond>
                                <XCondAnd>
                                <XCondOr>
                                <XCond name="IsEqual">
                                <XFilterParam
name="fieldname" value="socketid"/>
                                <XFilterParam
name="fieldvalue" value="0x31"/>
                                </XCond>
                                <XCond name="IsEqual">
                                <XFilterParam
name="fieldname" value="socketid"/>

```

```

name="fieldvalue" value="0x33"/>
</XCond>
</XCondOr>
<XCondOr>
  <XCond name="IsEqual">
    <XFilterParam
name="fieldname" value="amtype"/>
    <XFilterParam
name="fieldvalue" value="0x0B"/>
  </XCond>
  <XCond name="IsEqual">
    <XFilterParam
name="fieldname" value="amtype"/>
    <XFilterParam
name="fieldvalue" value="0x0D"/>
  </XCond>
</XCondOr>
</XCondAnd>
</XCondOr>
<XCond name="IsEqual">
  <XFilterParam name="fieldname"
value="board_id"/>
  <XFilterParam name="fieldvalue"
value="0x85"/>
</XCond>
<XCondOr>
  <XCond name="IsEqual">
    <XFilterParam name="fieldname"
value="packet_id"/>
    <XFilterParam name="fieldvalue"
value="0x86"/>
  </XCond>
  <XCond name="IsEqual">
    <XFilterParam name="fieldname"
value="packet_id"/>
    <XFilterParam name="fieldvalue"
value="0x06"/>
  </XCond>
</XCondOr>
</XCondAnd>
</XFilter>
<XDataSinks>
  <XDataSink name="Generic Print Datasink">
    <XDSPParam name="printstring" value="MTS400
[sensor data converted to engineering units]:\n  health:      node
id = %s\n  battery:      = %s mv \n  humid:      = %s
%% \n  Temperature:      = %s degC \n  IntersemaTemperature: =
%s degC \n  IntersemaPressure: = %s mbar \n  Light :
= %s lux\n  X-axis Accel:      = %s mg \n  Y-axis Accel:
= %s mg "/>
    <XDSPParam name="printfields"
value="nodeid,voltage,humid,humtemp,prtemp,press,taoch0,accel_x,accel_
y"/>
  </XDataSink>
  <XDataSink name="Open Log Datasink">
    <XDSPParam name="tablename_1"
value="mts400_results"/>
  </XDataSink>
</XDataSinks>

```

```

        <XDSPParam name="createsql_1" value="CREATE
TABLE mts400_results ( result_time timestamp without time zone,epoch
integer, nodeid integer, parent integer,voltage integer, humid
integer, humtemp integer,prtemp integer,press integer,taosch0
integer,taosch1 integer,accel_x integer, accel_y integer, calibW0
integer, calibW1 integer, calibW2 integer, calibW3 integer)"/>

        <XDSPParam name="tablename_2"
value="mts400_results_1"/>
        <XDSPParam name="createsql_2" value="CREATE
TABLE mts400_results_1 ( result_time timestamp without time zone,epoch
integer, nodeid integer, parent integer,voltage integer, humid
integer, humtemp integer,prtemp integer,press integer,taosch0
integer,taosch1 integer,accel_x integer, accel_y integer, calibW0
integer, calibW1 integer, calibW2 integer, calibW3 integer)"/>

        <XDSPParam name="rulesql_2" value="CREATE RULE
cache_mts400_results AS ON INSERT TO mts400_results DO ( DELETE FROM
mts400_results_1 WHERE nodeid = NEW.nodeid;INSERT INTO
mts400_results_1 VALUES (NEW.*); )"/>
        <XDSPParam name="insertsql_2" value="INSERT
into mts400_results
(result_time,nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,
taosch1,accel_x,accel_y,calibW0,calibW1,calibW2,calibW3) values
(now(),%i,%i,%i,%i,%i,%i,%i,%i,%i,%i,%i,%i,%i,%i,%i)"/>
        <XDSPParam name="insertfields_2"
value="nodeid,parent,voltage,humid,humtemp,prtemp,press,taosch0,taosch
1,accel_x,accel_y,calibW0,calibW1,calibW2,calibW3"/>
        <XDSPParam name="tablename_3"
value="mts400_results_i"/>
        <XDSPParam name="createsql_3" value="CREATE
TABLE mts400_results_i ( nodeid integer ,x_coord integer, y_coord
integer, z_coord integer,mote_name text, calib bytea , boardid
integer)"/>
        <XDSPParam name="updatesql_3" value="update
mts400_results_i set calib='\\%o\\%o\\%o\\%o\\%o\\%o\\%o\\%o' where
nodeid=%i"/>
        <XDSPParam name="updatefields_3"
value="calibB0,calibB1,calibB2,calibB3,calibB4,calibB5,calibB6,calibB7
,nodeid"/>
        <XDSPParam name="insertsql_3" value="INSERT
into mts400_results_i ( nodeid,calib) values
(%i,'\\%o\\%o\\%o\\%o\\%o\\%o\\%o\\%o')"/>
        <XDSPParam name="insertfields_3"
value="nodeid,calibB0,calibB1,calibB2,calibB3,calibB4,calibB5,calibB6,
calibB7"/>
    </XDataSink>
    <XDataSink name="Generic test Datasink">
        <XDSPParam name="voltage" value="[2500,3500]"/>
        <XDSPParam name="humid" value="[0,100]"/>
        <XDSPParam name="humtemp" value="[-5,45]"/>
        <XDSPParam name="prtemp" value="[-5,45]"/>
        <XDSPParam name="press" value="[900,1100]"/>
        <XDSPParam name="accel_x" value="[-
2000,2000]"/>
        <XDSPParam name="accel_y" value="[-
2000,2000]"/>
        <XDSPParam name="taoch0" value="[0,500]"/>
    </XDataSink>
</XDataSinks>

```

```

        </XFieldExtractor>
        <XFieldExtractor name="XMTS400 Multihop PK8 Config XML"
order="3">
            <XFields>
                <XField name="amtype" byteoffset="2" length="1"
type="uint8"/>
                <XField name="nodeid" byteoffset="7" length="2"
type="uint16" specialtype="nodeid"/>
                <XField name="parent" byteoffset="14" length="2"
type="uint16" specialtype="parentid"/>
                <XField name="group" byteoffset="3" length="1"
type="uint8"/>
                <XField name="socketid" byteoffset="11" length="1"
type="uint8"/>
                <XField name="board_id" byteoffset="12" length="1"
type="uint8" specialtype="boardid"/>
                <XField name="packet_id" byteoffset="13" length="1"
type="uint8"/>
                <XField name="voltage" byteoffset="16" length="2"
type="uint16">
                    <XConversion function="(1252352/x)"
returntype="uint16">
                        <XConvParam variablename="x"
fieldname="voltage" type="float"/>
                    </XConversion>
                </XField>
            </XFields>
            <XFilter>
                <XCondAnd>
                    <!--
                        Check if this is a sensor type
                        packet...AMTYPE_MULTIHOP or AMTYPE_XDEBUG
                        The Byte offsets defined above assume
                        that there is a multihop header
                    /-->
                    <XCondOr>
                        <XCond name="IsEqual">
                            <XFilterParam name="fieldname"
value="amtype"/>
                            <XFilterParam name="fieldvalue"
value="0x31"/>
                        </XCond>
                        <XCond name="IsEqual">
                            <XFilterParam name="fieldname"
value="amtype"/>
                            <XFilterParam name="fieldvalue"
value="0x33"/>
                        </XCond>
                    </XCondAnd>
                    <XCondOr>
                        <XCond name="IsEqual">
                            <XFilterParam
name="fieldname" value="socketid"/>
                            <XFilterParam
name="fieldvalue" value="0x31"/>
                        </XCond>
                        <XCond name="IsEqual">
                            <XFilterParam
name="fieldname" value="socketid"/>

```

```

name="fieldvalue" value="0x33"/>
</XCond>
</XCondOr>
<XCondOr>
  <XCond name="IsEqual">
    <XFilterParam
name="fieldname" value="amtype"/>
    </XCond>
  <XCond name="IsEqual">
    <XFilterParam
name="fieldvalue" value="0x0B"/>
    </XCond>
  <XCond name="IsEqual">
    <XFilterParam
name="fieldname" value="amtype"/>
    </XCond>
  <XCond name="IsEqual">
    <XFilterParam
name="fieldvalue" value="0x0D"/>
    </XCond>
  </XCondOr>
</XCondAnd>
</XCondOr>
<XCond name="IsEqual">
  <XFilterParam name="fieldname"
value="board_id"/>
  <XFilterParam name="fieldvalue"
value="0x85"/>
</XCond>
<XCondOr>
  <XCond name="IsEqual">
    <XFilterParam name="fieldname"
value="packet_id"/>
    <XFilterParam name="fieldvalue"
value="0x88"/>
  </XCond>
  <XCond name="IsEqual">
    <XFilterParam name="fieldname"
value="packet_id"/>
    <XFilterParam name="fieldvalue"
value="0x08"/>
  </XCond>
</XCondOr>
</XCondAnd>
</XFilter>
<XDataSinks>
  <XDataSink name="Generic Print Datasink">
    <XDSPParam name="printstring" value="MTS400
[sensor data converted to engineering units]:\n  health:      node
id = %s\n  battery:      = %s mv  "/>
    <XDSPParam name="printfields"
value="nodeid,voltage"/>
  </XDataSink>
  <XDataSink name="Open Log Datasink">
    <XDSPParam name="tablename_1"
value="mts400_results"/>
    <XDSPParam name="createsql_1" value="CREATE
TABLE mts400_results ( result_time timestamp without time zone,epoch
integer, nodeid integer, parent integer,voltage integer, humid
integer, humtemp integer,prtemp integer,press integer,taosch0

```

```

integer,taosch1 integer,accel_x integer, accel_y integer)"/>
        <XDSPParam name="tablename_2"
value="mts400_results_1"/>
        <XDSPParam name="createsql_2" value="CREATE
TABLE mts400_results_1 ( result_time timestamp without time zone,epoch
integer, nodeid integer, parent integer,voltage integer, humid
integer, humtemp integer,prtemp integer,press integer,taosch0
integer,taosch1 integer,accel_x integer, accel_y integer)"/>

        <XDSPParam name="rulesql_2" value="CREATE RULE
cache_mts400_results AS ON INSERT TO mts400_results DO ( DELETE FROM
mts400_results_1 WHERE nodeid = NEW.nodeid;INSERT INTO
mts400_results_1 VALUES (NEW.*); )"/>
        <XDSPParam name="insertsql_2" value="INSERT
into mts400_results (result_time,nodeid,parent,voltage) values
(now(),%i,%i,%i)"/>
        <XDSPParam name="insertfields_2"
value="nodeid,parent,voltage"/>
        </XDataSink>
        <XDataSink name="Generic test Datasink">
        <XDSPParam name="voltage" value="[2500,3500]"/>
        </XDataSink>
        </XDataSinks>
    </XFieldExtractor>

</XServeConfig>

```

Περιεχόμενα αρχείου **mts400_sensor_config.xml**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE XServeConfig SYSTEM "../xserve_config.dtd">
<XServeConfig>
  <XFieldExtractor name="XMTS40 Multihop Config XML" order="3">
    <XFields>
      <XField name="amtype" byteoffset="2" length="1"
type="uint8"/>
      <XField name="nodeid" byteoffset="7" length="1"
type="uint8" specialtype="nodeid"/>
      <!--
      <XField name="parent" byteoffset="14" length="2"
type="uint16" specialtype="parentid"/>
      /-->
      <XField name="group" byteoffset="3" length="1"
type="uint8"/>
      <XField name="board_id" byteoffset="5" length="1"
type="uint8" specialtype="boardid"/>
      <XField name="packet_id" byteoffset="6" length="1"
type="uint8"/>
      <XField name="rsvd" byteoffset="8" length="1"
type="uint8"/>
      <XField name="voltage" byteoffset="9" length="2"
type="uint16">
        <XConversion function="(1252352/x)"
returntype="uint16">
          <XConvParam variablename="x"
fieldname="voltage" type="float"/>
        </XConversion>
      </XField>
      <XField name="humid" byteoffset="11" length="2"
type="uint16">
        <XConversion function="(-38.4 + 0.0098*y -
25.0)*(0.01 + 0.00008 * x) - 4.0 + 0.0405 * x - 0.0000028 * x * x"
returntype="uint16">
          <XConvParam variablename="x"
fieldname="humid" type="float"/>
          <XConvParam variablename="y"
fieldname="humtemp" type="float"/>
        </XConversion>
      </XField>
      <XField name="humtemp" byteoffset="13" length="2"
type="uint16">
        <XConversion function="(-38.4 + 0.0098*x)"
returntype="uint16">
          <XConvParam variablename="x"
fieldname="humtemp" type="float"/>
        </XConversion>
      </XField>
      <XField name="calibW0" byteoffset="15" length="2"
type="uint16"/>
      <XField name="calibW1" byteoffset="17" length="2"
type="uint16"/>
      <XField name="calibW2" byteoffset="19" length="2"
type="uint16"/>
      <XField name="calibW3" byteoffset="21" length="2"
type="uint16"/>
    </XFields>
  </XFieldExtractor>
</XServeConfig>
```

```

        <XField name="prtemp" bytearrayoffset="23" length="2"
type="uint16">
            <XConversion function="(200.0+(x-(8*((a &
1) &lt;&lt; 10) | (b &gt;&gt; 6))+20224))*((b &
63)+50.0)/1024.0)/10.0" returntype="float">
                <XConvParam variablename="x"
fieldname="prtemp" type="float"/>
                <XConvParam variablename="a"
fieldname="calibw0" type="uint16"/>
                <XConvParam variablename="b"
fieldname="calibw1" type="uint16"/>
            </XConversion>
        </XField>
        <XField name="press" bytearrayoffset="25" length="2"
type="uint16">
            <XConversion function="(((a &gt;&gt; 1)+(d
&gt;&gt; 6)*(y-(8*((a & 1) &lt;&lt; 10) | (b &gt;&gt;
6))+20224))/1024.0+24576)*(x-7168.0))/16384 - (((c & 63) &lt;&lt;
6) | (d & 63))*4 + ((c &gt;&gt; 6)-512.0)*(y-(8*((a & 1)
&lt;&lt; 10) | (b &gt;&gt; 6))+20224))/1024))/32.0 +250.0"
returntype="float">
                <XConvParam variablename="x"
fieldname="press" type="float"/>
                <XConvParam variablename="y"
fieldname="prtemp" type="float"/>
                <XConvParam variablename="a"
fieldname="calibw0" type="uint16"/>
                <XConvParam variablename="b"
fieldname="calibw1" type="uint16"/>
                <XConvParam variablename="c"
fieldname="calibw2" type="uint16"/>
                <XConvParam variablename="d"
fieldname="calibw3" type="uint16"/>
            </XConversion>
        </XField>
        <XField name="taosch0" bytearrayoffset="27" length="2"
type="uint16"/>
        <XField name="taosch1" bytearrayoffset="29" length="2"
type="uint16"/>
        <XField name="accel_x" bytearrayoffset="31" length="2"
type="uint16">
            <XConversion function="1000.0 * (1.0 - (500 -
x)/((500 - 400)/2))" returntype="float">
                <XConvParam variablename="x"
fieldname="accel_x" type="uint16"/>
            </XConversion>
        </XField>
        <XField name="accel_y" bytearrayoffset="33" length="1"
type="uint8">
            <XConversion function="1000.0 * (1.0 - (500 -
(x+y*256))/((500 - 400)/2))" returntype="float">
                <XConvParam variablename="x"
fieldname="accel_y" type="uint8"/>
                <XConvParam variablename="y"
fieldname="rsvd" type="uint8"/>
            </XConversion>
        </XField>
        <XField name="taoch0" bytearrayoffset="27" length="2"
type="uint16">

```

```

        <XConversion function="(16.5*((1 &lt;&lt; ((x
&amp; 112) &gt;&gt; 4))-1) + (x &amp; 15)*((1 &lt;&lt; ((x &amp; 112)
&gt;&gt; 4))-1))*0.46 / exp(3.13*((16.5*((1 &lt;&lt; ((y &amp; 112)
&gt;&gt; 4))-1) + (y &amp; 15)*(( 1 &lt;&lt; ((y &amp; 112) &gt;&gt;
4))-1)))/(16.5*((1 &lt;&lt; ((x &amp; 112) &gt;&gt; 4))-1) + (x &amp;
15)*((1 &lt;&lt; ((x &amp; 112) &gt;&gt; 4))-1)))"
returntype="float">
        <XConvParam variablename="x"
fieldname="taosch0" type="float"/>
        <XConvParam variablename="y"
fieldname="taosch1" type="float"/>
    </XConversion>
    </XField>
    <XField name="calibB0" byteoffset="15" length="1"
type="uint8"/>
    <XField name="calibB1" byteoffset="16" length="1"
type="uint8"/>
    <XField name="calibB2" byteoffset="17" length="1"
type="uint8"/>
    <XField name="calibB3" byteoffset="18" length="1"
type="uint8"/>
    <XField name="calibB4" byteoffset="19" length="1"
type="uint8"/>
    <XField name="calibB5" byteoffset="20" length="1"
type="uint8"/>
    <XField name="calibB6" byteoffset="21" length="1"
type="uint8"/>
    <XField name="calibB7" byteoffset="22" length="1"
type="uint8"/>
    </XFields>
    <XFilter>
        <XCondAnd>

            <!--
                Check if this is a sensor type
                packet...AMTYPE_MULTIHOP or AMTYPE_XDEBUG
                The Byte offsets defined above assume
                that there is a multihop header
            /-->
            <XCondOr>
                <XCond name="IsEqual">
                    <XFilterParam name="fieldname"
value="amtype"/>
                    <XFilterParam name="fieldvalue"
value="0x00"/>
                </XCond>

                <XCond name="IsEqual">
                    <XFilterParam name="fieldname"
value="amtype"/>
                    <XFilterParam name="fieldvalue"
value="0x32"/>
                </XCond>

            </XCondOr>
            <XCond name="IsEqual">
                <XFilterParam name="fieldname"
value="board_id"/>
                <XFilterParam name="fieldvalue"
value="0x85"/>
            </XCond>

```

```

        </XCond>
        <XCondOr>
            <XCond name="IsEqual">
                <XFilterParam name="fieldname"
value="packet_id"/>
                <XFilterParam name="fieldvalue"
value="0x81"/>
            </XCond>

            <XCond name="IsEqual">
                <XFilterParam name="fieldname"
value="packet_id"/>
                <XFilterParam name="fieldvalue"
value="0x01"/>
            </XCond>
        </XCondOr>
    </XCondAnd>
</XFilter>
<XDataSinks>
    <XDataSink name="Generic Print Datasink">
        <XDSPParam name="printstring" value="MTS400
[sensor data converted to engineering units]:\n  health:      node
id = %s\n  battery:      = %s mv \n  humid:      =
%s %% \n  Temperature:  = %s degC \n  IntersemaTemperature:
= %s degC \n  IntersemaPressure:  = %s mbar \n  Light :
= %s lux\n  X-axis Accel:      = %s mg \n  Y-axis Accel:
= %s mg "/>
        <XDSPParam name="printfields"
value="nodeid,voltage,humid,humtemp,prtemp,press,taoch0,accel_x,accel_
y"/>
    </XDataSink>
    <XDataSink name="Open Log Datasink">
        <XDSPParam name="tablename_1"
value="mts400_results"/>
        <XDSPParam name="createsql_1" value="CREATE
TABLE mts400_results ( result_time timestamp without time zone,epoch
integer, nodeid integer, parent integer,voltage integer, humid
integer, humtemp integer,prtemp integer,press integer,taosch0
integer,taosch1 integer,accel_x integer, accel_y integer, calibW0
integer, calibW1 integer, calibW2 integer, calibW3 integer)"/>

        <XDSPParam name="tablename_2"
value="mts400_results_1"/>
        <XDSPParam name="createsql_2" value="CREATE
TABLE mts400_results_1 ( result_time timestamp without time zone,epoch
integer, nodeid integer, parent integer,voltage integer, humid
integer, humtemp integer,prtemp integer,press integer,taosch0
integer,taosch1 integer,accel_x integer, accel_y integer, calibW0
integer, calibW1 integer, calibW2 integer, calibW3 integer)"/>

        <XDSPParam name="rulesql_2" value="CREATE RULE
cache_mts400_results AS ON INSERT TO mts400_results DO ( DELETE FROM
mts400_results_1 WHERE nodeid = NEW.nodeid;INSERT INTO
mts400_results_1 VALUES (NEW.*); )"/>
        <XDSPParam name="insertsql_2" value="INSERT
into mts400_results
(result_time,nodeid,voltage,humid,humtemp,prtemp,press,taosch0,taosch1
,accel_x,accel_y,calibW0,calibW1,calibW2,calibW3) values
(now(),%i,%i,%i,%i,%i,%i,%i,%i,%i,%i,%i,%i,%i,%i)"/>

```

```

        <XDSPParam name="insertfields_2"
value="nodeid,voltage,humid,humtemp,prtemp,press,taoch0,taoch1,accel_x
,accel_y,calibw0,calibw1,calibw2,calibw3 "/>
        <XDSPParam name="tablename_3"
value="mts400_results_i"/>
        <XDSPParam name="createsql_3" value="CREATE
TABLE mts400_results_i ( nodeid integer ,x_coord integer, y_coord
integer, z_coord integer,mote_name text, calib bytea , boardid
integer)"/>
        <XDSPParam name="updatesql_3" value="update
mts400_results_i set calib='\\%o\\%o\\%o\\%o\\%o\\%o\\%o\\%o' where
nodeid=%i"/>
        <XDSPParam name="updatefields_3"
value="calibB0,calibB1,calibB2,calibB3,calibB4,calibB5,calibB6,calibB7
,nodeid"/>
        <XDSPParam name="insertsql_3" value="INSERT
into mts400_results_i ( nodeid,calib) values
(%i,'\\%o\\%o\\%o\\%o\\%o\\%o\\%o\\%o')"/>
        <XDSPParam name="insertfields_3"
value="nodeid,calibB0,calibB1,calibB2,calibB3,calibB4,calibB5,calibB6,
calibB7"/>
    </XDataSink>
    <XDataSink name="Generic test Datasink">
        <XDSPParam name="voltage" value="[2500,3500]"/>
        <XDSPParam name="humid" value="[0,100]"/>
        <XDSPParam name="humtemp" value="[-5,45]"/>
        <XDSPParam name="prtemp" value="[-5,45]"/>
        <XDSPParam name="press" value="[900,1100]"/>
        <XDSPParam name="accel_x" value="[-
2000,2000]"/>
        <XDSPParam name="accel_y" value="[-
2000,2000]"/>
        <XDSPParam name="taoch0" value="[0,500]"/>
    </XDataSink>
</XDataSinks>
</XFieldExtractor>

</XServeConfig>

```

Παράρτημα Ε

Πιο κάτω παρατίθεται ένα σύντομο tutorial για να γίνει η συνένωση του SQL Server με κάποιο άλλο linked server:

1) Download & install 64bit version of PostgreSQL psqlODBC (if server running x64) from:

<http://www.postgresql.org/ftp/odbc/versions/msi/>

I downloaded this version:

psqlodbc_09_01_0100-x64.zip

2) Two ways to create a linked server in SQL Server 2005:

2i) Execute the following query:

```
EXEC master.dbo.sp_addlinkedserver @server = N'LINKED_MOTES',  
@srvproduct=N'PostgreSQL',  
@provider=N'MSDASQL.1', @provstr=N'Driver=PostgreSQL  
ANSI(x64);uid=tele;port=5432;Server=ServerIP;database=task;pwd=tiny'
```

IMPORTANT:

From SQL Server 2005 > Server objects > Right click on LINKED_MOTES > Properties > Security

Then check "Be made using this security context:", Remote login: tele, With password: tiny

If you forget this step, probably stored procedures that access the linked server will fail!

2ii) Set up ODBC data source

2.1 Start > Run... and Enter in Open textbox:
%SystemRoot%\system32\odbcad32.exe and click OK.

or

2.1 Start > Control Panel > System and Security > Administrative Tools

2.2 Click Add to add a new data source.

2.3 Select driver for PostgreSQL (e.g. PostgreSQL ANSI(x64)) and then click finish.

2.4 Set up data source connection.

2.4.1 Name the data source as you like or leave default (e.g. PostgreSQL30).

2.4.2 Set the name of database you want to access.

2.4.3 Set the server IP address, where PostgreSQL server is running or localhost.

2.4.4 Set the user name and password to authenticate access.

2.4.5 Default port of PostgreSQL is 5432.

2.4.6 Leave SSL mode to disable.

2.4.7 Click test to see if everything is ok. A popup window must show up telling you "Connection successful".

If not check that PostgreSQL server firewall has an exception for port 5432 and is allowing TCP remote connections.

IMPORTANT:

To access the sensors database I have used the following:

- a) Data Source: PostgreSQL30
- b) Database: task
- c) Server: My router's IP (I configured a port forwarding to router's settings, TCP 5432 --> PostgreSQL Server local IP)
- d) User name: tele
- e) Password: tiny
- f) SSL mode: disable
- g) Port: 5432

2.5 From SQL Server 2005 > Server objects > Right click on Linked Servers and click "New linked server..."

2.6 For linked server name enter what you like.

2.7 For provider select Microsoft OLE DB provider for ODBC drivers.

2.8 Product name enter PostgreSQL.

2.9 Data source must match with the name of step 2.4.1.

2.10 Leave blank and click ok.

2.11 If error occurred proceed with the 2i.

3) Check the linked server by executing the query:

```
SELECT a.*  
FROM OPENQUERY(LINKED_MOTES, 'SELECT table_name FROM  
information_schema.tables') As a;
```

Information:

LINKED_MOTES is the name I gave to the linked server