

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ**



**ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΠΟΛΥΜΕΣΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ:**

**ΑΝΑΠΤΥΞΗ ΑΣΥΡΜΑΤΟΥ ΔΙΚΤΥΟΥ ΕΠΙΤΗΡΗΣΗΣ ΦΥΣΙΚΩΝ ΜΕΓΕΘΩΝ ΜΕ  
ΧΡΗΣΗ ΤΕΧΝΟΛΟΓΙΑΣ GOOGLE API ΚΑΙ WEB 2.0**

**ΓΑΛΑΝΑΚΗΣ ΝΙΚΟΣ Α.Μ.1628**

**ΠΡΩΤΟΨΑΛΤΗΣ ΔΗΜΗΤΡΙΟΣ Α.Μ. 1557**

**ΕΙΣΗΓΗΤΗΣ : ΡΗ.Δ ΒΛΙΣΙΔΗΣ ΑΝΔΡΕΑΣ**



## **Ευχαριστίες**

Αρχικά θα θέλαμε να ευχαριστήσουμε το καθηγητή μας Κ. Ανδρέα Βλησίδα ο οποίος ήταν ο εισηγητής της πτυχιακής μας εργασίας . Μας βοήθησε πάρα πολύ με την απόκτηση των υλικών που είχαμε ανάγκη να χρησιμοποιήσουμε για να αναπτύξουμε την πλατφόρμα αλλά και με δική του παρότρυνση σχεδιάστηκε , αναπτύχθηκε και υλοποιήθηκε όλο το project . Ευχαριστούμε επίσης το συνεργάτη του εργαστηρίου και πτυχιούχο της ίδιας σχολής τον Σταύρο Χαρακόπουλος ο οποίος εργάζεται σε μια από τις πιο ανεπτυγμένες εταιρείες της Ευρώπης Πλαστικά Κρήτης Α.Ε. του οποίου πείρα , το ευρύ φάσμα γνώσεών του και η καλή του θέληση να μας προσφέρει την κάθε είδους βοήθεια τον καθιστά ως βασικό παράγοντα επίτευξης του project.Τέλος ευχαριστούμε τις οικογένειες μας που μας έδωσαν απλόχερα την ευκαιρία να σπουδάσουμε κάτι το οποίο μας αρέσει ώστε να έχουμε τα προσόντα να εργαστούμε σε αυτόν τον τομέα.

Και πάλι ευχαριστούμε.

## ***Abstract***

*The advances on miniaturization techniques have made possible the development of a new network paradigm, the Wireless Sensor Networks (WSNs). The main purpose of a Wireless Sensor Network as a whole is to serve as an interface to the real world, providing physical information such as temperature, light, radiation, and others, to a computer system. These types of networks have a simple structure: there are dozens or 100s of elements, called “sensor nodes” that are able to sense the physical features of their surroundings. After such information is processed by these nodes, it is sent through a wireless channel to a central system, called “Base Station”. It is possible to abstract the nodes as the “sensing cells” of a living system, where the Base Station can be considered as the “central brain”. The services offered by a WSN can be classified into three major categories: monitoring, alerting, and information “On-Demand”. Sensor nodes can continuously monitor certain features of their surroundings (e.g. measuring the radiation level). Sensors can also check whether certain physical conditions (e.g. a radiation leak) are taking place, alerting the users of the system if an alarm is triggered. Finally, the network can be queried about the actual levels of a certain feature, providing information “On-Demand”. It is important to note that the computational capabilities of the nodes allow to automatically reconfigure their internal operation during the lifetime of the network, or even use them as a distributed computing platform or communication platform under extreme circumstances .Our purpose was to built a platform based on WSN that is monitoring a delimited forest area to be able to watch every parameter that be able to start “bad event” like fire. Early detection of forest fires is the primary way of minimizing their damages .As a result of completing the above platform was that we could prevent in any situation that was coming up .Saving the all the parameters and monitoring them all the time the fire protection of forest region could be easier and with no human risk .Also the patrols of firemen would be more effective and at the same time less .Can you imagine the period of summer with no fires at ours forests ?This is the best result of our platform .In future we could add in our platform a new prevention system that has a mechanism which throws water in fire situation. So we conclude that the technology of WSNs could help the human resources and relevant government departments with extreme directly way for the common good and for the protection of sources of oxygen.*



## Περιεχόμενα

Πρόλογος.....	9
ΚΕΦ.1 Τα Nodes.....	10
1.1 Sensor board.....	10
1.1.1 Αισθητήριο Θερμοκρασίας-Υγρασίας.....	11
1.1.2 Αισθητήριο βαρομετρικής πίεσης και θερμοκρασίας.....	11
1.1.3 Αισθητήρας φωτός.....	12
1.1.4 Axis Accelerometer.....	13
1.2 Processor/Radio Board.....	14
1.2.1 Χαρακτηριστικά του IRIS Processor Radio Board.....	14
1.2.2 Επεξεργαστής και πλατφόρμα επικοινωνίας.....	15
1.3 Το Base Station αποτελείτε και αυτό από δύο κομμάτια.....	15
1.3.1 Processor/Radio Board.....	15
1.3.2 USB PC Interface Board.....	15
ΚΕΦ.2 WSN.....	17
2.1 Γενικά.....	17
2.2 Εφαρμογές.....	18
2.2.1 Παρακολούθηση περιοχής.....	18
2.2.2 Παρακολούθηση της ρύπανσης του αέρα.....	18
2.2.3 Ανίχνευση δασικών πυρκαγιών.....	18
2.2.4 Παρακολούθηση θερμοκηπίου.....	18
2.2.5 Ανίχνευση κατολίσθησης.....	18
2.3 Βιομηχανική παρακολούθηση.....	19
2.3.1 Παρακολούθηση της “υγείας” των μηχανών.....	19
2.3.2 Παρακολούθηση ύδρευσης / αποχέτευσης.....	19
2.3.3 Γεωργία.....	19
2.3.4 Παρακολούθηση υποδομών.....	19
2.4 Χαρακτηριστικά.....	19
2.5 ZigBee.....	20
2.5.1 Γενικά.....	20
2.5.2 Τεχνική επισκόπηση.....	20

2.5.3 Χρησιμότητα.....	21
2.5.4 Συσκευές ZigBee.....	22
2.6 Τοπολογίες Δικτύου ZigBee.....	22
2.6.1 Τοπολογία Αστέρα.....	22
2.6.2 Τοπολογία Δέντρου ή Συστάδας ( Cluster – Tree Topology ).....	22
2.6.3 Τοπολογία Πλέγματος (Mesh Topology, Peer to Peer).....	23
ΚΕΦ.3 Επισκόπηση του MoteView.....	24
3.1 Γενικά.....	24
3.2 Καρτέλες Οπτικοποίησης .....	25
3.2.1 Data .....	25
3.2.2 Command .....	25
3.2.3 Chart .....	26
3.2.4 Health .....	27
3.2.5 Histogram .....	27
3.2.6 Scatterplot .....	28
3.2.7 Topology .....	29
3.3 Connect to WSN .....	29
Κεφ.4 Λειτουργικό Σύστημα.....	31
4.1 Γενικά.....	31
4.2 Τα βασικά στοιχεία του TinyOs .....	31
4.2.1 Components .....	31
4.2.2 Commands ( εντολές ).....	33
4.2.3 Events (γεγονότα).....	33
4.2.4 Tasks (διεργασίες).....	33
4.2.5 Χρονοπρογραμματιστής.....	34
4.3 nesC .....	34
4.3.1 Interface .....	34
4.3.2 Module .....	35
4.3.3 Configuration.....	37
4.3.4 Generic Components.....	38
4.3.5 Tasks .....	39

4.3.6 Async συναρτήσεις και Atomic Statements .....	39
Κεφ.5 Java Application.....	40
5.1 Λίγα λόγια για την Java .....	40
5.1.1 Ολοκληρωμένο περιβάλλον ανάπτυξης(IDE).....	41
5.1.2 Εισαγωγικά για την εφαρμογή.....	41
5.2. Πληροφορίες σχετικά με την εφαρμογή του GUI. ....	41
5.2.1 Κάνοντας Back-up της βάσης από το Menu του API.....	42
5.2.1.1 Εισαγωγή JDBC.....	43
5.2.1.2 Αποθήκευση σε XML αρχείο .....	45
5.2.2 Ενσωμάτωση χάρτη στην εφαρμογή μας. ....	48
5.2.3 Δημιουργία Γραφημάτων.....	52
5.2.4 Δημιουργία Alerts για την εφαρμογή .....	56
Επίλογος .....	60
Παράρτημα.....	61



## Πρόλογος

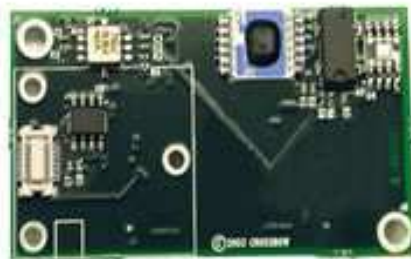
Με στόχο να εκπληρώσουμε την υποχρέωση μας σαν φοιτητές του τμήματος Εφαρμοσμένης Πληροφορικής και Πολυμέσων την εκπόνηση της πτυχιακής εργασίας μας επιλέξαμε ένα θέμα που σαφώς έχει άμεση σχέση με τον κόσμο της Πληροφορικής αλλά αφορά και διάφορους κρατικούς και κοινωνικούς φορείς των οποίων η δράση έχει άμεση και έμμεση σχέση με την προστασία του φυσικού μας περιβάλλον . Αν σκεφτούμε και το γεγονός ότι καθημερινά γίνονται ατασθαλίες και απεισκευσίες εις βάρος του καταλήγουμε ότι χρήζει ανάγκη να δημιουργηθούν κάποια αποτελεσματικά αντίμετρα . Η πλατφόρμα που δημιουργήσαμε έχει σαν βάση την χρήση ενός δικτύου ασύρματων αισθητηρίων που επικοινωνούν μεταξύ τους και όλα μαζί στέλνουν σε μια κεντρική μονάδα που δέχεται και αποθηκεύει τις αντίστοιχες τιμές των αισθητηρίων δηλαδή όλα αυτά που αποτελείται ένα Wireless Sensor Network(WSN).Επίσης σαν δεύτερο κύριο στοιχείο της πλατφόρμας μας είναι μια Java desktop application η οποία με τα εργαλεία που μας παρέχει η Java σαν γλώσσα προγραμματισμού υλοποιήσαμε ένα γραφικό περιβάλλον το οποίο δίνει την δυνατότητα στο κάθε χρήστη να κάνει την εποπτεία του χώρου όπου είναι τοποθετημένα οι ασύρματες μονάδες . Από την εφαρμογή αυτή μπορεί να δει γραφήματα με τις τιμές των αισθητηρίων σε σχέση με το χρόνο , να παραμετροποιήσει τις τιμές που είναι αποδεκτές και δεν δημιουργούν κάποιο alert και το βασικό είναι ότι ο χρήστης θα ενημερώνεται άμεσα και αποτελεσματικά από το σύστημα για τον αν συμβεί κάποιον συμβάν που θα δημιουργήσει πρόβλημα π.χ. η τάση της μπαταρίας του node φτάσει σε χαμηλά επίπεδα , δημιουργηθούν συνθήκες που είναι προφανές ότι θα υπάρξει ο κίνδυνος πυρκαγιάς .Η χρησιμότητα της ανάπτυξης της πλατφόρμας είναι προφανές ότι έχει θετικό αντίκτυπο και σε όλο το κοινωνικό σύνολο αλλά και φέρνει την ανάπτυξη και τη χρήση των τεχνολογικών επιτευγμάτων με πολύ φιλικό τρόπο κοντά στην προστασία του περιβάλλοντός μας . Και αυτό πρέπει όλοι να το συνειδητοποιήσουμε και να προσπαθούμε ο καθένας όπως μπορεί να γίνεται υλοποιήσιμη κάθε τέτοια εφαρμογή της τεχνολογίας.

## ΚΕΦ.1 Τα Nodes

Για την υλοποίηση του προγράμματος μας χρησιμοποιήσαμε ένα πακέτο τριών Nodes το οποίο αποτελείται από δύο Sensor nodes και ένα Base station node . Τα συγκεκριμένο πακέτο έχει τέτοια χαρακτηριστικά ως προς το φάσμα που εκπέμπει την πολύ χαμηλή κατανάλωση που έχει σε ενέργεια αλλά και την ισχυρή προγραμματιστική ισχύ που μας προσφέρει . Αυτά τα χαρακτηριστικά τα οποία αναλύονται παρακάτω το κάνει κατάλληλο για τη χρησιμοποίηση στο project μας.

### 1.1 Sensor board

Το Sensor node αποτελείται από δύο κομμάτια .Το MTS400 μας προσφέρει πέντε βασικά αισθητήρια με την επιλογή και GPS στο μοντέλο MTS420.Η συγκεκριμένη πλακέτα προσφέρετε για ένα ευρύ φάσμα εφαρμογών , από ένα απλό ασύρματο μετεωρολογικό σταθμό έως ένα πολύπλοκο ασύρματο δίκτυο αισθητήρων (WSN) για επιτήρηση διαφόρων περιοχών (βιομηχανίες, δάση, σπίτια κ.α. ). Το MTS400 είναι τελευταίας ενεργειακής κλάσης έτσι παρατείνει τη ζωή των μπαταριών στο μέγιστο χρησιμοποιώντας λογισμικό για low-Voltage.



Εικόνα 1 : MTS400

Περιλαμβάνει αισθητήρια για :

Θερμοκρασία

Υγρασία

Βαρομετρική πίεση

Φωτεινότητα

### 1.1.1 Αισθητήριο Θερμοκρασίας-Υγρασίας

Το αισθητήριο Θερμοκρασίας υγρασίας είναι το SHT11 το οποίο είναι ένα Chip με διπλό αισθητήριο που έχει μια βαθμονομημένη ψηφιακή έξοδο, 14-bit είσοδο με αναλογικό σε ψηφιακό μετατροπέα. Συνδέετε στην υπόλοιπη πλακέτα με μία σειριακή διεπαφή. Η ενέργεια που χρειάζεται για να λειτουργήσει ελέγχεται από το λογισμικό του προγραμματιστή. Ο μετατροπέας από αναλογικό-ψηφιακό προγραμματίζετε έτσι ώστε να μας δώσει τις ψηφιακές τιμές που θέλουμε για θερμοκρασία και υγρασία.

#### Προδιαγραφές SHT11

Sensor Type	Sensirion SHT11	
Channels	Humidity	Temperature
Range	0 to 100%	-40°C to 80°C
Accuracy	± 3.5% RH (typical)	± 2°C
Operating Range	3.6 to 2.4 volts	
Interface	Digital interface	

### 1.1.2 Αισθητήριο βαρομετρικής πίεσης και θερμοκρασίας

Το αισθητήριο MS55ER της INERSEMA είναι μια SMD-υβριδική συσκευή η οποία περιλαμβάνει παιδομετρικό αισθητήρα πίεσης και μια ADC διεπαφή. Μας παρέχει μια 16 bit λέξη που περιλαμβάνει μέτρηση πίεσης και θερμοκρασίας. Όλες οι επικοινωνίες με τη βάση γίνονται μέσω μίας διεπαφής τριών καλωδίων. Όπως και στο SHT11 η ενέργεια λειτουργίας του αισθητηρίου και η ποσότητα των μετρήσεων εξαρτάτε από το λογισμικό που δημιουργεί ο προγραμματιστής. Με τον analog-to-digital μετατροπέα που υπάρχει στο αισθητήριο γίνετε και η μετατροπή των τιμών πίεσης και θερμοκρασίες σε ψηφιακές τιμές.

#### Προδιαγραφές MS55ER

<b>Sensor Type</b>	Intersema MS5534
<b>Channels</b>	Pressure and Temperature
<b>Range</b>	Pressure: 300 to 110 mbar Temperature: -10°C to 60°C
<b>Accuracy</b>	Pressure: ± 3.5% Temperature: ± 2°C
<b>Operating Range</b>	3.6 to 2.2 volts
<b>Interface</b>	Digital interface

### 1.1.3 Αισθητήρας φωτός

Ο TLS2550(αισθητήριο φωτός) είναι ένας ψηφιακός αισθητήρας φωτός με δύο καλωδιώσεις SMBus συριακή θύρα .Είναι κατασκευασμένο από την TAOS . Συνδυάζει δύο φωτοδιόδους σε μια σύνθεση με τον μετατροπέα analog-to-digital σε ένα ενιαίο ολοκληρωμένο κύκλωμα CMOS για την παροχή φωτομετρήσεων με αποτελεσματικό εύρος 12-bit. Η ισχύς και τα σήματα ελέγχου του αισθητηρίου ελέγχονται μέσω κατασκευής προγράμματος από τον προγραμματιστή. Ο μετατροπέας analog-to-digital κάνει τη μετατροπή από “φως” σε ψηφιακές μονάδες.

#### Προδιαγραφές TLS2550

<b>Sensor Type</b>	Taos TSL2550
<b>Channels</b>	Light
<b>Range</b>	400 – 1000 nm
<b>Operating Range</b>	3.6 to 2.7 volts

<b>Interface</b>	Digital interface
------------------	-------------------

#### 1.1.4 Axis Accelerometer

Το επιταχυνσιόμετρο άξονα είναι μια συσκευή με τεχνολογία MEMS (Micro-Electro-Mechanical Systems ) surface δύο αξόνων και έως +- 2g επιτάχυνση . Έχει πολύ χαμηλή κατανάλωση ρεύματος (<1mA) και μπορεί να χρησιμοποιηθεί για διάφορες λειτουργίες όπως ανίχνευση ,κλίση του “note”, κίνηση , δόνηση η ακόμα και για σεισμικές μετρήσεις . Η έξοδος του αισθητήρα είναι συνδεδεμένος στο ADC κανάλι του board.

#### Προδιαγραφές αισθητήρα φωτός

<b>Sensor Type</b>	Analog Devices ADXL202JE
<b>Channels</b>	X (ADC1), Y (ADC2)
<b>Range</b>	±2 G (1 G = 9.81 m/s <sup>2</sup> )
<b>Sensitivity</b>	167 mV/G, ±17 %
<b>Resolution</b>	2 mG (0.002 G) RMS
<b>Offset</b>	VBATTERY/2 ±0.4 V
<b>Operating Range</b>	3.6 to 3.0 V
<b>Interface</b>	Analog interface

## 1.2 Processor/Radio Board

Σε αυτό το κομμάτι έχουμε ένα IRIS Processor / radio board (χρησιμοποιείτε και στο Base station) το οποίο λειτουργεί στα 2.4GHz με δυνατότητα λειτουργίας χαμηλής ισχύος. Το IRIS Mote διαθέτει πολλές νέες δυνατότητες που βελτιώνουν τη συνολική λειτουργικότητα των προϊόντων ασύρματης δικτύωσης αισθητήρων.



*Εικόνα 2 : IRIS Processor / Radio Board*

### 1.2.1 Χαρακτηριστικά του IRIS Processor Radio Board

- Έως και τρεις φορές μεγαλύτερη εμβέλεια και διπλάσια program memory σε σχέση με τα προηγούμενα MICA motes.
- Σε περίπτωση που τα IRIS motes έχουν οπτική επαφή έχει γίνει δοκιμή ότι μπορούν να επικοινωνήσουν σε απόσταση πεντακοσίων μέτρων χωρίς καμία ενίσχυση.
- Το IEEE 802.15.4 που χρησιμοποιεί είναι συμβατό με RF πομποδέκτη.
- Στα 2.4 – 2.8 Ghz που λειτουργεί είναι συμβατή με το παγκόσμιο ISM φάσμα .
- Προσφέρει στο σήμα μας μεγάλη ανθεκτικότητα στις παρεμβολές και παρέχει εγγενή ασφάλεια δεδομένων.
- 250 kbps ρυθμό δεδομένων



*Εικόνα 3 : IRIS Processor / Radio Board*

### **1.2.2 Επεξεργαστής και πλατφόρμα επικοινωνίας**

Στο IRIS Processor/Radio Board χρησιμοποιήστε έναν επεξεργαστή XM2110CA ο οποίος είναι βασισμένος σε έναν ATmega 128. Ο ATmega 128 είναι ένας χαμηλής κατανάλωσης επεξεργαστής που προγραμματίζεται μέσω της εσωτερικής του μνήμης. Ένας τέτοιος επεξεργαστής έχει τη δυνατότητα να τρέχει τις εφαρμογές μας όσο αφορά τα αισθητήρια αλλά και τη δικτυακή επικοινωνία μεταξύ και των υπόλοιπων nodes.

## **1.3 Το Base Station αποτελείτε και αυτό από δύο κομμάτια**

### **1.3.1 Processor/Radio Board**

Είναι ακριβώς το ίδιο με το κομμάτι του sensor board μόνο εδώ χρησιμοποιείται για διαφορετικές λειτουργίες. Το IRIS λειτουργεί σαν Base station εάν το συνδέσουμε στον υπολογιστή με μια usb θύρα.

### **1.3.2 USB PC Interface Board**

Το MIB520 usb interface προσφέρει μια USB θύρα για τη μεταφορά των δεδομένων αλλά και για τον over the air ή μέσω usb προγραμματισμό των Sensor Nodes. Για αυτό το σκοπό μας προσφέρει δύο ports ένα για το προγραμματισμό των nodes και ένα για την μεταφορά των δεδομένων στον υπολογιστή μέσω θύρας usb.



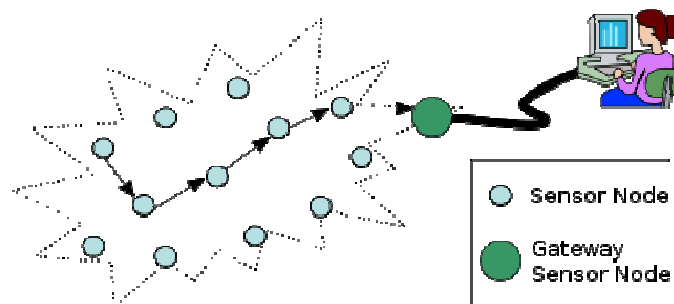
*Εικόνα 4 : MIB520*



## ΚΕΦ.2 WSN

### 2.1 Γενικά

Ένα ασύρματο δίκτυο αισθητήρων αποτελείται από χωρικά διάσπαρτους αυτόνομους αισθητήρες οι οποίοι χρησιμοποιούνται για την παρακολούθηση φυσικών ή περιβαλλοντολογικών συνθηκών , όπως θερμοκρασία , ήχος , δόνηση , πίεση , κ.α. ,καθώς και να περάσει συνεταιριστικά τα δεδομένα μέσω του δικτύου σε μια κύρια τοποθεσία. Τα πιο σύγχρονα δίκτυα είναι διπλής κατεύθυνσης δηλαδή επιτρέπουν και τον έλεγχο της δραστηριότητας των αισθητήρων . Η ανάπτυξη του WSN υποκινήθηκε για στρατιωτικές εφαρμογές όπως επιτήρηση του πεδίου μάχης . Σήμερα τέτοιου είδους WSNs χρησιμοποιούνται σε πολλές βιομηχανικές και καταναλωτικές εφαρμογές , όπως η παρακολούθηση και ο έλεγχος της βιομηχανικής διαδικασίας , την υγεία των βιομηχανικών μηχανών και ου το καθεξής.



Εικόνα 5 : WSN (Wireless sensor network)

Το WSN αποτελείται από “κόμβους” από λίγους έως αρκετές εκατοντάδες ακόμα και χιλιάδες , όπου κάθε κόμβος συνδέεται με αισθητήρες (μπορεί και με έναν μόνο). Κάθε τέτοιος κόμβος του δικτύου αισθητήρων αποτελείται συνήθως από κάποια συγκεκριμένα μέρη : ένα ασύρματο πομποδέκτη συνδεδεμένο με μία εσωτερική κεραία ή με κάποια εξωτερική , ένα μικροελεκτή , ένα ηλεκτρονικό κύκλωμα για τη διασύνδεση με τους αισθητήρες και τη πηγή ενέργειας που συνήθως είναι μπαταρίες η μια ενσωματωμένη μορφή συγκομιδής ενέργειας (π.χ ηλιακό πάνελ). Οι κόμβοι αισθητήρων ποικίλουν σε μέγεθος , μπορούν να έχουν το από το μέγεθος ενός μικρού κουτιού έως ενός κόκκου σκόνης . Στα εργαστήρια προωθούνται προγράμματα για την κατασκευή πραγματικά μικροσκοπικών διαστάσεων κόμβων αισθητήρων. Το κόστος τους είναι επίσης μεταβλητό κυμαίνεται από μερικές εκατοντάδες ευρώ μέχρι αρκετά μικρά ποσά εξαρτάτε από την πολυπλοκότητα των μεμονωμένων κόμβων αισθητήρων . Το μέγεθος και το κόστος κατασκευής των κόμβων αισθητήρων οδηγεί σε αντίστοιχο περιορισμό των δυνατοτήτων τους όσο αφορά ενέργεια , μνήμη ,υπολογιστική ισχύ , εύρος επικοινωνίας . Η τοπολογία ενός WSN μπορεί να ποικίλει από ένα απλό δίκτυο αστέρα έως ένα προηγμένο multi-hop ασύρματο δίκτυο πλέγματος . Τέλος η τεχνική μετάδοσης δεδομένων μεταξύ κόμβων και βάσης μπορεί να είναι είτε με μέθοδο δρομολογητή είτε με μέθοδο πλημμύρα.

## 2.2 Εφαρμογές

### 2.2.1 Παρακολούθηση περιοχής

Η παρακολούθηση περιοχής είναι μια τυπική εφαρμογή των WSNs . Σε αυτό το πλαίσιο το WSN αναπτύσσεται σε μια περιοχή όπου κάποια φαινόμενα πρέπει να παρακολουθούνται. Ένα στρατιωτικό παράδειγμα είναι η χρήση αισθητήρων για την ανίχνευση εισβολής του εχθρού . Ένα πολιτικό παράδειγμα η γέο - περιφραξη φυσικών πόρων ενός κράτους όπως φυσικό αέριο , πετρέλαιο , κ.α. Όταν οι αισθητήρες ανιχνεύουν ένα “γεγονός” στη παρακολούθηση περιοχής τότε το γεγονός αυτό αναφέρεται σε έναν από τους σταθμούς βάσης (base station) και η βάση με τη σειρά της λαμβάνει τα κατάλληλα μέτρα (π.χ : στέλνει ένα μήνυμα στο διαδίκτυο). Επίσης τα WSNs μπορούν να χρησιμοποιήσουν μια σειρά από διάφορους αισθητήρες ώστε να χρησιμοποιηθούν για την παρακολούθηση οχημάτων από μοτοσυκλέτες έως και τρένα .

### 2.2.2 Παρακολούθηση της ρύπανσης του αέρα

Υπάρχουν WSNs που έχουν αναπτυχθεί σε διάφορες πόλεις όπως Στοκχόλμη , Λονδίνο , Μπρίσμαν κ.α που συλλέγουν πληροφορίες για τη συγκέντρωση επικίνδυνων αερίων για τούς πολίτες

### 2.2.3 Ανίχνευση δασικών πυρκαγιών

Ένα WSN μπορεί να εγκατασταθεί σε ένα δάσος για να ελέγχει μια πυρκαγιά εφόσον έχει αρχίσει . Οι κόμβοι πρέπει να είναι εξοπλισμένοι με αισθητήρες θερμοκρασίας , υγρασίας και αισθητήρια ανίχνευσης αερίων που προέρχονται από την καύση των δέντρων και άλλων ειδών βλάστησης . Η έγκαιρη ανίχνευση είναι ζωτικής σημασίας για την επιτυχή δράση των ομάδων πυρόσβεσης χάρη στα WSNs η πυροσβεστική υπηρεσία θα είναι σε θέση να γνωρίζει τότε μια πυρκαγιά ξεκίνησε και πως εξελίσσεται .

### 2.2.4 Παρακολούθηση θερμοκηπίου

Τα WSNs επίσης χρησιμοποιούνται για να ελέγχουν τα επίπεδα θερμοκρασίας και υγρασίας στο εσωτερικό των θερμοκηπίων . Όταν η θερμοκρασία και η υγρασία πέσει κάτω από συγκεκριμένα επίπεδα ο διαχειριστής του θερμοκηπίου πρέπει να ενημερώνεται είτε μέσω ηλεκτρονικού ταχυδρομείου είτε μέσω κάποιου γραπτού μηνύματος στο κινητό του τηλέφωνο . Κατόπιν ενεργεί εκείνος είτε κάποιοι μηχανισμοί οι οποίοι είναι συνδεδεμένοι με το WSN του θερμοκηπίου .

### 2.2.5 Ανίχνευση κατολίσθησης

Ένα σύστημα ανίχνευσης κατολίσθησης , χρησιμοποιεί που έχει τη δυνατότητα ανίχνευσης μικρών κινήσεων του εδάφους αλλά και των αλλαγών διαφόρων παραμέτρων που μπορεί να συμβούν πριν η κατά τη διάρκεια μιας κατολίσθησης . Μέσα από αυτά τα στοιχεία που συγκεντρώνονται μπορεί να είναι δυνατό να γνωρίζουμε την εμφάνιση των κατολισθήσεων πολύ πριν αυτό συμβεί στην πραγματικότητα.

## 2.3 Βιομηχανική παρακολούθηση

### 2.3.1 Παρακολούθηση της “υγείας” των μηχανών

Έχουν αναπτυχτεί WSNs για να ελέγχουν τη κατάσταση των βιομηχανικών μηχανών με βάση τη συντήρηση έτσι προσφέρουν σημαντική εξοικονόμηση χρημάτων . Τώρα σε σχέση με ένα παρόμοιο αλλά ενσύρματο σύστημα το WSN υπερτερεί κατά κράτος διότι έχει φθηνότερο κόστος τοποθέτησης και εφαρμογής και δεν περιορίζεται π.χ: κινητά μέρη μηχανής , περιστρεφόμενα μέρη μηχανής , ζώνες περιορισμένης πρόσβασης κ.α.

### 2.3.2 Παρακολούθηση ύδρευσης / αποχέτευσης

Υπάρχουν πολλές ευκαιρίες για τη χρήση των WSNs στα δίκτυα ύδρευσης και αποχέτευσης . Εγκαταστάσεις οι οποίες δε καλωδιώνονται για παροχή ενέργειας και μεταφοράς δεδομένων μπορούν να ελέγχονται με τη χρήση βιομηχανικών ασύρματων συσκευών εισόδου/εξόδου και αισθητήρες οι οποίοι διοχετεύονται με ενέργεια μέσω ηλιακών πάνελ η μπαταριών.

### 2.3.3 Γεωργία

Η χρησιμοποίηση των WSNs στα πλαίσια του γεωργικού τομέα γίνεται όλο και πιο δεδομένη . Χρησιμοποιώντας WSNs ελευθερώνεται ο γεωργός από τη διατήρηση των καλωδίων σε ένα δύσκολο περιβάλλον. Συστήματα ζύγισης σε συνδυασμό με πομπές πίεσεως μπορούν να χρησιμοποιηθούν για τη παρακολούθηση της στάθμης των δεξαμενών νερού όπως επίσης μπορούν να ελέγχονται ποσότητες νερού που διέρχονται από της αντλίες ύδρευσης . Ακόμα και τα συστήματα ύδρευσης μπορούν να ελέγχονται από τα WSNs (π.χ κάθε πότε πρέπει να ποτίζει).

### 2.3.4 Παρακολούθηση υποδομών

Ασύρματοι αισθητήρες μπορούν να χρησιμοποιηθούν για την παρακολούθηση της κυκλοφορίας στο εσωτερικό κτιρίων και υποδομών , όπως γέφυρες , υπόγειες διαβάσεις , σήραγγες κ.α. . Ένα άλλο κομμάτι σε αυτό το τομέα είναι ότι με τις απαραίτητες μηχανολογικές πρακτικές μπορεί να γίνει παρακολούθηση περιουσιακών στοιχείων ξαποστάσεις.

## 2.4 Χαρακτηριστικά

Τα κύρια χαρακτηριστικά του περιλαμβάνουν WSN είναι τα εξής :

- Η κατανάλωση ενέργειας δε περιορίζει τους κόμβους γιατί χρησιμοποιούν μπαταρίες ή μηχανισμούς συγκομιδής ενέργειας.
- Κινητικότητα των κόμβων.
- Δυναμική τοπολογία του δικτύου.
- Δεν επηρεάζονται από τις διακοπές της επικοινωνίας μεταξύ τους.
- Ετερογένεια των κόμβων.
- Δυνατότητα ανάπτυξης του δικτύου σε μεγάλη κλίμακα .

- Ικανότητα να αντέχει σε σκληρές περιβαλλοντικές συνθήκες
- Ευκολία στη χρήση
- Λειτουργία χωρίς επίβλεψη.

Κόμβους μπορείτε να φανταστείτε ως μικρούς υπολογιστές, ιδιαίτερα όσον αφορά τις βασικές τους διασυνδέσεις και των εξαρτημάτων τους. Συνήθως αποτελείται από μια μονάδα επεξεργασίας με περιορισμένη υπολογιστική ισχύ και περιορισμένη μνήμη, αισθητήρες ή **MEMS** (συμπεριλαμβανομένων ειδικών κυκλωμάτων κλιματισμού), μια συσκευή επικοινωνίας (συνήθως ασύρματοι σταθμοί ή εναλλακτική οπτική επικοινωνία ), και μια πηγή ενέργειας συνήθως με τη μορφή μιας μπαταρίας. Άλλες πιθανές μορφές ενέργειας είναι μηχανισμοί συγκομιδής ενέργειας , και, ενδεχομένως, δευτεροβάθμια συσκευές επικοινωνίας (π.χ. RS-232 ή USB ).

Οι σταθμοί βάσης είναι ένας ή περισσότεροι κόμβοι που διακρίνονται από πολύ μεγαλύτερη υπολογιστική ισχύ και κατανάλωση ενέργειας σε σχέση με τους κόμβους αισθητήρων. Ενεργούν ως μια πύλη μεταξύ των κόμβων αισθητήρων και του τελικού χρήστη, καθώς συνήθως παρέχει δεδομένα από τα WSN με ένα διακομιστή. Άλλες ειδικές συστατικά δρομολόγησης με βάση τα δίκτυα είναι δρομολογητές, έχουν σχεδιαστεί για τον υπολογισμό, και τη διανομή τους μέσω πινάκων δρομολόγησης. Πολλές από τις τεχνικές που χρησιμοποιούνται για τη σύνδεση με τον έξω κόσμο, συμπεριλαμβανομένων των δικτύων κινητής τηλεφωνίας, δορυφορικά τηλέφωνα, modems ραδιόφωνο, υψηλής ισχύος WiFi συνδέσεις κλπ χρησιμοποιούνται και στα WSNs.

## 2.5 ZigBee

### 2.5.1 Γενικά

Το ZigBee είναι μια σειρά από υψηλού επιπέδου πρωτοκόλλων επικοινωνίας που χρησιμοποιούν μικρά ψηφιακά ραδιόφωνα ώστε να πετύχει χαμηλή κατανάλωση , βασίζεται στο πρότυπο IEEE 802 για τα προσωπικά δίκτυα . Οι εφαρμογές περιλαμβάνουν ασύρματους διακόπτες, μετρητές ηλεκτρικής κατανάλωσης και άλλων καταναλωτικό και βιομηχανικό εξοπλισμό που απαιτεί μικρής εμβέλειας ασύρματη μεταφορά δεδομένων σε σχετικά χαμηλά ποσοστά . Η τεχνολογία που ορίζεται από την προδιαγραφή ZigBee προορίζεται να είναι απλούστερη και λιγότερο δαπανηρή από ό, τι άλλες , όπως το Bluetooth . ZigBee απευθύνεται σε (RF) εφαρμογές που απαιτούν ένα χαμηλό ρυθμό δεδομένων, μεγάλη διάρκεια ζωής της μπαταρίας και ασφαλή δικτύωση.

### 2.5.2 Τεχνική επισκόπηση

Το ZigBee είναι χαμηλού κόστους, χαμηλής ισχύος , πρότυπο ασύρματων δικτύων πλέγματος . Το χαμηλό κόστος επιτρέπει η τεχνολογία που θα αναπτυχθεί ευρέως στις ασύρματες εφαρμογές ελέγχου και παρακολούθησης. Η χαμηλή κατανάλωση επιτρέπει μεγαλύτερη διάρκεια ζωής με μικρότερες μπαταρίες. Η δικτύωση Mesh παρέχει υψηλή αξιοπιστία και μεγαλύτερη έκταση. Η τεχνολογία προορίζεται να είναι απλούστερη και λιγότερο δαπανηρή από ότι άλλες WPANs όπως Bluetooth .

Το ZigBee δραστηριοποιείται στις βιομηχανικές, επιστημονικές και ιατρικές (ISM) ραδιοσυχνότητες. 868 MHz στην Ευρώπη, 915 MHz στις ΗΠΑ και την Αυστραλία, και 2,4 GHz στις περισσότερες χώρες σε όλο τον κόσμο. Οι ταχύτητες μετάδοσης δεδομένων διαφέρουν από 20 έως 250 kilobits / δευτερόλεπτο.

Το επίπεδο δικτύου του πρωτοκόλλου ZigBee βασίζεται στο πρότυπο IEEE 802.15.4, υποστηρίζει δίκτυο αστέρα και το χαρακτηριστικό τύπου δένδρο δίκτυο, και γενικά τα δίκτυα πλέγματος. Κάθε δίκτυο πρέπει να έχει μία συσκευή συντονιστή, επιφορτισμένη με τη δημιουργία του, τον έλεγχο των παραμέτρων της, και βασική συντήρηση. Στο δίκτυο αστέρα, ο συντονιστής πρέπει να είναι ο κεντρικός κόμβος. Οι ZigBee δρομολογητές σε όλα τα δίκτυα πλέγματος επιτρέπουν την επέκταση της επικοινωνίας σε επίπεδο δικτύου.

Το IEEE 802.15.4 είναι ένα πρότυπο που ορίζει το φυσικό επίπεδο (physical layer) και τον έλεγχο πρόσβασης μέσου (medium access control) για μικρής εμβέλειας δίκτυα χαμηλής ταχύτητας (Low-rate personal area networks). Πάνω σε αυτό στηρίζεται το πρωτόκολλο ZigBee, που προσφέρει την ολοκληρωμένη λύση για δίκτυο παρέχοντας τα υπόλοιπα επίπεδα που δεν ορίζονται από το πρότυπο. Το ZigBee προσθέτει τα δύο επίπεδα που υπολείπονται αυτά είναι το επίπεδο δικτύου και το επίπεδο εφαρμογών. Εκτός από τα δύο επίπεδα που προανέφερα υπάρχει ακόμα ένα που βρίσκεται πάνω από το επίπεδο εφαρμογών και είναι χαρακτηριστικό του πρωτοκόλλου ZigBee το ZigBee device objects (ZDO'S). Το ZDO'S είναι υπεύθυνο για έναν αριθμό από καθήκοντα όπως: το ρόλο κάθε συσκευής του δικτύου, τη διαχείριση των αιτημάτων των συσκευών, την εύρεση καινούριων συσκευών και την ασφάλεια του δικτύου. Τα δίκτυα Zigbee μπορούν να λειτουργήσουν είτε σε λειτουργία περιοδικής εκπομπής ενός σήματος συντονισμού, είτε σε λειτουργία μη εκπομπής. Στην πρώτη περίπτωση ένα σήμα αποστέλλεται περιοδικά από το συντονιστή, το οποίο σαν επακόλουθο έχει να «ξυπνά» όλες τις συσκευές του δικτύου οι οποίες πρέπει να ενημερώσουν τον συντονιστή αν έχουν κάποιο μήνυμα να αποστείλουν. Εάν όχι, τότε η κάθε συσκευή επιστρέφει σε κατάσταση αναμονής. Στην άλλη περίπτωση, όταν δεν υπάρχει αυτή η περιοδική εκπομπή του σήματος από τον συντονιστή, το δίκτυο το οποίο δημιουργείται είναι λιγότερο συντονισμένο, καθώς η κάθε τερματική συσκευή εκπέμπει ένα σήμα το οποίο θα πρέπει να φτάσει στο συντονιστή περνώντας από όλους τους ενδιάμεσους κόμβους του δικτύου. Σε αυτή την περίπτωση, ο συντονιστής θα πρέπει να είναι συνεχώς σε λειτουργία για να είναι έτοιμος σε κάθε σήμα που μπορεί να ληφθεί, καταναλώνοντας έτσι μεγαλύτερα ποσά ενέργειας. Σε κάθε περίπτωση όμως, ένα δίκτυο αποτελούμενο από συσκευές που ενσωματώνουν το πρωτόκολλο IEEE802.15.4 διατηρεί την κατανάλωση ισχύος σε χαμηλά επίπεδα διότι η πλειοψηφία των συσκευών του δικτύου παραμένουν ανενεργές για μεγάλα χρονικά διαστήματα.

### 2.5.3 Χρησιμότητα

Το Zigbee είναι σχεδιασμένο έτσι ώστε να μπορεί να ενσωματωθεί σε ένα πλήθος συσκευών στο σπίτι ή το γραφείο, για παράδειγμα σε φωτισμούς, διακόπτες, εισόδους και ηλεκτρικές συσκευές. Αυτές οι συσκευές μπορούν να αλληλεπιδράσουν χωρίς την χρήση καλωδιώσεων και μπορούν να ελεγχθούν από μία και μόνη συσκευή η οποία μπορεί να είναι ένα κινητό τηλέφωνο ή ένα τηλεχειριστήριο. Παρά το γεγονός ότι η τεχνολογία που εισάγει δεν είναι επαναστατική, προχωράει ένα βήμα παραπέρα από τις παραδοσιακές ασύρματες επικοινωνίες όπως ο απλός τηλεχειρισμός για το άνοιγμα της γκαραζόπορτας ή το άναμμα του φωτισμού. Το σημείο που διαφοροποιείται από αυτές τις εφαρμογές είναι το γεγονός ότι το πρωτόκολλο 802.15.4 επιτρέπει την επικοινωνία δύο δρόμων μεταξύ όλων των συσκευών στις οποίες

ενσωματώνεται, δηλαδή τα φώτα, τους διακόπτες, τους θερμοστάτες, τον κλιματισμό και λοιπά.

#### 2.5.4 Συσκευές ZigBee

Υπάρχουν δύο τύποι συσκευών που ορίζει η προτυποποίηση IEEE 802.15.4. Η πρώτη είναι η πλήρης IEEE 802.15.4 συσκευή (Full Function Device – FFD) και μπορεί να εκτελέσει όλες τις απαιτούμενες από το δίκτυο λειτουργίες. Μία τυπική FFD, συνήθως τροφοδοτείται από ανεξάντλητη πηγή (τροφοδοτικό AC από την ηλεκτρική εγκατάσταση). Πρέπει να είναι συνεχώς ενεργοποιημένη και συνδεδεμένη με το ασύρματο δίκτυο. Ο δεύτερος τύπος συσκευών είναι οι συσκευές περιορισμένων δυνατοτήτων (Reduce Function Device – RFD). Οι εργασίες που μπορούν να εκτελέσουν περιορίζονται στον έλεγχο εξωτερικών συσκευών και διακοπών και στη δειγματοληψία αισθητήρων. Συνήθως, επειδή τροφοδοτούνται μέσω μπαταριών, είναι προγραμματισμένες να κοιμούνται για μεγάλο χρονικό διάστημα. Το πρωτόκολλο ZigBee παίρνει τους ορισμούς των συσκευών FFD και RFD του IEEE 802.15.4 και ορίζει τρεις τύπους δικών του συσκευών. Ο Συντονιστής δικτύου (ZigBee Coordinator), είναι μία FFD συσκευή, μοναδική ανά δίκτυο ZigBee, και είναι αυτή που το δημιουργεί. Μόλις ο Συντονιστής ορίσει το δίκτυο, αναθέτει διευθύνσεις δικτύου στις συσκευές που επιτρέπεται να συνδεθούν σε αυτό. Επίσης, διαχειρίζεται τον πίνακα δικτύωσης και δρομολογεί τα μηνύματα μεταξύ των RFD. Στη συνέχεια, έχουμε το Τερματικό (ZigBee End Device). Η συσκευή αυτή, είναι ο κόμβος του δικτύου που είναι συνδεδεμένος με αισθητήρες ή εκτελεί εργασίες ελέγχου εξωτερικά συνδεδεμένων συσκευών. Το Τερματικό μπορεί να είναι είτε FFD είτε RFD. Αυτό καθορίζεται από τη φύση των εργασιών που προορίζεται να εκτελεί. Αν, για παράδειγμα, ο αισθητήρας πρέπει να δειγματοληπτείται συνέχεια, επειδή ελέγχει κάποιο κρίσιμο μέγεθος, επιλέγεται FFD. Το τρίτο είδος συσκευής του δικτύου, είναι ο Δρομολογητής (ZigBee Router) και η παρουσία του είναι προαιρετική. Ο Δρομολογητής, είναι μία FFD συσκευή, η οποία επιτρέπει να συνδεθούν στο δίκτυο περισσότεροι κόμβοι. Έτσι, με τη χρήση Δρομολογητών, είναι δυνατό να επεκτείνουμε το μέγεθος και το εύρος του δικτύου, καθώς, συσκευές που βρίσκονται εκτός της εμβέλειας του Συντονιστή, μέσω Δρομολογητών μπορούν να συνδεθούν κανονικά.

## 2.6 Τοπολογίες Δικτύου ZigBee

### 2.6.1 Τοπολογία Αστέρα

Τοπολογία Αστέρα (Star Network) : Η τοπολογία αστέρα περιλαμβάνει ένα συντονιστή (Coordinator) και μια ή περισσότερες τελικές συσκευές (τερματικά). Στην τοπολογία αστέρα, όλες οι τελικές συσκευές επικοινωνούν μόνο με τον Συντονιστή. Αν κάποια τελική συσκευή χρειαστεί να μεταφέρει δεδομένα σε μια άλλη τελική συσκευή, στέλνει τα δεδομένα στον Συντονιστή. Αυτός στην συνέχεια, τα προωθεί στον τελικό αποδέκτη. Τα τερματικά, είναι φυσικά και ηλεκτρικά απομονωμένα μεταξύ τους, και ο μόνος τρόπος για να ανταλλάξουν πληροφορίες είναι μέσω του Συντονιστή. Ο Αστέρας, θεωρείται δίκτυο μονής αναπήδησης (single hop), καθώς υπάρχει μόνο ένα επιτρεπτό μονοπάτι ανάμεσα σε οποιοδήποτε τερματικό και το Συντονιστή. Ένα σημαντικό μειονέκτημα της τοπολογίας αυτής, είναι ότι όλοι οι κόμβοι πρέπει να είναι εντός της εμβέλειας του Συντονιστή.

### 2.6.2 Τοπολογία Δέντρου ή Συστάδας ( Cluster – Tree Topology )

Μια άλλη τοπολογία δικτύου είναι αυτή του δέντρου συμπλεγμάτων (Cluster Tree). Σε αυτή την τοπολογία, οι τελικές συσκευές μπορούν να συνδεθούν είτε στον Συντονιστή του δικτύου

είτε σε κάποιο Δρομολογητή. Οι Δρομολογητές, επιτελούν δύο λειτουργίες . Μια είναι η αύξηση του μέγιστου αριθμού των συσκευών που μπορούν να υπάρχουν στο δίκτυο. Η άλλη είναι η αύξηση της φυσικής εμβέλειας του δικτύου. Με την προσθήκη ενός Δρομολογητή, η τελική συσκευή δεν χρειάζεται να βρίσκεται εντός της εμβέλειας του Συντονιστή. Όλα τα μηνύματα σε αυτή την τοπολογία κινούνται με την 12ιεραρχία δέντρου. Το Δέντρο, είναι ουσιαστικά πολλοί Αστέρες, όπου οι κεντρικοί κόμβοι είναι συνδεδεμένοι μεταξύ τους. Όπως είναι εύκολα αντιληπτό, τα τερματικά δεν έχουν άμεση επικοινωνία μεταξύ τους, αλλά, όλα τα μηνύματα πρέπει να περάσουν από τουλάχιστον ένα Δρομολογητή ή το Συντονιστή. Η τοπολογία αυτή, θεωρείται πολλαπλών αναπηδήσεων (multi hop), καθώς υπάρχουν πολλά μονοπάτια επικοινωνίας ενός κόμβου με το Συντονιστή .

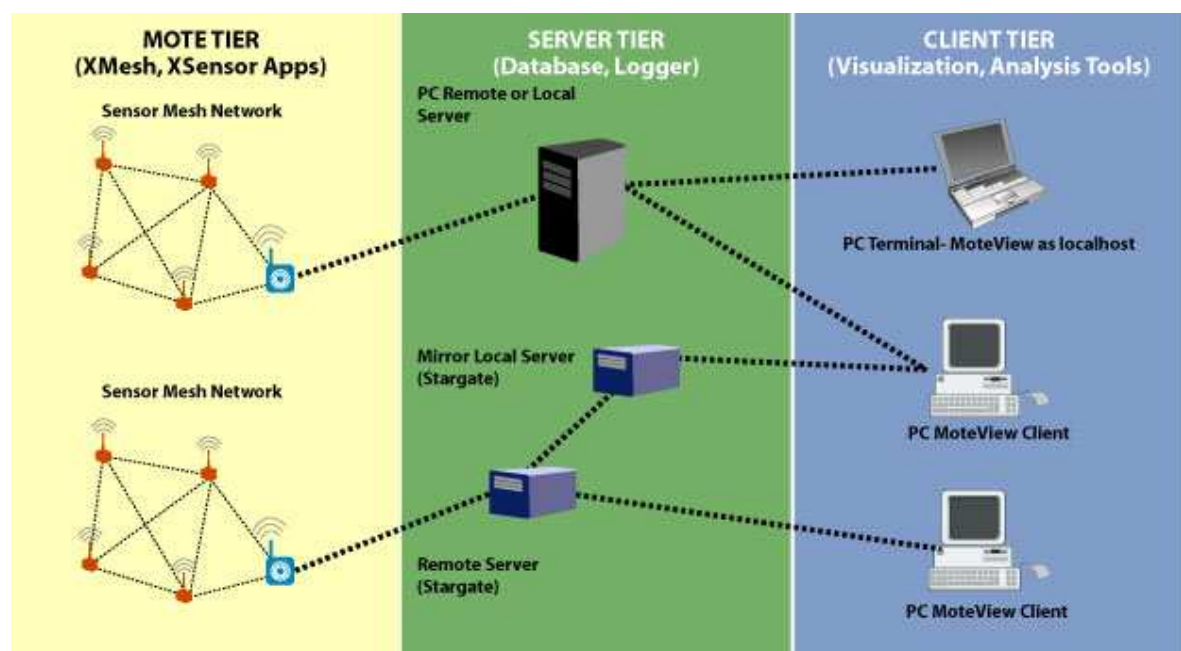
### **2.6.3 Τοπολογία Πλέγματος (Mesh Topology, Peer to Peer)**

Ένα δίκτυο πλέγματος (mesh), που αποτελεί και τη γενικότερη μορφή ενός δικτύου ZigBee, είναι παρόμοιο με αυτό του τύπου συμπλέγματος δέντρου, με τη διαφορά ότι τα FFD μπορούν να μεταβιβάσουν τα μηνύματα απευθείας σε άλλα FFD χωρίς να ακολουθηθεί η ιεραρχία δέντρου. Για τυχόν μηνύματα που πρέπει να μεταδοθούν εκτός εμβέλειας, η πληροφορία αναπηδά από κόμβο σε κόμβο μέχρι τον τελικό προορισμό της. Ωστόσο , τα μηνύματα προς τα RFD πρέπει και πάλι να περάσουν από τη γονική συσκευή. Τα πλεονεκτήματα αυτής της τοπολογίας είναι ότι μικραίνει η καθυστέρηση στη μεταβίβαση των μηνυμάτων και υπάρχει μεγαλύτερη αξιοπιστία.

## ΚΕΦ.3 Επισκόπηση του MoteView

### 3.1 Γενικά

Το MoteView είναι σχεδιασμένο να είναι μια διεπαφή ανάμεσα στον χρήστη και το ασύρματο δίκτυο αισθητήρων . Το Mote View παρέχει τα εργαλεία για την απλή ανάπτυξη και επίβλεψη του δικτύου αισθητήρων. Επίσης μέσω του Mote View γίνεται πολύ εύκολα η σύνδεση με τη βάση δεδομένων , η ανάλυση και τα γραφήματα των διαφόρων τύπων τιμών που διαβάζονται και στέλλονται από τα αισθητήρια .



Εικόνα 6 : Πλαίσιο τριών σημείων

Η εικόνα 6 απεικονίζει ένα πλαίσιο τριών σημείων που χρειάζονται για την ανάπτυξη ενός ασύρματου δικτύου αισθητήρων. Το πρώτο κομμάτι είναι το Mote Layer ή το mesh δίκτυο αισθητήρων . Τα Motes είναι προγραμματισμένα με το λογισμικό XMESH/TIYNYOS για να γίνουν ακριβής μονάδες μετρήσεων (πχ παρακολούθηση καιρικών φαινομένων , παρακολούθηση κίνησης στους δρόμους κ.α. ). Το δεύτερο κομμάτι η Server Tier παρέχει τις λειτουργίες της βάσης δεδομένων και την καταγραφή των δεδομένων . Δηλαδή στο Server Tier φτάνουν τα δεδομένα από τα αισθητήρια στο Base Station (πχ MIB510 , MIB520 κ.α.) και αποθηκεύονται στον Server . Στο τρίτο και τελευταίο κομμάτι η Client Tier υπάρχει το λογισμικό το οποίο παρέχει εργαλεία οπτικοποίησης , παρακολούθησης και ανάλυσης των δεδομένων που παίρνονται από τον Server . Αυτά τα εργαλεία παρέχει το Mote View τα οποία θα τα αναλύσουμε παρακάτω .

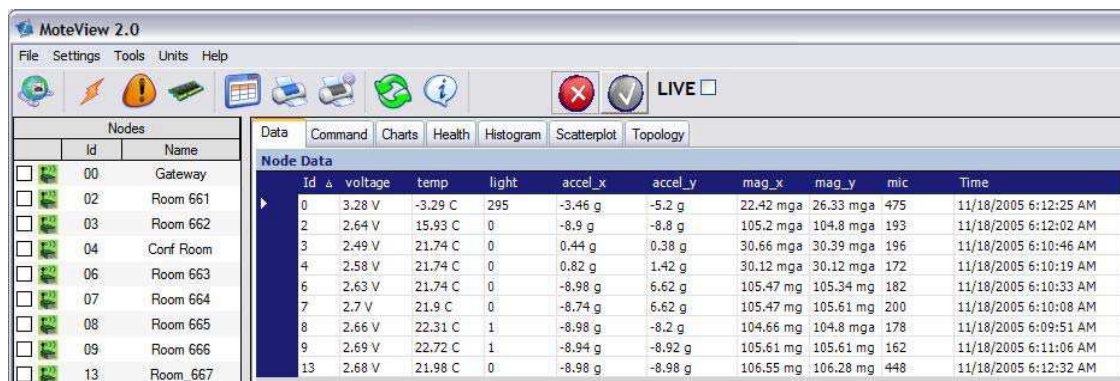


## 3.2 Καρτέλες Οπτικοποίησης

Στο Mote View έχουμε επτά καρτέλες οπτικοποίησης (Data , Commands , Charts , Histogram , Scatterplot and Topology) οι οποίες παρέχουν διαφορετικές μεθόδους για την οπτικοποίηση των δεδομένων .

### 3.2.1 Data

Η καρτέλα data μας δείχνει τη τελευταία μέτρηση των αισθητήρων του κάθε node αναλυτικά σε ένα πίνακα δέκα στηλών μαζί με το id του node . Οι μετρήσεις που βλέπουμε σε αυτό τον πίνακα είναι τροποποιημένες στις μονάδες μέτρησης τις οποίες θέλουμε (πχ βαθμούς κελσίου , μπαρ κ.τ.λ.). Το πίνακα αυτό έχουμε την δυνατότητα να τον αρχικοποιήσουμε όπως θέλουμε (πχ κατά τη θερμοκρασία) . Στην εικόνα 7 βλέπουμε ένα πραγματικό παράδειγμα αυτού του πίνακα .

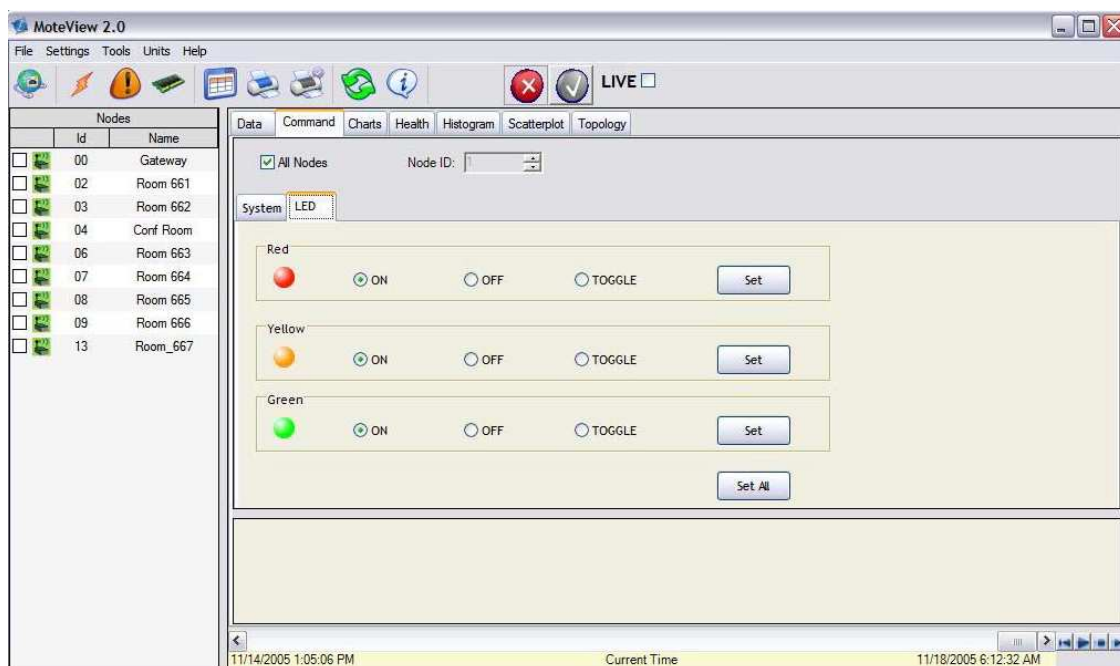
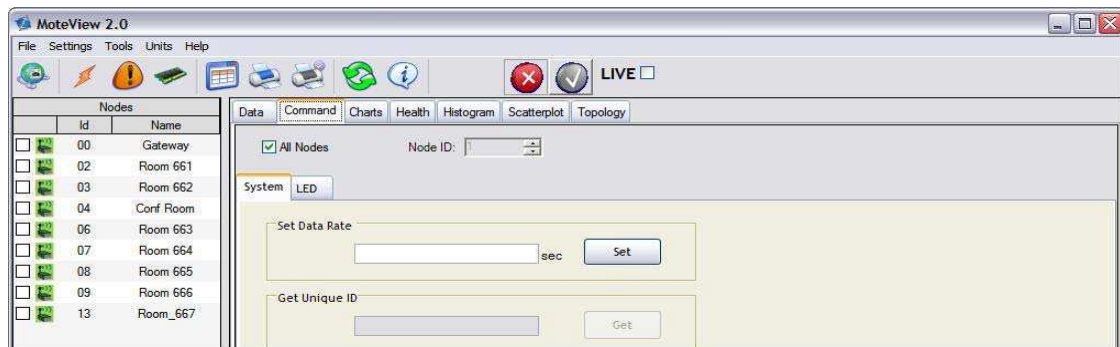


Id	voltage	temp	light	accel_x	accel_y	mag_x	mag_y	mic	Time
0	3.28 V	-3.29 C	295	-3.46 g	-5.2 g	22.42 mga	26.33 mga	475	11/18/2005 6:12:25 AM
2	2.64 V	15.93 C	0	-8.9 g	-8.8 g	105.2 mga	104.8 mga	193	11/18/2005 6:12:02 AM
3	2.49 V	21.74 C	0	0.44 g	0.38 g	30.66 mga	30.39 mga	196	11/18/2005 6:10:46 AM
4	2.58 V	21.74 C	0	0.82 g	1.42 g	30.12 mga	30.12 mga	172	11/18/2005 6:10:19 AM
6	2.63 V	21.74 C	0	-8.98 g	6.62 g	105.47 mg	105.34 mg	182	11/18/2005 6:10:33 AM
7	2.7 V	21.9 C	0	-8.74 g	6.62 g	105.47 mg	105.61 mg	200	11/18/2005 6:10:08 AM
8	2.66 V	22.31 C	1	-8.98 g	-8.2 g	104.66 mg	104.8 mga	178	11/18/2005 6:09:51 AM
9	2.69 V	22.72 C	1	-8.94 g	-8.92 g	105.61 mg	105.61 mg	162	11/18/2005 6:11:06 AM
13	2.68 V	21.98 C	0	-8.98 g	-8.98 g	106.55 mg	106.28 mg	448	11/18/2005 6:12:32 AM

Εικόνα 7 : Απεικόνιση Data

### 3.2.2 Command

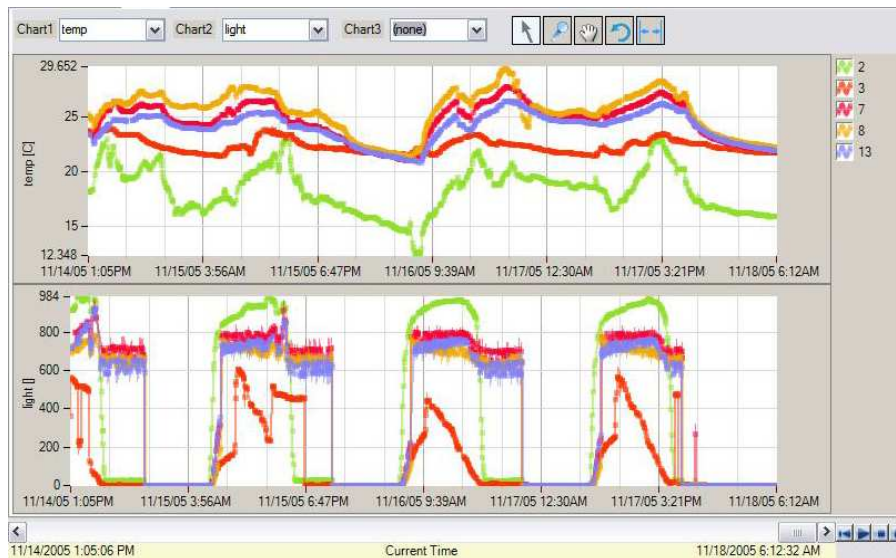
Η καρτέλα command δίνει τη δυνατότητα στο χρήστη να αλλάξει παραμέτρους των nodes μέσω του ασύρματου δικτύου . Μπορούμε να αλλάξουμε τη συχνότητα με την οποία στέλνονται τα δεδομένα των nodes στον server και την κατάσταση των leds . Στην εικόνα 8 βλέπουμε δύο στιγμιότυπα αυτής της καρτέλας .



Εικόνα 8 : Απεικόνιση Command

### 3.2.3 Chart

Μέσο της καρτέλας chart μπορούμε να δημιουργήσουμε γραφήματα των δεδομένων που λαμβάνουμε από τα nodes . Είναι πολύ χρήσιμη καρτέλα γιατί μπορούμε να κατασκευάσουμε συνδυαστικά γραφήματα έως και για εικοσιτέσσερα διαφορετικά nodes και για τρία διαφορετικά αισθητήρια σε ένα μόνο γράφημα έτσι μας είναι πολύ ευκολότερη η σύγκριση τιμών. Σε αυτά τα γραφήματα που γεννιούνται μέσο του mote view όπως θα δείτε και στην εικόνα 9 πάντα στον οριζόντιο άξονα είναι η ώρα και η ημερομηνία και στον κάθετο άξονα είναι ο τύπος των τιμών που θέλουμε να δημιουργήσουμε γράφημα.



Εικόνα 9 : Απεικόνιση Chart

### 3.2.4 Health

Μπορούμε να δημιουργήσουμε κώδικα έτσι ώστε το Xmesh χώρια από τα κανονικά πακέτα δεδομένων να μας στέλνει και Health πακέτα . Αυτά τα πακέτα μας ενημερώνουν για την κατάσταση των nodes αλλά και του δικτύου , μας δίνουν πληροφορίες όπως κατάσταση μπαταρίας , επί τις εκατό για καλύτερη κατανόηση την υγεία των πακέτων και άλλα.

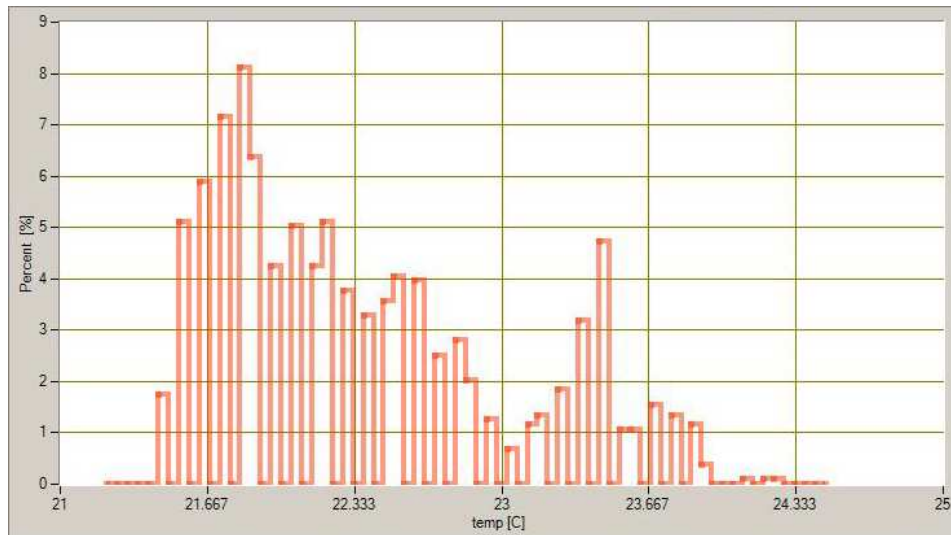
Node Data													
Id	health_pkt%	node_pkt%	forwarded	dropped	retries	battery	power_surr	board_id	quality_tx	quality_rx	path_cost	parent_rssi	Time
1	5.89 %	25.84 %	2.82 %	71.34 %	84.99 %	2.6 v	0 mAHr	131	0 %	0 %	1279	217	8/31/2006 13:00
2	7.93 %	42.93 %	50.66 %	6.4 %	2.8 %	2.5 v	0 mAHr	131	100 %	93.33 %	4	229	8/31/2006 13:00
3	2.84 %	15.78 %	70.14 %	14.08 %	66.1 %	2.8 v	0 mAHr	131	100 %	100 %	4	225	8/23/2006 21:00
4	0.47 %	97.54 %	0 %	2.46 %	6.83 %	2.7 v	0 mAHr	131	13.33 %	93.33 %	40	220	8/31/2006 13:00
5	5.5 %	47.53 %	48.73 %	3.75 %	1.67 %	2.7 v	0 mAHr	131	46.67 %	93.33 %	10	219	8/31/2006 13:00
6	3.46 %	14.32 %	60.41 %	25.27 %	4.28 %	2.7 v	0 mAHr	131	80 %	100 %	9	221	8/31/2006 13:00
7	5.63 %	48.58 %	31.12 %	20.3 %	3.98 %	2.6 v	0 mAHr	131	80 %	93.33 %	11	233	8/31/2006 13:00
8	10.14 %	87.51 %	8.69 %	3.8 %	5.09 %	2.9 v	0 mAHr	131	100 %	93.33 %	14	221	8/31/2006 13:00
9	14.9 %	59.09 %	37.25 %	3.66 %	4.09 %	2.9 v	0 mAHr	131	100 %	93.33 %	13	213	8/31/2006 13:00
10	8.07 %	69.56 %	24.66 %	5.78 %	7.58 %	2.5 v	0 mAHr	131	86.67 %	93.33 %	15	208	8/31/2006 13:00
11	9.18 %	79.05 %	16.34 %	4.61 %	4.58 %	2.7 v	0 mAHr	131	100 %	93.33 %	14	222	8/31/2006 13:00
12	3.04 %	26.13 %	65.98 %	7.88 %	6.75 %	2.6 v	0 mAHr	131	93.33 %	80 %	15	216	8/31/2006 13:00
13	2.47 %	35.3 %	44.36 %	20.34 %	8.93 %	2.6 v	0 mAHr	131	66.67 %	80 %	8	210	8/31/2006 13:00
14	2.23 %	29.57 %	39.73 %	30.7 %	6.92 %	2.6 v	0 mAHr	131	86.67 %	86.67 %	11	223	8/31/2006 13:00
15	5.03 %	44.02 %	48.8 %	7.19 %	12.94 %	2.7 v	0 mAHr	131	40 %	86.67 %	18	211	8/31/2006 13:00

Εικόνα 10 : Απεικόνιση Health

### 3.2.5 Histogram

Η καρτέλα Histogram συνοψίζει μια στατιστική κατανομή ενός μόνο αισθητήρα και μας τη παρουσιάζει σε ένα γράφημα μπάρας. Μέσο αυτού του γραφήματος μπορούμε να μελετήσουμε την εξάπλωση την ασυμμετρία και άλλα στοιχεία ενός μόνο αισθητήρα. Έχουμε την δυνατότητα να κάνουμε και σύγκριση του ίδιου αισθητήρα σε διαφορετικά Nodes (έως 24). Η απεικόνιση γίνεται ως εξής στον κάθετο άξονα έχουμε τα δεδομένα του αισθητηρίου σε

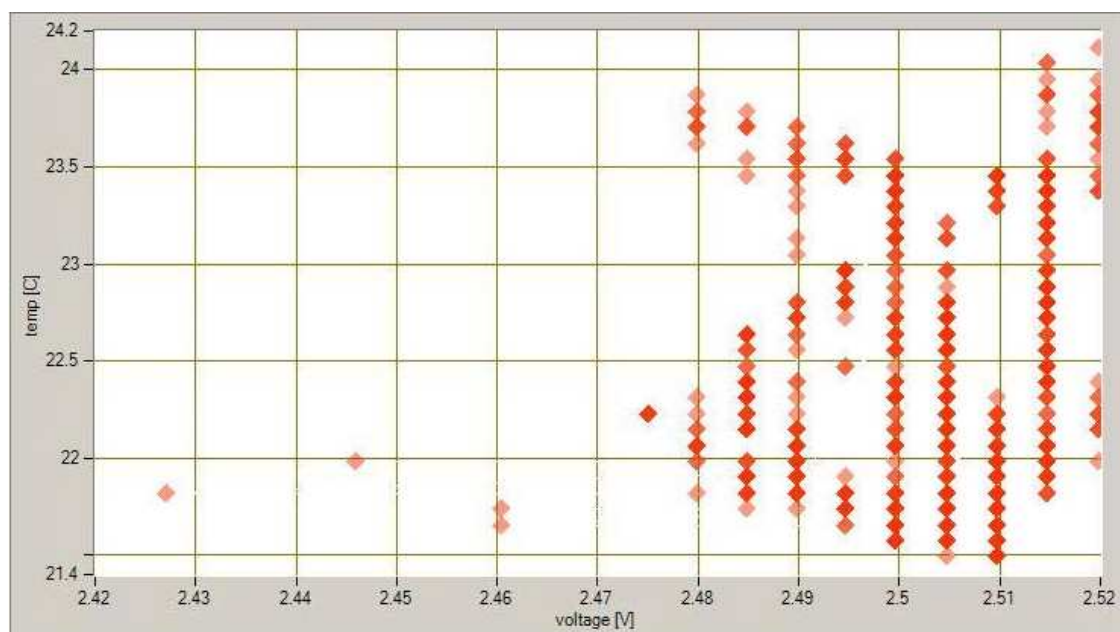
βιομηχανική μορφή και στον κάθετο άξονα παρουσιάζεται την κατανομή του αισθητηρίου σε ποσοστό επί τις εκατό όπως φαίνεται στην παρακάτω εικόνα 11 .



Εικόνα 11 : Απεικόνιση Histogram

### 3.2.6 Scatterplot

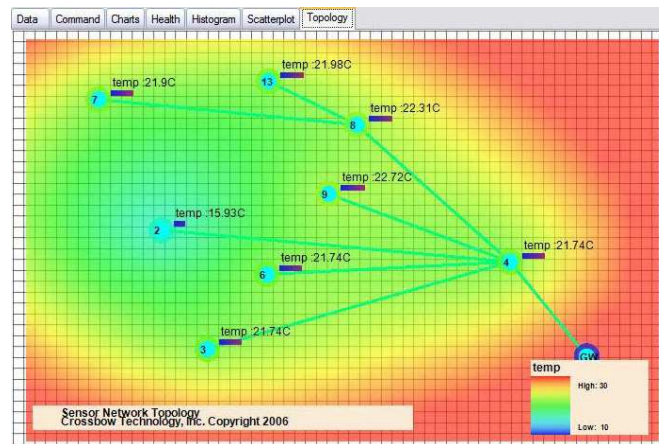
Στην καρτέλα scatterplot έχουμε την απεικόνιση ενός γραφήματος που αποτελείτε από τις τιμές δύο διαφορετικών αισθητήριων (πχ υγρασία , θερμοκρασία) που παίρνονται από διάφορα Nodes. Σε αυτό το γράφημα επειδή βλέπουμε τις τιμές δύο διαφορετικών αισθητήριων έχουμε τη δυνατότητα να συγκρίνουμε κατά πόσο η τιμή του ενός επηρεάζει την τιμή του άλλου , αυτός είναι και ο βασικός λόγος ύπαρξης αυτής της καρτέλας . Μία ιδέα για τη δομή αυτού του γραφήματος τη βλέπουμε στην παρακάτω εικόνα 12 .



Εικόνα 12 : Απεικόνιση Scatterplot

### 3.2.7 Topology

Η καρτέλα topology μας δείχνει ουσιαστικά την τοπολογία του ασύρματου δικτύου αισθητήριων που έχουμε δημιουργήσει . Στο κάθε node φαίνονται και κάποιες τιμές βέβαια μπορούμε να δούμε και όλες τις τιμές εάν κάνουμε “κλικ” πάνω στο node . Αρχικά είναι μια προσομοίωση της πραγματικής τοπολογίας του δικτύου αλλά μπορεί να γίνει και πραγματική σε περίπτωση που μπορούμε να κάνουμε scan για παράδειγμα την κάτοψη του κτηρίου που έχουμε αναπτύξει το δίκτυο . Η πιο σημαντική λειτουργία αυτής της καρτέλας είναι ότι μας δείχνει την πραγματική πάντα πορεία των πακέτων και έτσι έχουμε έναν χάρτη του Mesh δικτύου μας.



Εικόνα 13 : Απεικόνιση Topology

### 3.3 Connect to WSN

Το πιο βασικό χαρακτηριστικό του Mote view είναι το κουμπί Connect to WSN . Με το πάτημα αυτού του κουμπιού ανοίγει ένας οδηγός ο οποίος βήμα βήμα δημιουργεί το Ασύρματο δίκτυο αισθητήριων μαζί με όλες τις παραμέτρους του . Πατώντας τον οδηγό ανοίγει ένα παράθυρο με τέσσερις καρτέλες .

#### 1η καρτέλα Mode

Αυτή η καρτέλα επιτρέπει στον χρήστη να επιλέξει Operation mode και Acquisition Type. Έχει τις εξής επιλογές αν θα συνδεθεί σε κανονικό δίκτυο η σε κάποιο εικονικό , και μέσω Acquisition Type προσδιορίζουμε που βρίσκεται η βάση δεδομένων για την αποθήκευση των τιμών ως προς το base station mode που τραβάει δεδομένα από το δίκτυο .

#### 2η Καρτέλα

Επιτρέπει στο χρήστη να επιλέξει το μοντέλο του Gateway Mode και την θήρα επικοινωνίας . Το μοντέλο Gateway Mode μπορεί να είναι Ethernet , usb η Serial . Υπάρχει και η περίπτωση να είναι τύπου Xserve εάν θέλουμε να συνδεθούμε σε ένα δίκτυο αποκρησμένα .

### **3η καρτέλα .**

Σε αυτή την καρτέλα παίρνουμε τις ρυθμίσεις για τη βάση δεδομένων ( PostgreSQL ) που αποθηκεύουμε τα δεδομένα που τραβάμε από το δίκτυο . Αυτές οι ρυθμίσεις είναι οι εξής όνομα της βάσης , ip , αριθμός θήρας , username και password . Η PostgreSQL είναι μια σχεσιακή βάση δεδομένων ανοικτού κώδικα με πολλές δυνατότητες. Η ανάπτυξη της διαρκεί ήδη πάνω από δύο δεκαετίες και βασίζεται σε μια αποδεδειγμένα καλή αρχιτεκτονική η οποία έχει δημιουργήσει μια ισχυρή αντίληψη των χρηστών της γύρω από την αξιοπιστία, την ακεραιότητα δεδομένων και την ορθή λειτουργία. Η PostgreSQL τρέχει σε όλα τα βασικά λειτουργικά συστήματα, στα οποία περιλαμβάνονται το Linux, το UNIX (AIX, BSD, HP-UX, SGI, IRIX, MAC OS X, Solaris, Tru64) και τα Windows. Είναι συμβατή με ACID, και συμπεριλαμβάνει τους περισσότερους SQL92 και SQL99 τύπους δεδομένων συμπεριλαμβανομένων INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL και TIMESTAMP. Επίσης υποστηρίζει αποθήκευση μεγάλων δυαδικών αντικειμένων (binary), όπως εικόνες, ήχοι ή βίντεο. Διαθέτει επίσης περιβάλλοντα προγραμματισμού για τις γλώσσες προγραμματισμού C, C++, Java, Perl, Python, Ruby, Tcl, και υποστήριξη για την πλατφόρμα .NET και το πρότυπο ODBC, ενώ περιλαμβάνει και εξαιρετικό εγχειρίδιο χρήσης

### **4η Καρτέλα**

Τέλος σε αυτή την καρτέλα διαλέγουμε το sensor board το οποίο χρησιμοποιούμε έτσι ώστε το mote view να μπορεί να κάνει τα σωστά scaling και να μπορεί να αποθήκευση σωστά στη βάση τα δεδομένα .

Όπως θα δούμε στο επόμενο κεφάλαιο είναι αυτή η δυνατότητα που μας δίνει το mote view να δημιουργούμε ουσιαστικά ένα καινούργιο δίκτυο είναι πάρα πολύ σημαντική διότι εάν δεν υπήρχε αυτή η λειτουργία ακόμα και μια επανεκκίνηση του δικτύου μας θα ήταν μεγάλος μπελάς και θα χρειαζόντουσαν πολύ εξειδικευμένες γνώσεις προγραμματισμού από το χρήστη . Η ουσία της δημιουργίας αυτού του project είναι να μπορεί ο καθένας με βασικές γνώσεις χειρισμού υπολογιστή να είναι σε θέση να κάνει εύκολα επιβλέψει και τις βασικές λειτουργίες του δικτύου .

## Κεφ.4 Λειτουργικό Σύστημα

### 4.1 Γενικά

Το TinyOs είναι το πιο διαδεδομένο λειτουργικό σύστημα ανοικτού κώδικα για τα ασύρματα δίκτυα αισθητήρων. Πρώτη του έκδοση παρουσιάστηκε στο πανεπιστήμιο του Berkeley το 2000 πολύ γρήγορα όμως άρχισε να χρησιμοποιείται από όλη την ακαδημαϊκή κοινότητα . Για τον προγραμματισμό των δικών μας nodes χρησιμοποιήσαμε και εμείς αυτό το λειτουργικό . Παρακάτω θα αναφερθούμε σε πολύ σημαντικά κομμάτια της σχεδίασης του λειτουργικού ώστε να καταλάβουμε τη λειτουργία του .

Οι σχεδιαστές του TinyOs ήταν πολύ περιοσμένοι στο τρόπο σχεδίασης του λειτουργικού αυτού συστήματος διότι είχαν περιορισμούς όπως : ελάχιστη κατανάλωση ενέργειας , πολύ μικρό σε μέγεθος , αξιόπιστο , να υποστηρίζει επαναπρογραμματισμό κ.α. Με βάση ακόμα και αυτούς τους περιορισμούς οι σχεδιαστές κατάφεραν να φτιάξουν ένα λειτουργικό σύστημα το οποίο να καλύπτει από τις δυνατότητες των κόμβων και των ασύρματων δικτύων. Χαρακτηριστικό του TinyOS είναι η απουσία του multithreading και η χρησιμοποίηση του event-driven μοντέλου . Το event-driven εξασφαλίζει παράλληλη εκτέλεση διεργασιών χρησιμοποιώντας ελάχιστη ενεργεία και μνήμη . Το σύστημα εξασφαλίζει αυτή την ενέργεια και την μνήμη διότι βρίσκετε συνέχεια σε sleep mode έως ότου να βρεθεί μια διεργασία και να το ξυπνήσει η οποία είναι συνήθως από ένα event . Άρα σε αντίθεση με ένα polling σύστημα το οποίο κρατάει συνέχεια ξύπνιο τον επεξεργαστή σπαταλώντας ενέργεια και κάνοντας άσκοπες ερωτήσεις (“Υπάρχει κάτι να κάνω”), το event-driven σύστημα προκαλώντας ένα event ξυπνάει τον επεξεργαστή τον χρησιμοποιεί όσο θέλει και Μέτα πέφτει πάλι σε ύπνο .

Στο TinyOs δεν ορίζονται βασικές έννοιες όπως στα γενικού σκοπού λειτουργικά συστήματα (πυρήνας , εικονική μνήμη κα) , η κατανομή της μνήμης γίνεται στατικά κατά τη διαδικασία μεταγλώττισης αυτό έχει σαν πλεονέκτημα την αποφυγή συγκρούσεων (overheads) και η διαχείριση του hardware γίνεται άμεσα αφού δεν υπάρχει ο πυρήνας (kernel). Η δομή του TinyOs απαρτίζεται από ένα χρονοπρογραμματιστή (scheduler) και μια στοίβα από components.

### 4.2 Τα βασικά στοιχεία του TinyOs

#### 4.2.1 Components

Τα components μοιάζουν αρκετά με τις διεργασίες που εκτελούνται στα γενικού σκοπού ΛΣ. Κάθε component έχει ένα πλαίσιο μνήμης σταθερού μεγέθους το οποίο καθορίστηκε κατά τη

μεταγλώττιση έτσι ώστε να διατηρεί τη κατάσταση του και τα δεδομένα του . Τα components μπορούν να περιέχουν συναρτήσεις (command handlers , event handler , tasks).

Θέλοντας να κρύψουμε το hardware από το χρήστη /προγραμματιστή τον οποίο όταν θέλει να στείλει ένα πακέτο δεν τον ενδιαφέρει ποιο hardware το κάνει , ταυτόχρονα πρέπει και ο κώδικας να είναι ανεξάρτητος των Nodes . Έτσι το tinyOs μέσω της προγραμματιστικής αφάιρεσης δίνει τη λύση με την αφάιρεση σε τρία διακριτά στρώματα:

α. HPL-hardware presentation layer: Στη δομή του ΛΣ είναι το χαμηλότερο στρώμα και δουλειά του είναι να παρουσιάζει στο επόμενο στρώμα τις ικανότητες του hardware. Είναι εξαρτώμενο από το hardware και διαφέρει από πλατφόρμα σε πλατφόρμα . Το σημαντικό είναι ότι σε αυτό το στρώμα τα components έχουν μια συγκεκριμένη λειτουργία , δομή και χρησιμοποιούνται για συγκεκριμένο σκοπό .Εντολές για αρχικοποίηση , απενεργοποίηση και έναρξη του hardware. Εντολές <<get>> και <<set>> κατευθείαν στους registers . Ξεχωριστές εντολές για τις διαδικασίες που χρησιμοποιούνται πιο συχνά .Εντολές για ενεργοποίηση/απενεργοποίηση των interrupts. Ρουτίνες που θα καλούνται ανάλογα με κάποιο interrupt που προκύπτει.

β.HAL – hardware adaption layer: Είναι το ενδιάμεσο στρώμα και χρησιμοποιεί raw interfaces από το HPL δημιουργώντας χρήσιμες αφαιρέσεις οι οποίες είναι συγκεκριμένες για κάθε πλατφόρμα. Μέσο αυτών των αφαιρέσεων επιτυγχάνουμε την απαιτούμενη ενεργειακή απόδοση .

γ. HIL – hardware interface layer: Το τελευταίο στρώμα χρησιμοποιώντας platform-specific abstraction από το HAL τις μετατρέπει σε hardware independent abstraction. Έχουμε δύο ειδών HIL components:

1.Strong/real HILs : Σε αυτή την περίπτωση με τον ίδιο κώδικα αναμένουμε όλες οι πλατφόρμες να συμπεριφέροντε το ίδιο (πχ Timers , Leds κα).

2.Weak HILs : Χρησιμοποιώντας αφαιρέσεις από weak HILs μπορούμε να γράψουμε φορητές εφαρμογές αλλά η συμπεριφορά της πλατφόρμας ενδέχεται να διαφέρει.

Με την αφάιρεση σε αυτά τα τρία στρώματα το λειτουργικό γίνεται ευέλικτο και ανεξάρτητο οποιασδήποτε πλατφόρμας . Η ουσιαστική ευελιξία του tinyOs είναι ότι αν η εφαρμογή δε χρειάζεται βελτιώσεις σε κατανάλωση η οτιδήποτε άλλο μπορεί να χρησιμοποιήσει components από το ανώτερο επίπεδο. Ο προγραμματιστής βέβαια έχει πρόσβαση και στα κατώτερα components σε περίπτωση που θέλει να κάνει οποιαδήποτε αλλαγή.



#### 4.2.2 Commands ( εντολές )

Τα commands είναι συναρτήσεις που υλοποιούνται στα components και χρησιμοποιούνται για να διενεργήσουν αιτήσεις σε χαμηλότερου επιπέδου components. Τα commands εκτελούνται άμεσα και μπορούν να διακοπουν μόνο από ασύγχρονα events. Μέσα από τα commands μπορούν να κληθούν άλλα commands ή tasks αλλά δε μπορούν να σηματοδοτηθούν events.

#### 4.2.3 Events (γεγονότα)

Υπάρχουν δύο κατηγορίες events τα σύγχρονα και τα ασύγχρονα events. Τα σύγχρονα events συνήθως σηματοδοτούν την λήξη μιας λειτουργίας . Τα ασύγχρονα events προκύπτουν από hardware interrupts για παράδειγμα ένα event θα μπορούσε να είναι το τέλος της λειτουργίας boot του κόμβου . Κάτι πολύ σημαντικό που πρέπει να αναφέρουμε για τα events είναι ότι δε μπορούν να διακοπουν παρά μόνο από ανώτερης σημασίας ασύγχρονο event . Τα events και τα commands είναι τα εργαλεία που χρησιμοποιούν τα components για να επικοινωνήσουν μεταξύ τους . Η χρήση των commands και events υλοποιεί την split – phase operation που ακολουθεί το tinyOs . Για παράδειγμα εάν θέλουμε να διαβάσουμε την τιμή από έναν αισθητήρα θα πρέπει να γίνει σε δύο φάσεις . Πρώτον να καλεί το command που διαβάζει πχ call read(); έτσι ξεκινάει η διαδικασία του διαβάσματος και όταν διαβαστεί η τιμή πρέπει να φωνάξουμε το event “τέλος διαβάσματος ” πχ event read.ReadDone(); τότε μπορούμε να πάρουμε το αποτέλεσμα .

#### 4.2.4 Tasks (διεργασίες)

Τα events και τα commands είναι διεργασίες μικρές αλλά απόλυτα αναγκαίες . Τα υπόλοιπα υπολογιστικά κομμάτια μιας εφαρμογής που απαιτούν περισσότερο χρόνο εκτέλεσης αλλά είναι χαμηλότερης σημασίας υλοποιούνται με τα tasks . Τα tasks καλούνται από commands ή events και δεν εκτελούνται όσο υπάρχουν events και commands σε εκτέλεση από το λειτουργικό . Το σημαντικό είναι ότι ένα task δεν μπορεί να σταματήσει τη λειτουργία από ένα άλλο task αλλά ούτε από σύγχρονο event. Άρα ο σωστός τρόπος δημιουργίας ενός task από ένα σωστό προγραμματιστή του tinyOs είναι να τα κρατάει μικρά σε μέγεθος έτσι ώστε να μην έρθει στη δυσάρεστη θέση να χάνει πακέτα επειδή εκτελώ το λειτουργικό μεγάλο task.

#### 4.2.5 Χρονοπρογραμματιστής

Είναι υπεύθυνος για την ομαλή λειτουργία των components λειτουργεί με βάση μία στοίβα στην οποία καταχωρεί tasks τα οποία όταν καλούνται από το σύστημα παίρνονται από αυτή τη στοίβα με λογική FIFO , αυτό δεν ισχύει απόλυτα διότι σε περίπτωση που εμφανιστεί ένα ασύγχρονο event διακόπτεται η λειτουργία του task για να εκτελεστεί το event . Ο χρονοπρογραμματιστής όσο υπάρχουν task τροφοδοτεί με αυτά τον επεξεργαστή όταν αδειάσει η στοίβα του βάζει τον επεξεργαστή σε sleep mode.

### 4.3 nesC

Ο αρχικός προγραμματισμός του tinyOs έγινε σε C , αλλά χρειάστηκε μια καινούργια γλώσσα για να μπορέσει να υποστηρίξει την component – based λογική του και έτσι δημιουργήθηκε η nesC (network embedded system C).

Η nesC είναι component – based , event-driven γλώσσα προγραμματισμού . Η nesC σχεδιάστηκε ως προέκταση της γλώσσας C για την δημιουργία εφαρμογών , για την πλατφόρμα του tinyOs , που αποτελείται από components συνενωμένα μεταξύ τους , με μία διαδικασία που ονομάζεται wiring . Παρακάτω θα αναλύσουμε τα βασικά στοιχεία που απαρτίζουν ένα πρόγραμμα γραμμένο σε nesC.

#### 4.3.1 Interface

Τα interfaces είναι λειτουργίες που παρέχονται από κάποια components σε components . Κάθε component που παρέχει μια λειτουργία ονομάζεται provider ( παροχές ) και κάθε component που χρησιμοποιεί το interface ονομάζεται user ( χρήστης ) . Στα interfaces απλά δηλώνονται οι συναρτήσεις ( commands και events ). Η υλοποίηση των commands είναι υποχρέωση των providers που παρέχουν το συγκεκριμένο interface , ενώ η υλοποίηση των events είναι υποχρέωση του user component . Παρακάτω αναλύεται το read interface από τα interfaces του tinyOS .

```
interface Read<val_t>
{
    command error_t read();
}
```

- Η λέξη interface ορίζει την έναρξη του interface.
- Η λέξη read που ακολουθεί είναι το όνομα του interface.
- Μετά από το όνομα ακολουθούν οι

```
event void readDone(error_t  
result, val_t val);
```

παράμετροι ως εξής: <παράμετρος\_1 , παράμετρος\_2 , .....>. Αυτό που πρέπει να προσέξουμε στις παραμέτρους είναι ότι η παράμετρος που θέτει ο provider πρέπει να είναι η ίδια με αυτή που εισάγεται στον user. Πχ. Αν ο provider έχει Read<unit15\_r> ο user πρέπει να έχει Read<unit15\_r> αν βάλει κάτι διαφορετικό θα μας «χτυπήσει» σφάλμα κατά τη διαδικασία του debugging.

- Στη συνέχεια δηλώνουμε τα commands και τα events . Η λέξη «command» δηλώνει command η συνέχεια της εντολής command δηλώνει τον τύπο που επιστρέφει το συγκεκριμένο command στον κώδικα δίπλα είναι το error\_t ( τύπος σφάλματος του tinyOS ) , ακολουθεί το όνομα του command «read» το οποίο παρατηρούμε ότι δεν παίρνει παραμέτρους για αυτό το λόγο και είναι άδειο .
- Η λέξη event δηλώνει event το οποίο ακολουθεί ο τύπος που επιστρέφει , όπως μας είναι γνωστός από τη C « void » . Στη συνέχεια ακολουθεί το όνομα του event « readDone » και τέλος οι παράμετροι που δέχεται το event : το result και το val που είναι τύπου error\_t και val\_t αντίστοιχα.

#### 4.3.2 Module

Τα modules είναι το πρώτο από τα δύο components που υπάρχουν στο nesC . Όπως έχουμε αναφέρει και παραπάνω τα modules μπορούν να είναι providers η users ενός interface ή ακόμα και τα δύο μαζί . Το module αποτελείται από δύο μέρη , στο πρώτο μέρος δηλώνονται πια interfaces χρησιμοποιούνται και ποια παρέχονται και στο δεύτερο μέρος γίνεται η υλοποίηση

του component module . Στον παρακάτω πίνακα αναλύεται η δομή και οι εντολές που απαρτίζουν ένα module .

```
module myModule
{
    user interface Boot;
    provides interface StdControl;
}
```

Implementation

```
{
    event void Boot.booted()
{
    call StdControl.start();
    call StdControl.stop();
}
    command result_t
    StdControl.start()
{
    //do something to start
}
    command result_t
    StdControl.stop()
{
    //do something to stop
}
```

➤ Πάντα ξεκινάμε με τη λέξη module και στη συνέχεια το όνομα του « myModule » Μέσα στις αγκύλες δηλώνονται τα interfaces τα οποία θα παρέχει το module «provides» και θα χρησιμοποιεί « user ».

➤ Το συγκεκριμένο κομμάτι ξεκινάει με τη δεσμευμένη λέξη implementation , μέσα στις αγκύλες που ακολουθούν γράφεται το

βασικό πρόγραμμα μιας εφαρμογής. Σε αυτό το κομμάτι του κώδικα θα δούμε να υλοποιούνται τα events και τα commands των interfaces που δηλώθηκαν παραπάνω. Όπως παρατηρούμε και στον κώδικα αριστερά μας έχει χρησιμοποιηθεί το event Boot στο οποίο είναι uses (χρήστης) το module myModule μαζί με όλα τα events που διέπουν αυτό το interface. Στο interface StdControl το myModule είναι provider άρα πρέπει να υλοποιήσει όλα τα commands τα οποία δηλώνει .

```
    }  
}  
  
StdControl.nc  
  
interface StdControl  
{  
    command error_t start();  
    command error_t stop();  
}
```

### 4.3.3 Configuration

Το Configuration είναι το δεύτερο από τα δύο components της NesC . Κάθε TinyOS εφαρμογή έχει τουλάχιστον ένα βασικό configuration component το οποία πρέπει να συνδέσει το component της εφαρμογής μας με άλλα component του συστήματος . Παρατηρώντας τον παραπάνω κώδικα όταν δηλώσαμε τα interface που παρέχει και χρησιμοποιεί το myModule (uses interface Boot;) δημιουργείτε το ερώτημα που ξέρει το module που να κοιτάξει να βρει αυτό το interface. Αυτή τη δουλειά αναλαμβάνει να κάνει το configuration component το οποίο έχει παρόμοια σύνταξη με το module και χρησιμοποιεί μόνο τρεις τελεστές : -> , <- , = . Παρακάτω γράφεται ο κώδικας που του configuration module που αντιστοιχεί στο παραπάνω module.

```
configuration myModuleAppC  
{.....}
```

- Όπως και στο module ξεκινάμε με τη δεσμευμένη λέξη configuration και συνεχίζουμε το όνομα του συγκεκριμένου configuration . Ανάμεσα στις αγκύλες δεν υπάρχει τίποτα.

#### Implementation

```
{  
    components myModuleC, MainC;  
    user.interface -> provider.interface  
    myModuleC.Boot ->MainC.Boot;  
    MainC.StdControl -> myModuleC.Boot;  
}
```

- Και πάλι όπως και στο module ξεκινάει με τη λέξη implementation . Τώρα μέσα στις αγκύλες αρχικά καλούνται τα components τα οποία καλείτε να διάσυνδεσε το συγκεκριμένο configuration μεταξύ τους . Στη συνέχεια του κώδικα όπως βλέπουμε γίνεται το wiring χρησιμοποιώντας κάθε φορά τον κατάλληλο τελεστή δείχνοντας στον compiler ποιος είναι ο user και ποιος provider . Παρατηρούμε ότι το interface boot που χρησιμοποιείται στο myModule παρέχεται από την MainC.Boot. Αυτή ακριβώς τη δουλειά κάνει το Implementation .

#### 4.3.4 Generic Components

Τα components είναι μοναδικά και τα configuration που αναφέρονται σε ένα component αναφέρονται σε στο ίδιο κομμάτι κώδικα . Στις τελευταίες εκδόσεις του TinyOS (tinyos2.x ,) προστεθήκαν τα generic components . Στα generic components με την χρησιμοποίηση της λέξης new μπορούμε να δημιουργήσουμε καινούργια στιγμιότυπα του ίδιου component . Τα generic components έχουν την δυνατότητα να δέχονται παραμέτρους . Η δήλωση generic component γίνεται με τη χρησιμοποίηση της λέξης generic .

generic configuration TimerMilliC{ } ή generic module VirtualizeTimerC( )

Υπάρχουν δύο ειδών generic components τα generics modules τα οποία είναι κομμάτια επαναχρησιμοποιούμενου κώδικα και τα generics configurations είναι επαναχρησιμοποιήσιμο σύνολο σχέσεων που διαμορφώνουν μια υψηλού επιπέδου αφαίρεση.

#### 4.3.5 Tasks

Στο προηγούμενο κεφάλαιο έχει γίνει αναφορά και εξήγηση της έννοιας των tasks , σε αυτό το κεφάλαιο θα δούμε πως μέσω της nesC μπορούμε να χρησιμοποιήσουμε tasks .

```
task void taskName() ;
```

Ένα task δε δέχεται ποτέ παραμέτρους και επιστρέφει void. Στη συνέχεια το task χωρίζεται ως εξής :

```
task void taskName()
```

```
{//Εδώ γράφουμε τον κώδικα που καθορίζει τη λειτουργία του task για να τελειώσει τη λειτουργία //του το task ακολουθεί η παρακάτω εντολή.
```

```
    signal taskNameDone();
```

```
}
```

Ένα task χρησιμοποιεί τη λέξη post για να καλέσει ένα command η event . Η post τοποθετεί στη στοιβάδα του χρονοπρογραμματιστή την task η οποία θα εκτελεστεί την ώρα που πρέπει . Επιστρέφει success εάν το συγκεκριμένο task δεν υπάρχει στο task query και fail αν υπάρχει στη στοιβάδα και δεν έχει ακόμα εκτελεστεί .

#### 4.3.6 Async συναρτήσεις και Atomic Statements

Επειδή τα task είναι non – preemptive και αυτό θα μπορούσε να προκαλέσει πολλά προβλήματα τη λύση την φέρνει η nesC δίνοντας μας τη δυνατότητα να δημιουργήσουμε συναρτήσεις ασύγχρονες οι οποίες αν χρειαστεί σταματάνε τα task για να εκτελεστούνε . Η σύνταξη τους είναι απλή βάζουμε μπροστά από τη συνάρτηση τη λέξη async όπως στο παράδειγμα που ακολουθεί.

```
interface AsyncStdControl
```

```
{
```

```
    async command error_t
```

```
    start();
```

```
    async command error_t stop();
```

}

Τα atomic statements φτιάχτηκαν για να δίνουν στο χρήστη τη δυνατότητα προστασίας μεταβλητών που είναι προσβάσιμες από ασύγχρονες διεργασίες . Για παράδειγμα αν μια ασύγχρονη διαδικασία μετράει την τιμή και την αποθηκεύει στην μεταβλητή myvalue , στη συνέχεια καλεί μια σύγχρονη διαδικασία η οποία αναλαμβάνει να στείλει το αποτέλεσμα σε έναν άλλο κόμβο , αν ακριβώς πριν την αποστολή προκύπτει ασύγχρονη διαδικασία τότε η myvalue παίρνει άλλη τιμή και η σύγχρονη διαδικασία θα στείλει αυτή την τιμή . Βάζοντας μπροστά από την μεταβλητή τη λέξη atomic (είναι η λέξη που χρησιμοποιεί το atomic statement) αποφεύγουμε αυτό το πρόβλημα .

```
atomic myValue=getValue();
```

Η

```
atomic
```

```
{
```

```
    myValue1 = getValue1();
```

```
    myValue2 = getValue2();
```

```
}
```

## Κεφ.5 Java Application

### 5.1 Λίγα λόγια για την Java

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού η οποία σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems.Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας .Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows,Linux,Macintosh χωρίς να χρειαστεί να ξαναγίνει compiling ή να αλλάξει ο πηγαίος



κώδικας για κάθε λειτουργικό σύστημα .Η αιτία αυτής της δυνατότητας που μας δίνει η Java είναι η εικονική μηχανή της Java (JVM).Αφού γραφτεί ένα πρόγραμμα σε Java μεταγλωττίζεται μέσω του javac μεταγλωττιστή ο οποίος παράγει ένα αριθμό από αρχεία .class(byte ή bytecode).Στη συνέχεια το JVM θα αναλάβει να διαβάσει τα αρχεία .class και θα τα μετατρέψει σε κώδικα μηχανής που υποστηρίζεται από το λειτουργικό σύστημα και επεξεργαστή προεξοφλώντας ότι το πρόγραμμα δεν θα έχει καταστροφικά αποτελέσματα για τον υπολογιστή.

### 5.1.1 Ολοκληρωμένο περιβάλλον ανάπτυξης(IDE)

Για να γράψει κανείς Java το μόνο που χρειάζεται είναι είτε ένα notepad είτε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) το οποίο βοηθάει πολύ στην εύρεση σφαλμάτων .Συγκεκριμένα θα ασχοληθούμε με το Netbeans το οποίο προσφέρει ένα περιβάλλον όπου ο χρήστης έχει την δυνατότητα να δημιουργήσει οποιαδήποτε είδους Java Application.Το πρόγραμμα είναι free και open-source και έχει ένα πολύ ευρύ φάσμα χρηστών Java developers αλλά και Java Beginners . Πληροφορίες σχετικά με το κατέβασμα και την εγκατάστασή του μπορεί ο καθένας να τις βρει στο site της [ORACLE](#).

### 5.1.2 Εισαγωγικά για την εφαρμογή

Η Java περιέχει την βιβλιοθήκη *Swing* η οποία περιέχει όλα αυτά που θέλει να εισάγει σε ένα Application Programming Interface(API) είτε σε κάτι που θα τρέχει **τοπικά** όπως στη δικιά μας περίπτωση είτε θα τρέχει με την μορφή site στο **Internet** .Η *Swing* είναι το πρωταρχικό εργαλείο της Java για την δημιουργία γραφικής διεπαφής χρήστη(GUI) Java program.Το NETBEANS παρέχει ένα πολύ δυνατό εργαλείο τη Palette το οποίο περιέχει όλα τα είδη αντικειμένων που μπορούμε εμείς να τα ενσωματώσουμε στην GUI(Graphical User Interface) εφαρμογή μας . Συγκεκριμένα χρησιμοποιήσαμε τα εξής αντικείμενα για να χτίσουμε την εφαρμογή μας: MENUBAR, MENUITEM, JBUTTON, JPANEL, JFRAME κ.α..Το καθένα από αυτά παραγοντοποιήθηκε κατάλληλα έτσι ώστε το αποτέλεσμα να είναι σε μορφή μιας καθαρά φιλική ,κατανοητή και αλληλεπιδραστική προς το χρήστη.

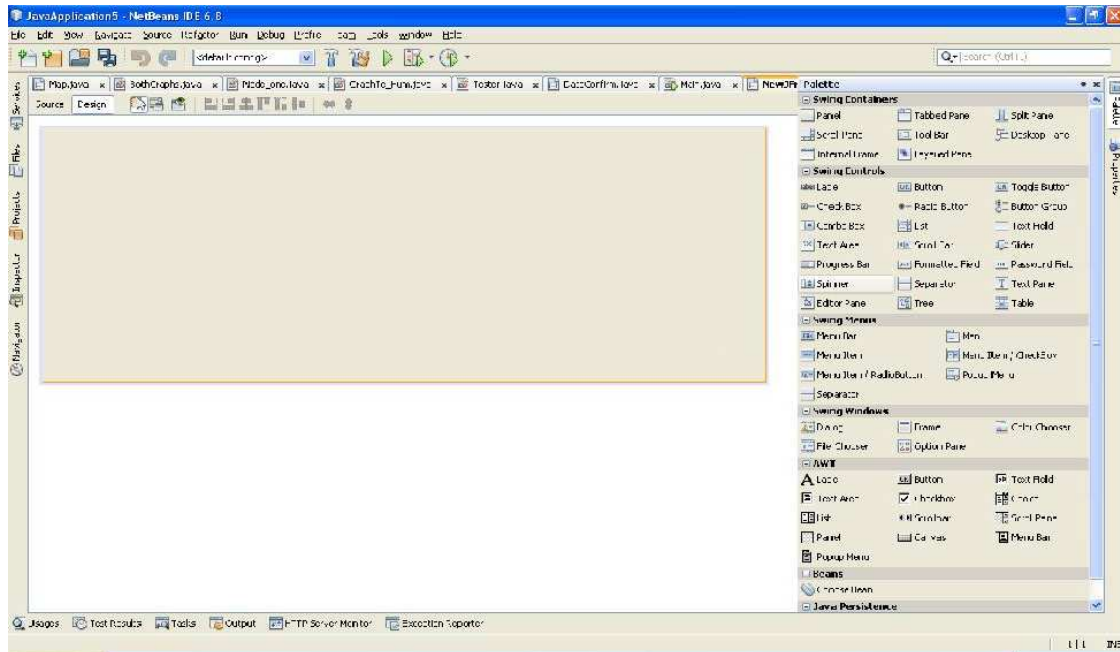
Στόχος της υλοποίησης του GUI είναι η άμεση απεικόνιση των τιμών των αισθητηρίων των Nodes στο παράθυρο του API.Τα Nodes έχουν σταθερή τοποθεσία και παίρνουν την μορφή *waypoint* <sup>1</sup> σε ένα χάρτη που απεικονίζεται στο API.Επίσης θα είναι δυνατό ο χρήστης θα μπορεί να βλέπει κάποια γραφήματα από τα οποία θα βγάζει συμπεράσματα για τον χώρο επιτήρησης των nodes.Δηλαδή θα έχουμε μια real-time παρακολούθηση των τιμών των αισθητηρίων και μέσα από ανάλογο χειρισμό του API θα βγάζουμε τα συμπεράσματα από μια αξιόπιστη πηγή πληροφοριών.

## 5.2. Πληροφορίες σχετικά με την εφαρμογή του GUI.

---

<sup>1</sup> Το waypoint είναι ένα αντικείμενο που αντιπροσωπεύει στο χάρτι ένα node

Αφού ακολουθήσουμε το Menu του Netbeans και φτιάξουμε ένα καινούργιο project του Desktop Application θα φτάσουμε σε ένα σημείο που μας δείχνει η παρακάτω εικόνα.



Εικόνα 14 : Απεικόνιση Netbeans

Όπως βλέπουμε στο κύριο μέρος του παράθυρου είναι το Design και στα δεξιά είναι η Palette όπου με drag and drop μπορούμε να χρησιμοποιήσουμε καθετί που μας προσφέρει η Swing. Δίπλα στο Design είναι το Source απ' όπου έχουμε την δυνατότητα να γράψει ο χρήστης κώδικα σχετικά με τα αντικείμενα που έχει ενσωματώσει στην εφαρμογή . Επίσης πάνω από την Palette είναι και τα Properties όπου μπορούμε αντί να γράψουμε κώδικα με απλά βήματα να κάνουμε ακριβώς το ίδιο που θα κάναμε με κώδικα αλλά προφανώς αρκετά συντομότερα .Εμείς θα χρησιμοποιήσουμε στην αρχή ένα JMenuItem ένα αντικείμενο που κληρονομείται από την κλάση του JFrame.Εκεί θα έχουμε τις περισσότερες 'ενέργειες' που θα ο χρήστης θα μπορεί να κάνει όπως να επιλέξει ποιανού node γραφήματα θέλει να δει ,να κάνει κάποιο back-up της βάσης ,να ορίσει τις ανώτερες τιμές (alerts) των αισθητηρίων του κάθε node που επιθυμεί και τέλος να τερματίσει την εφαρμογή.

### 5.2.1 Κάνοντας Back-up της βάσης από το Menu του API.

Έχοντας δημιουργήσει στην μπάρα μας το Menu **File** προσθέτουμε το **JMenuItem** με ονομασία **jMenuItemBackUP**.Αυτό γίνεται κάνοντας δεξί click πάνω στο **File** στη συνέχεια διαλέγουμε **add from palette** →**MenuItem**.Όπως αναφέραμε και πιο πάνω μπορούμε στο **Properties** να παραμετροποιήσουμε όπως επιθυμούμε το **MenuItem**.Έπειτα πρέπει να του προσθέσουμε μια ενέργεια(Action).Κάνοντας πάλι δεξί click αυτή την φορά στο **MenuItem**

διαλέγουμε **Events**→**Action**→**actionPerfomed** το IDE μας εμφανίζει στην οθόνη μας το σημείο όπου όταν θα προσθέσουμε κατάλληλο κώδικα θα έχουμε ολοκληρώσει επιτυχώς την ενέργεια που θέλουμε για το συγκεκριμένο **MenuItem**. Στην παρακάτω εικόνα παραθέτουμε τον κώδικα που θέλουμε να εκτελεστεί όταν πατήσουμε το **MenuItem(Back up DB)**. Στην ουσία μέσα σε ένα block try catch λέμε να δημιουργηθεί ένα αντικείμενο τύπου ColOldDB.

```
273 | Generated Code
527 |
528 | private void jMenuItemBackUPActionPerformed(java.awt.event.ActionEvent evt) {
529 |     // TODO add your handling code here:
530 |     try{
531 |         ConOldDB oldDb=new ConOldDB();
532 |     }catch (Exception ex) {
533 |         System.out.println("CAUSE IS from map: "+ex.getCause());
534 |         System.out.println(ex);
535 |     }
536 | }
537 |
```

Εικόνα 15 : Απεικόνιση Κώδικα

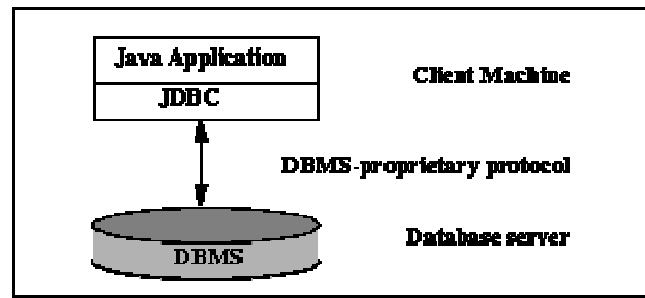
Μέσα στην κλάση ColOldDB έχουμε γράψει κώδικα όπου με την βοήθεια του JDBC διαβάζουμε τα δεδομένα του table όπου αποθηκεύονται οι τιμές των αισθητηρίων και τα αποθηκεύουμε σε ένα XML αρχείο.

### 5.2.1.1 Εισαγωγή JDBC

Το JDBC είναι ένα Java API το οποίο μπορεί να έχει πρόσβαση σε κάθε είδους δεδομένα που σώζονται σε πίνακες και ειδικά σε σχεσιακές βάσεις. Το JDBC βοηθάει σε Java applications οι οποίες διαχειρίζονται 3 προγραμματιστικές δραστηριότητες:

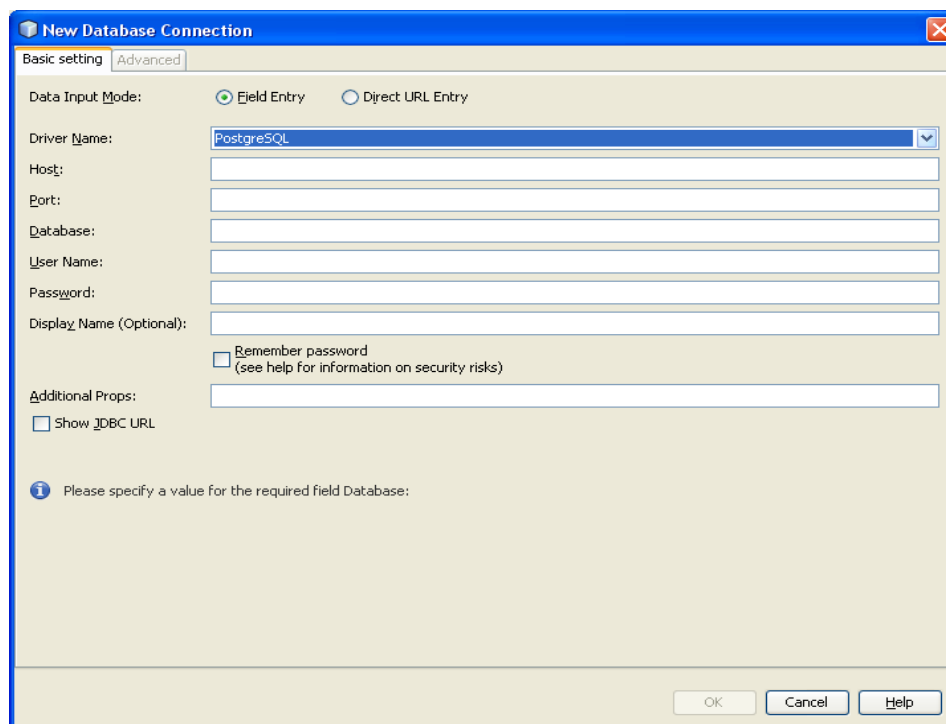
- Σύνδεση σε data source(database)
- Αποστολή queries και ενημερωτικών δηλώσεων σε κάποια βάση
- Ανάκτηση και διαχείριση δεδομένων που παράγονται από την εκτέλεση των queries.

Σε αυτό το κομμάτι μπορούμε να αναφέρουμε ότι το JDBC API “μίλαει” κατευθείαν στη βάση και ονομάζεται *two-tier Architecture*.



Εικόνα 16 : Στοιβία Java Application

Σε ένα two-tier model η java application “μιλαει” κατευθείαν στη πηγή δεδομένων .Αυτό απαιτεί ένα JDBC driver που μπορεί να επικοινωνήσει με τα δεδομένα που έχει πρόσβαση .Στον κόσμο της πληροφορικής υπάρχουν διάφορες βάσεις δεδομένων όπως η Oracle,MySQL ,PostGres κ.α. οι οποίες όλες έχουν κοινό background από τη γλώσσα SQL.Η κάθε μία της έχει και το δικό της JDBC Driver.Στην δική μας περίπτωση αφού τα δεδομένα μας σώζονται σε μια PostGres βάση θα χρειαστούμε τον αντίστοιχο Driver.Αυτό γίνεται άμεσα από το IDE πηγαίνοντας στο *Services* κάνοντας δεξί click στο **Databases**→**New Connection**.Πατώντας OK θα μας εμφανίσει το ακόλουθο window στο οποίο πρέπει να εισάγουμε κάποιες πολύ σημαντικές παραμέτρους που αφορούν την βάση που θέλουμε να συνδεθούμε.



Εικόνα 17 : DataBase Connection

- Στο Driver Name βάζουμε PostgreSQL για το προφανές λόγο.
- Host : localhost
- Port :5432
- Database : task
- User Name : tele
- Password : tiny

Στην συνέχεια γράφουμε κώδικα στην συνάρτηση όπου θέλουμε να εκτελέσουμε από το application ένα query.Στην παρακάτω εικόνα θα δούμε πως μπορούμε μέσα από κώδικα Java να εκτελέσουμε ένα query και αυτό βεβαίως δεν έχει μεγάλες αλλαγές αν είχαμε κάποιου άλλου είδους βάση.

```
39 public void dBconnection() throws SQLException, TransformerException(  
40     try(  
41         System.out.println("KSEKINAEI H SUNDESH ME TH VASH TASK");  
42         Class.forName("org.postgresql.Driver"); //o driver gia thn vash  
43         System.out.println("pire to driver:");  
44         myCon=DriverManager.getConnection("jdbc:postgresql://localhost:5432/task?user=tele&password=tiny"  
45         System.out.println("pire to con");  
46         myStatement=myCon.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_UPDATABLE);  
47         System.out.println("pire to state");  
48  
49         myRs=myStatement.executeQuery("Select * from public.mts400_results");//O 400 einai  
50         System.out.println("pire to query");
```

Εικόνα 18 : Απεικόνιση Κώδικα

### 5.2.1.2 Αποθήκευση σε XML αρχείο

Τι είναι η XML ;

XML<sup>2</sup> (Extensible Marked Language) είναι μια γλώσσα σήμανσης που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων ,δημιουργήθηκε από τον διεθνή οργανισμό προτύπων W3C(World Wide Web Consortium).Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα ,στην γενικότητα και τη χρησιμότητα στο Διαδίκτυο .Είναι μια μορφοποίηση δεδομένων κειμένου με υποστήριξη Unicode για όλες τις γλώσσες του κόσμου .Αν και η σχεδίαση της XML εστιάζει στα κείμενα χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων που προκύπτουν για παράδειγμα στις

<sup>2</sup> Πήγη από την [WIKI](#)

υπηρεσίες Ιστού .Με λίγα λόγια η XML σχεδιάστηκε για να μεταφέρει και να αποθηκεύει δεδομένα.

### **Βασική δομή της XML 1.0 <sup>3</sup>.**

**Χαρακτήρας Unicode :**Εξ ορισμού ένα κείμενο XML είναι μια ακολουθία χαρακτήρων .Σχεδόν κάθε χαρακτήρας Unicode μπορεί να εμφανίζεται σε ένα κείμενο XML.

**Επεξεργαστής και Εφαρμογή :**Είναι το λογισμικό που επεξεργάζεται ένα κείμενο XML.Είναι αναμενόμενο ότι ένας επεξεργαστής δουλεύει για μια εφαρμογή .Υπάρχουν μερικές πολύ συγκεκριμένες απαιτήσεις σχετικά με το τί μπορεί και τί δεν μπορεί να κάνει ένας επεξεργαστής XML,αλλά καμία ,όσον αφορά στην συμπεριφορά της εφαρμογής .Ο επεξεργαστής πολύ συχνά ονομάζεται XML Parser.

**Σήμανση και Περιεχόμενο :**Οι χαρακτήρες που απαρτίζουν ένα κείμενο XML αποτελούν είτε τη *σήμανσή* του είτε το *περιεχόμενό* του .Η σήμανση και το περιεχόμενο μπορούν να επισημανθούν και να διακριθούν ύστερα από την εφαρμογή κάποιων απλών συντακτικών κανόνων .Όλα τα αλφαριθμητικά που συνιστούν σήμανση είτε ξεκινούν με το χαρακτήρα ‘<’ και καταλήγουν στον χαρακτήρα ‘>’,είτε ξεκινούν με το χαρακτήρα ‘&’ και καταλήγουν με το χαρακτήρα ‘;’.Ακολουθίες χαρακτήρων που δεν συνιστούν σήμανση αποτελούν το περιεχόμενο ενός κειμένου XML.

**Ετικέτα:** Ένα στοιχείο σήμανσης που ξεκινά με το χαρακτήρα ‘<’ και καταλήγει με το χαρακτήρα ‘>’.Υπάρχουν τρία είδη ετικέτας : *ετικέτα αρχής* (<section>),*ετικέτες τέλους* (</section>) και *ετικέτες χωρίς περιεχόμενο*(<line-break/>).

**Στοιχείο:** Ένα λογικό απόσπασμα ενός κειμένου, που είτε ξεκινά με μία ετικέτα-αρχής και καταλήγει σε μία ετικέτα-τέλους, είτε αποτελείται μόνο από μία ετικέτα-χωρίς-περιεχόμενο. Οι χαρακτήρες που υπάρχουν, αν υπάρχουν, μεταξύ μιας ετικέτας-αρχής και μιας ετικέτας-τέλους, συνιστούν το *περιεχόμενο* του στοιχείου, το οποίο μπορεί να περιέχει σήμανση, συμπεριλαμβανομένων και άλλων στοιχείων, που ονομάζονται *στοιχεία-παιδιά*. Ένα παράδειγμα ενός στοιχείου είναι το <Greeting>Hello, world.</Greeting>. Ένα άλλο είναι το <line-break/>.

**Χαρακτηριστικό:** Ένα στοιχείο σήμανσης που αποτελείται από ένα ζευγάρι *όνομα/τιμή*, το οποίο υπάρχει μέσα σε μία ετικέτα-αρχής ή σε μία ετικέτα-χωρίς-περιεχόμενο. Στο παράδειγμα παρακάτω, το στοιχείο *img* έχει δύο χαρακτηριστικά, τα *src* και *alt*: . Ένα άλλο παράδειγμα θα ήταν το<step number="3">Connect A to B.</step>, όπου το όνομα του χαρακτηριστικού είναι "number" και η τιμή του είναι "3".

Τα κείμενα XML αποτελούνται εξ ολοκλήρου από χαρακτήρες Unicode. Εκτός από ένα μικρό αριθμό, ειδικά εξαιρούμενων χαρακτήρων ελέγχου, κάθε χαρακτήρας που ορίζεται στο Unicode, μπορεί να εμφανίζεται στο περιεχόμενο ενός κειμένου XML. Το σύνολο των

---

<sup>3</sup> Πηγή από την [WIKI](#)

χαρακτήρων που μπορούν να εμφανίζονται στη σήμανση, αν και κάπως περιορισμένο, παραμένει μεγάλο. Η XML παρέχει κάποιες διευκολύνσεις για την ταυτοποίηση της κωδικοποίησης των χαρακτήρων Unicode που απαρτίζουν ένα κείμενο και για την απεικόνιση χαρακτήρων που, για τον έναν ή τον άλλο λόγο, δεν μπορούν να χρησιμοποιηθούν ευθέως.

### Συγγραφή κώδικα για την δημιουργία του XML αρχείου.

Παρακάτω θα δούμε στην εικόνα τον κώδικα που χρησιμοποιήσαμε για να δημιουργήσουμε το XML file.

```
77 public void xmlcreation() throws SQLException, TransformerConfigurationException, TransformerException{
78     System.out.println("tha ftiaksei to xml arxeio");
79     System.out.println("Eisai stin\n"+myRs.getRow() + "grammi");
80     try{
81
82         dbfac = DocumentBuilderFactory.newInstance();// Edw ksekiname na ftiaxnoume tin domi tou xml
83         docBuilder = dbfac.newDocumentBuilder();
84         doc = docBuilder.newDocument();
85         root = doc.createElement("data");
86         doc.appendChild(root);
87         child_node1 = doc.createElement("node");
88         child_node1.setAttribute("id", Integer.toString(idnodes[0])); // NODE1
89         // child_node1.setAttribute("Latitude", Float.toString( (float) 35.311679));
90         // child_node1.setAttribute("Longitude", Float.toString( (float) 25.101265));
91         root.appendChild(child_node1);
92         child_node2=doc.createElement("node");
93         child_node2.setAttribute("id", Integer.toString(idnodes[1]));
94         // child_node2.setAttribute("Latitude", Float.toString( (float) 35.312414));// NODE2
95         // child_node2.setAttribute("Longitude", Float.toString( (float) 25.100986));
96         root.appendChild(child_node2);
97
98
99
```

Εικόνα 19 : Απεικόνιση Κώδικα

Στην παραπάνω εικόνα ορίζουμε την βασική δομή του XML αρχείου. Δηλαδή ορίζουμε ως root το data και δύο παιδιά του data το node με καθένα από τα παιδιά να έχουν διαφορετικό attribute(id) με την αντίστοιχη τιμή του κάθε node.

Συνεχίζουμε με τον κώδικα όπου παίρνουμε κάθε γραμμή του πίνακα και τα αποθηκεύουμε στο XML αρχείο. Συγκεκριμένα όσο ο δείκτης που μας δείχνει κάθε γραμμή του πίνακα δεν ισούται με την τιμή null (ΕΛΔ. Ο δείκτης ξεκινάει από την πρώτη γραμμή μέχρι την τελευταία του πίνακα) παίρνουμε τα δεδομένα κάθε στήλης που μας ενδιαφέρουν και τα αποθηκεύουμε στο XML αρχείο. Χρησιμοποιήσουμε σε κάθε γραμμή ως κριτήριο την στήλη που αναφέρεται στο id του node για να προσθέσουμε στο αντίστοιχο node(element) τα δεδομένα μας. Παρακάτω φαίνεται και στην πράξη πως είναι αυτό δυνατόν.

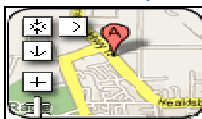
```
106 while(myRs.next()){
107     System.out.println("Eisai stin\n"+myRs.getRow() +"grammi");
108     try{
109         if (myRs.getInt(3)==idnodes[0]){
110             c_child=doc.createElement("values");
111             Timestamp stime=myRs.getTimestamp(1);
112             child_node1.appendChild(c_child);
113             // c_child.setAttribute("timestamp", stime.toString());
114
115             timestamp=doc.createElement("timestamp");
116             timestamp.appendChild(doc.createTextNode(stime.toString()));
117             c_child.appendChild(timestamp);
118             temp=doc.createElement("temperature");
119             temp.appendChild(doc.createTextNode(Float.toString(float) (-39.7 + 0.01 * myRs.getInt(7))));
120             c_child.appendChild(temp);
121             hum=doc.createElement("humidity");
122             hum.appendChild(doc.createTextNode(Float.toString(float) (-2.0468 + (0.0367 * myRs.getInt(6)) + (-1.2955 * 0.000001 * myR
123             c_child.appendChild(hum);
124             voltage=doc.createElement("voltage");
125             voltage.appendChild(doc.createTextNode(Float.toString(float) (1252.352 / myRs.getInt(5))));
126             c_child.appendChild(voltage);
127             lat=doc.createElement("Latitude");
128             lat.appendChild(doc.createTextNode(Float.toString(float) 35.311679));//
129             c_child.appendChild(lat); // NODE 1!!!!!!!!!!!!
130             lon=doc.createElement("Longitude"); //
```

Εικόνα 20 : Απεικόνιση Κώδικα

Τέλος αφού σαρώσουμε όλο τον πίνακα και πάρουμε όλα τα δεδομένα που χρειαζόμαστε με τις ακόλουθες εντολές αποθηκεύουμε το XML αρχείο μας στο σκληρό δίσκο ούτως ώστε να το χρησιμοποιήσουμε αργότερα.

```
TransformerFactory transformerFactory = TransformerFactory.newInstance();  
Transformer transformer = transformerFactory.newTransformer();  
DOMSource source = new DOMSource(doc);  
StreamResult result = new StreamResult(new File("D:\\back_up.xml"));  
transformer.transform(source, result);
```

### 5.2.2 Ενσωμάτωση χάρτη στην εφαρμογή μας.



Καταρχάς πρέπει να αναφέρουμε την σπουδαιότητα ενός χάρτη στην εφαρμογή μας απ' όπου θα εποπτεύουμε τα nodes μας .Όλο και περισσότερο πληθαίνουν οι ιστόχωροι (sites)τα οποία παρέχουν κάποιες υπηρεσίες στο σύνολο χρηστών που τους επισκέπτεται ο κάθε χρήστης βλέποντας ένα χάρτη του είναι του είναι πολύ πιο κατανοητό για το που βρίσκεται το σημείο ενδιαφέροντός του. Οπότε καταλαβαίνει και την απόσταση που έχει από αυτόν και μπορεί να πάρει ως κριτήριο επιλογής του την απόσταση που έχει από τον ίδιο ή ακόμα και να πάρει πληροφορίες για το πού βρίσκεται ακριβώς έτσι ώστε να μεταβεί ο ίδιος στην τοποθεσία .Ένα από τα σημαντικότερα site που παρέχουν geodata είναι και το Wikipedia.

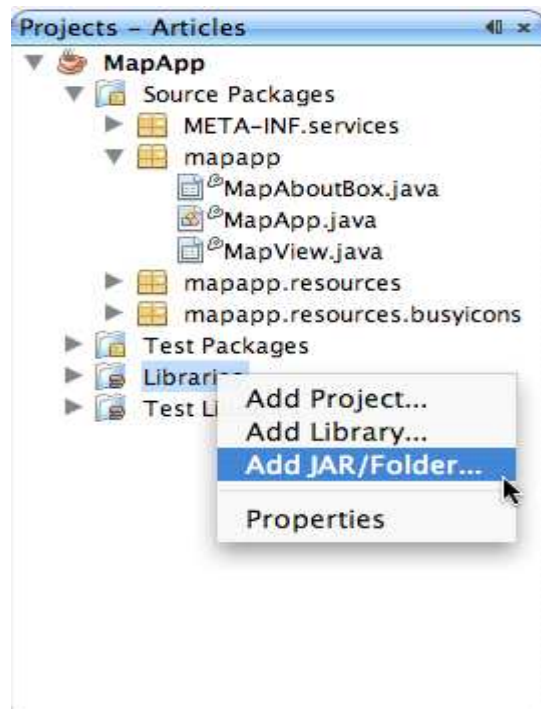


Από τις πρώτες ημέρες της πλοήγησης οι χάρτες έχουν σημαντικό ρόλο στο εμπόριο Ένας ναύτης χωρίς χάρτη θα ήταν χαμένος .Χωρίς χάρτη ένας ιδιοκτήτης γης δεν θα ήξερε τί θα του άνηκε .Χάρη στις καινοτομίες στις συσκευές, στα δίκτυα, στους δορυφόρους η τεχνολογία χαρτογράφησης είναι έτοιμη να παράγει νέες ιδέες και εφαρμογές .Η [Google](#) με ξεκίνησε την έκρηξη της χαρτογράφησης με την έκδοση [Google Maps](#) και την απόκτηση της εταιρείας Keyhole(Google Earth) της οποίας στοιχείο της είναι η 3D χαρτογράφηση.

Παρόλο τους πολλούς τρόπους με τους οποίους οι άνθρωποι παράγουν και χρησιμοποιούν γεωγραφικά δεδομένα υπάρχει ένα κοινός component: map viewer.Όπως η Google έχει το δικό της έτσι και Swing έχει το δικό της component χαρτογράφησης το: *open source* JXMapView.Το JXMapView δίνει αρκετές ικανότητες χρησιμοποιώντας το σε Swing based Java applications όπως δηλαδή και η δικιά μας εφαρμογή .Το JXMapView είναι ένας ανοιχτού κώδικα Swing component που σχεδιάστηκε από τους java developers στο [SwingLabs](#).Χάρη στο JXMapView είμαστε σε θέση να χρησιμοποιούμε τεχνολογίες χαρτογράφησης στις java desktop applications.Στον πυρήνα του είναι ένα ειδικό JPanel που ξέρει πώς να φορτώνει «πλακάκια χάρτη» από ένα διακομιστή εικόνων(image server).Όλες οι πληροφορίες σχετικά με το πώς μετατρέπονται οι συντεταγμένες σε pixels,cache tiles και βελονιάζονται όλα αυτά στην οθόνη του χρήστη με σαφήνεια προς αυτόν είναι κρυμμένες στο JXMapView API.Το μόνο που χρειάζεται να κάνει ο developer που θέλει να βάλει στην εφαρμογή του ένα χάρτη είναι να το προσθέσει όπως κάθε άλλο JPanel.

### **Βήμα 1°.**

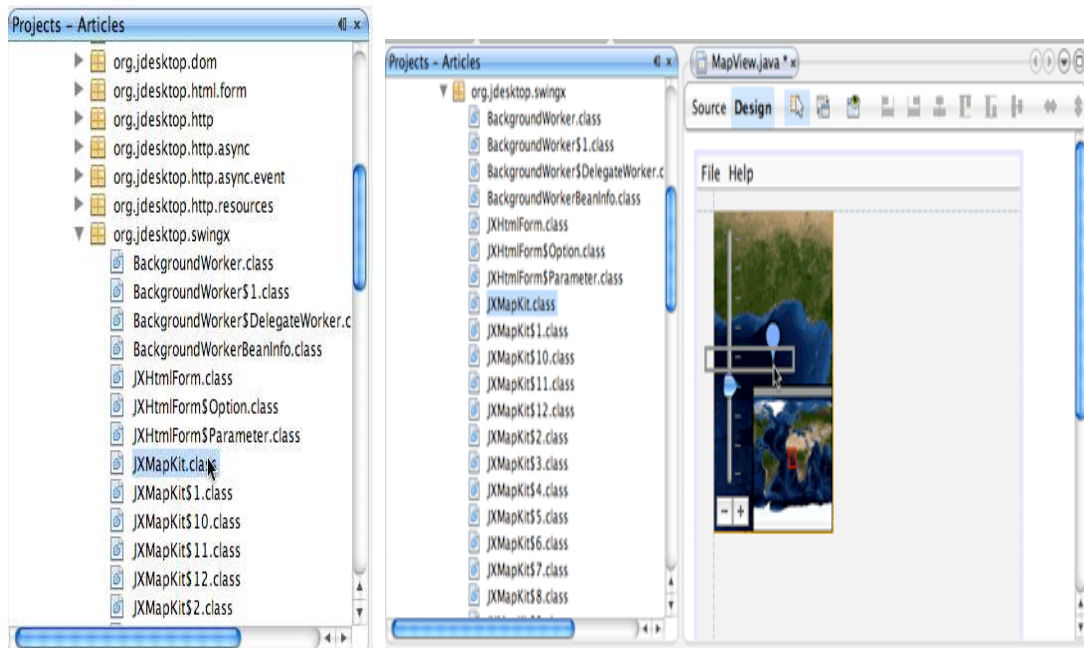
Αφού έχουμε δημιουργήσει ένα project στο IDE και έχουμε φτιάξει μια form όπως την εικόνα 1 πρέπει να κατεβάσουμε την τελευταία έκδοση της βιβλιοθήκης swingx-ws από το [site](#) και το κάνουμε unzip μέσα στο φάκελο lib του project ή με την βοήθεια των αρχείων .jars κάνοντας δεξί click στο Libraries Node στο Project inspector και να διαλέξουμε Add Jar/Folder (Εικόνα 8).Πρέπει επίσης να προσθέσουμε τα εξής .jars αρχεία swingx-bean.jar, swingx.jar, and swing-worker.jar μέσα στο directory του project : [//lib/swingx-ws-bin/lib/cobundle](#) .



Εικόνα 21 : Πρόσθεση Jar αρχείων

## Βήμα 2 °.

Αφού προσθέσαμε τα .jars αρχεία μπορούμε να εισάγουμε *JXMapKit* στην φόρμα μας .Με drag and drop προσθέτουμε το *JXMapKit* βρίσκοντας την κλάση *JXMapKit.class* σύμφωνα με την εικόνα 21 και οδηγούμαστε στο αποτέλεσμα που μας δείχνει η εικόνα 22.



Εικόνα 22 : Πρόσθεση Jar αρχείων

Στην εφαρμογή μας δεν χρησιμοποιήσαμε το JXMapView αλλά την kit έκδοση επειδή περιέχει zoom button, zoom slider και mini view. Η μόνη διαφορά τους είναι ότι το kit περιέχει τα χαρακτηριστικά που χρησιμοποιούνται ευρέως. Στην συνέχεια θα παραμετροποιήσουμε το χάρτη ούτως ώστε να πάρουμε το επιθυμητό αποτέλεσμα.

### **Προσθέτοντας Waypoints.**

Για να κάνουμε τον χάρτη χρήσιμο πρέπει να ζωγραφίσουμε σημεία σε συγκεκριμένες τοποθεσίες. Στον κόσμο της χαρτογράφησης συντεταγμένες που αντιπροσωπεύουν φυσικές θέσεις ονομάζονται waypoints. Waypoints παρουσιάζονται στην κλάση Waypoint. Το JXMapView έχει ένα ειδικό «ζωγράφο» (Painter) ο οποίος μπορεί να ζωγραφίσει σημεία στο χάρτη. Painters είναι κλάση που κάνει implement στο interface Painter και μπορεί να καθοριστεί (set-αριστεί) σε ένα component της Swingx. Το JXMapView μπορεί να προσθέσει painters με την μέθοδο setOverPainter(). Παρακάτω θα δούμε σχετικό κώδικα με τον οποίο θα προσθέσουμε στον χάρτη μας δυο waypoints που αντιστοιχούν στα δυο μας nodes.

```
56 //DIMIOURGIA WAYPOINTS
57 Set<Waypoint> waypoints = new HashSet<Waypoint>();
58 waypoints.add(new Waypoint(35.31168,25.101265));//suntetagmenes tou node1
59 waypoints.add(new Waypoint(35.312412,25.100986));//suntetagmenes tou node2
60 //crate a WaypointPainter to draw the points
61 final WaypointPainter painter = new WaypointPainter();
62 painter.setWaypoints(waypoints);
63 //gia to label tou node1
64 jXMapKit1.getMainMap().setOverlayPainter(painter);
```

Εικόνα 23 : Απεικόνιση Κώδικα

### Αλλάζοντας Image server.

Από default το JXMapKit χρησιμοποιεί ένα αντίγραφο του NASA’s Blue Marble satellite images<sup>4</sup> που βρίσκεται σε ένα server του SwingLabs<sup>5</sup>. Τα δεδομένα του Blue Marble έχουν ανάλυση 8-km. Άλλη επιλογή είναι ο OpenStreetMap.org tile(πλακάκι) server ο οποίος προσφέρει ένα διανυσματικό χάρτη του κόσμου ο οποίος δημιουργήθηκε από εθελοντές που κατέχουν δέκτες GPS. Το JXMapKit είναι προδιαμορφωμένα και με τους δύο servers οπότε μπορούμε να κάνουμε την επιλογή μας ή να χρησιμοποιήσουμε το δικό μας map server το οποίο είναι πιο περίπλοκο. Παρακάτω στην εικόνα 24 θα δούμε τον κώδικα που γράψαμε για την επιλογή του server, σε ποιά περιοχή να μας δείχνει στο χάρτη καθώς και πόσο αρχικό zoom θα έχουμε, την επιλογή να μας εμφανίζεται στο κάτω μέρος του χάρτη μια μικρογραφία του και τέλος την επιλογή να μας εμφανίζονται στο χάρτη οι διευθύνσεις των οδών.

```
49 //Edw tha grasw kwdika gia ton prosdiorismo tou xarti(suntetagn
50 jXMapKit1.setDefaultProvider(DefaultProviders.OpenStreetMaps);
51 jXMapKit1.setAddressLocationShown(true);//eksafanish arxikou parser
52 jXMapKit1.setCenterPosition(new GeoPosition(35.31168,25.101265));// jX
53 jXMapKit1.setZoom(1);
54 jXMapKit1.setMiniMapVisible(true);
```

Εικόνα 24 : Απεικόνιση Κώδικα

Για περισσότερες πληροφορίες σχετικά με το τί μπορούμε να κάνουμε με το JXMapViewer βρίσκονται στο site [java.net](http://java.net).

### 5.2.3 Δημιουργία Γραφημάτων

Με τα γραφήματα είναι η πιο άνετη για να παρουσιάσει κανείς πληροφορίες με πολύ απλό τρόπο. Χρησιμοποιώντας τα γραφήματα βοηθάει αυτούς που τα βλέπουν να καταλάβουν και να ερμηνεύσουν την πληροφορία πιο εύκολα και άνετα χωρίς κάποια κουραστική διαδικασία.

<sup>4</sup> Η NASA παίρνει photo την Γη από το διάστημα και τις αποθηκεύει στους servers της.

<sup>5</sup> Είναι ένα ανοιχτού κώδικα project προτείνοντας επεκτάσεις της Java Swing GUI toolkit.

Στην δική μας desktop application θα χρησιμοποιήσουμε την ανοιχτού κώδικα βιβλιοθήκη JFreeChart η οποία χρησιμοποιείται περίπου από 40000-50000 java developers.Οι developers μπορούν να προσθέσουν επαγγελματική ποιότητα στις Swing-based applications.

#### **Εισαγωγή βιβλιοθήκης μέσα στο project μας.**

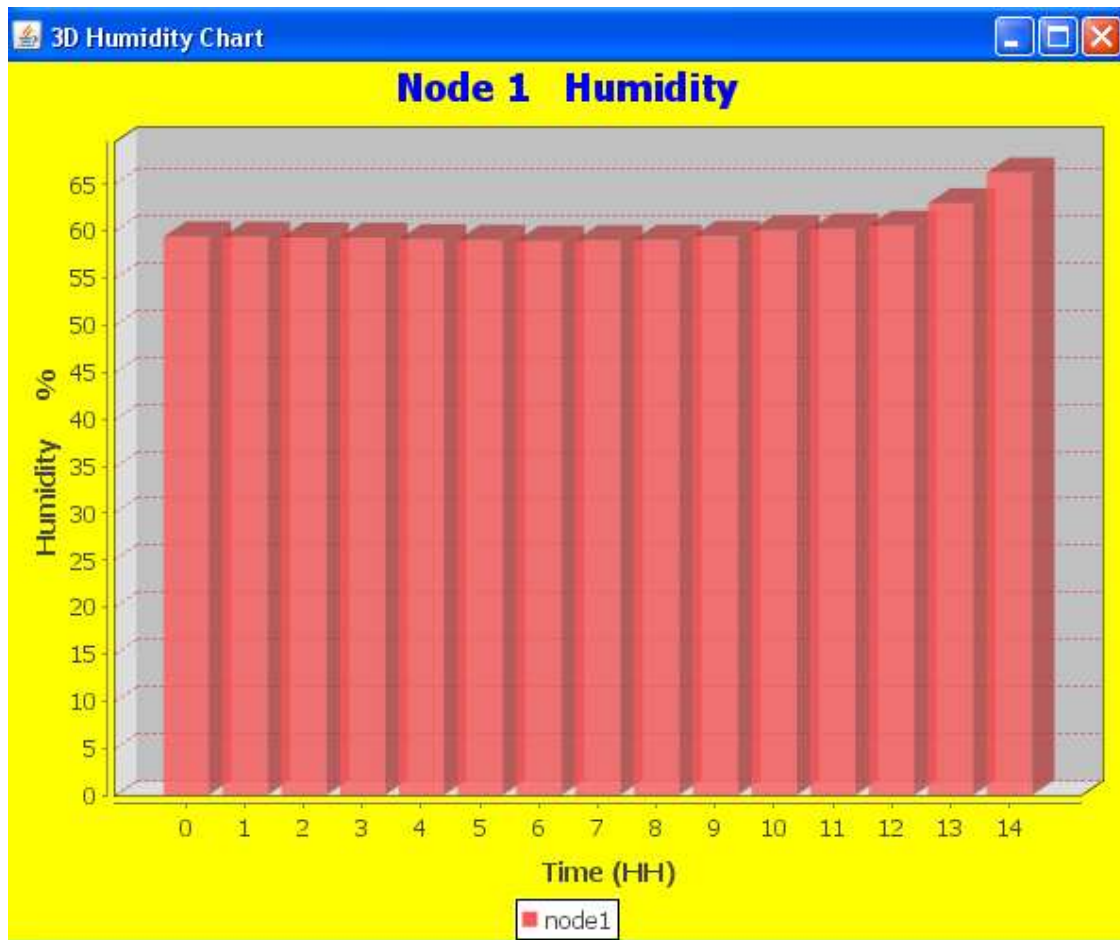
Σαν πρώτο βήμα πρέπει να εισάγουμε ένα αρχείο .jar που να έχει την βιβλιοθήκη jFreeChart αλλά απαιτείται να κατεβάσουμε και την βιβλιοθήκη JCommon για να υλοποιήσουμε τα γραφήματα που επιθυμούμε.Και τις δύο βιβλιοθήκες τις κατεβάζουμε από το site [αυτό](#).Οι ενέργειες που κάνουμε για τις προσθέσουμε στο project μας είναι οι ίδιες που κάναμε με την πρόσθεση της βιβλιοθήκης Swinx-ws.

#### **Επιλογή δεδομένων που θέλει ο χρήστης να του παρουσιαστούν στο γράφημα.**

Η εφαρμογή για να είναι ικανοποιητική και να εξυπηρετεί τους σκοπούς της πρέπει να προσφέρει την δυνατότητα στο χρήστη το τί θέλει να δει.Κυρίως στο κομμάτι των γραφημάτων πρέπει να είναι πολύ ευέλικτη με εύκολο χειρισμό.Οπότε είναι αναγκαίο όταν επιλέξει ο χρήστης από την μπάρα του menu να δει κάποιο γράφημα να εμφανίζεται ένα παράθυρο όπου θα παραμετροποιεί τις επιλογές του.

#### **Κλάση CurrentXML .**

Μία από τις δυο επιλογές του χρήστη ως προς τα διαγράμματα που θέλει να δει θα είναι η επιλογή να δει τα διαγράμματα των nodes (θερμοκρασία,τάση,υγρασία) τα οποία θα περιέχουν τιμές από τις 00:00 μέχρι και την ώρα που αιτήθηκε από την εφαρμογή να δει τα διαγράμματα.Για παράδειγμα αν ο χρήστης <<ζητήσει>> από την εφαρμογή στις 17:00 να δει το γράφημα της θερμοκρασίας για οποιαδήποτε node τότε θα εμφανιστούν από τις 00:00 μέχρι και τις 17:00 και στο διάγραμμα θα φαίνονται 17 μπάρες με την κάθε μπάρα να έχει τον μέσο όρο των τιμών την αντίστοιχης ώρας.Κάτι αντίστοιχο δείχνει και η παρακάτω εικόνα.



Εικόνα 25 : Γράφημα

Η κλάση *CurrentXML* αυτό που κάνει είναι να δημιουργεί ένα xml file και να του προσθέτει elements κάθε φορά που γίνεται αποθήκευση στην βάση. Η δομή της είναι η ίδια με την κλάση που κάνει Back-up την βάση μας με την μόνη διαφορά ότι έχει elements μιας μέρας μόνο και αυτά ανανεώνονται με το πέρασμά της.

### Κλάση *Node\_two*

Εδώ γίνεται η χρήση του xml file που δημιουργεί η παραπάνω κλάση. Η κλάση αυτή περιέχει 3 συναρτήσεις δηλ όσες είναι και οι τιμές που παίρνουμε από τα nodes. Η κάθε συνάρτηση όταν καλείται δημιουργεί και το αντίστοιχο γράφημα όπου αναφέρεται. Παρακάτω παραθέτουμε ένα κομμάτι του κώδικά μας ,το πιο σημαντικό ,όπου περιγράφεται η συνάρτηση που δημιουργεί το γράφημα της τάσης του node.

```
dataset = new DefaultCategoryDataset();
```

```
fstNmElmnLst1=fstElmnt.getElementsByTagName("voltage");
```

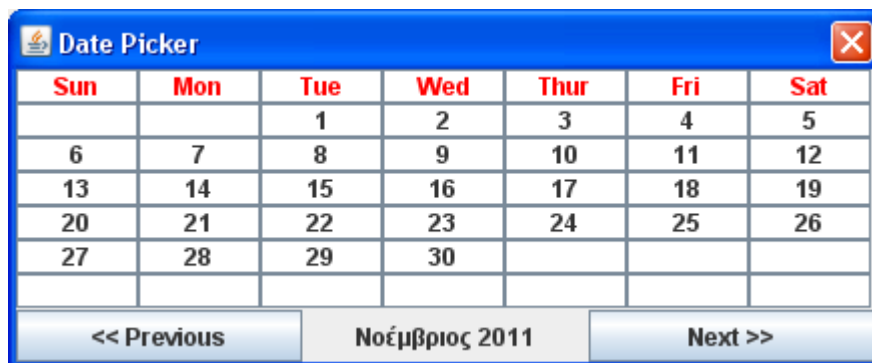
```
dataset.setValue(aDouble,"node"+(y+1),  
Integer.toString(cal.get(Calendar.HOUR_OF_DAY)));
```

```
JFreeChart chart1 = ChartFactory.createBarChart3D ("Node "+(y+1) + " Voltage","Time  
(HH)", "Voltage Volt", dataset, PlotOrientation.VERTICAL, true,true, false);
```

Είναι σαφές όμως ότι δεν θα είναι αρκετό η εφαρμογή μας να μας παρέχει διαγράμματα μόνο για την μέρα που θα χρησιμοποιήσουμε αυτήν επιλογή.Για αυτό η δεύτερη επιλογή του GUI ως προς τα διαγράμματα είναι να δούμε όποιο γράφημα επιθυμούμε οποιαδήποτε μέρας.

### Κλάση DatePicker

Αυτή η κλάση με πολύ ωραίο και χρήσιμο τρόπο εμφανίζει στην οθόνη του χρήστη ένα ημερολόγιο όπου διαλέγει όποια μέρα θέλει για να δει το αντίστοιχο γράφημα του node που έχει επιλέξει.Το πως φαίνεται θα δούμε στην παρακάτω εικόνα.



Εικόνα 26 : Απεικόνιση κλάσης DatePicker

Δεν θα αναφερθούμε στον κώδικα αυτό του παραθύρου καθώς υπάρχουν πληθώρα παραδείγματα στο Διαδίκτυο και ουσιαστικά με την βοήθειά του και εμείς κατασκευάσαμε το συγκεκριμένο παράθυρο.Πιο σημαντικό θέμα είναι να αναλύσουμε το σημείο μετά την επιλογή οποιαδήποτε ημερομηνίας δλδ τί θα γίνει περαιτέρω.

### Κλάση AnyDay

Αυτή η κλάση παίρνει ο constructor δυο ορίσματα id και την ημερομηνία που διάλεξε ο χρήστης.Μέσα στον constructor υλοποιείται μια σύνδεση στην βάση όπου αποθηκεύονται οι τιμές των nodes.Έπειτα καλείται μια συνάρτηση για να υπολογίσει την επόμενη μέρα από αυτή που ζήτησε ο χρήστης(μας είναι χρήσιμο για το ερώτημα που θα γίνει στην βάση).Το πιο σημαντικό κομμάτι αυτής της κλάσης είναι το ερώτημα που θα κάνουμε στην βάση δεδομένων.Είναι μεγάλης σημασίας το ερώτημα να μας <<φέρει>> τα σωστά

δεδομένα. Παρακάτω θα δούμε περιληπτικά τον κώδικα όπου υλοποιεί τα παραπάνω που αναφέραμε.

```
myCon=DriverManager.getConnection("jdbc:postgresql://host:port/task?user=username&password=password");

rs = myStatement.executeQuery("select * from public.mts400_results where nodeid = "+id+"
and result_time between (select date_trunc('day',timestamp '"+ts+"') and (select
date_trunc('day',timestamp '"+tsmp+"'))order by result_time ASC ");

Timestamp time=rs.getTimestamp(1);

cal.setTime(timestamp);

sumHum=sumHum + (float) (-2.0468 + (0.0367 * rs.getInt(6)) + (-1.2955 * 0.000001 *
rs.getInt(6) * rs.getInt(6)));

dataset.setValue(Double.parseDouble(Float.toString(sumHum/sa)),"node"+id,
Integer.toString(cal.get(Calendar.HOUR_OF_DAY)));

JFreeChart chart1 = ChartFactory.createBarChart3D ("Node "+id + " Humidity","Time
(HH)", "Humidity %", dataset, PlotOrientation.VERTICAL, true,true, false);
```

Η μεταβλητή [myCon](#) είναι τύπου Connection και χρησιμοποιείται για να συνδεθούμε στη βάση που είτε θα τρέχει στο PC μας είτε σε ένα άλλο μες στο δίκτυο. Στην δικιά μας περίπτωση ο server της βάσης δεδομένων μας είναι σε ένα PC στο τοπικό δίκτυο μας για αυτό και στην θέση του host πρέπει να βάλουμε την IP διεύθυνση που έχει ο server και μετά το αντίστοιχο port όπου δέχεται αιτήσεις για σύνδεση στη βάση. Φυσικά για να συνδεθούμε στη βάση πρέπει να προσθέσουμε τις παραμέτρους όπως το όνομα της βάσης το username και password του χρήστη.

Η επόμενη μεταβλητή έχει μεγάλη σημασία γιατί μας δείχνει([rs](#) είναι ένας δείκτης) το πίνακα που προκύπτει κάνοντας ένα ερώτημα (query) στη βάση.

Τέλος η μεταβλητή [sumHum](#) είναι τύπου float και χρησιμοποιείται για έχουμε το άθροισμα των τιμών της υγρασίας. Αυτές οι πράξεις που βλέπουμε είναι για να κάνουμε την τιμή του αισθητηρίου που είναι αποθηκευμένη στη βάση σε ένα αριθμό που να δηλώνει το ποσοστό της υγρασίας. Ανάλογες πράξεις γίνονται και στις τιμές της θερμοκρασίας για να παίρνουμε σε βαθμούς Κελσίου αλλά και στην τάση σε Volt.

Η τελευταία γραμμή του κώδικα που παρουσιάζουμε αφορά καθαρά το γράφημα και κάποιες σημαντικές παραμέτρους που πρέπει να ορίσουμε για το παράθυρο που θα εμφανιστεί το γράφημα. Στην *Εικόνα 25* μπορούμε να διακρίνουμε που συμβάλουν αυτές οι παράμετροι.

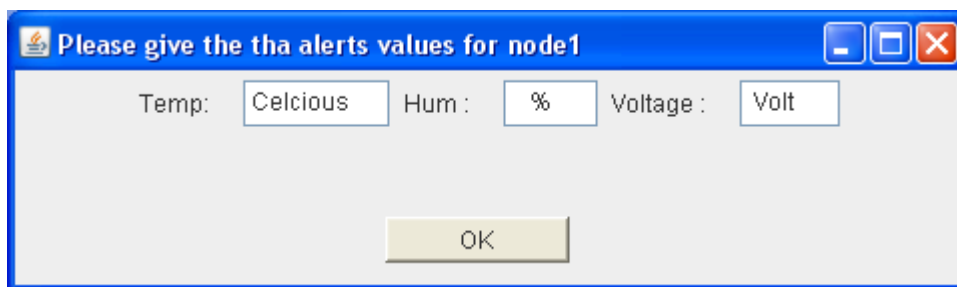
#### 5.2.4 Δημιουργία Alerts για την εφαρμογή



Η αλληλεπίδραση της εφαρμογής με το χρήστη είναι αναγκαίο να υφίσταται και όταν καταλήγουμε στην περίπτωση που κάποιο έχει κάποια μη επιτρεπτή τιμή. Στην δικιά μας εφαρμογή σύμφωνα με τα αισθητήρια που διαθέτουν τα nodes δηλαδή της θερμοκρασίας και της υγρασίας μπορούμε να ορίσουμε τις επιτρεπόμενες τιμές τους και θα προσθέσουμε σε αυτά την τάση του ρεύματος που έχουν κάθε στιγμή. Δηλαδή εμείς θα επιτηρούμε τις τιμές της θερμοκρασίας αλλά και τη τάση του κάθε node. Με την υγρασία δεν θα ασχοληθούμε καθόλου γιατί θα ήταν χρήσιμη μόνο με το συνδυασμό και κάποιου άλλου αισθητηρίου για να εξάγουμε κάποια συμπεράσματα. Παρακάτω θα προσπαθήσουμε να αναλύσουμε τις κλάσεις-αντικείμενα όπου μας βοηθάνε να κάνουμε την επιτήρηση των τιμών, να αποφασίζει ο χρήστης ποια θα είναι τα επιτρεπόμενα όρια και τέλος την ανάδραση που θα έχει η εφαρμογή όποτε ισχύει η περίπτωση που οι τιμές των αισθητηρίων ξεπεράσουν τα όρια που έχει ορίσει ο χρήστης.

### Κλάση Alarm\_node

Μέσα από το γραφικό περιβάλλον της εφαρμογής ο χρήστης μπορεί να επιλέξει την παραμετροποίηση των ορίων των τιμών των nodes. Όπως αναφέραμε και παραπάνω σημασία για μας θα έχει η θερμοκρασία και η τάση του κάθε node. Παρακάτω θα δούμε πώς φαίνεται το παράθυρο όπου ο χρήστης θα ορίζει τα όρια.



Εικόνα 27 : Απεικόνιση κλάσης Alarm\_node

Αφού ο χρήστης πληκτρολογήσει τις τιμές που επιθυμεί με το πάτημα του Button OK θα δημιουργείται ένα xml file με ανάλογη ονομασία. Το αρχείο αυτό θα έχει την ίδια περίπου δομή με αυτό το αρχείο που έχει όλες τις τιμές της βάσης απλά τα elements temperature, voltage, humidity θα δηλώνουν τα όρια και προφανώς σαν μέγεθος αρχείο θα είναι πολλές φορές μικρότερο. Παρακάτω παραθέτουμε τον ανάλογο κώδικα που υλοποιεί τα παραπάνω.

```
dbfac = DocumentBuilderFactory.newInstance();
docBuilder = dbfac.newDocumentBuilder();
doc = docBuilder.newDocument();
root = doc.createElement("Limits");
doc.appendChild(root);
child_node1 = doc.createElement("node");
child_node1.setAttribute("id", Integer.toString(d));
root.appendChild(child_node1);
c_child=doc.createElement("Alerts");
child_node1.appendChild(c_child);
temp=doc.createElement("Temprature");
temp.appendChild(doc.createTextNode(textTemp.getText()));
c_child.appendChild(temp);
volt=doc.createElement("Voltage");
volt.appendChild(doc.createTextNode(textVolt.getText()));
c_child.appendChild(volt);
hum=doc.createElement("Humidity");
hum.appendChild(doc.createTextNode(textHum.getText()));
c_child.appendChild(hum);
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
DOMSource source = new DOMSource(doc);
StreamResult result = new StreamResult(new File("D:\\alert_NODE1.xml"));
Toolkit.getDefaultToolkit().beep();//dimiourgw beep sound otan dimiourgithe to arxeio XML
transformer.transform(source, result);
```

*Εικόνα 28 : Απεικόνιση Κώδικα*

### **Κλάση AutoAlarm**

Είναι απολύτως αναγκαίο ο έλεγχος των τιμών των αισθητηρίων καθόλα την διάρκεια που εκτελείται η εφαρμογή επιτήρησης . Αυτή είναι η κύρια λειτουργία αυτής της κλάσης που θα περιγράψουμε παρακάτω . Αύτη η κλάση δημιουργεί κάθε 30 sec ένα αντικείμενο που ελέγχει τις τιμές της θερμοκρασίας και της τάσης των node.Στην περίπτωση που είμαστε μες στα όρια των τιμών η εφαρμογή απλά τυπώνει ένα μήνυμα και ενημερώνει το χρήστη ότι είναι όλα φυσιολογικά . Στην περίπτωση όμως που ξεπεράσει κάποια τιμή τα όρια η εφαρμογή κάνει τις εξής ενέργειες:

- Ακούγονται 10 beep
- Στέλνεται ένα email που ενημερώνει για τις τιμές των αισθητηρίων
- Τα textfields που αναγράφουν τις τιμές παίρνουν σαν background color το κόκκινο.

Παρακάτω θα αναφερθούμε στην υλοποίηση του αντικειμένου που στέλνει το notification email.

### **Κλάση SendEmail**

Μέσα από την κλάση μπορούμε να συνειδητοποιήσουμε την σπουδαιότητα,την χρησιμότητα και το μεγάλο εύρος δυνατοτήτων που μας προσφέρει η χρήση της Java για την κατασκευή χρήσιμων εφαρμογών.Κύριο βήμα είναι να κάνουμε import με το συνήθη τρόπο την

βιβλιοθήκη mail.jar στο project μας.Παρακάτω θα δούμε κάποιες γραμμές κώδικα της Java που set-άρουν την αποστολή email από μια desktop εφαρμογή.

```
props.put("mail.smtp.host", "smtp.gmail.com");
```

Εδώ ορίζουμε το πρωτόκολλο και το server που θα χρησιμοποιηθεί για να σταλεί το email.

```
message.setSubject("Notification From Sensor Monitoring Java Programm");
```

Εδώ ορίζουμε το θέμα που θα φέρει το email.

```
t.sendMessage(message, message.getAllRecipients());
```

Τέλος από τις πιο σημαντικές εντολές που γράφτηκαν η οποία ουσιαστικά στέλνει το email.

Με την βοήθεια που μας προσφέρει το Διαδίκτυο και τις υπεράριθμες πηγές πληροφοριών του (sites,blogs) μπορούμε να δούμε πως η mail.jar βιβλιοθήκη μας δίνει τις ίδιες επιλογές που μας προσφέρει και η αποστολή email χρησιμοποιώντας ένα πρόγραμμα περιήγηση στον Ιστό.

## Επίλογος

Αφού ολοκληρώθηκε όλη η πλατφόρμα επιτυχώς οφείλουμε να αναφέρουμε τα συμπεράσματα στα οποία καταλήξαμε. Το configuration των nodes θέλει αρκετό χρόνο στα manual της Crossbow και πολύ «ψάξιμο» πηγών από το Διαδίκτυο. Επίσης όσον αφορά την εφαρμογή της Java απαιτούσε την εύρεση των κατάλληλων βιβλιοθηκών για να υλοποιηθούν όλες οι διεργασίες κάτι το οποίο ήταν μια πολύωρη ασχολία κατά την διάρκεια της εκπόνησης της πτυχιακής μας εργασίας. Τα προβλήματα που ήρθαν στο προσκήνιο προερχόταν κυρίως από τις ελλειπείς γνώσεις μας για την τεχνολογία που χρησιμοποιήσαμε αλλά εν τέλει φτάσαμε στο σημείο να κατανοήσουμε πλήρως τα λάθη μας και να κάνουμε μια επιτυχημένη αποσφαλμάτωση. Για προφανείς λόγους η ενασχόληση και η ανάπτυξη αυτών των εφαρμογών μόνο θετικά πράγματα μπορούν να προσφέρουν στην κοινωνία και στην βιομηχανία. Η άμεση επίβλεψη των φυσικών μεγεθών έχει ως σκοπό να δώσει μια σημαντική βοήθεια στην λήψη αποφάσεων και αν σκεφτούμε ότι πιθανόν να υπάρχουν ήδη αποφάσεις για κάποια προ-σχεδιασμένα σενάρια ο διαχειριστής του συστήματος θα κάνει πιο εύκολα και πιο σωστά την δουλειά του. Τέτοιες παρεμφερές εφαρμογές βλέπουμε και αναπτύσσονται ευρέως στο εξωτερικό π.χ. στην γεωργία, σε πόλεις για επιτήρηση της ατμοσφαιρικής ρύπανσης, σε συστήματα ανίχνευσης κατολίσθησης, σε μηχανές που χρησιμοποιούνται στην βιομηχανία ως προς την «υγεία» τους, σε σύστημα παρακολούθησης υδρεύσης ή αποχέτευσης. Τέλος, όλο το εκπαιδευτικό σύστημα που υπολειτουργεί στην χώρα μας οφείλει με την άμεση ενίσχυση από τους κατάλληλους φορείς να δημιουργήσει συνθήκες ούτως ώστε να αναπτύσσονται στα ιδρύματα τέτοιες εφαρμογές μιας και αυτά μπορούν άμεσα να συνδεθούν με το καταναλωτικό κοινό.

## Παράρτημα

1. <http://el.wikipedia.org/wiki/PostgreSQL>
2. <http://en.wikipedia.org/wiki/Tinyos>
3. <http://en.wikipedia.org/wiki/NesC>
4. <http://www.memsic.com/products/wireless-sensor-networks/development-kits.html>
5. <http://www.xubuntu.org/>
6. <http://today.java.net/pub/a/today/2007/10/30/building-maps-into-swing-app-with-jxmapviewer.html#using-alternate-map-servers>
7. [java.net](http://java.net)
8. <http://netbeans.dzone.com/nb-ckeditor-javamail-tutorial>
9. HEAD FIRST JAVA (O'REILLY)
10. <http://java.sun.com/developer/onlineTraining/JavaMail/contents.html#JavaMailIntro>
11. <http://nesc.sourceforge.net/papers/nesc-ref.pdf>
12. <http://www.tinyos.net/tinyos-1.x/doc/tutorial/lesson1.html>