

# Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών *Πληροφοριακά και  
Επικοινωνιακά Συστήματα*

Μεταπτυχιακή Διατριβή



Διασύνδεση Αισθητήρα Θερμοκρασίας και Υγρασίας σε  
Ασύρματο Κόμβο για την Αποφυγή Συγκέντρωσης Δρόσου  
σε Καλλιέργειες Θερμοκηπίου

Αικατερίνη Προύσαλη

Επιβλέπων Καθηγητής  
Χρήστος Γκουμόπουλος

Οκτώβριος 2015

# **Ανοικτό Πανεπιστήμιο Κύπρου**

## **Σχολή Θετικών και Εφαρμοσμένων Επιστημών**

**Μεταπτυχιακό Πρόγραμμα Σπουδών Πληροφοριακά και  
Επικοινωνιακά Συστήματα**

### **Μεταπτυχιακή Διατριβή**

**Διασύνδεση Αισθητήρα Θερμοκρασίας και Υγρασίας σε  
Ασύρματο Κόμβο για την Αποφυγή Συγκέντρωσης Δρόσου  
σε Καλλιέργειες Θερμοκηπίου**

**Αικατερίνη Προύσαλη**

**Επιβλέπων Καθηγητής  
Χρήστος Γκουμόπουλος**

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών Στα Πληροφοριακά και Επικοινωνιακά Συστήματα από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών του Ανοικτού Πανεπιστημίου Κύπρου.

**Οκτώβριος 2015**



## Περίληψη

Ένα σημαντικό πρόβλημα που μπορεί να προκύψει εντός του χώρου του θερμοκηπίου είναι η δημιουργία συνθηκών που ευνοούν τη συγκέντρωση δρόσου στις καλλιέργειες που αναπτύσσονται μέσα σε αυτό. Η δημιουργία δρόσου ευνοεί την ανάπτυξη ασθενειών στα φυτά που μπορούν να οδηγήσουν ακόμη και σε καταστροφή της παραγωγής. Στην παρούσα διατριβή εξετάζεται η δυνατότητα χρήσης Ασύρματων Δικτύων Αισθητήρων (WSN) με σκοπό την ανίχνευση κι επεξεργασία των συνθηκών υγρασίας και θερμοκρασίας και στόχο την έγκαιρη επέμβαση όταν αυτές υποδεικνύουν την τάση δημιουργίας δρόσου. Στο πλαίσιο αυτό μελετήθηκε η τεχνολογία και οι εφαρμογές των WNS καθώς και οι εφαρμογές των Τεχνολογιών Πληροφορίας και Επικοινωνιών (ΤΠΕ) στη γεωργία που έχουν οδηγήσει στην υιοθέτηση του όρου «γεωργία ακριβείας». Με αυτά τα δεδομένα, επιλέχθηκαν κατάλληλες διατάξεις μέτρησης (αισθητήρες) οι οποίες χρησιμοποιήθηκαν στην ανάπτυξη ενός ασύρματου αισθητήριου κόμβου βασισμένο στην πλατφόρμα ανοικτού υλικού Arduino. Ο κόμβος, κάνοντας με τη χρήση των αισθητήρων συλλέγει κι αποστέλλει ασύρματα τα περιβαλλοντικά δεδομένα. Τα δεδομένα αυτά λαμβάνονται, επεξεργάζονται, αποθηκεύονται και παρουσιάζονται στο χρήστη μέσω μιας εφαρμογής που επίσης αναπτύχθηκε για το σκοπό αυτό και που μπορεί να εκτελείται σε οποιοδήποτε Η/Υ.

## **Summary**

An important issue that may arise inside a greenhouse area is the creation of conditions that permit the formation of dew in agricultures that grow inside. Dew creation enables for plant diseases development that can even lead to crop destruction. This dissertation examines the use of Wireless Sensor Networks (WSN) in sensing and processing humidity and temperature conditions as a mean for immediate intervention when these indicate dew formation tendency. In this context WSN technology and applications were studied along with the application of Information and Communication Technology (ICT) in agriculture that led to the introduction of the term “precision agriculture”. Considering these, appropriate sensing devices (sensors) were chosen and used in building a sensor node based on Arduino open hardware platform. The node uses the sensors to collect and a communication module to wireless send the environmental data. These are received, processed, stored and presented to the user via a desktop application that was also developed for this purpose and can be executed in any personal computer.

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους τους ανθρώπους που με την υπομονή και την καλή τους διάθεση στάθηκαν δίπλα μου σε όλη την προσπάθεια ολοκλήρωσης αυτού του κύκλου σπουδών. Ιδιαίτερες ευχαριστίες απευθύνονται στον επιβλέποντα της παρούσης εργασίας κο. Γκουμόπουλο για τη στήριξη και την υπομονή.

# Περιεχόμενα

Περίληψη .....	iv
Summary .....	v
Ευχαριστίες .....	vi
Περιεχόμενα .....	vii
Κεφάλαιο 1 .....	1
1.1 Πρόλογος.....	1
1.2 Σκοπός.....	2
1.3 Δομή εργασίας.....	2
Κεφάλαιο 2 .....	3
2.1 Κατηγορίες Ασύρματων Δικτύων Αισθητήρων .....	5
2.1.1 Επίγεια WSN.....	5
2.1.2 Υπόγεια WSN .....	5
2.1.3 Υποθαλάσσια WSN.....	6
2.1.4 Πολυμεσικά WSN .....	6
2.1.5 Κινητά WSN .....	7
2.2 Χαρακτηριστικά Γνωρίσματα Ασύρματων Δικτύων Αισθητήρων .....	8
2.2.1 Ανοχή σε σφάλματα .....	8
2.2.2 Κλιμάκωση.....	9
2.2.3 Κόστος Παραγωγής.....	9
2.3 Αρχιτεκτονική WSN .....	12
2.3.1 Φυσικό Επίπεδο.....	13
2.3.2 Επίπεδο Ζεύξης Δεδομένων .....	14
2.3.3 Επίπεδο Δικτύου.....	15
2.3.4 Επίπεδο Μεταφοράς.....	16
2.3.5 Επίπεδο Εφαρμογής .....	17
2.4 Πρότυπα και Πρωτόκολλα WSN .....	17
2.4.1 Bluetooth (IEEE 802.15.1).....	17
2.4.2 ZigBee .....	18
2.4.3 UWB ( IEEE 802.15.3) .....	20
2.4.4 Wi-Fi (IEEE 802.11).....	21
2.4.5 Άλλα πρωτόκολλα.....	22
2.5 Εφαρμογές Ασύρματων Δικτύων Αισθητήρων.....	22
2.5.1 Στρατιωτικές Εφαρμογές.....	23

2.5.2 Περιβαλλοντικές Εφαρμογές .....	24
2.5.3 Εφαρμογές Υγείας.....	24
2.5.4 Οικιακές Εφαρμογές .....	24
2.5.5 Βιομηχανικές Εφαρμογές.....	25
2.5.6 Άλλες εμπορικές Εφαρμογές.....	26
Κεφάλαιο 3 .....	27
3.1 Εφαρμογές WSNs στη Γεωργία Ακριβείας.....	27
Ενδεικτικές εφαρμογές σε κάθε μια από τις παραπάνω κατηγορίες είναι [130]:.....	27
3.1.1 Εφαρμογές σε συστήματα συλλογής δεδομένων .....	27
3.1.2 Εφαρμογές σε συστήματα ελέγχου άρδευσης.....	28
3.1.3 Εφαρμογές σε συστήματα παροχής πληροφοριών .....	29
3.1.4 Εφαρμογές σε συστήματα ελέγχου θερμοκηπίου.....	30
3.2 Συνθήκες Θερμοκηπίου.....	30
3.2.1 Θερμοκρασία.....	31
3.2.2 Σχετική Υγρασία .....	31
3.2.3 Σημείο Δρόσου.....	32
Κεφάλαιο 4 .....	34
4.1 Αισθητήρες Θερμοκρασίας.....	34
4.1.1 Μέτρηση θερμοκρασίας με χρήση θερμίστορ.....	36
4.2 Υγρασία και σημείο δρόσου .....	37
Κεφάλαιο 5 .....	40
5.1 Αισθητήριοι κόμβοι.....	40
5.1.1 Arduino .....	41
5.1.2 Σύνδεση θερμίστορ .....	43
5.1.3 Σύνδεση αισθητήρα υγρασίας .....	46
5.1.4 Δημιουργία κόμβου και Έλεγχος λειτουργίας.....	48
5.2 Διασύνδεση Κόμβων.....	50
5.2.1 XBee.....	51
5.2.2 Σύνδεση XBee.....	53
5.2.2 Ρύθμιση XBees.....	53
5.2.3 Σύνδεση τροφοδοσίας .....	55
5.3 Λήψη και Διαχείριση Δεδομένων .....	57
5.3.1 Υλοποίηση εφαρμογής σε Java .....	57
5.3.2 Βάση Δεδομένων.....	59
5.3.3 Σύνδεση με XBee .....	60
5.3.4 Υπολογισμός Σημείου Δρόσου .....	61
5.3.2 Εκκίνηση Προγράμματος.....	62



5.4	Λειτουργία Εφαρμογής .....	62
5.5	Έλεγχος Λειτουργίας Συστήματος .....	66
Κεφάλαιο 6	.....	68
Παράρτημα Α	.....	70
A.1	Arduino .....	70
A.2	JAVA.....	71
Βιβλιογραφία	.....	80

# Κεφάλαιο 1

## Εισαγωγή

### 1.1 Πρόλογος

«*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.*» Με τη φράση αυτή ο Marc Weiser [39] περιγράφει την εποχή της ήρεμης τεχνολογίας, η οποία μετά την κυριαρχία των mainframes και των προσωπικών υπολογιστών χαρακτηρίζει τη νέα εποχή υπολογιστικής, στην οποία κυριαρχούν συστήματα διάχυτου υπολογισμού (ubiquitous computing) τα οποία βρίσκονται πανταχού παρόντα στο παρασκήνιο της ανθρώπινης δραστηριότητας.

Ένα βασικό χαρακτηριστικό των συστημάτων αυτών είναι η αλληλεπίδρασή τους με το πλαίσιο (context). Το πλαίσιο ορίζεται [1] ως η πληροφορία που μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει την κατάσταση κάθε οντότητας που εντάσσεται σε αυτό. Η πληροφορία αυτή, για να είναι αξιοποιήσιμη, πρέπει πριν την επεξεργασία της να συλλεγεί. Η συλλογή των παραμέτρων πλαισίου μπορεί να γίνει με διάφορους τρόπους, όπως μέσω της κατάστασης δικτύου και συσκευών, του προφίλ ενός χρήστη ή μιας συσκευής, ή με χρήση ειδικών διατάξεων που ονομάζονται αισθητήρες (sensors) [10]. Οι τιμές των παραμέτρων πλαισίου που μπορούν να συλλεχθούν μέσω των αισθητήρων μπορούν απεικονίζουν φυσικά μεγέθη, όπως θερμοκρασία, υγρασία, βαρομετρική πίεση, ήχος, ταχύτητα, επιτάχυνση, μαγνητικό πεδίο, ρύπους κ.α.

Ένας ή περισσότεροι, ενεργειακά αυτόνομοι, κόμβοι με ενσωματωμένους αισθητήρες, οι οποίοι παρατηρούν, καταγράφουν, και μεταδίδουν τα δεδομένα ασύρματα σε έναν ή περισσότερους σταθμούς βάσης για περαιτέρω επεξεργασία συνθέτουν ένα Ασύρματο Δίκτυο Αισθητήρων (Wireless Sensor Network – WSN). Τα δίκτυα αυτά βρίσκουν εφαρμογή σε πολλούς τομείς της καθημερινής δραστηριότητας [19], ακόμη και σε παραδοσιακούς τομείς της οικονομίας, όπως η αγροτική παραγωγή, και αποτελούν ένα βασικό αρωγό στη διαμόρφωση και ανάπτυξη του Διαδικτύου των Πραγμάτων (Internet of Things – IoT) [8].

## 1.2 Σκοπός

Η εργασία εκπονείται στο πλαίσιο του Μεταπτυχιακού Προγράμματος Σπουδών στα Πληροφοριακά Συστήματα. Σκοπός της εργασίας είναι η μελέτη ενός ιδιαίτερου τομέα συστημάτων διάχυτου υπολογισμού, βασικού συστατικού του Διαδικτύου των Πραγμάτων, τα Ασύρματα Δίκτυα Αισθητήρων (Wireless Sensor Networks). Σε συνέχεια της βασικής μελέτης, θα εξεταστούν οι δυνατότητες που προσδίδει η χρήση τους στον τομέα της αγροτικής παραγωγής. Ακόμη, η εργασία θα επεκταθεί στην κατασκευή ενός συστήματος, βασισμένου σε αισθητήρες, το οποίο, εφαρμοζόμενο σε καλλιέργειες εντός θερμοκηπίου, θα έχει τη δυνατότητα έγκαιρης προειδοποίησης για την αποφυγή δρόσου σε καλλιέργειες εντός θερμοκηπίου.

## 1.3 Δομή εργασίας

Μετά την παρούσα εισαγωγή, στο Κεφάλαιο 2 εξετάζονται τα Ασύρματα Δίκτυα Αισθητήρων (Wireless Sensor Networks – WSN) ως προς τα χαρακτηριστικά και τις εφαρμογές τους. Στη συνέχεια, στο Κεφάλαιο 3, η εργασία επικεντρώνεται στον τομέα της Γεωργίας Ακριβείας (Precision Agriculture) ο οποίος προϋποθέτει την ενσωμάτωση τεχνολογιών πληροφορίας και επικοινωνιών (Information and Telecommunication Technology – ICT) στον τομέα της αγροτικής παραγωγής. Το Κεφάλαιο 4 προτείνει την υλοποίηση ενός WSN με χρήση της ανοικτής πλατφόρμας υλικού Arduino, η οποία ελέγχει κατάλληλους αισθητήρες και συλλέγει και αποστέλλει τα δεδομένα σε κατάλληλα διαμορφωμένο σταθμό βάσης. Η υλοποίηση του πρότυπου δικτύου και ενός σχετικού συστήματος λογισμικού παρουσιάζεται στο Κεφάλαιο 5. Τέλος, το Κεφάλαιο 6 αποτελεί τον επίλογο της εργασίας.

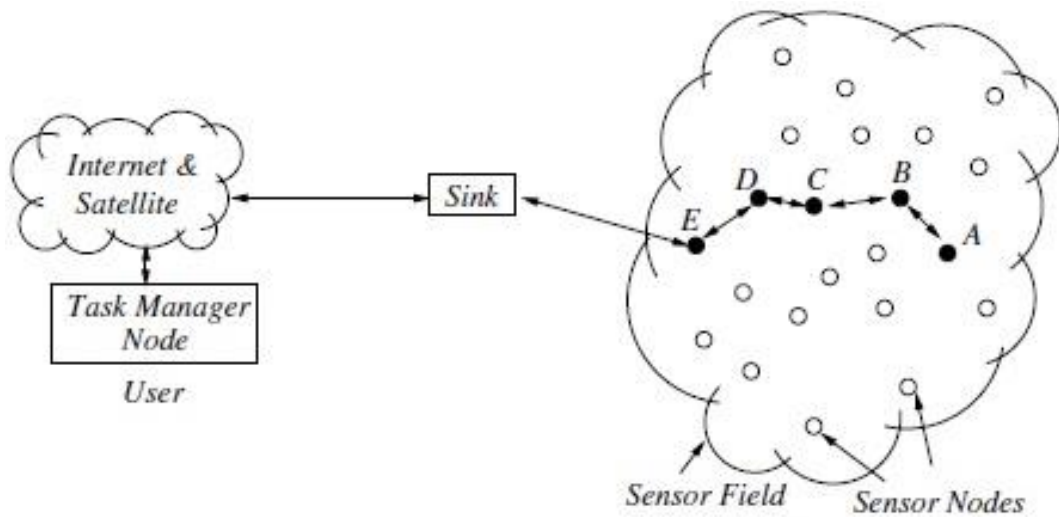
# Κεφάλαιο 2

## Ασύρματα Δίκτυα Αισθητήρων

Τα δίκτυα αισθητήρων, όπως και άλλες τεχνολογικές λύσεις, με χαρακτηριστικό παράδειγμα το διαδίκτυο (internet) έχουν τις ρίζες τους σε εφαρμογές που αναπτύχθηκαν για στρατιωτικούς σκοπούς. Οι εφαρμογές αυτές, έπειτα από αρκετά χρόνια έρευνας και πειραματικής λειτουργίας κατέληξαν να έχουν μια ευρύτερη απήχηση. Η αρχική εφαρμογή των δικτύων αισθητήρων, αναφέρεται να υπήρξε το Σύστημα Ηχητικής Παρακολούθησης SOSUS (Sound Surveillance System), το οποίο χρησιμοποιήθηκε στις αρχές τις δεκαετίας του 1950 κατά τη διάρκεια του ψυχρού πολέμου για την ανίχνευση και τον εντοπισμό σοβιετικών υποβρυχίων μέσω ειδικών ακουστικών αισθητήρων (υδρόφωνα) [13].

Τα τελευταία χρόνια, τα ασύρματα δίκτυα αισθητήρων απέκτησαν μια παγκόσμια απήχηση. Σε αυτό συνέβαλε καθοριστικά, η ραγδαία ανάπτυξη της επιστήμης των Μικρο-Ηλεκτρο-Μηχανικών συστημάτων (MEMS) καθώς και των ασύρματων επικοινωνιών, από την δεκαετία του 1980 και έπειτα, η οποία διευκόλυνε την ανάπτυξη των έξυπνων αισθητήρων, ώστε να έχουν μεταξύ τους την δυνατότητα της ασύρματης επικοινωνίας [38].

Ένα WSN αποτελείται από κόμβους που διακρίνονται στις πηγές (sources), οι οποίες μπορούν να λαμβάνουν, να αποστέλλουν και να προωθούν τις πληροφορίες στο δίκτυο με προορισμό τους αποδέκτες (sinks), οι οποίοι τις λαμβάνουν και τις αποθηκεύουν για τη μετέπειτα επεξεργασία τους. Ο αποδέκτης μπορεί να αποτελείται από έναν απλό κόμβο ή κάποια οντότητα που δεν ανήκει στο δίκτυο αισθητήρων (π.χ. Laptop ή Pda) ή κάποιον κόμβο γέφυρα (gateway) που προωθεί την πληροφορία σε ένα μεγαλύτερο δίκτυο (π.χ. Internet) [3].



**Εικόνα 1.** Δομή WSN [3]

Τυπικά, διακρίνονται δύο τύποι ασύρματων δικτύων αισθητήρων, το δομημένο (structured) και το αδόμητο (unstructured) [40]. Ένα αδόμητο ασύρματο δίκτυο αισθητήρων περιέχει πυκνά και τυχαία διασπαρμένους στο πεδίο, κόμβους αισθητήρων (Sensor Nodes - SN), οι οποίοι σχηματίζουν μεταξύ τους ένα αυτο-διαμορφούμενο, αδόμητο (μέσω κατά βούλησης -ad-hoc-συνδέσεων) δίκτυο. Ένα ιδιαίτερα σημαντικό μειονέκτημα αυτού του δικτύου είναι η έλλειψη διαχείρισης και συνοχής καθώς και η δυσκολία στον εντοπισμό σφαλμάτων λόγω του μεγάλου αριθμού ασύρματων κόμβων, τυχαία τοποθετημένων. Ο τρόπος αυτός της τοποθέτησης των ασύρματων κόμβων μπορεί να οδηγήσει στην αδυναμία πλήρους κάλυψη της περιοχής ενδιαφέροντος, και κατά συνέπεια την αδυναμία λήψης επαρκών μετρήσεων.

Αντίθετα, ένα δομημένο ασύρματο δίκτυο αισθητήρων περιέχει κόμβους τοποθετημένους, μετά από μελέτη, σε προεπιλεγμένες θέσεις στο πεδίο, παρέχοντας κάλυψη σε συγκεκριμένα σημεία του πεδίου μέτρησης. Το πλεονέκτημα αυτού του δικτύου είναι ότι λόγω της άρτιας τοποθέτησης είναι δυνατή η χρήση λιγότερων κόμβων αισθητήρων. Το γεγονός αυτό, σε συνδυασμό με τη γνώση της θέσης κάθε κόμβου έχει ως αποτέλεσμα το χαμηλότερο κόστος συντήρησης και διαχείρισης του δικτύου.

Συνήθως, σε WSN που καλύπτουν μεγάλη έκταση, η άμεση επικοινωνία μεταξύ μιας πηγής και του sink είναι αδύνατη. Για το λόγο αυτό ο σχεδιασμός και η ανάπτυξη ενός δικτύου αισθητήρων επιβάλλει και την παρεμβολή ενδιάμεσων κόμβων αισθητήρων, οι οποίοι λειτουργούν ως γέφυρες. Οι ενδιάμεσοι αυτοί κόμβοι συλλέγουν τα δεδομένα, ενδεχομένως μπορεί να τα επεξεργαστούν και τα αναμεταδίδουν στη συνέχεια σε μια επόμενη γέφυρα ή στον αποδέκτη

(sink). Το ιδιαίτερο χαρακτηριστικό ενός κόμβου γέφυρα είναι πως απαιτείται μεγαλύτερη ενεργειακή πηγή, μεγαλύτερη επεξεργαστική ικανότητα, και μεγαλύτερη εμβέλεια μετάδοσης.

## 2.1 Κατηγορίες Ασύρματων Δικτύων Αισθητήρων

Τα ασύρματα δίκτυα αισθητήρων μπορούν να αναπτυχθούν εξίσου επίγεια, υπόγεια και υποθαλάσσια. Ανάλογα με το πεδίο υλοποίησης και το σκοπό που επιτελούν, μπορούν να διακριθούν στις ακόλουθες κατηγορίες [18, 40]:

- Επίγεια (terrestrial) WSN
- Υπόγεια (underground) WSN
- Υποθαλάσσια (underwater) WSN
- Πολυμεσικά (multimedia) WSN
- Κινητά (mobile) WSN

### 2.1.1 Επίγεια WSN

Τα επίγεια (terrestrial) WSN αποτελούνται από εκατοντάδες ως χιλιάδες οικονομικούς κόμβους αισθητήρων, οι οποίοι αναπτύσσονται σε μια δεδομένη περιοχή είτε με τη μορφή ενός αδόμητου ή με τη μορφή ενός δομημένου δικτύου. Σε ένα επίγειο WSN είναι πολύ σημαντική η αξιόπιστη επικοινωνία, ειδικά σε ένα πυκνό περιβάλλον όπου δύνανται να παρεμβάλλονται φυσικά ή τεχνητά εμπόδια: Οι κόμβοι θα πρέπει να είναι σε θέση να επικοινωνήσουν αποτελεσματικά με τον σταθμό βάσης. Στην περίπτωση που η ισχύς της κύριας πηγής ενέργειας είναι περιορισμένη και δεν είναι επαναφορτιζόμενη, οι κόμβοι είναι καλό να εξοπλίζονται με μια δευτερεύουσα πηγή ενέργειας όπως είναι τα ηλιακά κύτταρα (solar cells).

### 2.1.2 Υπόγεια WSN

Τα υπόγεια (underground) WSN αποτελούνται από ένα συγκεκριμένο αριθμό κόμβων αισθητήρων, οι οποίοι τοποθετούνται υπογείως είτε μέσα στο χώμα, είτε μέσα σε ένα όρυγμα ή ένα ορυχείο, ώστε να παρατηρούνται και να καταγράφονται οι συνθήκες που επικρατούν. Σε

αυτή την περίπτωση χρησιμοποιούνται απαραίτητα και επίγειες γέφυρες-κόμβοι (sink nodes) ώστε να μπορεί να αναμεταδοθεί αξιόπιστα η πληροφορία στο σταθμό βάσης. Συγκριτικά με τα επίγεια WSN, πρέπει να τονιστεί ότι η εγκατάσταση τους είναι πιο ακριβή λόγω του εξειδικευμένου εξοπλισμού και της συντήρησής του. Υπάρχουν αρκετά φυσικά εμπόδια, όπως είναι το έδαφος, το νερό, διάφορα πετρώματα και ορυκτά, τα οποία αποτελούν πρόκληση για την αξιόπιστη ασύρματη μετάδοση δεδομένων καθώς παρατηρούνται υψηλά επίπεδα εξασθένησης ή ακόμη και απώλειας του σήματος. Έτσι οι απαιτήσεις για τον σχεδιασμό και την ανάπτυξη ενός υπόγειου WSN αυξάνονται λόγω της μεγαλύτερης ανάγκης για κατανάλωση ενέργειας και του μεγαλύτερου κόστους. Όπως και στα επίγεια WSN, βασικός στόχος είναι η εξοικονόμηση ενέργειας ώστε να παραταθεί η διάρκεια ζωής του κόμβου. Αυτό μπορεί να επιτευχθεί με την εφαρμογή ενός πιο αποδοτικού πρωτοκόλλου επικοινωνίας.

### **2.1.3 Υποθαλάσσια WSN**

Τα υποθαλάσσια (underwater) WSN αποτελούνται από έναν μικρό αριθμό, λόγω του μεγάλου τους κόστους, κόμβων αισθητήρων οι οποίοι τοποθετούνται μέσα στο νερό, καθώς και από κάποια αυτόνομα υποβρύχια οχήματα, τα οποία χρησιμοποιούνται για την εξερεύνηση ή την συλλογή δεδομένων από τους κόμβους. Στα υποθαλάσσια WSN χρησιμοποιείται μια αραιή διάταξη των κόμβων αισθητήρων και η ασύρματη επικοινωνία εγκαθιδρύεται μέσω των ακουστικών κυμάτων. Οι προκλήσεις που προκύπτουν στα υποθαλάσσια WSN είναι το περιορισμένο εύρος ζώνης, η μεγάλη καθυστέρηση διάδοσης του σήματος καθώς και η εξασθένηση του. Έτσι οι υποθαλάσσιοι κόμβοι θα πρέπει να είναι αυτόνομοι και προσαρμοσμένοι στις δύσκολες συνθήκες του ωκεανού. Το μεγαλύτερο ζήτημα στα υποθαλάσσια WSN είναι η διατήρηση ενέργειας και περιλαμβάνει την ανάπτυξη κατάλληλων τεχνικών υποβρύχιας επικοινωνίας και δικτύωσης.

### **2.1.4 Πολυμεσικά WSN**

Η μείωση του κόστους και του μεγέθους σε υλικά όπως κάμερες και μικρόφωνα έχουν δώσει ιδιαίτερη ώθηση στα πολυμεσικά (multimedia) δίκτυα αισθητήρων [2] τα οποία χρησιμοποιούνται για την παρακολούθηση χώρου (video surveillance), την παρακολούθηση συνθηκών περιβάλλοντος, εφαρμογές υγείας κ.α με σκοπό να επιτρέπουν την ανίχνευση και την

παρακολούθηση γεγονότων με την μορφή πολυμέσων, όπως είναι το βίντεο, ο ήχος και η εικόνα. Οι κόμβοι διασυνδέονται μεταξύ τους ασύρματα με σκοπό την ανάκτηση δεδομένων, την επεξεργασία, τη συσχέτιση και τέλος την αποστολή τους μετά από συμπίεση (για εξοικονόμηση αποθηκευτικού χώρου κι εύρους ζώνης). Γενικά, αναπτύσσονται σε μια περιοχή με τη μορφή ενός δομημένου δικτύου ώστε να επιτυγχάνεται πλήρης κάλυψη. Οι προκλήσεις που προκύπτουν αφορούν απαιτήσεις για υψηλό εύρος ζώνης, υψηλή κατανάλωση ενέργειας, παροχή ποιότητας υπηρεσιών (QoS), τεχνικές επεξεργασίας και συμπίεσης δεδομένων και πολυεπίπεδος σχεδιασμός. Συγκεκριμένα, το περιεχόμενο ενός πολυμέσου, όπως είναι η ροή ενός βίντεο απαιτεί υψηλό εύρος ζώνης προκειμένου να παραδοθεί στον παραλήπτη και αυτό έχει ως αποτέλεσμα ο μεγάλος ρυθμός δεδομένων να οδηγεί σε υψηλή κατανάλωση ενέργειας. Είναι σημαντικό να τονιστεί ότι πρέπει να επιτευχθεί ένα συγκεκριμένο επίπεδο στην παροχή ποιότητας υπηρεσίας (QoS) ώστε να μεταδοθεί αξιόπιστα το πολυμεσικό περιεχόμενο, λόγω της μεταβαλλόμενης καθυστέρησης (jitter) και της καθυστέρησης λόγω ανεπαρκούς χωρητικότητας καναλιού (latency). Η απόδοση βελτιώνεται σημαντικά υπό την προϋπόθεση φιλτραρίσματος της περιττής πληροφορίας και συμπίεσης του περιεχομένου.

### **2.1.5 Κινητά WSN**

Τα κινητά (mobile) WSN αποτελούνται από ένα σύνολο κόμβων αισθητήρων, οι οποίοι είναι αυτοκινούμενοι, αλληλοεπιδρούν με το φυσικό περιβάλλον και έχουν την ικανότητα της ανίχνευσης, του υπολογισμού και της επικοινωνίας όπως ακριβώς λειτουργούν και οι στατικοί κόμβοι αισθητήρων. Μία βασική τους διαφορά είναι ότι οι κινητοί κόμβοι έχουν την ικανότητα να επανατοποθετούνται και να αυτο-οργανώνονται μέσα στο δίκτυο. Σε ένα κινητό WSN, οι κόμβοι διανέμονται έτσι ώστε να συγκεντρώσουν τις πληροφορίες και επικοινωνούν μεταξύ τους εφόσον βρίσκονται εντός εμβέλειας. Ακόμη μία σημαντική διαφορά είναι η κατανομή των δεδομένων στο δίκτυο, καθώς στα κινητά WSN χρησιμοποιείται η δυναμική δρομολόγηση (dynamic routing). Οι δυσκολίες που προκύπτουν περιλαμβάνουν το σχεδιασμό, την τοποθέτηση, την αυτο-οργάνωση, τον έλεγχο διακίνησης της πληροφορίας, την πλήρη κάλυψη της περιοχής, την ενέργεια, τη συντήρηση και τέλος την επεξεργασία των δεδομένων. Οι εφαρμογές των κινητών WSN περιλαμβάνουν την παρακολούθηση του περιβάλλοντος, την ανίχνευση στόχου, την αναζήτηση και διάσωση και την παρακολούθηση σε πραγματικό χρόνο κάποιου επικίνδυνου υλικού.



## 2.2 Χαρακτηριστικά Γνωρίσματα Ασύρματων Δικτύων Αισθητήρων

Σε αντίθεση με τα παραδοσιακά δίκτυα, ένα ασύρματο δίκτυο αισθητήρων έχει τους δικούς του ιδιαίτερους περιορισμούς και σχεδιασμό. Οι περιορισμοί περιλαμβάνουν:

- την περιορισμένη διαθέσιμη ποσότητα ενέργειας
- την περιορισμένη εμβέλεια επικοινωνίας
- το χαμηλό εύρος ζώνης και
- την περιορισμένη δυνατότητα επεξεργασίας κι αποθήκευσης των δεδομένων στον κόμβο.

Όσο αφορά τον σχεδιασμό, οι περιορισμοί εξαρτώνται από την εφαρμογή και βασίζονται στο περιβάλλον και τα χαρακτηριστικά του. Το περιβάλλον αποτελεί καθοριστικό παράγοντα στον σχεδιασμό της δομής και του μέγεθος του δικτύου καθώς και στην τοπολογία δικτύου που θα επιλεγεί να χρησιμοποιηθεί. Όταν η εφαρμογή αφορά εσωτερικούς και κλειστούς χώρους απαιτούνται λιγότεροι κόμβοι για να υλοποιηθεί το δίκτυο, ενώ όταν αφορά υπαίθριους εξωτερικούς χώρους η απαίτηση σε κόμβους για να καλυφθεί μια ευρύτερη περιοχή αυξάνεται. Στην περίπτωση δε που το δίκτυο αποτελείται από εκατοντάδες ή χιλιάδες κόμβους αισθητήρων, καθώς και όταν το περιβάλλον είναι απροσπέλαστο από τον ανθρώπινο παράγοντα, τότε είναι προτιμότερη η ανάπτυξη ενός αδόμητου δικτύου. Επίσης, διάφορα ιδιαίτερα χαρακτηριστικά του περιβάλλοντος χώρου μπορεί να περιορίσουν την επικοινωνία μεταξύ των κόμβων κι έτσι να επηρεαστεί σημαντικά η συνδεσιμότητα, άρα κι η απαιτούμενη τοπολογία του δικτύου.

### 2.2.1 Ανοχή σε σφάλματα

Συνήθως σε ένα δίκτυο WSN με μεγάλο αριθμό κόμβων είναι πολύ πιθανό να προκύψουν σφάλματα από απώλεια ή αλλοίωση της πληροφορίας. Οι αιτίες που μπορεί να προκαλέσουν αυτά τα φαινόμενα είναι η έλλειψη της απαιτούμενης ενέργειας, η φυσική καταστροφή των κόμβων, οι παρεμβολές που δημιουργούνται στο ασύρματο μέσο και γενικότερα οι άσχημες

περιβαλλοντικές συνθήκες που μπορεί να επικρατούν. Έτσι, για να είναι αξιόπιστη η λειτουργία του δικτύου θα πρέπει να υπάρχει μια ανοχή σε τέτοιου είδους σφάλματα. Η πιθανότητα της αξιόπιστης λειτουργίας  $R_k(t)$  ενός κόμβου  $k$  σε ένα χρονικό διάστημα  $(0,t)$  ορίζεται από την εξίσωση Poisson [3]:

$$R_k(t) = \exp(-\lambda_k t),$$

Όπου  $\lambda_k$  ο ρυθμός αστοχίας του κόμβου  $k$  και  $t$  ο χρόνος.

### 2.2.2 Κλιμάκωση

Ο αριθμός των κόμβων αισθητήρων που μπορεί να συμμετέχουν στη μελέτη ενός φαινομένου μπορεί να είναι πάρα πολύ μεγάλος. Από εκατοντάδες ως χιλιάδες. Η αρχιτεκτονική του δικτύου, τα πρωτόκολλα του δικτύου, καθώς και ο αλγόριθμος δρομολόγησης που θα χρησιμοποιηθεί θα πρέπει να είναι σε θέση να διαχειριστούν αυτό το μεγάλο πλήθος των κόμβων και τη μεγάλη χωρική πυκνότητά τους. Η χωρική πυκνότητα υπολογίζεται ως εξής [3]:

$$\mu(R) = \frac{N\pi R^2}{A}$$

όπου  $N$  ο αριθμός των διασκορπισμένων κόμβων αισθητήρων σε μια περιοχή εμβαδού  $A$  και  $R$  η απόσταση εκπομπής.

### 2.2.3 Κόστος Παραγωγής

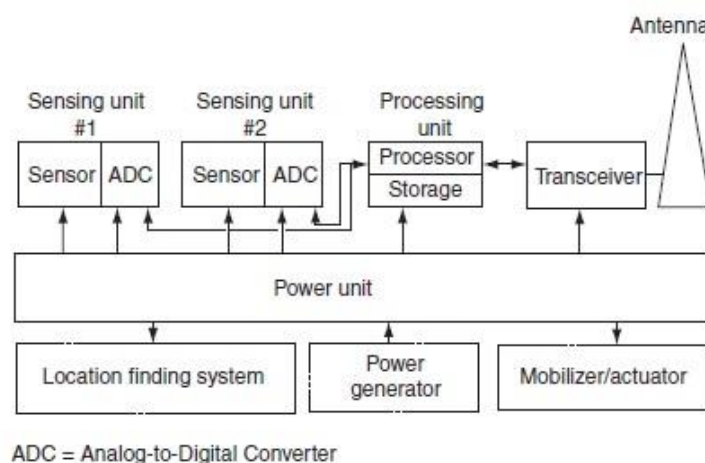
Καθώς τα δίκτυα αισθητήρων αποτελούνται από ένα μεγάλο αριθμό αισθητήρων κόμβων, το κόστος ενός μόνο κόμβου είναι πολύ σημαντικός παράγοντας ώστε να υπολογιστεί το συνολικό κόστος υλοποίησης και λειτουργίας του δικτύου. Συγκριτικά, εάν το κόστος του δικτύου αποβεί πιο ακριβό σε σχέση με το κόστος παραδοσιακών μεθόδων, καταλήγουμε στο συμπέρασμα ότι η ανάπτυξη ενός τέτοιου δικτύου δεν είναι συμφέρουσα. Συμπερασματικά, το κόστος του κάθε αισθητήριου κόμβου θα πρέπει να είναι όσο το δυνατό μικρότερο.

## 2.2.4 Αρχιτεκτονική του αυτόνομου κόμβου

Οι αυτόνομοι κόμβοι είναι συσκευές μικρές σε μέγεθος, έχουν χαμηλή κατανάλωση ενέργειας και χαμηλό κόστος. Αποτελούνται από έναν ή περισσότερους αισθητήρες (sensors), μία μονάδα επεξεργασίας (processor and memory), μια μονάδα παροχής ενέργειας (power supply), έναν πομποδέκτη (radio) κι έναν ενεργοποιητή (actuator). Επιπρόσθετα, ανάλογα με την εφαρμογή και τις παραμέτρους της, το σύστημα μπορεί να περιέχει επιπλέον ένα σύστημα ανίχνευσης θέσης (Location Finding System), μια μονάδα παραγωγής ενέργειας (Power Generator) καθώς και έναν μηχανισμό κίνησης (mobilizer) στην περίπτωση που απαιτείται η κινητικότητα των κόμβων για την παρατήρηση κάποιου φαινομένου.

Αναλυτικότερα, η αρχιτεκτονική του αυτόνομου κόμβου είναι:

1. **Ένας ή περισσότεροι αισθητήρες (sensors).** Ανάλογα με την εφαρμογή, επιλέγονται οι κατάλληλοι αισθητήρες, οι οποίοι θα παρατηρήσουν και θα συλλέξουν τα δεδομένα. Επίσης, σε αυτή τη μονάδα περιλαμβάνεται κι ένα σύστημα μετατροπής των δεδομένων από αναλογική σε ψηφιακή μορφή έτσι ώστε να είναι εφικτή η μετέπειτα επεξεργασία τους από τη μονάδα επεξεργασίας.
2. **Μία μονάδα επεξεργασίας (Processing unit).** Η μονάδα αυτή ενσωματώνει ένα μικροεπεξεργαστή (processor) και μια μνήμη (storage) για την επεξεργασία και την αποθήκευση των δεδομένων.
3. **Μία μονάδα παροχής ενέργειας (power unit).** Η μονάδα αυτή συνήθως αποτελείται από μια πηγή ενέργειας (μπαταρία), ώστε να παρέχεται μια περιορισμένη ενεργειακή αυτονομία.
4. **Μία μονάδα αμφίδρομης επικοινωνίας (radio or transceiver).** Η μονάδα αυτή αποτελείται από έναν πομποδέκτη ραδιοσυχνοτήτων (Radio Frequency – RF), ο οποίος διασυνδέει τον αισθητήριο κόμβο ασύρματα στο δίκτυο.
5. **Έναν ενεργοποιητή (Actuator),** ο οποίος συνήθως εκτελεί κάποια πολύ απλή διαδικασία όπως το άνοιγμα και το κλείσιμο ενός διακόπτη ή τη μετατροπή του ηλεκτρικού σήματος σε κάποια άλλη μορφή.



**Εικόνα 2.** Αρχιτεκτονική Κόμβου [3]

Από την ανάλυση και περιγραφή του αισθητήριου κόμβου προκύπτουν κάποιοι αυστηροί περιορισμοί του υλικού μέρους οι οποίοι είναι:

- Το απαιτούμενο μέγεθος του θα πρέπει να είναι μικρό και το βάρος του ελάχιστο.
- Θα πρέπει να καταναλώνει εξαιρετικά χαμηλή ενέργεια.
- Θα πρέπει να λειτουργεί σε υψηλή ογκομετρική πυκνότητα.
- Θα πρέπει να έχει μικρό κόστος παραγωγής και να είναι αυτόνομος.
- Θα πρέπει να προσαρμόζεται ανάλογα στο περιβάλλον.

### 2.2.5 Κατανάλωση Ενέργειας

Κάθε αισθητήριος κόμβος είναι μια μικρή ηλεκτρονική συσκευή κι εξοπλίζεται με μια περιορισμένη πηγή ενέργειας, που συνήθως είναι μια μπαταρία. Έτσι η διάρκεια ζωής ενός αισθητήριου κόμβου εξαρτάται άμεσα από την διάρκεια ζωής της πηγής ενέργειας, για την οποία, ανάλογα με την εφαρμογή, μπορεί να καταστεί αδύνατη η μελλοντική αντικατάστασή της. Σε ένα δίκτυο ad-hoc πολλαπλών βημάτων, κάθε κόμβος αναλαμβάνει την ευθύνη της αποστολής και της δρομολόγησης των πακέτων δεδομένων. Στην περίπτωση της δυσλειτουργίας κάποιων κόμβων, θα πρέπει να αναδιοργανωθεί το δίκτυο και να επαναδρομολογηθούν τα πακέτα δεδομένων. Συνεπώς η διατήρηση και η διαχείριση της ενέργειας είναι σημαντικοί παράγοντες στο σχεδιασμό ενός ασύρματου δικτύου αισθητήρων κι η κατανάλωση ενέργειας θα μπορούσε να κατανεμηθεί σε τρεις τομείς:

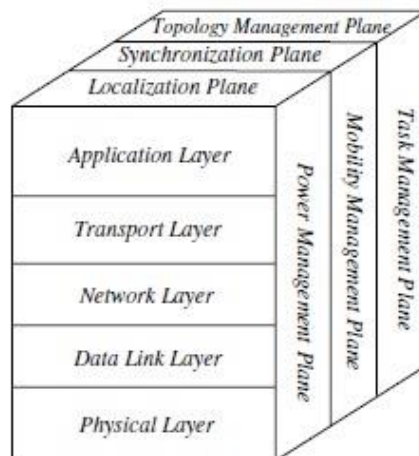
- στην ανίχνευση,
- στην επικοινωνία και
- στην επεξεργασία δεδομένων.

## 2.3 Αρχιτεκτονική WSN

Η στοίβα πρωτοκόλλων χρησιμοποιείται από τον κεντρικό σταθμό βάσης (sink) κι από όλους τους κόμβους αισθητήρων σε ένα δίκτυο. Ακολουθεί το μοντέλο OSI και αποτελείται από τα ακόλουθα πέντε επίπεδα:

- το φυσικό επίπεδο,
- το επίπεδο ζεύξης δεδομένων,
- το επίπεδο δικτύου,
- το επίπεδο μεταφοράς,
- το επίπεδο εφαρμογής

Επίσης παράλληλα υλοποιούνται τα επίπεδα διαχείρισης ενέργειας, διαχείρισης κίνησης, διαχείρισης εργασιών και το επίπεδο εντοπισμού, το επίπεδο συγχρονισμού και το επίπεδο διαχείρισης τοπολογίας.



**Εικόνα 3.** Στοίβα Πρωτοκόλλων [3]

Η λειτουργία των επιπέδων τους διαχείρισης ενέργειας, κίνησης και εργασιών είναι ιδιαίτερα σημαντική κι εστιάζεται στην επιτήρηση της ενέργειας, της κίνησης και της κατανομής εργασιών μεταξύ των αισθητήριων κόμβων ώστε να εξασφαλιστεί η σωστή και αποδοτικά ενεργειακή

λειτουργία τους, η δρομολόγηση τα δεδομένων σε ένα ασύρματο δίκτυο αισθητήρων και ο διαμοιρασμός των διαθέσιμων πόρων μεταξύ τους με σκοπό τη συνολική μείωση στην κατανάλωση ενέργειας και την παράταση ζωής του δικτύου. Αναλυτικότερα:

### **1. Επίπεδο Διαχείρισης Ενέργειας (Power Management Plane):**

Διαχειρίζεται τον τρόπο με τον οποίο ένας αισθητήριος κόμβος καταναλώνει την ενέργεια του. Για παράδειγμα, ένας κόμβος μετά τη λήψη ενός μηνύματος μπορεί να κλείσει κάποιον γειτονικό του δέκτη. Αυτό συμβαίνει ώστε να αποφευχθούν τα διπλότυπα μηνύματα. Επίσης, εάν διαπιστωθεί ότι το επίπεδο ενέργειας είναι χαμηλό, τότε ο αισθητήριος κόμβος εκπέμπει στους γειτονικούς του ότι δεν μπορεί να συμμετέχει στη δρομολόγηση των μηνυμάτων. Έτσι, η ενέργεια που υπολείπεται χρησιμοποιείται απλώς για ανίχνευση.

### **2. Επίπεδο Διαχείρισης Κίνησης (Mobility Management Plane):**

Εντοπίζει και καταγράφει την κίνηση του κάθε αισθητήριου κόμβου, έτσι ώστε να εξασφαλίζεται πάντα η διαδρομή πίσω στο χρήστη κι οι αισθητήριοι κόμβοι να μπορούν να παρακολουθούν τους γειτονικούς τους. Με τον τρόπο αυτό μπορεί να αντισταθμιστεί το ποσοστό της κατανάλωσης ενέργειας σε σχέση με τον αριθμό των εργασιών που λαμβάνουν χώρα.

### **3. Επίπεδο Διαχείρισης Εργασιών (Task Management Plane):**

Προγραμματίζει τις εργασίες ανίχνευσης που πραγματοποιούνται σε μια συγκεκριμένη περιοχή. Για την απόδοση και την εκτέλεση της εργασίας δεν απαιτείται η λειτουργία όλων των αισθητήριων κόμβων ταυτόχρονα. Αυτό έχει ως αποτέλεσμα κάποιοι αισθητήριοι κόμβοι να εκτελούν μια εργασία περισσότερες φορές σε σχέση με άλλους, πάντα λαμβάνοντας υπόψη το επίπεδο της διαθέσιμης ενέργειας τους,

Οι λειτουργίες που επιτελούνε τα επίπεδα δικτύου, είναι συνοπτικά:

#### **2.3.1 Φυσικό Επίπεδο**

Το φυσικό επίπεδο (physical layer) είναι υπεύθυνο για την υλοποίηση των σωστών τεχνικών της διαμόρφωσης, της μετάδοσης και της λήψης του σήματος. Αρμοδιότητες των πρωτοκόλλων

φυσικού επιπέδου είναι: Η επιλογή της συχνότητας, η δημιουργία της φέρουσας συχνότητας, η ανίχνευση του σήματος, η διαμόρφωση και αποδιαμόρφωση του σήματος.

### **2.3.2 Επίπεδο Ζεύξης Δεδομένων**

Το επίπεδο ζεύξης δεδομένων (Data Link Layer) είναι υπεύθυνο για τη για την πολυπλεξία της ροής δεδομένων, για την ανίχνευση των πλαισίων δεδομένων και τη διασφάλιση της αξιόπιστης επικοινωνίας μέσω τεχνικών ελέγχου σφαλμάτων και της διαχείρισης πρόσβασης στο μέσο (Medium Access Control - MAC) με στόχο να ελαχιστοποιηθούν οι συγκρούσεις με γειτονικούς κόμβους. Εξασφαλίζει την αξιόπιστη σύνδεση σημείου προς σημείο (point to point) και σημείου προς πολλά σημεία (point to multipoint) σε ένα δίκτυο επικοινωνίας.

#### **MAC**

Το πρωτόκολλο MAC έχει ως σκοπό την επίτευξη δύο στόχων. Ο πρώτος στόχος αφορά τη δημιουργία της δομής του δικτύου. Χιλιάδες αισθητήριοι κόμβοι μπορεί να είναι πυκνά διασκορπισμένοι σε μία περιοχή. Για τους κόμβους αυτούς θα πρέπει να εδραιωθούν σύνδεσμοι επικοινωνίας για την μεταφορά των δεδομένων. Ο δεύτερος στόχος αφορά την δίκαιη και αποδοτική κατανομή των πόρων του μέσου μεταξύ των αισθητήριων κόμβων. Οι πόροι περιλαμβάνουν χρονική διάρκεια, ενέργεια και συχνότητα. Ανεξάρτητα από το σύστημα πρόσβασης μέσου, η ενεργειακή απόδοση είναι υψίστης σημασίας και το πρωτόκολλο MAC θα πρέπει να υποστηρίζει τη τρόπους εξοικονόμησης ενέργειας για τους αισθητήριους κόμβους. Ο πιο προφανής τρόπος είναι να τεθεί εκτός λειτουργίας ο πομποδέκτης όταν δεν απαιτείται η χρήση του. Αν και αυτή η μέθοδος φαίνεται ότι παρέχει σημαντικό ενεργειακό κέρδος, μπορεί να παρεμποδίσει τη συνδεσιμότητα του δικτύου, καθώς αν ο πομποδέκτης τεθεί εκτός λειτουργίας δεν θα μπορεί να λάβει πλαίσια δεδομένων από τους γειτονικούς κόμβους και ουσιαστικά θα αποσυνδεθεί από το δίκτυο. Στην πράξη, εφόσον ο πομποδέκτης τεθεί εκτός λειτουργίας σε μια αδρανή χρονική διάρκεια ένας άλλος αισθητήριοι κόμβος που πιθανώς τον χρησιμοποιούσε, θα καταλήξει να ξοδέψει περισσότερη ενέργεια για την αποστολή πακέτων δεδομένων ως ένα πιο απομακρυσμένο κόμβο. Οπότε εξάγεται το συμπέρασμα πως ο συγκεκριμένος τρόπος εξοικονόμησης ενέργειας είναι αποδοτικός μόνο αν ο χρόνος που αναλώνεται είναι μεγαλύτερος από ένα ορισμένο χρονικό όριο (κατώφλι).

## ΕΛΕΓΧΟΣ ΣΦΑΛΜΑΤΩΝ

Άλλη μία σημαντική λειτουργία του επιπέδου ζεύξης δεδομένων είναι ο έλεγχος σφαλμάτων κατά την μετάδοση δεδομένων. Υπάρχουν δύο τεχνικές ελέγχου και διόρθωσης σφαλμάτων, η διόρθωση λαθών στον δέκτη (Forward Error Correction - FEC) και η αυτόματη αίτηση επανάληψης (Automatic Repeat Request - ARQ). Η χρησιμότητα της τεχνικής ARQ σε διάφορες εφαρμογές ενός δικτύου αισθητήρων περιορίζεται σημαντικά από το κόστος των επιπρόσθετων αναμεταδόσεων. Από την άλλη, η πολυπλοκότητα της αποκωδικοποίησης είναι μεγαλύτερη στην τεχνική FEC καθώς πρέπει να αναπτυχθούν ικανότητες όπως είναι η διόρθωση σφαλμάτων. Για το σχεδιασμό ενός τέτοιου συστήματος είναι σημαντική η καλή γνώση των χαρακτηριστικών του καναλιού μετάδοσης.

### 2.3.3 Επίπεδο Δικτύου

Στο επίπεδο δικτύου (Network Layer) υλοποιείται η δρομολόγηση κι η προώθηση που παρέχονται από το επίπεδο μεταφοράς. Οι αισθητήριοι κόμβοι βρίσκονται διασκορπισμένοι σε μια περιοχή και τα δεδομένα που συλλέγονται μεταδίδονται σε έναν κεντρικό σταθμό βάσης (sink), ο οποίος μπορεί να βρίσκεται αρκετά μακριά από την περιοχή. Ωστόσο, η περιορισμένη απόσταση μετάδοσης από τους αισθητήριους κόμβους εμποδίζει την άμεση επικοινωνία μεταξύ αυτών και του σταθμού βάσης. Για να αντιμετωπιστεί το πρόβλημα, απαιτείται η χρήση αποδοτικών πρωτοκόλλων δρομολόγησης πολλαπλών βημάτων με ενδιάμεσους αισθητήριους κόμβους να λειτουργούν ως αναμεταδότες. Οι υπάρχουσες τεχνικές δρομολόγησης γι' ασύρματα δίκτυα ad-hoc δεν είναι συνήθως κατάλληλες να καλύψουν τις απαιτήσεις των δικτύων αισθητήρων, καθώς οι αρχές που πρέπει να λαμβάνονται υπόψη είναι:

- Η ενεργειακή απόδοση
- Τα δίκτυα αισθητήρων είναι συνήθως Δεδομένο-κεντρικά (Data-centric).
- Η δυνατότητα συγκέντρωσης δεδομένων από πολλαπλούς γειτονικούς κόμβους
- Λόγω του μεγάλου και μη σταθερού αριθμού κόμβων, δε μπορούν να παρέχονται μοναδικά αναγνωριστικά για κάθε ένα κόμβο, οπότε διευθυνσιοδοτούνται δυναμικά με βάση τα δεδομένα ή την τοποθεσία τους.

Το σημαντικό ζήτημα που αφορά τη δρομολόγηση είναι ότι βασίζεται σε δεδομένο-κεντρικά αιτήματα (data-centric queries). Ανάλογα με τις πληροφορίες που αιτείται ο χρήστης, το



πρωτόκολλο δρομολόγησης θα πρέπει να διευθυνσιοδοτεί διαφορετικούς κόμβους που θα παρέχουν τις αιτούμενες πληροφορίες.

Μία ακόμη σημαντική λειτουργία του επιπέδου δικτύου είναι η διαδικτύωση με εξωτερικά δίκτυα, όπως άλλα δίκτυα αισθητήρων, συστήματα εντολών και ελέγχου, καθώς και το διαδίκτυο. Οι κόμβοι αποδέκτες (sink nodes) μπορούν να χρησιμοποιηθούν ως πύλη (gateway) για άλλα δίκτυα, ή να δημιουργήσουν ένα δίκτυο κορμού (backbone), με πρόσβαση άλλα δίκτυα μέσω μίας πύλης (gateway).

### **2.3.4 Επίπεδο Μεταφοράς**

Το επίπεδο μεταφοράς (Transport Layer) διασφαλίζει τη διατήρηση της ροής των δεδομένων και είναι εξαιρετικά αναγκαίο όταν το δίκτυο χρειάζεται να έχει πρόσβαση στο διαδίκτυο ή σε άλλα εξωτερικά δίκτυα. Το πρωτόκολλο TCP δε μπορεί να αντιμετωπίσει τις προκλήσεις που θέτει ένα WSN. Σε αντίθεση με τα πρωτόκολλα όπως το TCP, τα συστήματα επικοινωνίας από άκρη σε άκρη (end-to-end) στα δίκτυα αισθητήρων δε βασίζονται σε μια παγκόσμια διευθυνσιοδότηση, αλλά θεωρούν ότι η διευθυνσιοδότηση χρησιμοποιεί τα πακέτα δεδομένων ή την τοποθεσία των κόμβων τους για τη δήλωση του τόπου προορισμού. Παράγοντες, όπως η κατανάλωση ενέργειας (power consumption) και η κλιμάκωση (scalability) και χαρακτηριστικά στοιχεία όπως είναι η δεδομένο-κεντρική δρομολόγηση δείχνουν ότι τα δίκτυα αισθητήρων χρειάζονται μια διαφορετική προσέγγιση στο επίπεδο μεταφοράς. Έτσι, αυτές οι απαιτήσεις τονίζουν την ανάγκη για νέους τύπους πρωτοκόλλων στο επίπεδο αυτό.

Η ανάπτυξη των πρωτοκόλλων του επιπέδου μεταφοράς είναι αρκετά απαιτητική διότι οι κόμβοι του δικτύου επηρεάζονται από τους περιορισμούς του υλικού μέρους, που αφορούν π.χ. στην διαθέσιμη ενέργεια και στον αποθηκευτικό χώρο της μνήμης. Αυτό έχει ως αποτέλεσμα, κάθε αισθητήριος κόμβος να μην έχει την δυνατότητα να αποθηκεύσει μεγάλες ποσότητες δεδομένων και οι επιβεβαιώσεις αποστολής και παραλαβής (acknowledgments) είναι αρκετά δαπανηρές για τα δίκτυα αισθητήρων. Για την διασφάλιση της επικοινωνίας μέσα σε ένα WSN, τα πρωτόκολλα επιπέδου μεταφοράς απαιτούν δύο κύριες λειτουργίες, την αξιοπιστία και τον έλεγχο συμφόρησης (congestion control). Ο περιορισμένος αριθμός πόρων και το υψηλό κόστος ενέργειας απαγορεύει στους μηχανισμούς αξιόπιστης επικοινωνίας των συστημάτων από άκρη σε άκρη να χρησιμοποιηθούν στα ασύρματα δίκτυα αισθητήρων και καθιστά απαραίτητους τους τοπικούς μηχανισμούς αξιοπιστίας. Επιπλέον, η συμφόρηση του δικτύου που μπορεί να

προκύψει εξαιτίας της μεγάλης διακίνησης στη διάρκεια των γεγονότων θα πρέπει να αμβλυνθεί από τα πρωτόκολλα επιπέδου μεταφοράς, τα οποία έχουν ως σκοπό να αξιοποιήσουν στο έπακρο τις δυνατότητες των αισθητήριων κόμβων.

### **2.3.5 Επίπεδο Εφαρμογής**

Το επίπεδο εφαρμογής (Application Layer) περιλαμβάνει την κύρια εφαρμογή που έχει αναπτυχθεί. Επιπρόσθετα, εκτός από τον κώδικα της εφαρμογής, ο οποίος υλοποιείται για κάθε εφαρμογή ξεχωριστά, συναντώνται επίσης τα αιτήματα επεξεργασίας και οι διάφορες λειτουργίες διαχείρισης του δικτύου.

Η αρχιτεκτονική στοίβα των επιπέδων αρχικά υιοθετήθηκε για την ανάπτυξη των δικτύων αισθητήρων λόγω της επιτυχημένης σύνδεσης με το διαδίκτυο. Ωστόσο, οι μεγάλες κλίμακες εφαρμογών των WSN φανερώνουν ότι το ασύρματο κανάλι μετάδοσης έχει μεγάλη επίδραση στα υψηλότερα επίπεδα πρωτοκόλλων. Επιπλέον, οι περιορισμοί των πόρων και η φύση της ειδικής εφαρμογής του WSN οδηγεί σε λύσεις cross-layer που ενσωματώνουν την πολυεπίπεδη στοίβα πρωτοκόλλων.

## **2.4 Πρότυπα και Πρωτόκολλα WSN**

Τα πρότυπα που αναφέρονται στα WSN αναπτύχθηκαν κυρίως με βάση την απαίτηση για χαμηλή κατανάλωση ενέργειας. Ένα πρότυπο καθορίζει τις λειτουργίες και τα πρωτόκολλα που είναι απαραίτητα για την διασύνδεση των κόμβων αισθητήρων με κάποιο εξωτερικό δίκτυο. Τα πιο διαδεδομένα πρωτόκολλα είναι το Bluetooth (IEEE 802.15.1) [20], το ZigBee (IEEE 802.15.4) [11], το Ultra-Wideband (IEEE 802.15.3) [36] και το Wi-Fi (IEEE 802.11) [15].

### **2.4.1 Bluetooth (IEEE 802.15.1)**

Το πρότυπο IEEE 802.15.1 καθορίζει τα χαρακτηριστικά του φυσικού επιπέδου (physical layer) και του ελέγχου πρόσβασης μέσου (medium access control, MAC) για Ασύρματα δίκτυα χαμηλού ρυθμού μετάδοσης δεδομένων (L-RWPAN). Τα πλεονεκτήματα ενός δικτύου L-RWPAN είναι η

εύκολη εγκατάσταση, η αξιόπιστη μεταφορά δεδομένων, η μικρή περιοχή εμβέλειας, το εξαιρετικά χαμηλό κόστος, μια αρκετά καλή διάρκεια ζωής της μπαταρίας και μια απλή και ευέλικτη στοίβα πρωτοκόλλου.

Το Bluetooth περιγράφεται από το πρότυπο IEEE 802.15.1 κι αναφέρεται συνήθως σε ασύρματα δίκτυα τοπικής περιοχής (Wireless Personal Area Networks, WPAN). Είναι ένα πρωτόκολλο ασύρματης μετάδοσης σημάτων μέσω μικροκυμάτων, στην περιοχή των 2,4GHz, σχεδιασμένο για μικρή και περιορισμένη εμβέλεια, και συνήθως συναντάται σε συσκευές όπως είναι το ποντίκι του Η/Υ, το πληκτρολόγιο, ο εκτυπωτής κ.α. Ένα δίκτυο Bluetooth μπορεί να ακολουθεί δύο πιθανές τοπολογίες σύνδεσης: Το piconet και το scatternet.

Το piconet είναι ένα WPAN στο οποίο εδράζεται ένας κόμβος Master επάνω στον οποίο συνδέονται ένας ως επτά κόμβοι slaves. Η μέθοδος μετάδοσης της πληροφορίας που χρησιμοποιείται είναι η διασπορά συχνότητας (frequency hopping). Η επικοινωνία είναι αμφίδρομη και επιτρέπει τις απευθείας συνδέσεις από συσκευή σε συσκευή (point to point) ή από συσκευή σε πολλές συσκευές (point to multipoint). Όταν βρίσκονται στον χώρο δύο ή περισσότερα piconet, είναι συνετό ένας κόμβος του ενός να συνδέεται με ένα κόμβο του άλλου, σχηματίζοντας ένα μεγαλύτερο WPAN, το scatternet.

#### **2.4.2 ZigBee**

Το ZigBee καθορίζει το υψηλότερο επίπεδο πρωτοκόλλων επικοινωνίας που έχουν αναπτυχθεί σύμφωνα με το πρότυπο IEEE 802.15.4. Το ZigBee είναι μια απλή, χαμηλού κόστους και χαμηλής κατανάλωσης ενέργειας ασύρματη τεχνολογία επικοινωνίας που χρησιμοποιείται συνήθως ενσωματωμένη σε εφαρμογές.

Η ZigBee Alliance είναι μία ένωση εταιρειών, οι οποίες αναπτύσσουν πρότυπα και προϊόντα για αξιόπιστη, με χαμηλό κόστος και χαμηλή ισχύ ασύρματη δικτύωση. Το ZigBee είναι γνωστό ως το πρωτόκολλο που στηρίζεται στο πρότυπο IEEE 802.15.4 για ασύρματα προσωπικά δίκτυα υπολογιστών χαμηλού ρυθμού μετάδοσης (Low-Rate Wireless Personal Area Networks, LR-WPAN) το οποίο ορίζει το φυσικό επίπεδο και το επίπεδο MAC και χρησιμοποιείται σε συσκευές με μικρή κατανάλωση ενέργειας σε μια μικρή γενικά περιοχή λειτουργίας. Τα πλεονεκτήματα ενός δικτύου LR-WPAN είναι η εύκολη εγκατάσταση του, η αξιόπιστη μεταφορά δεδομένων, η

μικρή περιοχή λειτουργίας του, το εξαιρετικά χαμηλό κόστος και μια ικανοποιητική διάρκεια στη ζωή της μπαταρίας του.

Το ZigBee παρέχει μια αξιόπιστη, αυτόνομη και πολλαπλών βημάτων δικτύωση (multihop wireless communication) με μεγαλύτερη διάρκεια σε ενέργεια. Για τους παραπάνω λόγους, είναι το πλέον διαδεδομένο πρωτόκολλο για ασύρματα δίκτυα αισθητήρων στη γεωργία και τη κτηνοτροφία καθώς τα χαρακτηριστικά του ικανοποιούν τις απαιτήσεις που εγείρονται. Το πρωτόκολλο ZigBee καθορίζει τις προδιαγραφές του επιπέδου δικτύου για την τοπολογία αστέρα, δέντρου και peer to peer.

Γενικά, ένα δίκτυο Zigbee ανάλογα με την τοπολογία μπορεί να αποτελείται από τις παρακάτω συσκευές:

- Από έναν μόνο κεντρικό συντονιστή (PAN Coordinator) σε κάθε δίκτυο, ο οποίος είναι μια συσκευή πλήρους λειτουργίας (Full Function Device - FFD) και είναι υπεύθυνος για τη δημιουργία και τον έλεγχο του δικτύου και βρίσκεται πάντα ανοιχτός σε κατάσταση αναμονής.
- Από τις τερματικές συσκευές (End Devices), οι οποίες μπορεί να είναι είτε συσκευές πλήρους λειτουργίας (Full Function Device – FFD), είτε συσκευές μειωμένης λειτουργίας (Reduced Function Device – RFD) και μπορούν απλά να παρακολουθούν και να διεκπεραιώνουν λειτουργίες ελέγχου. Οι συσκευές μειωμένης λειτουργίας (Reduced Function Device – RFD) έχουν περιορισμένες δυνατότητες και τροφοδοτούνται με μπαταρία οπότε, όταν βρίσκονται σε κατάσταση αναμονής είναι κλειστοί για να εξοικονομήσουν ενέργεια.
- Από τους ενδιάμεσους δρομολογητές (Routers), οι οποίοι είναι συσκευές πλήρους λειτουργίας (Full Function Device – FFD) και χρησιμοποιούνται για να αυξηθεί η εμβέλεια του δικτύου και να συνδεθούν περισσότεροι κόμβοι μεταξύ τους. Βρίσκονται πάντα ανοιχτοί σε κατάσταση αναμονής και μπορούν να παρακολουθούν και να διεκπεραιώνουν λειτουργίες ελέγχου.

Αναλυτικότερα για τα μοντέλα των τοπολογιών έχουμε:

### **1. Τοπολογία Αστέρα**

Στην τοπολογία αστέρα (star), μία συσκευή πλήρους λειτουργίας FFD αναλαμβάνει τον ρόλο του συντονιστή και όλες οι άλλες περιφερειακές συσκευές λειτουργούν ως

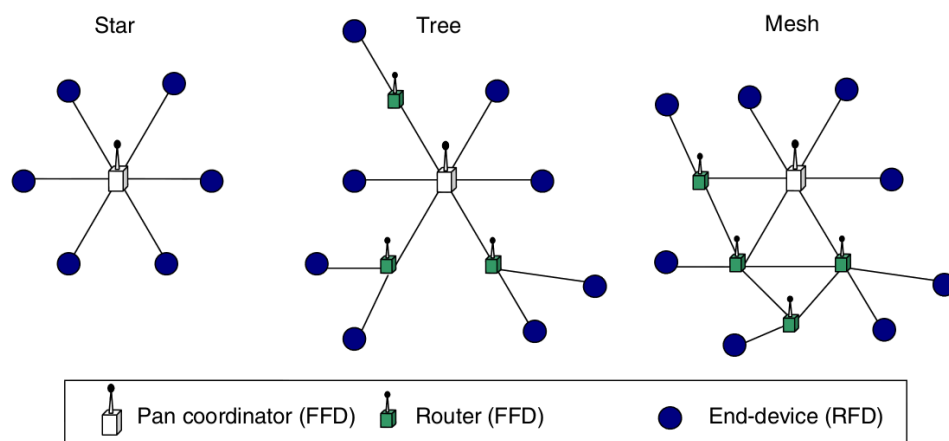
τερματικές, δηλαδή μπορούν να επικοινωνήσουν μόνο με τον διαχειριστή και όχι μεταξύ τους.

## 2. Τοπολογία Δέντρου

Στην τοπολογία δέντρου (tree), η οποία αποτελεί μια ειδική περίπτωση της τοπολογίας peer to peer υπάρχει μια συσκευή πλήρους λειτουργίας FFD όπου αναλαμβάνει τον ρόλο του συντονιστή. Όπως φαίνεται στην Εικόνα 4, τα κλαδιά του δέντρου που σχηματίζονται είναι κόμβοι ενδιάμεσων δρομολογητών και τα φύλλα είναι τερματικές συσκευές που επικοινωνούν μόνο με τον αποδέκτη και όχι μεταξύ τους (τοπολογία αστέρα).

## 3. Τοπολογία Mesh

Στην τοπολογία mesh, κάθε συσκευή πλήρους λειτουργίας FFD μπορεί να επικοινωνήσει με οποιονδήποτε ενδιάμεσο δρομολογητή (Router) στο δίκτυο. Όλες οι συσκευές μπορούν να επικοινωνήσουν μεταξύ τους μέσω του αλγόριθμου δρομολόγησης πολλαπλών βημάτων, ακόμα και αν βρίσκονται εκτός εμβέλειας (χρησιμοποιώντας κάποιο κοντινό διαθέσιμο κόμβο - router).



Εικόνα 4. Τοπολογίες ZigBee

### 2.4.3 UWB ( IEEE 802.15.3)

Το UWB είναι επίσης γνωστό ως το πρότυπο IEEE 802.15.3 για υψηλής ταχύτητας ασύρματα προσωπικά δίκτυα μικρής όμως περιοχής εμβέλειας. Ένα από τα πιο σημαντικά χαρακτηριστικά του UWB αφορά το εύρος ζώνης, το οποίο είναι μεγαλύτερο από 480 Mbps και είναι αποδεκτό για εφαρμογές πολυμέσων, όπως είναι η μετάδοση ήχου και βίντεο σε ένα τοπικό δίκτυο.

## 2.4.4 Wi-Fi (IEEE 802.11)

Το Wireless Fidelity (Wi-Fi) είναι επίσης γνωστό ως το πρότυπο IEEE 802.11 για ασύρματα τοπικά δίκτυα (WLAN). Επιτρέπει στους χρήστες να έχουν πρόσβαση στο δίκτυο μέσω τοπικών δικτύων μεγάλων ταχυτήτων αφού συνδεθούν σε ένα σημείο πρόσβασης (Access Point - AP) ή μεταξύ τους (ad-hoc).

Ο πυρήνας του IEEE 802.11 καθορίζει ένα σύνολο βασικών υπηρεσιών (BSS), το οποίο αποτελείται από κινητά ή σταθερά τμήματα. Εάν ένα τμήμα αφαιρεθεί από το δικό του σύνολο BSS τότε δε θα μπορεί πια να επικοινωνήσει με τα υπόλοιπα μέλη. Με βάση το σύνολο βασικών υπηρεσιών (BSS), διακρίνονται το ανεξάρτητο σύνολο βασικών υπηρεσιών (IBSS) και το εκτεταμένο σύνολο υπηρεσιών (ESS). Η λειτουργία του IBSS είναι δυνατή, όταν τα τμήματα μπορούν να επικοινωνήσουν απευθείας χωρίς κάποιο σημείο πρόσβασης (AP). Επειδή αυτός ο τύπος δικτύου συχνά χρησιμοποιείται χωρίς κάποιο προ-σχεδιασμό και μόνο για όσο χρονικό διάστημα χρειάζεται, αναφέρεται ως ένα ad-hoc δίκτυο.

Μια συγκριτική παρουσίαση των παραπάνω προτύπων, σύμφωνα με τους Lee et al. [25], παρουσιάζεται στον Πίνακα 1:

Standard	Bluetooth	UWB	ZigBee	Wi-Fi
IEEE spec.	802.15.1	802.15.3a *	802.15.4	802.11a/b/g
Frequency band	2.4 GHz	3.1-10.6 GHz	868/915 MHz; 2.4 GHz	2.4 GHz; 5 GHz
Max signal rate	1 Mb/s	110 Mb/s	250 Kb/s	54 Mb/s
Nominal range	10 m	10 m	10 - 100 m	100 m
Nominal TX power	0 - 10 dBm	-41.3 dBm/MHz	(-25) - 0 dBm	15 - 20 dBm
Number of RF channels	79	(1-15)	1/10; 16	14 (2.4 GHz)
Channel bandwidth	1 MHz	500 MHz - 7.5 GHz	0.3/0.6 MHz; 2 MHz	22 MHz
Modulation type	GFSK	BPSK, QPSK	BPSK (+ ASK), O-QPSK	BPSK, QPSK COFDM, CCK, M-QAM
Spreading	FHSS	DS-UWB, MB-OFDM	DSSS	DSSS, CCK, OFDM
Coexistence mechanism	Adaptive freq. hopping	Adaptive freq. hopping	Dynamic freq. selection	Dynamic freq. selection, transmit power control (802.11h)
Basic cell	Piconet	Piconet	Star	BSS
Extension of the basic cell	Scatternet	Peer-to-peer	Cluster tree, Mesh	ESS
Max number of cell nodes	8	8	> 65000	2007
Encryption	E0 stream cipher	AES block cipher (CTR, counter mode)	AES block cipher (CTR, counter mode)	RC4 stream cipher (WEP), AES block cipher
Authentication	Shared secret	CBC-MAC (CCM)	CBC-MAC (ext. of CCM)	WPA2 (802.11i)
Data protection	16-bit CRC	32-bit CRC	16-bit CRC	32-bit CRC

\* Unapproved draft.  
 • Acronyms: ASK (amplitude shift keying), GFSK (Gaussian frequency SK), BPSK/QPSK (binary/quadrature phase SK), O-QPSK (offset-QPSK), OFDM (orthogonal frequency division multiplexing), COFDM (coded OFDM), MB-OFDM (multiband OFDM), M-QAM (M-ary quadrature amplitude modulation), CCK (complementary code keying), FHSS/DSSS (frequency hopping/direct sequence spread spectrum), BSS/ESS (basic/extended service set), AES (advanced encryption standard), WEP (wired equivalent privacy), WPA (Wi-Fi protected access), CBC-MAC (cipher block chaining message authentication code), CCM (CTR with CBC-MAC), CRC (cyclic redundancy check).

Πίνακας 1. Σύγκριση πρωτοκόλλων επικοινωνίας WSN [25]

## 2.4.5 Άλλα πρωτόκολλα

- **WirelessHart**

Το πρότυπο WirelessHart [34] παρέχει ένα πρωτόκολλο δικτύου ασύρματης επικοινωνίας για εφαρμογές επεξεργασίας δεδομένων και ελέγχου. Το WirelessHart βασίζεται στο πρότυπο IEE 802.15.4 για χαμηλής ισχύς λειτουργία στα 2.4 GHz και είναι αξιόπιστο, ασφαλές και ενεργειακά αποδοτικό. Επίσης είναι συμβατό με όλες τις υπάρχουσες συσκευές και συστήματα. Υποστηρίζει τη δικτύωση mesh, την αναπήδηση καναλιού (channel hopping) και τα χρονικά συγχρονισμένα μηνύματα (time-synchronized messaging). Το δίκτυο επικοινωνίας είναι ασφαλές λόγω της διαχείρισης κρυπτογράφησης, επιβεβαίωσης και πιστοποίησης.

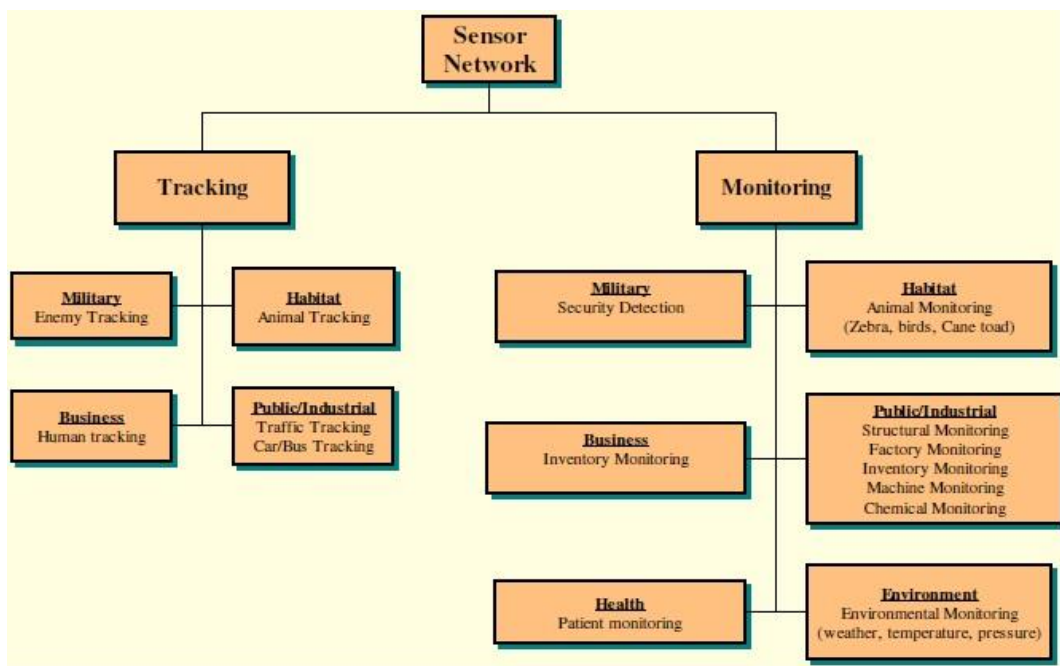
- **IETF 6LoWPAN**

Το πρότυπο IETF 6LoWPAN [28] παρέχει ένα πρωτόκολλο δικτύου ασύρματης επικοινωνίας για τη δημιουργία ενός ανεξάρτητου και κατακεκολλημένου δικτύου και χρησιμοποιείται για να διασφαλίσει την επικοινωνία σε εφαρμογές αυτοματισμών. Το IETF 6LoWPAN βασίζεται στο πρότυπο IEEE 802.15.4 για χαμηλής ισχύς λειτουργία στα 2.4 GHz, υποστηρίζει το πρωτόκολλο IPv6 Low Power Wireless Area Personal Network, βάση του οποίου το καθιστά συμβατό με πολλά δίκτυα επικοινωνιών (πρωτόκολλο IP) και μπορεί να υποστηρίξει έναν μεγάλο αριθμό συσκευών.

## 2.5 Εφαρμογές Ασύρματων Δικτύων Αισθητήρων

Οι εφαρμογές των Ασύρματων Δικτύων Αισθητήρων μπορούν να κατηγοριοποιηθούν με βάση τη λειτουργία που επιτελούν οι αισθητήριοι κόμβοι. Έτσι διακρίνονται WSN που χρησιμοποιούνται για παρακολούθηση (monitoring) και σε αυτά που χρησιμοποιούνται για ανίχνευση (tracking) [40]. Οι εφαρμογές παρακολούθησης περιλαμβάνουν την παρακολούθηση παραμέτρων πλαισίου για τυχόν αλλαγές, την παρακολούθηση στρατιωτικών στόχων, την παρακολούθηση της αυτοματοποιημένης διαδικασίας παραγωγής ενός εργοστασίου, την παρακολούθηση των σεισμικών δραστηριοτήτων μιας περιοχής και φυσικά στην βιοϊατρική επιστήμη, όπου διατάξεις όπως μετεγχειρητικά εμφυτεύματα αισθητήρων μπορούν να χρησιμοποιούνται για την παρακολούθηση της εξέλιξης της υγείας του ασθενούς. Αντίστοιχα, οι

εφαρμογές ανίχνευσης περιλαμβάνουν την ανίχνευση αντικειμένων, ζώων, ανθρώπων και οχημάτων.



Εικόνα 5. Κατηγοριοποίηση εφαρμογών WSN [40]

### 2.5.1 Στρατιωτικές Εφαρμογές

Τα ασύρματα δίκτυα αισθητήρων μπορούν να χρησιμοποιηθούν σε στρατιωτικά συστήματα [26, 40] για υπηρεσίες ανίχνευσης, επικοινωνίας, υπολογισμού και ελέγχου. Σε αυτό συνέβαλε καθοριστικά η ταχεία ανάπτυξη, η αυτο-οργάνωση καθώς και η ανοχή σφαλμάτων που τα χαρακτηρίζει. Με δεδομένο ότι τα ασύρματα δίκτυα αισθητήρων αποτελούνται από μια πυκνή διάταξη αναλώσιμων και χαμηλού κόστους κόμβων αισθητήρων, η αναμενόμενη καταστροφή κάποιων κόμβων από εχθρικές ενέργειες δε θα επηρεάσει άμεσα και σημαντικά τη στρατιωτική επιχείρηση και την επίβλεψή της. Κάποιες από τις στρατιωτικές εφαρμογές περιλαμβάνουν την επιτήρηση των στρατευμάτων, του εξοπλισμού και των πυρομαχικών των φιλικών συμμαχικών δυνάμεων, την παρακολούθηση του πεδίου μάχης, την ανίχνευση και την αναγνώριση του σημείου των εχθρικών στρατευμάτων, την εκτίμηση των απωλειών μιας μάχης και φυσικά την ανίχνευση και αναγνώριση μιας πυρηνικής, βιολογικής ή χημικής επίθεσης.



### **2.5.2 Περιβαλλοντικές Εφαρμογές**

Η ικανότητα των ασύρματων δικτύων αισθητήρων για αυτό-οργάνωση και αυτό-συντονισμό αξιοποιούνται μέσα από την υλοποίηση και τη χρήση ενός μεγάλου πλήθους περιβαλλοντικών εφαρμογών [40, 14]. Ορισμένες περιβαλλοντικές εφαρμογές περιλαμβάνουν την παρακολούθηση της κίνησης των πτηνών, μικρών ζώων και εντόμων, την παρακολούθηση των καλλιεργειών, της άρδευσης και των βοσκοτόπων, την παρακολούθηση και εξερεύνηση του πλανήτη με τη χρήση μηχανημάτων ευρείας κλίμακας, την ανίχνευση βιολογικών και χημικών κινδύνων, την παρακολούθηση της γεωργίας ακριβείας, την παρακολούθηση του θαλάσσιου περιβάλλοντος, την ανίχνευση των δασικών πυρκαγιών, την ανίχνευση πλημμυρών, την ανίχνευση ρύπανσης καθώς και την επιτήρηση της μετεωρολογικής ή της γεωφυσικής έρευνας.

### **2.5.3 Εφαρμογές Υγείας**

Η ταχύτατη ανάπτυξη των εμφυτευόμενων έξυπνων αισθητήρων και συσκευών στην επιστήμη της βιοϊατρικής έκανε εφικτή τη χρήση των ασύρματων δικτύων αισθητήρων σε διάφορες εφαρμογές της υγείας [40, 4]. Ορισμένες εφαρμογές υγείας περιλαμβάνουν την παρακολούθηση και την παροχή βοήθειας σε άτομα με ειδικές ανάγκες, την παρακολούθηση και διάγνωση ενός διασωληνωμένου ασθενή, την επιτήρηση και διαχείριση των φαρμάκων στα νοσοκομεία, την τηλεπαρακολούθηση των ανθρώπινων φυσιολογικών στοιχείων και δεδομένων [5] αλλά και την ανίχνευση και παρακολούθηση της κίνησης των γιατρών και των ασθενών μέσα στο χώρο του νοσοκομείου.

### **2.5.4 Οικιακές Εφαρμογές**

Καθώς η τεχνολογία εξελίσσεται ραγδαία, οι ασύρματοι έξυπνοι κόμβοι αισθητήρων καθώς και οι ενεργοποιητές μπορούν να ενσωματωθούν μέσα στα κυκλώματα διαφόρων οικιακών συσκευών [40] όπως είναι η ηλεκτρική σκούπα, ο φούρνος μικροκυμάτων, το ψυγείο, η συσκευή αναπαραγωγής DVD, κι άλλα. Οι κόμβοι των αισθητήρων που περιέχονται μέσα σε μια οικιακή συσκευή μπορεί να αλληλοεπιδράσουν μεταξύ τους και με κάποιο εξωτερικό απομακρυσμένο δίκτυο μέσω μιας οικιακής σύνδεσης στο διαδίκτυο ή κάποιας δορυφορικής σύνδεσης. Έτσι

επιτρέπει στους τελικούς χρήστες να μπορούν να διαχειριστούν πιο εύκολα τις οικιακές συσκευές και τοπικά αλλά και απομακρυσμένα. Συνεπώς, τα ασύρματα δίκτυα αισθητήρων επιτρέπουν την διασύνδεση διαφόρων συσκευών σε κατοικημένες περιοχές έχοντας ως αποτέλεσμα τον βολικό και πρακτικό έλεγχο των εφαρμογών από το σπίτι, επιτρέποντας την υλοποίηση της ιδέας του έξυπνου σπιτιού (smart home) [21].

## **2.5.5 Βιομηχανικές Εφαρμογές**

Τα δίκτυα των παραδοσιακών ενσύρματων αισθητήρων χρησιμοποιούνται ήδη για πολύ καιρό σε βιομηχανικούς τομείς, όπως στις εφαρμογές της βιομηχανικής ανίχνευσης και ελέγχου και στις εφαρμογές αυτοματοποίησης και ελέγχου πρόσβασης των κτιρίων. Ωστόσο, το μεγάλο κόστος της εγκατάστασης των συγκεκριμένων αισθητήρων περιόρισε σημαντικά την πρακτική δυνατότητα εφαρμογής αυτών των συστημάτων [40]. Ακόμη και στην περίπτωση που θα παραμεριζόταν το κόστος και θα προχωρούσε η ανάπτυξη ενός συστήματος αισθητήρων στη βιομηχανία, η αναβάθμισή του αργότερα θα ήταν απαγορευτική καθώς θα κόστιζε σχεδόν όσο η ανάπτυξη ενός νέου συστήματος. Επιπρόσθετα, σε συστήματα παρακολούθησης αισθητήρων έχει χρησιμοποιηθεί εξίσου σε βιομηχανικές εφαρμογές και η χειροκίνητη και μη – αυτόματη παρακολούθηση από ανθρώπινο δυναμικό με σκοπό την προληπτική συντήρηση του συστήματος. Η χειροκίνητη παρακολούθηση γενικά εκτελείται από εξειδικευμένο προσωπικό με τη χρήση μικρών συσκευών ανάλυσης. Τα δεδομένα των οποίων συλλέγονται από έναν κεντρικό σταθμό για μετέπειτα ανάλυση. Έτσι, το μειονέκτημα είναι η περιορισμένη ακρίβεια και η απαίτηση χρήσης προσωπικού σε σύγκριση με τα συστήματα των αισθητήρων όπου συναντάμε το υψηλό κόστος εγκατάστασης. Αντίθετα, τα ασύρματα δίκτυα αισθητήρων αποτελούν μια πολλά υποσχόμενη εναλλακτική επιλογή για αυτά τα συστήματα εξαιτίας της ευκολίας στην ανάπτυξη του συστήματος, της υψηλής ανάλυσης και της υψηλής ακρίβειας που παρέχεται μέσα από την ενεργειακή αυτονομία των ασύρματων μονάδων επικοινωνίας. Ορισμένες εμπορικές βιομηχανικές εφαρμογές περιλαμβάνουν την παρακολούθηση της φθοράς των υλικών, τη διαχείριση των αποθεμάτων, την παρακολούθηση της ποιότητας των προϊόντων, τον περιβαλλοντικό έλεγχο των γραφείων ενός κτιρίου, τον έλεγχο και την καθοδήγηση ενός μηχανήματος ρομπότ σε ένα αυτοματοποιημένο περιβάλλον παραγωγής καθώς και την ανίχνευση και παρακολούθηση ενός οχήματος.

## 2.5.6 Άλλες εμπορικές Εφαρμογές

Άλλες εμπορικές εφαρμογές όπου συναντώνται τα WSNs, σχετίζονται με αυτόματους ασύρματους μετρητές (AMR), οι οποίοι συλλέγουν δεδομένα για τις παροχές ρεύματος, νερού και αερίου. Η ασύρματη συγκέντρωση των δεδομένων για τα συστήματα έκδοσης λογαριασμών είναι αρκετά αποδοτική σε σχέση με το κόστος του συστήματος. Επίσης, υπάρχουν εταιρείες που παράγουν χαμηλού κόστους και χαμηλής ισχύος ολοκληρωμένα κυκλώματα και πομποδέκτες ειδικά για την υλοποίηση των ασύρματων εφαρμογών AMR. Τα ασύρματα δίκτυα αισθητήρων έχουν σημαντικό αντίκτυπο και σε ένα άλλο πεδίο, το οποίο αφορά εφαρμογές θέρμανσης, εξαερισμού και κλιματισμού (HVAC unit). Σε μεγάλα και εμπορικά κτίρια, είναι πολύ σύνηθες φαινόμενο ο έλεγχος πολλαπλών χώρων από μια μόνο μονάδα HVAC. Συνεπώς, τα συγκεκριμένα συστήματα ελέγχονται από έναν μόνο αισθητήρα τοποθετημένο σε έναν από τους χώρους. Ωστόσο, το χαμηλό κόστος της ασύρματης τεχνολογίας αισθητήρων, δίνει τη δυνατότητα να αντικατασταθεί ο αισθητήρας με ένα δίκτυο αισθητήρων, όπου θα υπάρχει τουλάχιστον ένας τοποθετημένος ανά χώρο [33].

# Κεφάλαιο 3

## Γεωργία ακριβείας

Τα τελευταία χρόνια η ανάπτυξη της τεχνολογίας, και συγκεκριμένα των συστημάτων διάχυτου υπολογισμού, έχει σταθεί αρωγός στην ολοένα κι αυξανόμενη εισαγωγή των τεχνολογιών πληροφορίας κι επικοινωνιών σε κάθε τομέα της ανθρώπινης δραστηριότητας. Η ενσωμάτωση τέτοιων τεχνολογιών, που αρχικά αναπτύχθηκαν για τη βιομηχανία, όπως τα συστήματα εντοπισμού θέσης, τα ενσωματωμένα συστήματα και τα δίκτυα αισθητήρων, και η κινητή υπολογιστική στο χώρο της γεωργικής παραγωγής οδήγησε στην εξέλιξη της γεωργίας ακριβείας (precision agriculture) η οποία έχει ως στόχο την αύξηση της παραγωγής του αγρότη και την προστασία του περιβάλλοντος [41]. Η τεχνολογία παρέχει στη γεωργία ακριβείας τα μέσα για την παρακολούθηση, την πρόσβαση και τον έλεγχο των καλλιεργειών και των νέων πρακτικών [9].

### 3.1 Εφαρμογές WSNs στη Γεωργία Ακριβείας

Οι εφαρμογές των WSNs στη γεωργία είναι πολλές και μπορούν να κατηγοριοποιηθούν ως εξής [12]:

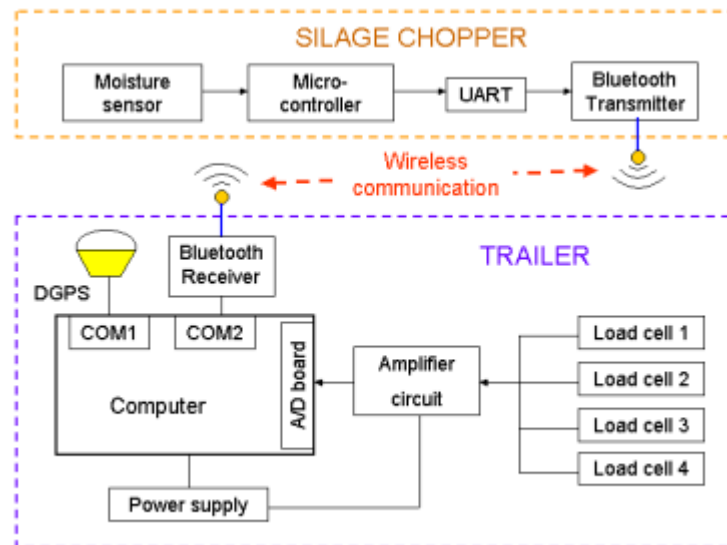
- Συστήματα συλλογής δεδομένων (spatial data collection)
- Συστήματα ελέγχου άρδευσης (irrigation control model)
- Συστήματα παροχής πληροφοριών στους αγρότες (Arrangement for supplying information to farmers)
- Συστήματα ελέγχου θερμοκηπίου (Greenhouse parameter control system).

Ενδεικτικές εφαρμογές σε κάθε μια από τις παραπάνω κατηγορίες είναι [130]:

#### 3.1.1 Εφαρμογές σε συστήματα συλλογής δεδομένων

Το 2002 οι Lee et al. [27] ανέπτυξαν το σύστημα Silage Yield Monitoring System. Ένα σύστημα παρακολούθησης της υγρασίας του χόρτου (που προορίζεται για ζωοτροφή) σε μια αγροτική

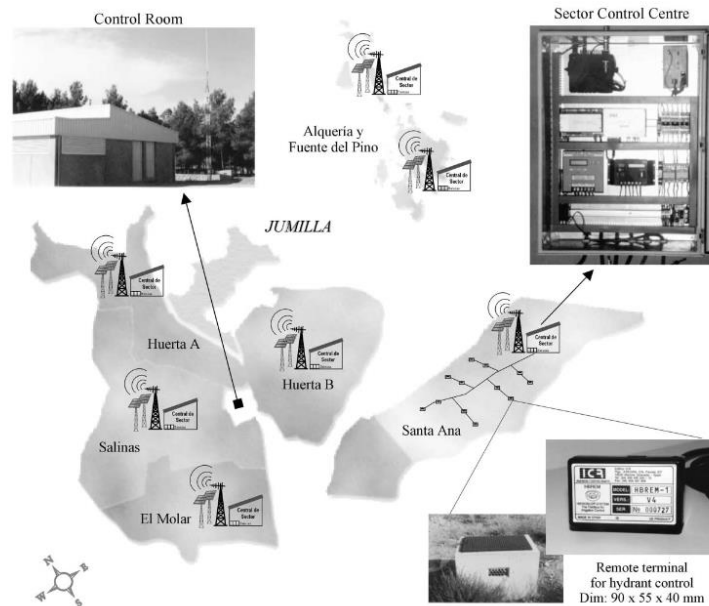
περιοχή. Τα δεδομένα υγρασίας, που επηρεάζει τη δυνατότητα αποθήκευσης και την ποιότητα της τροφής, λαμβάνονταν από αισθητήρες υγρασίας κατά τη συγκομιδή κι αποστέλλονταν με χρήση Bluetooth. Το κάθε δείγμα συνοδεύονταν από στοιχεία θέσης τα οποία λαμβάνονταν με χρήση DGPS



Εικόνα 6. Silage Yield Monitoring System [27]

### 3.1.2 Εφαρμογές σε συστήματα ελέγχου άρδευσης

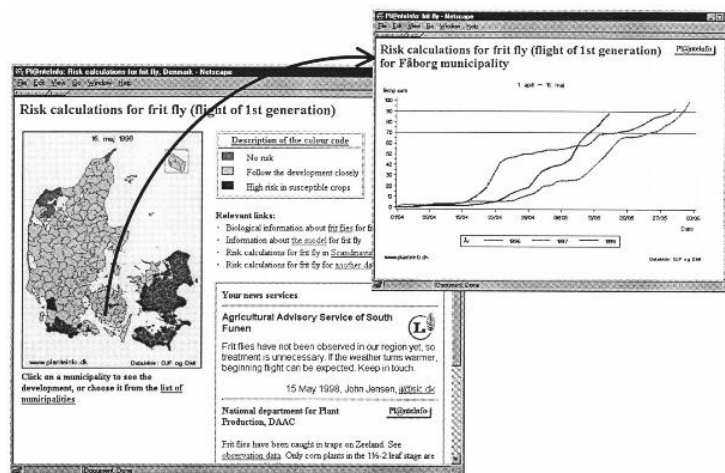
Το 2001 οι Damas et al. ανέπτυξαν το σύστημα HydroBus [139] μέσω σύμπραξης του πανεπιστημίου της Γρανάδα με μία ιδιωτική εταιρία με σκοπό τη διαχείριση της άρδευσης αρχικά σε μια περιοχή 1500 εκταρίων στην Ισπανία. Η περιοχή είχε χωριστεί συνολικά σε 7 υποπεριοχές, όπου είχαν εγκατασταθεί 1850 κρουνοί. Κάθε υποπεριοχή ελέγχονταν από τον κατά τόπους HB (HydroBus) Controller. Οι controllers με τη σειρά τους μετέφεραν τα δεδομένα στο κεντρικό σημείο ελέγχου. Αντίστοιχα συστήματα άρδευσης μπορούν να επιτύχουν εξοικονόμηση νερού κατά 30-60% [16].



Εικόνα 7. Σύστημα HydroBus [139]

### 3.1.3 Εφαρμογές σε συστήματα παροχής πληροφοριών

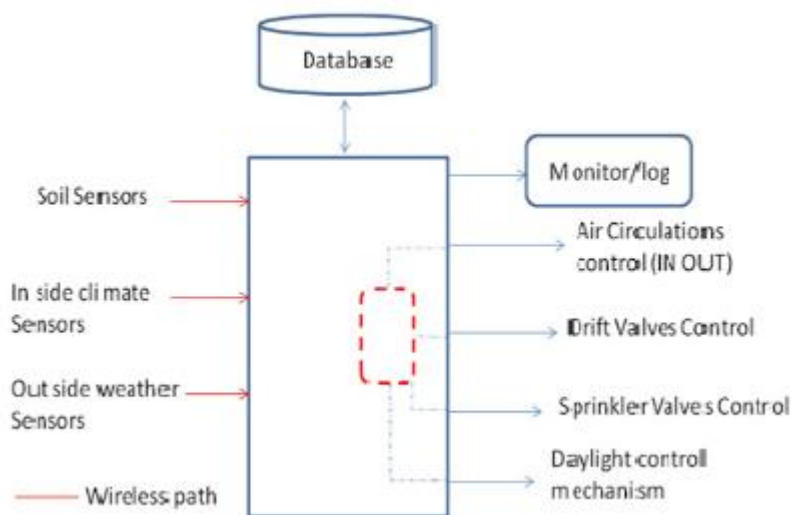
Το σύστημα Pl@nteInfo που αναπτύχθηκε από τους Jensen et al. [23] αποτελείται από ένα σύστημα λήψης αποφάσεων το οποίο αφού λάβει δεδομένα παρουσιάζει στους αγρότες πληροφορίες που τους ενδιαφέρουν, μέσω web pages. Αν και το συγκεκριμένο σύστημα δεν ήταν διασυνδεδεμένο με δίκτυα αισθητήρων, πληροφορίες από τα τελευταία θα μπορούσαν να αξιοποιηθούν.



Εικόνα 8. Σύστημα Pl@nteInfo [23]

### 3.1.4 Εφαρμογές σε συστήματα ελέγχου θερμοκηπίου

Οι Chaudhary et al εργάστηκαν σε μια ερευνητική εφαρμογή που αφορούσε τη δημιουργία ενός πρότυπου θερμοκηπίου με σκοπό τη σχεδίαση κι ανάπτυξη ενός WSN για τον έλεγχο και την παρακολούθηση των παραμέτρων. Αναζητήθηκε ο συνολικός αριθμός αισθητήρων και ενεργοποιητών, κάτι που εξαρτάται από το μέγεθος του θερμοκηπίου. Συνολικά χρησιμοποιήθηκαν τρεις τύποι αισθητήρων. Αισθητήρες θερμοκρασίας εξωτερικού περιβάλλοντος, αισθητήρες θερμοκρασίας εσωτερικού περιβάλλοντος και αισθητήρες εδάφους. Οι αισθητήρες εδάφους μπορούν επίσης να ελέγχουν το σύστημα άρδευσης και τοποθετούνται ανά δύο μέτρα μέσα στην γεωγραφική περιοχή μελέτης. Στο σύστημα αυτό οι έλεγχοι εκτελούνται με ασύγχρονο τρόπο και για την μετάδοση των δεδομένων χρησιμοποιείται η μέθοδος της δειγματοληψίας. Κάθε αισθητήρας μεταδίδει τα δεδομένα εφόσον η διαφορά μεταξύ της πρόσφατης τιμής μέτρησης και της προηγούμενης είναι μεγαλύτερη από ένα συγκεκριμένο τυπικό όριο [12].



Εικόνα 9. GreeHouse Block Diagram [140]

## 3.2 Συνθήκες Θερμοκηπίου

Όπως έχει ήδη αναφερθεί, με τη χρήση ασύρματων δικτύων αισθητήρων είναι δυνατή η λήψη ενός συνόλου παραμέτρων πλαισίου (υπό τη μορφή δεδομένων) από πολλά σημεία ενός καλλιεργήσιμου χώρου. Συγκεκριμένα, είναι δυνατή η παρακολούθηση και καταγραφή

παραμέτρων που σχετίζονται με το μικροκλίμα χώρου και επηρεάζουν τις καλλιέργειες. Τέτοιες παράμετροι είναι η θερμοκρασία, η υγρασία, το ηλιακό φως, η πίεση, η οξύτητα του εδάφους, τα επίπεδα διοξειδίου του άνθρακα στον αέρα (CO<sub>2</sub>) καθώς και η ταχύτητα και η κατεύθυνση του ανέμου. Καθώς η απόδοση της καλλιέργειας επηρεάζεται κυρίως από τις μη ελεγχόμενες αλλαγές στις συνθήκες, την ποσότητα του νερού και τα λιπάσματα που χρησιμοποιούνται κατά την άρδευση, η χρήση του θερμοκηπίου (greenhouse) θεωρείται μια ιδανική εναλλακτική στις ανοικτές καλλιέργειες καθώς το περιβάλλον εντός του είναι πλήρως ελεγχόμενο. Στον ελεγχόμενο αυτό χώρο είναι δυνατή η πρώιμη ανίχνευση ασθενειών στα φυτά και υπάρχει μια συνολική εκτίμηση της κατάστασης της παραγωγής.

### 3.2.1 Θερμοκρασία

Από τους κλιματικούς παράγοντες η θερμοκρασία είναι ο παράγοντας που έχει την πιο σημαντική και πολύπλοκη επίδραση στην ανάπτυξη και την παραγωγή των φυτών. Η ηλιακή ακτινοβολία αποτελεί μια φυσική πηγή θερμότητας του θερμοκηπίου, ευνοεί τη διαδικασία της φωτοσύνθεσης των φυτών και διαφοροποιείται ανάλογα με την εποχή. Ανάλογα με το είδος της καλλιέργειας θα πρέπει να υπάρχει σταθερή θερμοκρασία στο θερμοκήπιο κατά τη διάρκεια της ημέρας και της νύχτας. Η θερμοκρασία τη νύχτα θα πρέπει να είναι πάντα χαμηλότερη και για να επιτευχθεί μπορούν να χρησιμοποιηθούν διάφορες τεχνικές, όπως ο φυσικός εξαερισμός του χώρου, η σκίαση και η εγκατάσταση ενός συστήματος δροσισμού.

### 3.2.2 Σχετική Υγρασία

Ο αέρας σε φυσιολογικές συνθήκες περιέχει μια μικρή ποσότητα υγρασίας, δηλαδή κάποια γραμμάρια νερού ανά κυβικό μέτρο αέρα (g/m<sup>3</sup>). Η ποσότητα των υδρατμών στον αέρα μπορεί να φτάσει μία μέγιστη τιμή. Στην τιμή αυτή, για μια δεδομένη θερμοκρασία, ο αέρας είναι κορεσμένος από υδρατμούς ενώ στην περίπτωση μικρότερης συγκέντρωσης ο αέρας είναι ακόρεστος. Η λόγος της ποσότητας των υδρατμών  $e$  στον αέρα προς τη μέγιστη τιμή  $e_s$  (κορεσμού) ονομάζεται σχετική υγρασία (Relative Humidity - RH) κι αναφέρεται ως ποσοστό επί τοις εκατό [24].

$$RH = \frac{e}{e_s} 100 \quad (1)$$



Οι περισσότερες καλλιέργειες αναπτύσσονται σωστά όταν τα ποσοστά υγρασίας στο θερμοκήπιο κυμαίνονται μεταξύ 60-80%. Γενικά, τα υψηλά επίπεδα υγρασίας, πάνω από 80%, ευνοούν την ανάπτυξη ασθενειών στα φυτά, ενώ τα επίπεδα υγρασίας κάτω από 60% δεν ευνοούν την απόδοση καρπών. Στα θερμοκήπια, κατά τη διάρκεια της ημέρας, η σχετική υγρασία είναι συνήθως 25 – 70% ενώ κατά τη διάρκεια της νύχτας αυξάνεται και φτάνει 90 – 100%. Ειδικά το χειμώνα, σε βροχερές περιόδους, η σχετική υγρασία μπορεί παραμείνει 100% για αρκετές μέρες και νύχτες. Για να αποφευχθούν τα υψηλά επίπεδα υγρασίας στο θερμοκήπιο θα πρέπει τα φυτά να μην συνωστίζονται, να υπάρχει μια επαρκής απόσταση ανάμεσα τους και να χρησιμοποιείται ειδικός εξοπλισμός για τη συνεχή ανανέωση και κυκλοφορία του αέρα.

### 3.2.3 Σημείο Δρόσου

Σημείο δρόσου (Dew Point) είναι το σημείο της θερμοκρασίας τα κάτω από το οποίο όταν οι υδρατμοί ψυχθούν υπό σταθερή πίεση ο αέρας γίνεται κορεσμένος και η σχετική υγρασία έχει τιμή 100%.

$$e_s(t_d) = e(t) \quad (2)$$

Η θερμοκρασία δρόσου μπορεί να είναι οποιαδήποτε θερμοκρασία πάνω από 0°C και εξαρτάται από την απόλυτη υγρασία.

Η εμπειρική φόρμουλα που δίνει την πίεση των υδρατμών σε συνθήκες κορεσμού, γνωστή ως Magnus formula [24] είναι:

$$e_s = C \exp \frac{At}{B+t}, \quad (3)$$

με τιμές για τις σταθερές A, B και C: A=17.625, B=243.04°C και C=610.94 [24]

Αντικαθιστώντας τη σχέση (3) στη σχέση (2) έχουμε:

$$t_d = \frac{B \ln \left( \frac{e}{C} \right)}{A - \ln \left( \frac{e}{C} \right)} \quad (4)$$

Και από τις σχέσεις (3) και (4), απαλείφουμε το  $e$  κι έχουμε:

$$t_d = \frac{B \left( \ln \left( \frac{RH}{100} \right) + \frac{At}{B+t} \right)}{A - \ln \left( \frac{RH}{100} \right) - \frac{At}{B+t}}$$

Άρα με τη σχέση αυτή, με γνωστά τα  $A$ ,  $B$  τη σχετική υγρασία  $RH$  και τη θερμοκρασία χώρου  $t$  σε βαθμούς Celsius, είναι δυνατός ο υπολογισμός της τιμής θερμοκρασίας δρόσου  $t_d$  κάτω από την οποία ο αέρας θα κορεστεί. Γνωρίζοντας την τιμή αυτή, είναι δυνατή η ενεργοποίηση κατάλληλων μηχανισμών έτσι ώστε να μεταβληθούν οι συνθήκες ώστε να αποφευχθεί ο κορεσμός.

Στη συνέχεια, στο Κεφάλαιο 4 θα αναφερθούν οι τρόποι με τους οποίους είναι δυνατή η λήψη των τιμών  $RH$  και  $t$  και στο Κεφάλαιο 5 θα παρουσιαστεί ένα ολοκληρωμένο σύστημα έγκαιρης προειδοποίησης συνθήκης κορεσμού, που θα βασίζεται σε ένα ασύρματο δίκτυο αισθητήρων και μια κατάλληλη εφαρμογή λογισμικού.

# Κεφάλαιο 4

## Αισθητήρες Μέτρησης για τον Υπολογισμό Σημείου Δρόσου

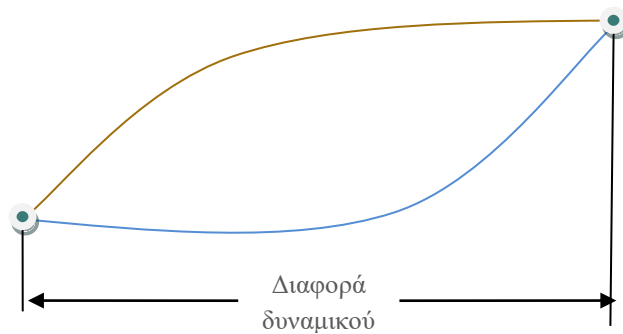
Στο προηγούμενο κεφάλαιο παρουσιάστηκε συνοπτικά η έννοια της γεωργίας ακριβείας και παρατέθηκαν οι απαραίτητοι παράμετροι πλαισίου που πρέπει να ελέγχονται σε ένα θερμοκήπιο με σκοπό την αποφυγή συγκέντρωσης δρόσου που ευνοεί την ανάπτυξη ασθενειών. Στη συνέχεια θα διερευνηθεί ο τρόπος μέτρησης των παραμέτρων αυτών με χρήση κατάλληλων αισθητήρων και θα εξεταστεί η δυνατότητα και ο τρόπος υλοποίησης ενός αισθητήριου κόμβου χαμηλού κόστους, βασισμένου σε ανοικτό υλικό, ο οποίος θα έχει τη δυνατότητα να συλλέγει και να αποστέλλει ασύρματα τα στοιχεία θερμοκρασίας και υγρασίας εντός του θερμοκηπίου.

### 4.1 Αισθητήρες Θερμοκρασίας

Η μέτρηση της θερμοκρασίας μπορεί να επιτευχθεί με τη χρήση διαφόρων τύπων αισθητήρων. Οι πιο διαδεδομένοι αισθητήρες μέτρησης [37] είναι τα θερμοζεύγη (thermocouples), οι αντιστάτες λευκόχρυσου (Platinum Resistance Thermometers – PRT) και τα θερμίστορ (thermistor). Οι αισθητήρες αυτοί παρουσιάζουν διαφοροποιήσεις τόσο ως προς τον τρόπο κατασκευής και λειτουργίας όσο και στην ακρίβεια αλλά και το κόστος [37].

- **Thermocouple**

Τα θερμοζεύγη αποτελούν ίσως το πιο διαδεδομένο αισθητήριο όργανο μέτρησης θερμοκρασίας. Η αρχή λειτουργίας τους βασίζεται στο φαινόμενο Seebeck κατά το οποίο όταν ενωθούν τα άκρα δύο μεταλλικών αγωγών διαφορετικών χαρακτηριστικών, τότε το κύκλωμα διαρρέεται από ρεύμα αν οι δύο επαφές (επαφή μέτρησης κι επαφή αναφοράς) βρίσκονται σε διαφορετική θερμοκρασία. Η θερμοκρασία μπορεί να υπολογιστεί από τη μέτρηση της διαφοράς δυναμικού μεταξύ των δύο επαφών, σε συνδυασμό με τα χαρακτηριστικά μετάλλων του θερμοζεύγους.



**Εικόνα 10.** Λειτουργία θερμοζεύγους

- **PRT**

Η αντίσταση ενός μετάλλου μεταβάλλεται με τη μεταβολή της θερμοκρασίας. Η αντίσταση  $R$  του μετάλλου στη θερμοκρασία  $T$  ( $R_T$ ) είναι συνάρτηση της θερμοκρασίας και δίδεται ως:

$$R_T = R_0 (1 + \alpha T + \beta T^2 + \gamma T^3 + \dots) ,$$

Όπου  $R_0$  η αντίσταση του μετάλλου σε θερμοκρασία  $0^\circ\text{C}$  και  $\alpha, \beta, \gamma$  διηλεκτρικές σταθερές που εξαρτώνται από το μέταλλο που χρησιμοποιείται.

Αν και ο λευκόχρυσος παρουσιάζει ιδιαίτερη γραμμικότητα στη μεταβολή της αντίστασής του σε σχέση με τη θερμοκρασία και αντοχή στο χρόνο, το αυξημένο κόστος έχει οδηγήσει στη δημιουργία αντίστοιχων αντιστατών όπου γίνεται χρήση άλλων μετάλλων όπως το νικέλιο και ο χαλκός.

Οι αισθητήρες PRT χρησιμοποιούνται συνήθως σε εφαρμογές όπου απαιτείται ακρίβεια και χαρακτηρίζονται από υψηλό κόστος.

- **Thermistor**

Το θερμίστορ, αποτελείται από ένα ημιαγώγιμο υλικό που μεταβάλλει την αντίστασή του σε σχέση με τη θερμοκρασία. Συγκεκριμένα η αντίσταση μειώνεται όσο η θερμοκρασία αυξάνεται (NTC Thermistors) ή το αντίθετο (PTC Thermistors). Η γενική, μη γραμμική, σχέση που συνδέει την αντίσταση σε θερμοκρασία  $T$  με τη θερμοκρασία  $T$  είναι [37]:

$$R_T = A \exp\left(\frac{\beta}{T}\right),$$

Όπου A και β σταθερές που χαρακτηρίζουν κάθε θερμίστορ και δίδονται από τον κατασκευαστή.

Τα χαρακτηριστικά των παραπάνω αισθητηρίων συνοψίζονται στον πίνακα 2 [35]:

Χαρακτηριστικό	Θερμοζεύγος	PRT	Θερμίστορ
Εύρος θερμοκρασιών	-200 ως 2000	-250 ως 850	-100 ως 300
Ακρίβεια	Χαμηλή (1°C)	Πολύ Υψηλή (0,03°C)	Μέση (0,1°C)
Γραμμικότητα	Μέση	Υψηλή	Χαμηλή
Απόκριση	Γρήγορη	Αργή	Μέση
Κόστος	Χαμηλό	Υψηλό	Χαμηλό
Ανοχή στο θόρυβο	Χαμηλή	Μέση	Υψηλή
Διάρκεια ζωής	Χαμηλή	Υψηλή	Μέση
Κόστος μέτρησης	Μέσο	Υψηλό	Χαμηλό

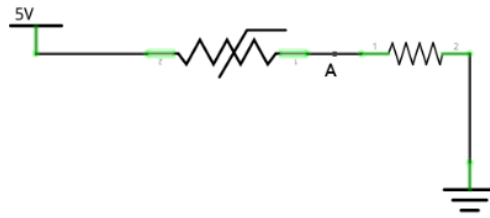
**Πίνακας 2** - Σύγκριση αισθητήρων θερμοκρασίας [35]

Παρατηρούμε, ότι βάση των παραπάνω, η επιλογή με το χαμηλότερο συνολικό κόστος κτήσης και μέτρησης σε σχέση με την ακρίβεια μέτρησης είναι το θερμίστορ, το οποίο και θα επιλεγεί για τη μέτρηση της θερμοκρασίας των φυτών του θερμοκηπίου.

#### 4.1.1 Μέτρηση θερμοκρασίας με χρήση θερμίστορ

Το θερμίστορ, όπως προαναφέρθηκε είναι ημιαγωγός που έχει τη δυνατότητα να μεταβάλλει την αντίστασή του σε σχέση με τη θερμοκρασία. Έτσι, γνωρίζοντας την αντίσταση στη θερμοκρασία αναφοράς, μετρώντας τη νέα αντίσταση το άκρο του, μπορεί να υπολογιστεί η θερμοκρασιακή μεταβολή που οδήγησε στη μεταβολή της αντίστασης, άρα η θερμοκρασία χώρου.

Μια τυπική διάταξη θερμίστορ είναι:



**Εικόνα 11.** Τυπική διάταξη θερμίστορ

Η διάταξη αυτή, γνωστή ως διαιρέτης τάσης, με μεταβλητή αντίσταση το θερμίστορ, παράγει μια τάση εξόδου  $V_{out}$  στο σημείο A ίση με:

$$V_{out} = V_{in} \frac{R}{R + R_1}$$

Με γνωστή την τάση  $V_{in}$ , την τιμή της αντίστασης  $R_1$  και μετρώντας την τάση εξόδου, είναι δυνατή η εύρεση της αντίστασης  $R$  του θερμίστορ:

$$R = R_1 \frac{V_{in} - V_{out}}{V_{out}}$$

Τέλος, με τη βοήθεια της σχέσης Steinhart-Hart [30], μπορεί να υπολογιστεί η θερμοκρασία  $T$ , δεδομένων των σταθερών  $A$ ,  $B$  και  $C$ , από τα τεχνικά χαρακτηριστικά του θερμίστορ:

$$\frac{1}{T} = A + B \ln R + C \ln^3 R$$

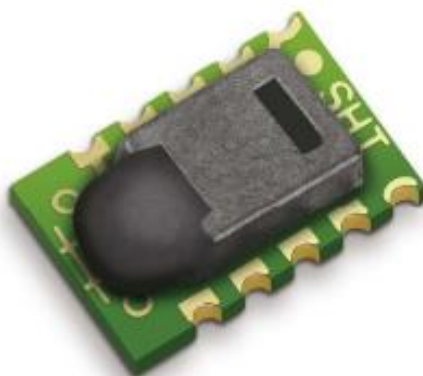
Η θερμοκρασία  $T$  δίδεται σε βαθμούς Kelvin, οι οποίοι μπορούν να μετατραπούν σε βαθμούς Celsius προσθέτοντας την τιμή 273,15 ( $C=T+273,15$ ).

## 4.2 Υγρασία και σημείο δρόσου

Η υγρασία του χώρου είναι σημαντικός παράγοντας που πρέπει να ελέγχεται σε εφαρμογές γεωργίας ακριβείας σε θερμοκήπια. Για τη μέτρησή της μπορούν να χρησιμοποιηθούν ειδικές διατάξεις που περιλαμβάνουν αισθητήρες υγρασίας, με πιο διαδεδομένες τη σειρά SHT1x που παρασκευάζεται από την εταιρία Sensirion [32] και τη σειρά DHTxx, με χαρακτηριστικότερο τον αισθητήρα υγρασίας-θερμοκρασίας DHT11 της Aosong (Guangzhou) Electronics, [6].

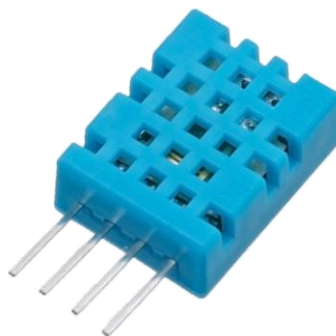
Η σειρά SHT1x αποτελείται από 3 προϊόντα αισθητήρων. Τους SHT10, SHT11 και SHT15. Ο SHT10 είναι αισθητήρας χαμηλού κόστους με απόκλιση 4,5%. Ο SHT11, που είναι ο πιο διαδεδομένος παρουσιάζει απόκλιση 3% και τέλος ο υψηλής ακριβείας SHT15 εμφανίζει απόκλιση μόλις 2%.

Οι αισθητήρες της σειράς SHT1x είναι μικροί σε μέγεθος, παρουσιάζουν μικρή κατανάλωση 80mW και διαθέτουν ψηφιακή σειριακή διεπαφή (serial interface).



**Εικόνα 12.** Αισθητήρες της σειράς SHT1x

Η σειρά DHTxx, αποτελείται από τους αισθητήρες DHT11 και DHT22. Οι αισθητήρες αυτοί παρουσιάζουν μικρότερη ακρίβεια (5% και 2-5% αντίστοιχα), μεγαλύτερη κατανάλωση (2mA) και μεγαλύτερο μέγεθος σε σχέση με την οικογένεια των SHT1x. Επίσης διαθέτουν ψηφιακή διεπαφή αλλά μικρότερη συχνότητα δειγματοληψίας (1Hz για τον DHT11). Στα υπέρ τους το πολύ μικρότερο κόστος που τους καθιστά ικανούς ανταγωνιστές σε εφαρμογές όπου η καθυστέρηση λήψης δεδομένων δεν είναι απαγορευτικός παράγοντας.



**Εικόνα 13.** Αισθητήρας DHT11

Και οι δύο οικογένειες αισθητήρων, έχουν την ικανότητα να μετρούν υγρασία και θερμοκρασία. Έτσι με τη χρήση των μετρήσεων των αισθητήρων αυτών μπορεί να υπολογιστεί το σημείο δρόσου, με χρήση της μεθόδου Magnus, η οποία παρουσιάστηκε στο Κεφάλαιο 3 και είναι:

$$t_d = \frac{B \left( \ln \left( \frac{RH}{100} \right) + \frac{At}{B+t} \right)}{A - \ln \left( \frac{RH}{100} \right) - \frac{At}{B+t}}$$



# Κεφάλαιο 5

## Υλοποίηση Συστήματος

Η προστασία των φυτών του θερμοκηπίου από περιπτώσεις κορεσμού και δημιουργίας δρόσου, απαιτεί τον συνεχή έλεγχο των παραμέτρων σχετικής υγρασίας και θερμοκρασίας χώρου αλλά και της θερμοκρασίας των φυτών, έτσι ώστε στην περίπτωση κατά την οποία παρατηρηθεί το φαινόμενο πτώσης της θερμοκρασίας σε κάποιο φυτό κοντά στο σημείο δρόσου, να υπάρξει η δυνατότητα προειδοποίησης ή αυτόματης ενέργειας για τη ρύθμιση των συνθηκών (άνοιγμα εξαερισμού, θέρμανση χώρου κοκ)

Η λήψη των τιμών σχετικής υγρασίας και θερμοκρασιών απαιτεί κατάλληλο αισθητήριο κόμβο, ο οποίος θα είναι αξιόπιστος, θα έχει τη δυνατότητα παραμετροποίησης και άμεσης δικτύωσης και το κόστος του δε θα είναι υψηλό.

Στο παρόν κεφάλαιο, στόχος είναι η υλοποίηση κατάλληλων ασύρματων κόμβων για τη συλλογή και αποστολή δεδομένων, καθώς και η ανάπτυξη εφαρμογής λήψης, αποθήκευσης και αξιοποίησης των δεδομένων αυτών.

### 5.1 Αισθητήριои κόμβοι

Οι αισθητήριои κόμβοι αποτελούν τη βασική διάταξη του συστήματος η οποία θα αναλάβει τη συγκέντρωση και την αποστολή των δεδομένων θερμοκρασίας και υγρασίας. Οι κόμβοι θα αποτελούνται από:

- Σύστημα μικροελεγκτή που υλοποιείται στην πλατφόρμα Arduino.
- Ψηφιακό αισθητήρα υγρασίας - θερμοκρασίας ο οποίος θα παρέχει τις τιμές για τον υπολογισμό του σημείου δρόσου.
- Αισθητήρες βασισμένους σε NTC Thermistors για τον υπολογισμό της θερμοκρασίας στα φύλλα των φυτών, ώστε να εντοπίζεται τυχόν προσέγγιση της θερμοκρασίας δρόσου.

## 5.1.1 Arduino

Το Arduino [7] είναι μια ανοιχτή, εύκολη στη χρήση, πλατφόρμα υλικού και λογισμικού. Η αρχική ιδέα γεννήθηκε στο Interaction Design Institute της πόλης Ivrea στη Βόρεια Ιταλία. Στόχος του έργου ήταν η δημιουργία μιας απλής πλατφόρμας προτυποποίησης (prototyping platform) για τη χρήση από φοιτητές χωρίς εξειδικευμένες γνώσεις ηλεκτρονικής [23]. Η πλατφόρμα αποτελείται από:

- **Το υλικό**

Το υλικό (hardware) μέρος του Arduino συντίθεται από μια μητρική πλακέτα η οποία φιλοξενεί ένα μικροελεγκτή (microcontroller), διάταξη μνήμης καθώς και αναλογικές και ψηφιακές εισόδους και εξόδους. Επίσης διαθέτει κύκλωμα τροφοδοσίας και σειριακή διεπαφή USB (Universal Serial Bus) που χρησιμοποιείται για την επικοινωνία με το λογισμικό ανάπτυξης εφαρμογών ή τη διασύνδεση κι ανταλλαγή δεδομένων με άλλα υπολογιστικά συστήματα. Ένα βασικό και καινοτόμο χαρακτηριστικό του Arduino είναι πως ο μικροελεγκτής δεν προγραμματίζεται μέσω κάποιου ειδικού programmer με χρήση μιας γλώσσας όπως η assembly, αλλά παρέχεται με εγκατεστημένο κατάλληλο boot loader ο οποίος εκκινεί το Arduino και στη συνέχεια εκτελεί το πρόγραμμα που ο χρήστης τοποθετεί στη μνήμη flash του συστήματος. Από τη στιγμή που η πλατφόρμα διαδόθηκε στην ευρύτερη διεθνή κοινότητα, σε συνδυασμό με την ανοικτότητά της που επιτρέπει στον καθένα τη χρήση, την προσαρμογή και την τροποποίηση, έχουν υλοποιηθεί διαφορετικές (γενικού σκοπού ή πιο εξειδικευμένες) υλοποιήσεις του υλικού, όπως φαίνεται στον Πίνακα 3.

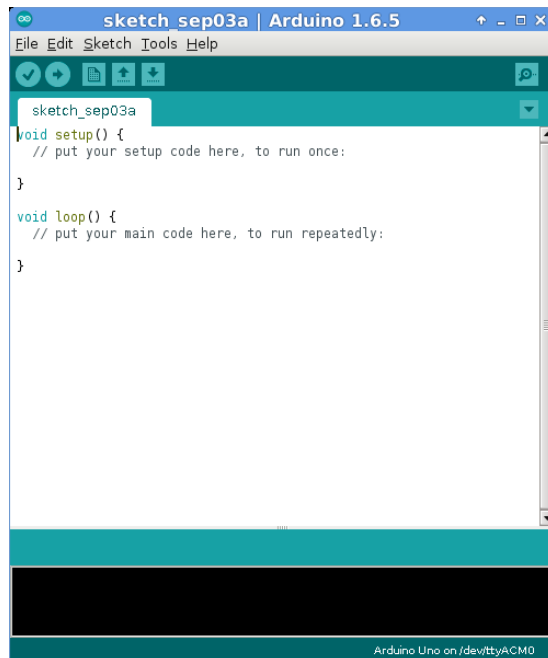
Name	Processor	Operating/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [KB]	SRAM [KB]	Flash [KB]	USB	UART
ArduinoBT	ATmega328P	5 V / 2.5-12 V	16 MHz	6/0	14/6	1	2	32	-	1
Due	ATSAM3X8E	3.3 V / 7-12 V	84 MHz	12/2	54/12	-	96	512	2 Micro	4
Esplora	ATmega32U4	5 V / 7-12 V	16 MHz	-	-	1	2.5	32	Micro	-
Ethernet	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/4	1	2	32	Regular	-
Fio	ATmega328P	3.3 V / 3.7-7 V	8 MHz	8/0	14/6	1	2	32	Mini	1
Gemma	ATtiny85	3.3 V / 4-16 V	8 MHz	1/0	3/2	0.5	0.5	8	Micro	0
Leonardo	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
LilyPad	ATmega168V ATmega328P	2.7-5.5 V / 2.7-5.5 V	8MHz	6/0	14/6	0.512	1	16	-	-
LilyPad SimpleSnap	ATmega328P	2.7-5.5 V / 2.7-5.5 V	8 MHz	4/0	9/4	1	2	32	-	-
LilyPad USB	ATmega32U4	3.3 V / 3.8-5 V	8 MHz	4/0	9/4	1	2.5	32	Micro	-
Mega 2560	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Mega ADK	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
Mini	ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	1	2	32	-	-
Nano	ATmega168 ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	0.512 1	1 2	16 32	Mini	1
Pro	ATmega168 ATmega328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	0.512 1	1 2	16 32	-	1
Pro Mini	ATmega328P	3.3 V / 3.35-12 V	8 MHz	6/0	14/6	0.512	1	16	-	1

		5 V / 5-12 V	16 MHz							
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1
Yún	ATmega32U4 AR9331 Linux	5 V	16 MHz 400MHz	12/0	20/7	1	2.5 16MB	32 64MB	Micro	1
Zero	ATSAMD21G18	3.3 V / 7-12 V	48 MHz	6/1	14/10	-	32	256	2 Micro	2

**Πίνακας 3.** Υλοποιήσεις Arduino (Πηγή: <http://www.arduino.cc>)

- **Το λογισμικό**

Το Arduino συνοδεύεται από περιβάλλον ανάπτυξης (Integrated Development Environment – IDE) ανοικτού κώδικα, το οποίο επιτρέπει την ανάπτυξη προγραμμάτων με δομή όμοια του Processing [29]. Κάθε πρόγραμμα που εκτελείται στο Arduino, περιέχει δύο βασικές μεθόδους (συναρτήσεις). Τη συνάρτηση `setup()`, η οποία εκτελείται μία και μοναδική φορά κατά την εκκίνηση του προγράμματος με σκοπό την αρχικοποίηση των μεταβλητών και του υλικού και τη `loop()` η οποία εκτελείται συνεχώς και χρησιμοποιείται για τον έλεγχο του υλικού.



**Εικόνα 14.** Το περιβάλλον ανάπτυξης εφαρμογών του Arduino

Για την υλοποίηση των αισθητήριων κόμβων του συστήματος, επιλέχθηκε το Arduino UNO, το οποίο είναι μια πλατφόρμα Arduino γενικής χρήσης, και το οποίο διαθέτει:

- 6 αναλογικές εισόδους
- 14 ψηφιακές διεπαφές εισόδου/εξόδου



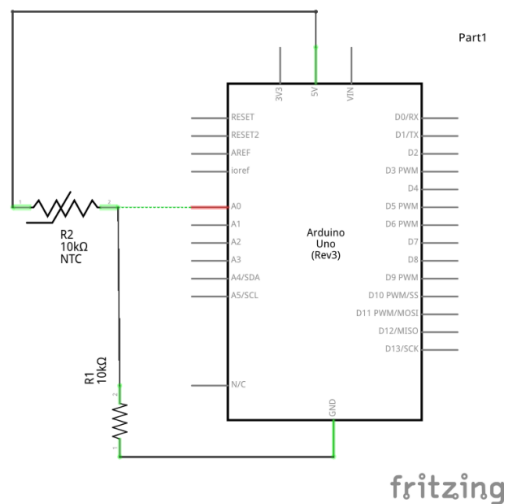
**Εικόνα 15.** Arduino UNO (Πηγή: <http://www.arduino.cc>)

### 5.1.2 Σύνδεση θερμίστορ

Όπως αναφέρθηκε στο Κεφάλαιο 4, το θερμίστορ είναι μια ημιαγώγιμη διάταξη η οποία μεταβάλλει την αντίστασή της με τη μεταβολή της θερμοκρασίας. Η μεταβολή της αντίστασης του θερμίστορ προκαλεί μεταβολή της τάσης εξόδου  $V_{out}$  στο διαιρέτη τάσης που σχηματίζεται από την αντίσταση  $R_1$  και το θερμίστορ. Μετρώντας την τάση στο σημείο  $V_{out}$ , μπορούμε να υπολογίσουμε την αντίσταση  $R$  του θερμίστορ:

$$R = R_1 \frac{V_{in} - V_{out}}{V_{out}} \quad (1)$$

Μπορούμε να μετρήσουμε την τάση στο σημείο  $V_{out}$  συνδέοντάς το σε μία αναλογική είσοδο του Arduino, όπως φαίνεται στην Εικόνα 16.



**Εικόνα 16.** Σύνδεση θερμίστορ στο Arduino

Κάθε αναλογική θύρα του Arduino δέχεται τιμές τάσης από 0 ως 5 Volt. Η είσοδος αυτή τροφοδοτείται στη συνέχεια εσωτερικά σε ένα μετατροπέα σήματος αναλογικού-σε-ψηφιακό (Analog-to-Digital Converter – ADC) 10 bit. Τα 10 αυτά bit αντιστοιχούν σε 1024 διαφορετικές τιμές τάσης εισόδου. Άρα τα 0 Volt αντιστοιχούν στην τιμή 0 ενώ τα 5Volt αντιστοιχούν στην τιμή 1024 ( $2^{10}$ ). Οι ενδιάμεσες τάσεις κβαντοποιούνται στις ενδιάμεσες 1022 τιμές.

Γνωρίζοντας την τιμή της αντίστασης της  $R_1=10K$ , της τροφοδοσίας τάσης του διαιρέτη (5 Volts) και την τιμή εξόδου, μπορούμε να χρησιμοποιήσουμε τη σχέση (1), οπότε έχουμε:

$$R = 10^4 \frac{1023 - V_{out}}{V_{out}} \quad (2)$$

Στο Arduino, η τιμή εισόδου μιας αναλογικής θύρας επιστρέφεται από τη συνάρτηση `analogRead(PORT)`, όπου PORT ο αριθμός που αντιστοιχεί στην αναλογική θύρα.

Άρα, η τάση εισόδου  $V$  στην αναλογική θύρα 0, θα επιστραφεί από τη συνάρτηση:

`V=analogRead(0);`

Και θα έχει τιμές από 0 ως 1023.

Το θερμίστορ που θα χρησιμοποιηθεί, είναι το χαμηλού κόστους NTC Thermistor με part number NTCLE232264063103. Ανατρέχοντας στο datasheet του κατασκευαστή, για την εύρεση των σταθερών  $A$ ,  $B$  και  $C$ , παρατηρούμε ότι ο κατασκευαστής προτείνει τη χρήση της σχέσης Steinhart-Hart στην οποία προστίθεται κι ένας τέταρτος συντελεστής. Η σχέση αυτή είναι:

$$\frac{1}{T} = A + B \ln \frac{R}{R_t} + C \ln^2 \frac{R}{R_t} + D \ln^3 \frac{R}{R_t}$$

Από το σχετικό πίνακα που παρέχεται στο datasheet εντοπίζουμε τους συντελεστές, οι οποίοι είναι:

- $A=0.003354016$
- $B = 0.0002569850$
- $C=0.000002620131$
- $D=0.0000000383091$

Έτσι, η συνάρτηση που επιστρέφει τη θερμοκρασία σε βαθμούς Celsius είναι:

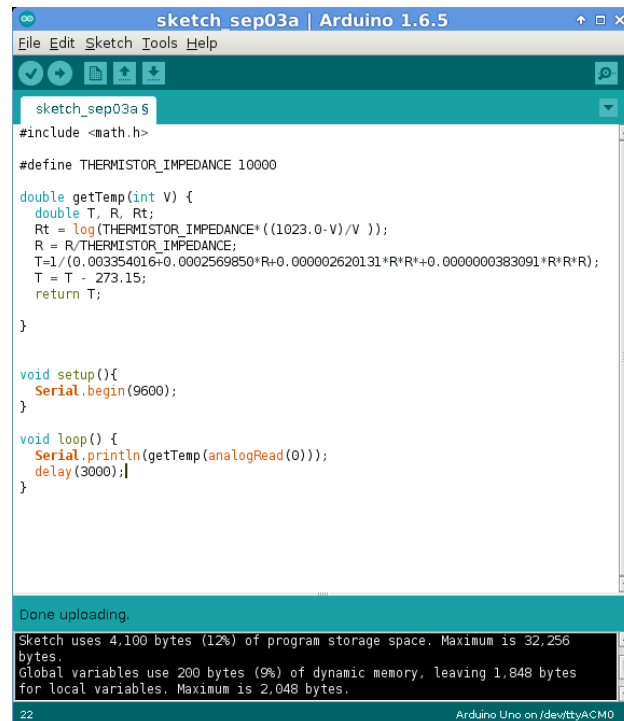
```

#define THERMISTOR_IMPEDANCE 10000

double getTemp(int V) {
  double T, R, Rt;
  Rt = log(THERMISTOR_IMPEDANCE*((1023.0-V)/V));
  R = Rt/THERMISTOR_IMPEDANCE;
  T=1/(0.003354016+0.0002569850*R+0.000002620131*R*R+0.0000000383091*R*R*R);
  T = T - 273.15;
  return T;
}

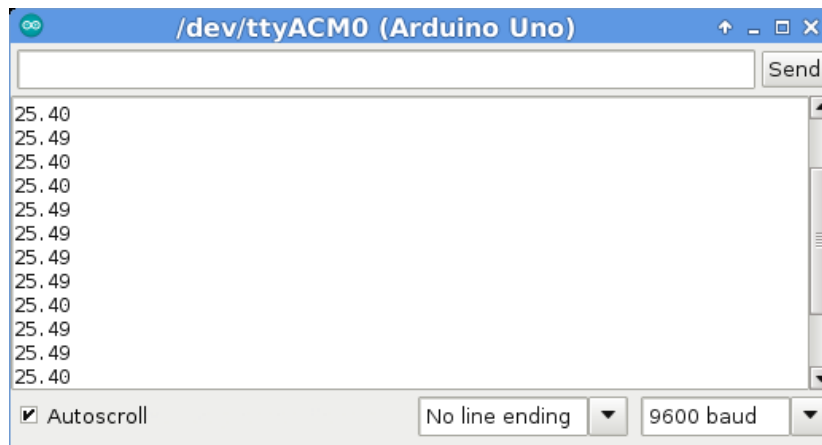
```

Με τον Arduino IDE μεταγλωττίζουμε κι εκτελούμε το πρόγραμμα στο Arduino.



**Εικόνα 17.** Μεταγλώττιση και μεταφορά προγράμματος στο Arduino

Για να ελέγξουμε τις τιμές που επιστρέφονται, αρχικοποιούμε στη συνάρτηση setup τη σειριακή διεπαφή με ταχύτητα 9600bps και στη συνέχεια, στη loop διαβάζουμε ανά 3 δευτερόλεπτα την τιμή εισόδου της αναλογικής πόρτας 0, το αποτέλεσμα της οποίας το περνάμε ως όρισμα στη συνάρτηση που μας δίνει τη θερμοκρασία. Μετά τη φόρτωση του προγράμματος, στο serial monitor τυπώνεται η θερμοκρασία του χώρου:



**Εικόνα 18.** Εμφάνιση θερμοκρασίας στην κονσόλα

Με τη βοήθεια ενός θερμομέτρου, επιβεβαιώνουμε την αλήθεια της μέτρησης:



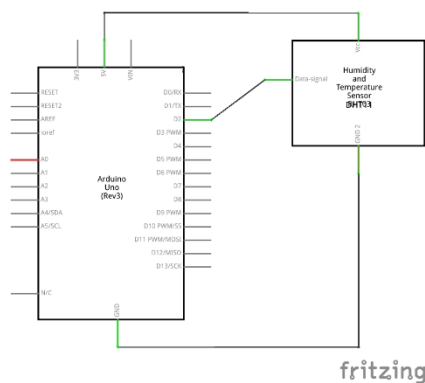
**Εικόνα 19.** Επιβεβαίωση μέτρησης με θερμόμετρο

Το NTC Thermistor θα χρησιμοποιηθεί για τη μέτρηση της θερμοκρασίας στα φύλλα των φυτών.

### 5.1.3 Σύνδεση αισθητήρα υγρασίας

Για τον υπολογισμό το σημείου δρόσου, είναι απαραίτητη η λήψη της τιμής της σχετικής υγρασίας του χώρου. Για τη λήψη της, θα χρησιμοποιηθεί ο αισθητήρας DHT11. Αν και δεν

παρουσιάζει την ακρίβεια του SHT11, και καταναλώνει περισσότερη ενέργεια, είναι διαθέσιμος ευρέως και η διαφορά στην τιμή κάνει την επιλογή του πιο συμφέρουσα.



Εικόνα 20. Σύνδεση DHT11

Ο DHT11, έχει 3 ενεργά pin. Τα δύο χρησιμοποιούνται για τη σύνδεση στην τροφοδοσία και τη γείωση ενώ το τρίτο για την αποστολή των δεδομένων θερμοκρασίας και υγρασίας. Η αποστολή των δεδομένων είναι ψηφιακή έτσι δεν απαιτείται καμιά μετατροπή.

Ένα από τα σημαντικά πλεονεκτήματα των ανοικτών τεχνολογιών (υλικού και λογισμικού) είναι η δημιουργία κοινοτήτων και η ανάπτυξη λύσεων διαθέσιμων στην υπόλοιπη κοινότητα αλλά κι έξω από αυτή. Έτσι, έχει δημιουργηθεί κατάλληλη βιβλιοθήκη η οποία διευκολύνει την ανάγνωση των δεδομένων που διαβάζονται από τον αισθητήρα. Η βιβλιοθήκη είναι διαθέσιμη στον ιστότοπο: <https://github.com/adafruit/DHT-sensor-library/blob/master/DHT.h>

Με τη χρήση της, αρκεί να δημιουργήσουμε ένα αντικείμενο από την κλάση DHT της βιβλιοθήκης και στη συνέχεια να ζητήσουμε την τιμή της public μεταβλητής temperature για τη θερμοκρασία και humidity για την υγρασία.

Άρα ο κώδικας είναι:

```
#include <dht.h>

dht DHT;

#define DHT11_PIN 2

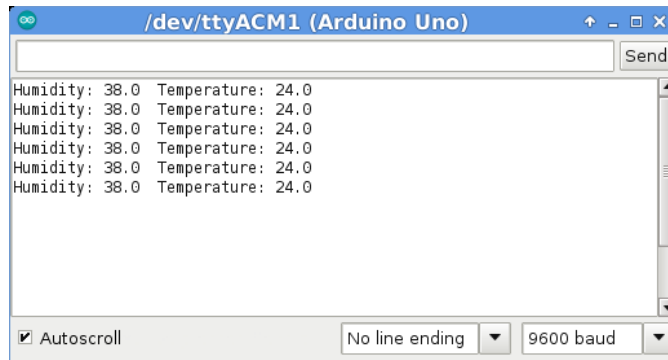
void setup() {
  Serial.begin(9600);
}

void loop() {
  int chk = DHT.read11(DHT11_PIN);
  Serial.print("Humidity: ");
  Serial.print(DHT.humidity, 1);
  Serial.print("Temperature: ");
```



```
Serial.print(DHT.temperature, 1);  
delay(30000);  
}
```

Οπότε εκτελώντας έχουμε:



**Εικόνα 21.** Εμφάνιση τιμών υγρασίας και θερμοκρασίας στην κονσόλα

#### 5.1.4 Δημιουργία κόμβου και Έλεγχος λειτουργίας

Μετά τη σύνδεση κι έλεγχο της ορθότητας του τρόπου χρήσης των αισθητήριων οργάνων, πλέον είναι δυνατή η σύνδεση του συνόλου τους στο Arduino. Συγκεκριμένα θα συνδεθεί ένας αισθητήρας DHT11 στην ψηφιακή πόρτα 2 του Arduino και έξι (6) NTC Thermistors, όσες και οι αναλογικές πόρτες σε αυτές.

Η βασική λειτουργία του κόμβου είναι η αποστολή των τιμών που καταγράφονται από τους αισθητήρες. Η αποστολή των τιμών θα γίνεται ως μια συμβολοσειρά (String) η οποία θα είναι της μορφής:

ΥγρασίαΧώρου:ΘερμοκρασίαΧώρου:ΘερμοκρασίαThermistor0:...:ΘερμοκρασίαThermistor5;

Με τον τρόπο αυτό, και με τη χρήση του διαχωριστή (delimiter) «:» ο λήπτης θα έχει τη δυνατότητα διαχωρισμού και κατανόησης των τιμών. Το String θα δημιουργηθεί από τη συνάρτηση `loop()` ως εξής:

```
Serial.print(DHT.humidity, 1);  
Serial.print(":");  
Serial.print(DHT.temperature, 1);  
Serial.print(":");  
Serial.print(getTemp(analogRead(0)));  
Serial.print(":");  
Serial.print(getTemp(analogRead(1)));
```

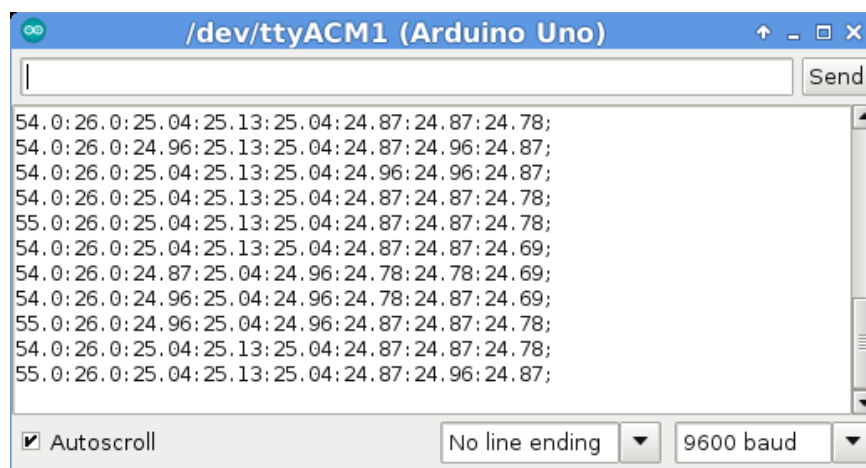
```

Serial.print(":");
Serial.print(getTemp(analogRead(2)));
Serial.print(":");
Serial.print(getTemp(analogRead(3)));
Serial.print(":");
Serial.print(getTemp(analogRead(4)));
Serial.print(":");
Serial.print(getTemp(analogRead(5)));
Serial.println(";");

```

Καθώς οι συνθήκες εντός του θερμοκηπίου θεωρούνται ελεγχόμενες η αποστολή αυτή μπορεί να γίνεται σε μικρή συχνότητα. Αρχικά η συχνότητα αυτή θα τεθεί στα 15 λεπτά της ώρας. Με τον τρόπο αυτό θα έχουμε 4 ωριαίες ενημερώσεις για τις συνθήκες του περιβάλλοντος. Στο Arduino η loop() εκτελείται διαρκώς. Μπορούμε να ορίσουμε το χρόνο καθυστέρησης μεταξύ δύο διαδοχικών εκτελέσεων με χρήση της συνάρτησης delay η οποία δέχεται ως όρισμα το χρόνο αυτό σε ms.

Εκτελούμε το πρόγραμμα κι ελέγχουμε στην κονσόλα πως το String δημιουργείται κι αποστέλλεται κανονικά:



**Εικόνα 22.** Συμβολοσειρά εξόδου

Παρατηρούμε ότι στην τελευταία αποστολή έχει καταγραφεί:

- Σχετική Υγρασία: 54%
- Θερμοκρασία: 26°C
- Θερμοκρασία στο θερμίστορ 0: 25,04°C
- Θερμοκρασία στο θερμίστορ 1: 25,13°C
- Θερμοκρασία στο θερμίστορ 2: 25,04°C
- Θερμοκρασία στο θερμίστορ 3: 24,87°C

- Θερμοκρασία στο θερμίστορ 4: 24,96°C
- Θερμοκρασία στο θερμίστορ 5: 24,87°C

Άρα η συμβολοσειρά δημιουργήθηκε με επιτυχία.

## 5.2 Διασύνδεση Κόμβων

Η ασύρματη διασύνδεση των κόμβων είναι ένας σημαντικός παράγοντας επιτυχίας στην εφαρμογή ενός WSN. Στο Κεφάλαιο 2 αναφέρθηκαν τα πιο γνωστά πρωτόκολλα που χρησιμοποιούνται για την επικοινωνία των κόμβων. Ειδικότερα στη γεωργία ακριβείας, το πιο διαδεδομένο πρωτόκολλο, είναι το πρωτόκολλο ZigBee καθώς σε σύγκριση με το Bluetooth παρουσιάζει ως πλεονεκτήματα [31]:

1. Μικρότερη κατανάλωση ενέργειας
2. Δυνατότητα υποστήριξης για πάνω από 65000 κόμβους στο ίδιο δίκτυο
3. Ευελιξία ως προς την τοπολογία

Το μειονέκτημά του που αφορά τις μικρότερες ταχύτητες επικοινωνίας, δεν αποτελεί ανασταλτικό παράγοντα καθώς ο όγκος της κίνησης δεδομένων είναι ελάχιστος.

Για την υλοποίηση του δικτύου, επιλέχθηκε η τοπολογία mesh έτσι ώστε να είναι δυνατή η υλοποίηση ακόμη και σε μεγάλη έκταση όπου δεν είναι δυνατή η άμεση επικοινωνία του τερματικού κόμβου (end device) με τον coordinator. Σε μικρότερες υλοποιήσεις μπορεί να χρησιμοποιηθεί τοπολογία αστέρα. Επίσης η χρήση mesh επιτρέπει να ερευνηθούν θέματα όπως η διαφορά στην κατανάλωση ενέργειας μεταξύ των διαφορετικών κόμβων:

- Ένας coordinator, ο οποίος θα είναι ο υπολογιστής (ή διακομιστής) στον οποίο θα αποστέλλονται τα δεδομένα και από τον οποίο θα λαμβάνονται οι αποφάσεις.
- Οι end devices (τερματικοί κόμβοι), οι οποίοι ως αισθητήριοι κόμβοι θα συλλέγουν και θα αποστέλλουν δεδομένα στον coordinator και θα βρίσκονται διασκορπισμένοι στο θερμοκήπιο και μπορούν να τροφοδοτούνται από αυτόνομες πηγές ενέργειας στα σημεία όπου δεν υπάρχει παροχή ρεύματος.
- Οι routers, οι οποίοι θα τοποθετηθούν σε σημεία όπου είναι δυνατή η παροχή ρεύματος και θα λειτουργούν ως αισθητήριοι κόμβοι. Η διαφοροποίηση με τους τερματικούς

κόμβους είναι πως οι routers είναι συνεχώς σε ενεργή κατάσταση καθώς έχουν τη δυνατότητα να προωθήσουν frames από άλλους κόμβους προς τον coordinator ή άλλο router.

Στην πειραματική διάταξη που αναπτύσσεται και παρουσιάζεται, θα υλοποιηθεί αρχικά ένας αισθητήριος κόμβος με χρήση Arduino, ο οποίος θα λειτουργεί ως router κι ένας κόμβος υλοποιημένος με λογισμικό σε Η/Υ ως coordinator.

Για την ασύρματη δικτύωση του Arduino, απαιτείται η χρήση ειδικής διάταξης, (wireless shield) η οποία λειτουργεί ως διεπαφή μεταξύ του Arduino board και των κατάλληλων modules που αναλαμβάνουν τη διαμόρφωση της πληροφορίας για την ασύρματη μετάδοσή της. Το wireless shield εφαρμόζεται πάνω στη βασική πλακέτα του Arduino, επεκτείνει τις αναλογικές και ψηφιακές διεπαφές του ώστε να μπορούν να χρησιμοποιηθούν και επάνω του μπορεί να συνδεθεί το κατάλληλο Module.



**Εικόνα 23.** Wireless Shield με module τοποθετημένο στο Arduino

### 5.2.1 XBee

Για την ασύρματη επικοινωνία θα χρησιμοποιηθούν τα XBee RF Modules [17] της Digi International. Τα XBee περιέχουν μικροεπεξεργαστή ο οποίος μπορεί να προγραμματιστεί μέσω λογισμικού καθιστώντας τα modules ιδιαίτερα εύελικτα.



**Εικόνα 24.** XBee module

Τα XBee modules μπορούν να λειτουργήσουν με δύο διαφορετικούς τρόπους (modes): API mode και Transparent (AT) mode. Στο AT mode η επικοινωνία είναι τύπου σημείο-προς-σημείο (point-to-point). Ο αποδέκτης των δεδομένων καθορίζεται κατά την παραμετροποίηση του module και δε μπορεί να επιλεγεί κατά βούληση από την εφαρμογή. Άρα η εφαρμογή απλά προωθεί τα πλαίσια στο XBee. Αντιθέτως στο API mode, υπάρχει κατάλληλη προγραμματιστική διεπαφή (Application Programming Interface – API) μέσω της οποίας η εφαρμογή μπορεί να αναγνωρίσει τον αποστολέα του πλαισίου που έλαβε αλλά και να καθορίσει τον παραλήπτη του πλαισίου προς αποστολή.

Σύμφωνα με το σχεδιασμό του συστήματος οι αισθητήριοι κόμβοι έχουν ως μοναδική εργασία τη συλλογή κι αποστολή δεδομένων, χωρίς να λαμβάνουν ποτέ με τη σειρά τους δεδομένα από τον Coordinator. Για το λόγο αυτό επιλέχθηκε η λειτουργία τους σε AT mode. Τα πλεονεκτήματα της επιλογής αυτής είναι:

1. Δεν απαιτείται καμιά τροποποίηση στο πρόγραμμα που εκτελείται στο Arduino καθώς το XBee συμπεριφέρεται ως σειριακή διεπαφή.
2. Δεν απαιτείται κατάλληλη παραμετροποίηση στον coordinator ώστε να αναγνωρίζει (acknowledge) τα πακέτα που λαμβάνει. Έτσι ο φόρτος εργασίας του μειώνεται.

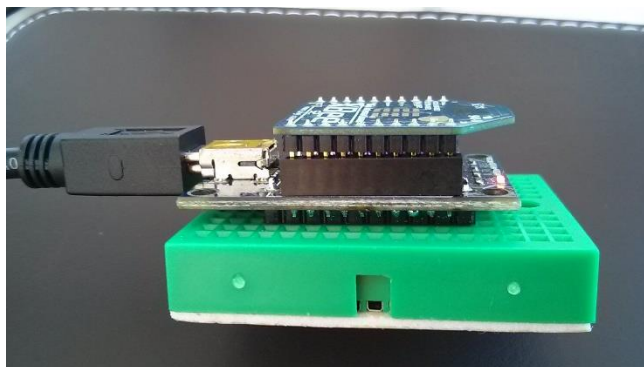
Αντιθέτως ο coordinator υπάρχει πιθανότητα να λαμβάνει πλαίσια (frames) από πολλούς αισθητήριους κόμβους. Η χρήση AT mode θα δυσκόλευε ιδιαίτερα την αναγνώριση των πλαισίων και θα καθιστούσε αδύνατη την περαιτέρω επεξεργασία τους, καθώς θα λαμβανόταν ένα σύνολο ASCII δεδομένων, χωρίς καμιά ιεραρχία. Τα πλαίσια που λαμβάνονται σε API mode

δεν είναι άμεσα αναγνώσιμα αλλά θα πρέπει να υπάρξει κατάλληλη διεπαφή λογισμικού (Application Programming Interface – API) ώστε να τα μετατρέψει καταλλήλως. Μια τέτοια προγραμματιστική διεπαφή έχει υλοποιηθεί από τον Andrew Rapp κι είναι διαθέσιμη στη διεύθυνση <http://code.google.com/p/XBee-api/> (τελ. Πρόσβαση: 31/08/2015)

### 5.2.2 Σύνδεση XBee

Η σύνδεση του XBee στο Arduino γίνεται μέσω του Wireless Shield, όπως αναφέρθηκε στην παράγραφο 5.2.1. Καθώς το XBee που θα συνδεθεί στο Arduino θα λειτουργεί σε transparent mode, το μόνο που απαιτείται είναι η αναδρομολόγηση της εξόδου μέσω του module κι όχι μέσω της θύρας USB. Αυτό επιτυγχάνεται με την επιλογή της θέσης Micro του διακόπτη Serial Select του Shield.

Για τη σύνδεση του coordinator module που θα λειτουργεί σε API mode, πέρα από τη χρήση της κατάλληλης προγραμματιστικής διεπαφής, πρέπει αρχικά να συνδεθεί με μια θύρα του Η/Υ. Η ίδια σύνδεση πρέπει επίσης να γίνει για τον προγραμματισμό των XBee, ο οποίος περιγράφεται στην επόμενη παράγραφο. Για τη σύνδεση αυτή θα χρησιμοποιηθεί ειδικός προσαρμογέας XBee σε USB, ο οποίος φαίνεται στην εικόνα 25.

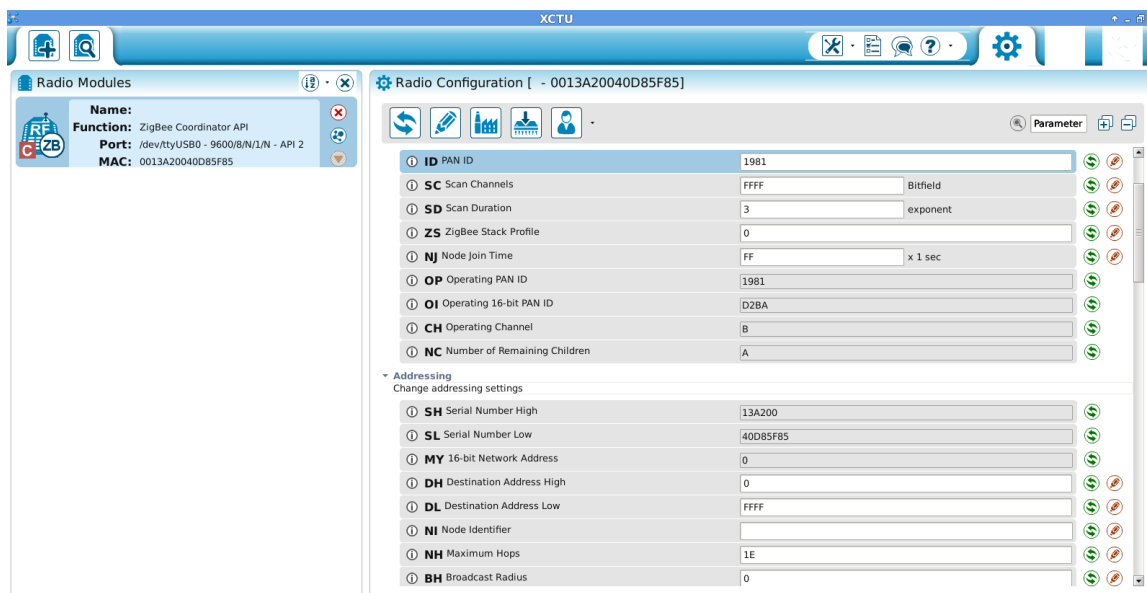


**Εικόνα 25.** Σύνδεση XBee μέσω USB

### 5.2.2 Ρύθμιση XBees

Πριν τη χρήση τους τα XBees πρέπει να ρυθμιστούν ως προς το mode λειτουργίας αλλά κι άλλες παραμέτρους. Η ρύθμιση μπορεί να γίνει με εντολές AT μέσω ενός τερματικού. Για να διευκολυνθεί όμως η επικοινωνία με το Module, η κατασκευάστρια εταιρία έχει διαθέσει την εφαρμογή X-CTU (XBee Configuration & Testing Utility) μέσω της οποίας είναι δυνατή η

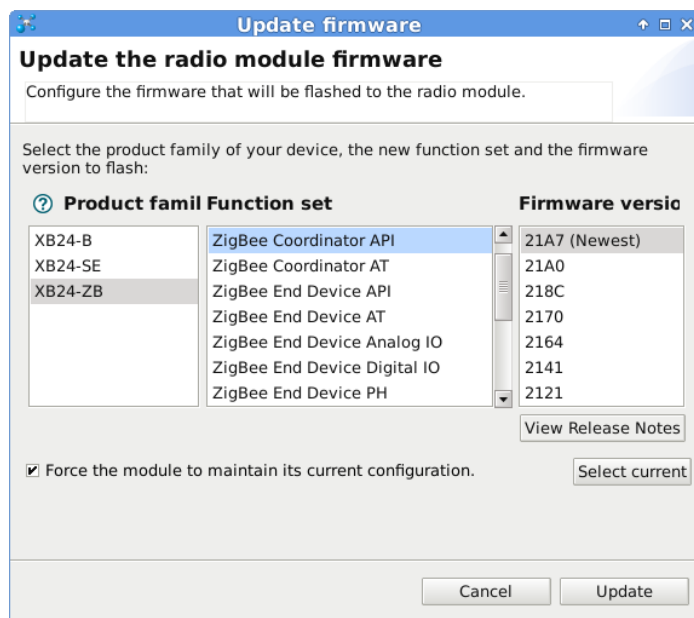
ανίχνευση, η ρύθμιση παραμέτρων λειτουργίας κι ο έλεγχος του XBee module με χρήση γραφικού περιβάλλοντος.



**Εικόνα 26.** Οθόνη ρύθμισης παραμέτρων XBee module

Η σημαντική παράμετρος που πρέπει να ρυθμιστεί και να είναι κοινή σε όλα τα modules είναι η διεύθυνση του δικτύου (PAN ID) που καθορίζει το δίκτυο στο οποίο ανήκει το module. Μόνο τα modules που ανήκουν στο ίδιο δίκτυο μπορούν να επικοινωνήσουν μεταξύ τους.

Επόμενη σημαντική παράμετρος είναι η διεύθυνση παραλήπτη (destination address). Η διεύθυνση του κάθε XBee αποτελείται από το serial number του, μεγέθους 64 bits, και χωρίζεται σε δύο ίσα τμήματα των 64 bits (στο X-CTU τα μηδενικά αρχικά bits δεν αναφέρονται). Ένα module, σε AT mode μπορεί να αποστείλει δεδομένα μόνο στο module του οποίου η διεύθυνση αναγράφεται ως destination. Έτσι στους routers, που λειτουργούν με το mode αυτό θα θέσουμε destination address τη διεύθυνση του coordinator. Ο coordinator στην εφαρμογή μας από τη μία δε στέλνει δεδομένα παρά μόνο λαμβάνει κι από την άλλη λειτουργεί σε API mode, οπότε η διεύθυνση προορισμού, αν χρειαστεί, μπορεί να τεθεί από το λογισμικό εφαρμογής. Ο τρόπος λειτουργίας των modules δεν τίθεται μέσω κάποια τιμής αλλά καθορίζεται από το firmware, άρα για την αλλαγή mode απαιτείται η αλλαγή του firmware που φέρει.



Εικόνα 27. Ενημέρωση firmware

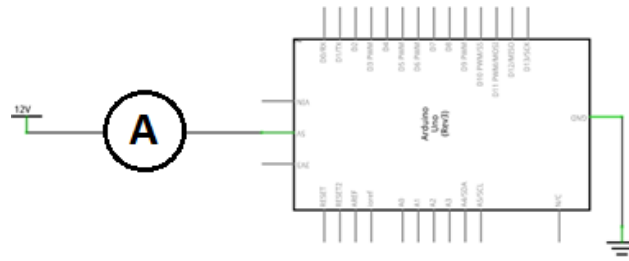
### 5.2.3 Σύνδεση τροφοδοσίας

Η επαρκής τροφοδοσία είναι σημαντική για τη σωστή λειτουργία του συστήματος. Παρότι σε ένα θερμοκήπιο η αντικατάσταση μιας μπαταρίας φαίνεται σχετικά απλή υπόθεση, εν τούτοις η αμέλεια μπορεί να οδηγήσει ένα κόμβο εκτός λειτουργίας με ενδεχόμενα προβλήματα για τα φυτά που ελέγχονται από αυτόν. Σε περιπτώσεις όπου υπάρχει δυνατότητα σύνδεσης παροχής ρεύματος, ο τρόπος αυτός θα ήταν προτιμότερος.

Το Arduino λειτουργεί με τροφοδοσία από 5 ως 12 V συνεχούς ρεύματος. Για να μετρήσουμε την κατανάλωση του κάθε κόμβου, μπορούμε να χρησιμοποιήσουμε αμπερόμετρο ώστε να μετρηθεί το ρεύμα που διαρρέει την τροφοδοσία για κάθε κατάσταση του κόμβου (ενεργός ή σε κατάσταση ύπνου).

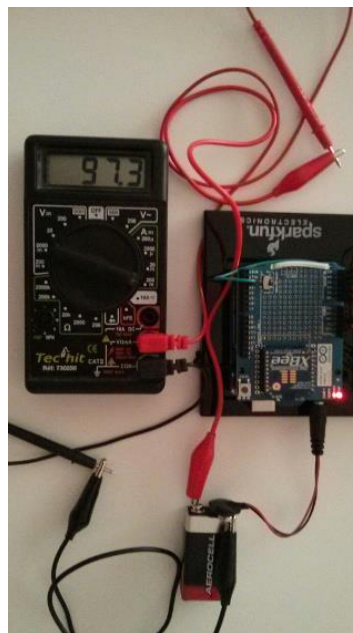
Υπολογίζοντας την κατανάλωση ρεύματος σε πλήρη ισχύ, μπορούμε να έχουμε μια πρόβλεψη για την ελάχιστη διάρκεια παροχής ισχύος της μπαταρίας. Ο υπολογισμός αυτός θα γίνει με χρήση αμπερομέτρου στην είσοδο της τροφοδοσίας του Arduino, το οποίο θα μετρήσει την ισχύ του ρεύματος που καταναλώνει ο κόμβος:





**Εικόνα 28.** Διάγραμμα σύνδεσης διάταξης αμπερόμετρου

Για τον κόμβο μας, με το XBee σε λειτουργία router συνδέουμε το αμπερόμετρο:



**Εικόνα 29.** Σύνδεση αμπερόμετρου

Κατά τη λειτουργία η μέγιστη τιμή ισχύος ρεύματος που μετρήθηκε με το XBee module ήταν 97,4mA. Άρα η διάρκεια ζωής μιας πηγής, που μπορεί να τροφοδοτήσει με n mAH, είναι:

$$\frac{n \cdot mAH}{97,4mA} = nh$$

Οπότε, μια τυπική αλκαλική μπαταρία, 9V, που συνήθως χαρακτηρίζεται από χωρητικότητα περίπου 500mAH, μπορεί να τροφοδοτήσει τον κόμβο για μόλις 5 ώρες.

Αντίστοιχη μέτρηση έγινε σε κόμβο στον οποίο το XBee που είχε ρυθμιστεί ως End Device, κάτι που σημαίνει πως ανάμεσα στις εκπομπές, άρα τον περισσότερο χρόνο, βρίσκεται σε κατάσταση

ύπνωσης. Η μέτρηση αυτή έδειξε μέγιστη τιμή 64mAH, κάτι που ανεβάζει τη διάρκεια τροφοδότησης στις περίπου 7,7 ώρες.

Από τα παραπάνω προκύπτει πως η χρήση των αισθητήριων κόμβων στο θερμοκήπιο πρέπει να συνοδευτεί από:

- Συνεχή πηγή τροφοδοσίας για τους κόμβους με ρολό router, ή
- Επαναφορτιζόμενες διατάξεις με μπαταρίες μεγαλύτερης χωρητικότητας, άνω των 1000mAH

## 5.3 Λήψη και Διαχείριση Δεδομένων

Τα XBee modules μπορούν να λειτουργήσουν σε API και AT mode. Σύμφωνα με το σχεδιασμό του συστήματος οι αισθητήριοι κόμβοι έχουν ως μοναδική εργασία τη συλλογή κι αποστολή δεδομένων, χωρίς να λαμβάνουν ποτέ με τη σειρά τους δεδομένα από τον Coordinator. Για το λόγο αυτό επιλέχθηκε η λειτουργία τους σε AT mode. Τα πλεονεκτήματα της επιλογής αυτής είναι:

1. Δεν απαιτείται καμιά τροποποίηση στο πρόγραμμα που εκτελείται στο Arduino καθώς το XBee συμπεριφέρεται ως σειριακή διεπαφή.
2. Δεν απαιτείται κατάλληλη παραμετροποίηση στον coordinator ώστε να αναγνωρίζει (acknowledge) τα πακέτα που λαμβάνει. Έτσι ο φόρτος εργασίας του μειώνεται.

Αντιθέτως ο coordinator υπάρχει πιθανότητα να λαμβάνει πλαίσια (frames) από πολλούς αισθητήριους κόμβους. Η χρήση AT mode θα δυσκόλευε ιδιαίτερα την αναγνώριση των πλαισίων και θα καθιστούσε αδύνατη την περαιτέρω επεξεργασία τους, καθώς θα λαμβανόταν ένα σύνολο ASCII δεδομένων, χωρίς καμιά ιεραρχία. Τα πλαίσια που λαμβάνονται σε API mode δεν είναι άμεσα αναγνώσιμα αλλά θα πρέπει να υπάρξει κατάλληλη διεπαφή λογισμικού ώστε να τα μετατρέψει καταλλήλως.

### 5.3.1 Υλοποίηση εφαρμογής σε Java

Σκοπός της της εφαρμογής που υλοποιήθηκε ε Java είναι η συλλογή των δεδομένων με στόχο:

1. Την επεξεργασία τους σε πραγματικό χρόνο για την ενημέρωση του χρήστη και τη λήψη αποφάσεων.
2. Την αποθήκευση για μελλοντική ανάκτηση κι αξιοποίηση.

Η εφαρμογή υλοποιήθηκε με χρήση του Java IDE Netbeans [146] και του designer που αυτός διαθέτει για τη δημιουργία της γραφικής διεπαφής χρήστη (User Graphical Interface – GUI). Επίσης εισήχθησαν στο NetBeans οι ακόλουθες βιβλιοθήκες:

- SQLite JDBC για τη χρήση της SQLite.
- XBee API για τη διασύνδεση με το δίκτυο ZigBee.
- Log4j που χρησιμοποιεί το XBee API για αποσφαλμάτωση.
- RXTXComm για την πρόσβαση στις θύρες εισόδου/εξόδου του υπολογιστή.

Οι κλάσεις που δημιουργήθηκαν χωρίστηκαν σε 4 πακέτα, που διαχωρίζουν την επιμέρους λειτουργικότητα και είναι:

1. Package PlantMonitor που περιέχει τις κλάσεις:
  - PlantMonitor η οποία περιέχει τη μέθοδο main και αρχικοποιεί τη σύνδεση με τη βάση και ξεκινά το GUI.
  - Dew, η οποία περιέχει τη μέθοδο για τον υπολογισμό του σημείου δρόσου
  - Επίσης περιέχεται το αρχείο παραμετροποίησης του log4j (log4j.properties) που καθορίζει τη συμπεριφορά του Log4J.
  - Settings, που καθορίζει τις επιλεγόμενες από το χρήστη ρυθμίσεις. Κάθε ρύθμιση είναι ένα ζευγάρι ονόματος και τιμής. Η κλάση settings περιέχει τα ονόματα αυτά.
2. Package db που περιέχει:
  - Την abstract κλάση Database. Η κλάση κληροδοτεί τις μεθόδους της σε κλάσεις για κάθε υποστηριζόμενη βάση δεδομένων. Με τον τρόπο αυτό, η εφαρμογή παρέχει πρόβλεψη για διαφορετικά συστήματα βάσεων δεδομένων, αρκεί για κάθε ένα από αυτά να έχει δημιουργηθεί η αντίστοιχη κλάση.
  - Την κλάση SQLite που επεκτείνει την Database και υλοποιεί τις μεθόδους της
3. Package XBee που περιέχει τις κλάσεις, τα αντικείμενα των οποίων αναλαμβάνουν την επικοινωνία με το module.
  - Η κλάση XBee ξεκινά ένα thread, έτσι ώστε η επικοινωνία με το XBee να μην παγώνει την κύρια εφαρμογή

- Η κλάση XBeeThread υλοποιεί το thread που αναλαμβάνει να διαβάσει τα δεδομένα από τη θύρα του υπολογιστή, να τα ετοιμάζει και να τα παρουσιάζει στο GUI αλλά και να τα αποθηκεύει στη βάση.
4. Package GUI, που περιέχει τις κλάσεις της γραφικής διεπαφής.

### 5.3.2 Βάση Δεδομένων

Ως βάση δεδομένων (ΒΔ) για τη δοκιμή του συστήματος χρησιμοποιήθηκε η SQLite [147], η οποία είναι μια πολύ απλή βάση δεδομένων ενός αρχείου. Στη ΒΔ θα υπάρχουν οι ακόλουθοι πίνακες:

- Πίνακας Conditions (id, serial, datetime, humidity, temperature)

Στον πίνακα αυτό αποθηκεύονται η διεύθυνση του κόμβου (serial), η ώρα λήψης των μετρήσεων (datetime), η σχετική υγρασία (humidity) και η θερμοκρασία (temperature)

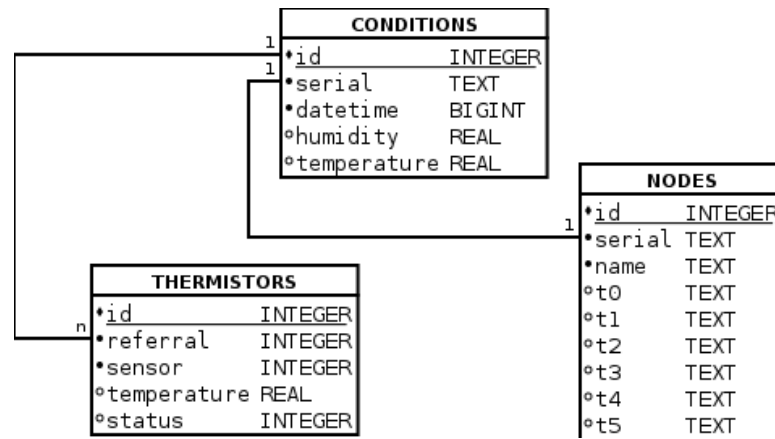
- Πίνακας Thermistors (id, referral, sensor, temperature, status)

Στον πίνακα αυτό αποθηκεύονται οι μετρήσεις στα φύλλα μέσω των thermistors. Κάθε μέτρηση αντιστοιχεί σε μια γραμμή ως εξής: Αναφορά στον πίνακα Conditions (referral), το thermistor που έλαβε (sensor) τη μέτρηση και η τιμή της καταγεγραμμένης θερμοκρασίας (temperature) στο συγκεκριμένο θερμίστορ. Επίσης η στήλη status προσδιορίζει αν η μέτρηση αντιστοιχούσε σε κανονικές συνθήκες (μακριά από το σημείο δρόσου) ή όχι. Το status βοηθά σε πιο εύκολη μελλοντική ανάκτηση των στιγμών που εντοπίστηκε πρόβλημα.

- Πίνακας Nodes (id, serial, name, t0, t1, t2, t3, t4, t5)

Βοηθητικός πίνακας που επιτρέπει την απόδοση φιλικών ονομάτων από το χρήστη στον κόμβο αλλά και σε κάθε ένα από τα θερμίστορ.

Το διάγραμμα συσχετίσεων φαίνεται στην εικόνα 30:



Εικόνα 30. Διάγραμμα οντοτήτων – συσχετίσεων βάσης δεδομένων

### 5.3.3 Σύνδεση με XBee

Η ανάγνωση των δεδομένων που λαμβάνονται στο XBee module απαιτεί πρόσβαση στη θύρα USB του υπολογιστή. Αρχικά η κλάση που αναλαμβάνει την επικοινωνία αυτή θα πρέπει να φορτώσει τη βιβλιοθήκη rxtxSerial. Επειδή το διάβασμα των δεδομένων από τη θύρα είναι συνεχές, πρέπει η διαδικασία να γίνεται σε διαφορετικό thread, το οποίο και καλεί η κλάση XBee.

```

public Xbee() throws XBeeException{
    // Load rxtx library
    System.loadLibrary("rxtxSerial");
    // Create thread
    xbeeThread = new XbeeThread(prefs.get("ComPort",
"/dev/ttyUSB0"), Integer.parseInt(prefs.get("ComBaud", "9600")));
    // Start thread
    xbeeThread.start();
}
  
```

Στη συνέχεια το νέο thread, αναλαμβάνει να διαβάσει τα frames που έχουν αποσταλεί από τους κόμβους και να εξάγει το serial number και το string που είχε δημιουργηθεί στο Arduino. Τα δεδομένα αυτά τέλος τα καταχωρεί στη βάση δεδομένων μαζί με την ημερομηνία κι ώρα λήψης

```

public void run(){
    try {
        // Open port to XBee module
        xbee.open(Port, baudRate);
        while (true) {
            try {
                XBeeResponse response=xbee.getResponse();
                // Inform user that XBee is connected
                MainGui.setStatus("Connected");

                if (response.getApiId() == ApiId.ZNET_RX_RESPONSE) {
  
```

```

        // Get data from XBee
        ZNetRxResponse rx=(ZNetRxResponse)response;
        // Get human-readable datetime and timestamp for DB
        Date date = new Date();
        SimpleDateFormat dt = new SimpleDateFormat("dd/MM/yyyy
HH:mm:ss");

        datetime = date.getTime();

        // Extract from string
        serial =
ByteUtils.toBase16(rx.getRemoteAddress64().getAddress()).replace("0x",
 "").replace(",","");
        data = ByteUtils.toString(rx.getData()).replace(";", "");
        readingData.put(serial, data);

        String[] recString = data.split(":", -1);
        String[] conditions = {recString[0], recString[1]};
        String[] temperatures = new String[recString.length-2];

        // Create new array with thermistor readings
        System.arraycopy(recString, 2, temperatures, 0,
recString.length-2);

        // Store to database
        Database.insertTemperatureReadings(Database.insertEnvironmentStatus(serial,
conditions, datetime), temperatures);

        } else {
            System.out.println("received unexpected packet " +
response.toString());
            MainGui.setStatus("Problem reading packet");
        }
        } catch (XBeeException | SQLException e) {
            MainGui.setStatus("Database or data error");
        }
    }
} catch (XBeeException ex) {
    Logger.getLogger(XbeeThread.class.getName()).log(Level.SEVERE, null,
ex);
    MainGui.setStatus("Unable to open port");
} finally {
    if (xbee.isConnected()) {
        xbee.close();
        MainGui.setStatus("Disconnected");
    }
}
}
}

```

### 5.3.4 Υπολογισμός Σημείου Δρόσου

Ο υπολογισμός του σημείου δρόσου γίνεται από τη μέθοδο `getDewPoint()` της κλάσης `Dev`.

```

public double getDewPoint() {
    double dewPoint = 0.0;
    DecimalFormat df = new DecimalFormat("##.##");
    dewPoint = B * (Math.log(humidity/100) + (A*temperature / (B+temperature))) / (A -
(Math.log(humidity/100) - (A*temperature / (B+temperature))));
}

```

```
        return Double.valueOf(df.format(dewPoint));  
    }
```

### 5.3.2 Εκκίνηση Προγράμματος

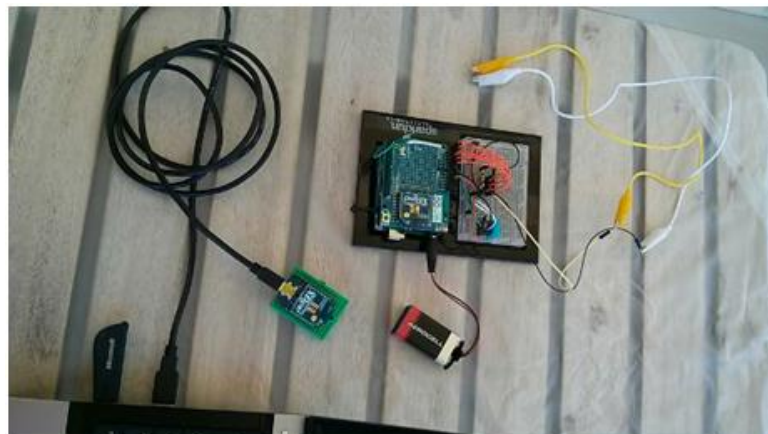
Κατά την εκκίνηση της εφαρμογής, εκτελείται η μέθοδος `main()` της αρχικής κλάσης `PlantMonitor` η οποία εκκινεί την αρχική οθόνη της γραφικής διεπαφής και τη σύνδεση με τη βάση.

```
public static void main(String[] args) {  
    /*  
    Start and show GUI  
    */  
    MainGui myGUI = new MainGui();  
    myGUI.setVisible(true);  
    /*  
    Init db  
    */  
    Database db = new SQLite();  
}
```

Το σύνολο του πηγαίου κώδικα, με τα απαραίτητα σχόλια παρατίθεται στο παράρτημα Α.

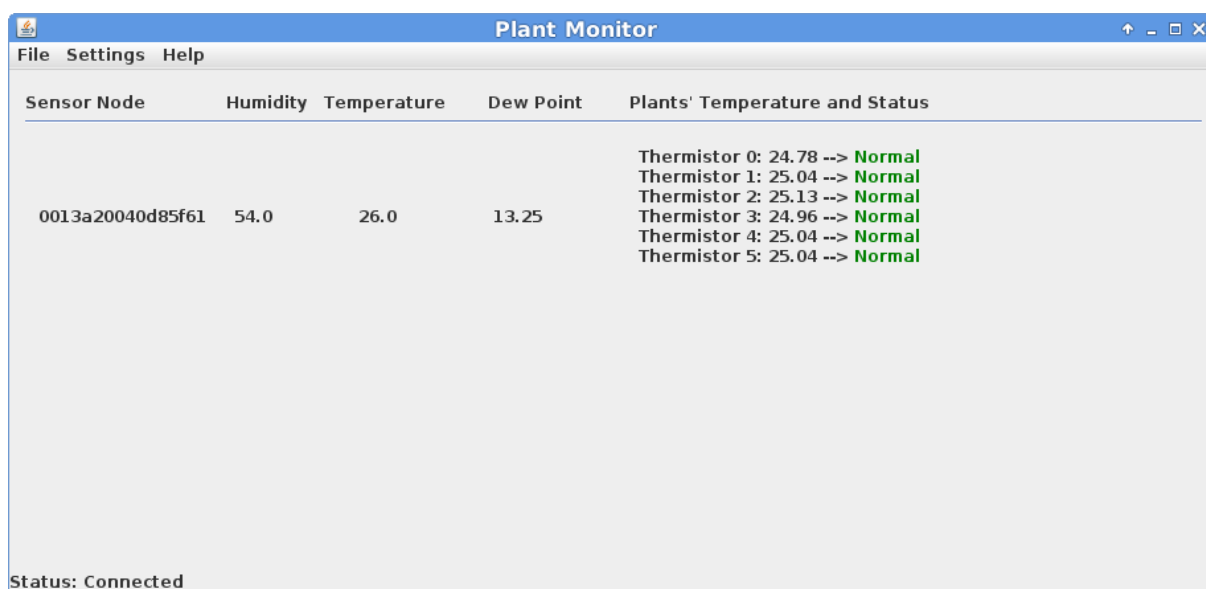
## 5.4 Λειτουργία Εφαρμογής

Αρχικά συνδέουμε τον coordinator μέσω του adapter σε μια USB θύρα του υπολογιστή και τροφοδοτούμε με ρεύμα τον κόμβο που φέρει τους αισθητήρες, όπως φαίνεται στην Εικόνα 31.



Εικόνα 31. Σύνδεση συστήματος

Εκκινώντας την εφαρμογή παρουσιάζεται στο χρήστη μια αρχική κονσόλα όπου εμφανίζονται οι συνδεδεμένοι κόμβοι. Για κάθε κόμβο η εφαρμογή παρουσιάζει τη διεύθυνσή του, τις τιμές σχετικής υγρασίας και θερμοκρασίας του χώρου καθώς και το σημείο δρόσου και τις τιμές της θερμοκρασίας των 6 θερμίστορ. Η εφαρμογή ελέγχει για νέες τιμές κάθε 30 δευτερόλεπτα και ενημερώνει το χρήστη για κάθε συμβάν που έχει να κάνει με την εφαρμογή μέσω της γραμμής κατάστασης, στο κάτω μέρος του παραθύρου:

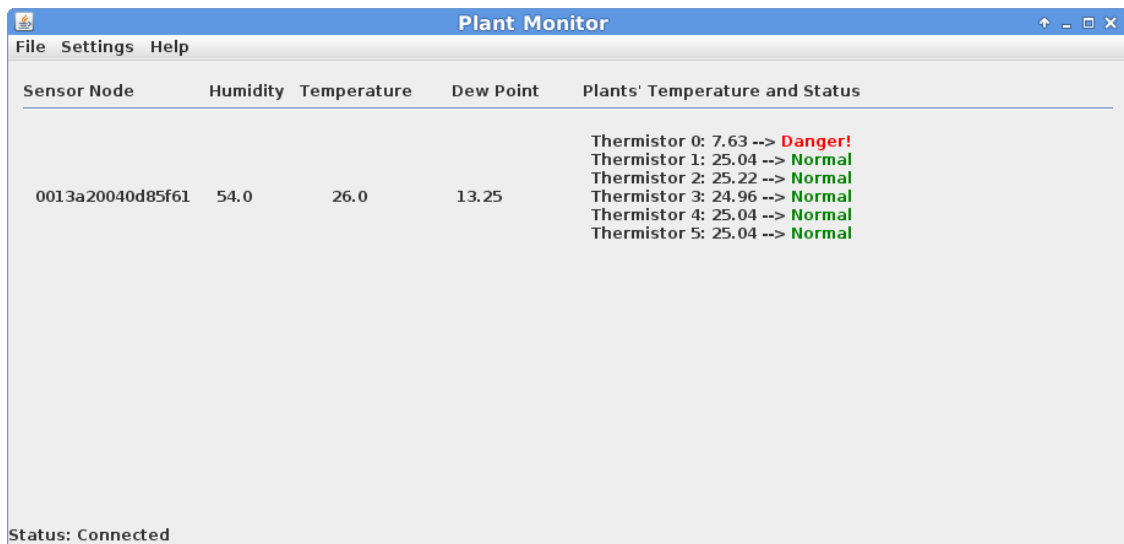


**Εικόνα 32.** Αρχική οθόνη εμφάνισης μετρήσεων

Δίπλα σε κάθε τιμή θερμοκρασίας αναφέρεται αν η τιμή αυτή είναι κανονική και δεν υπάρχει κίνδυνος (πράσινο Normal), αν η θερμοκρασία του θερμίστορ είναι λιγότερο από 4 βαθμούς πάνω από το σημείο δρόσου, οπότε εμφανίζεται η ένδειξη Warning, κι αν η θερμοκρασία έχει φτάσει 1 βαθμό πάνω από το σημείο δρόσου (και χαμηλότερα) εμφανίζεται κόκκινο Danger.

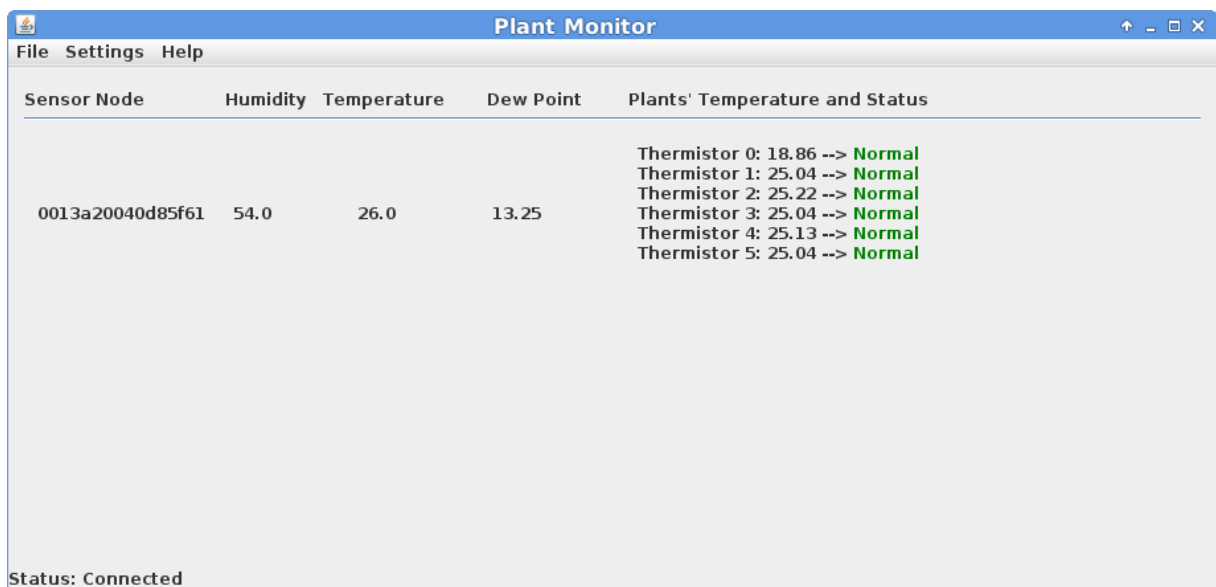
Παρατηρούμε αρχικά πως το σημείο δρόσου έχει υπολογιστεί σωστά. Επίσης παρατηρούμε πως οι 6 αισθητήρες, άρα και τα φυτά, βρίσκονται σε θερμοκρασία εντός κανονικών ορίων. Στη συνέχεια ψύχουμε με τη χρήση πάγου το 1<sup>ο</sup> θερμίστορ, στη θέση 0 ώστε να κατεβάσουμε τη θερμοκρασία του. Παρατηρούμε ότι έχοντας περάσει το κατώφλι, το σύστημα όντως αντιλαμβάνεται το πρόβλημα και μας προειδοποιεί:





**Εικόνα 33.** Προειδοποίηση για επικίνδυνη θερμοκρασία

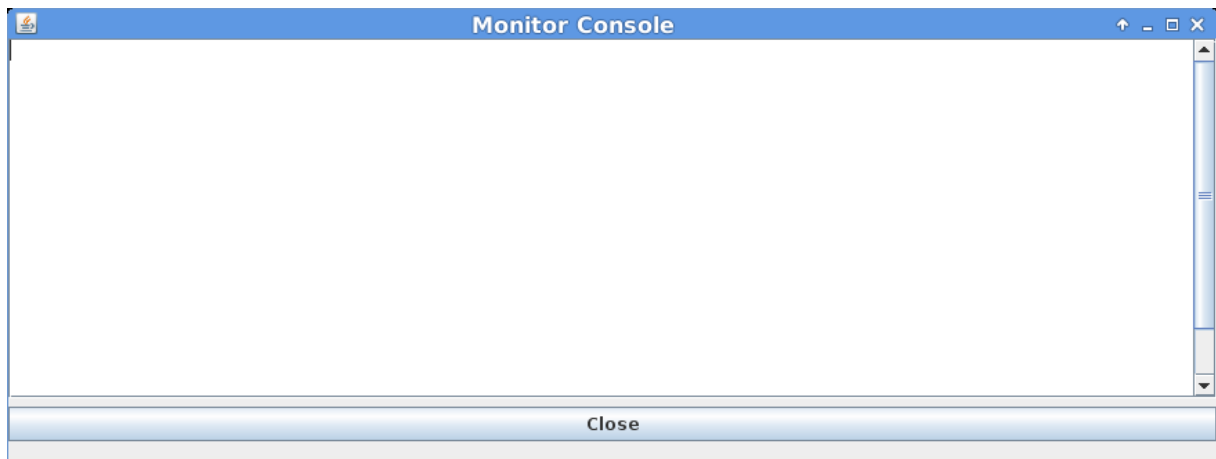
Απομακρύνοντας την πηγή ψύχους, η θερμοκρασία ανεβαίνει εκ νέου και το σύστημα επανέρχεται σε φυσιολογική λειτουργία:



**Εικόνα 34.** Επαναφορά σε κανονικές συνθήκες

Πέρα από αυτή τη βασική κονσόλα, η εφαρμογή περιέχει ένα μενού, από το οποίο παρέχονται οι ακόλουθες δυνατότητες.

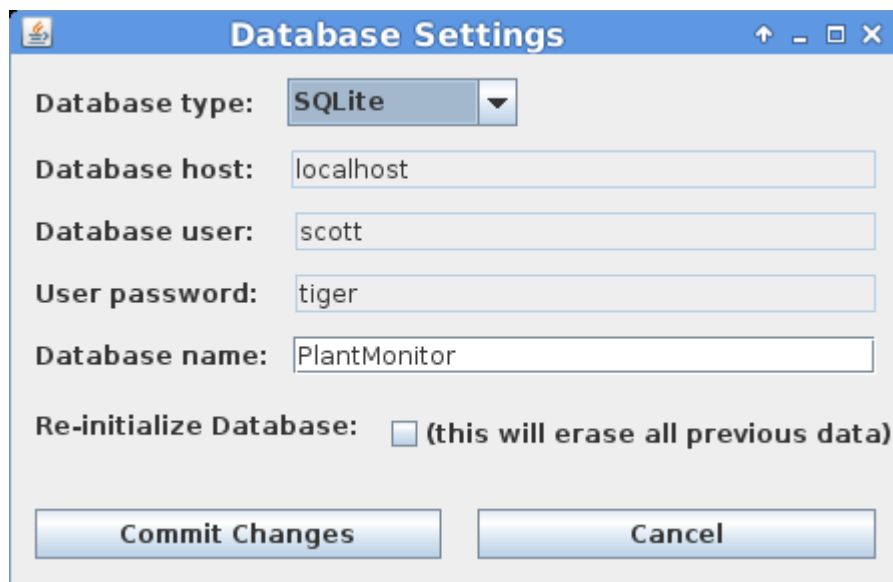
Εάν ο χρήστης θέλει να παρακολουθεί τα μηνύματα λάθους της εφαρμογής που προκύπτουν μετά από exceptions, η επιλογή File→Console αναδρομολογεί το System.out και το System.err στη γραφική κονσόλα, έτσι ώστε να είναι πιο εύκολη η αποσφαλμάτωση του συστήματος.



**Εικόνα 35.** Γραφικό περιβάλλον κονσόλας για τυχόν αποσφαλμάτωση

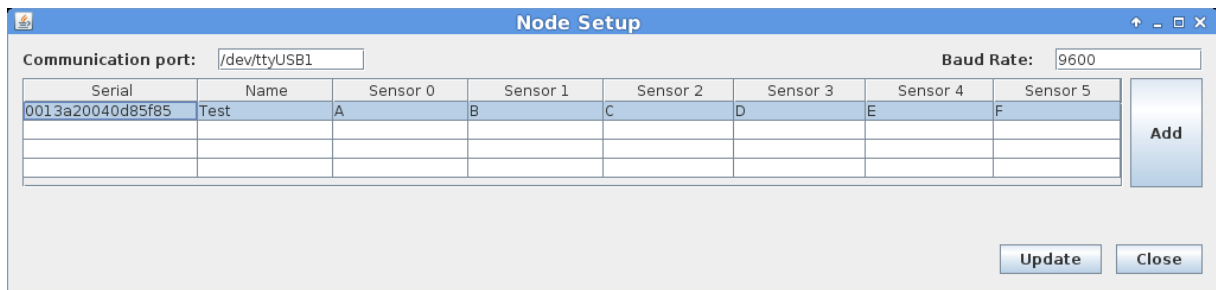
Η επιλογή File → History μπορεί να ανακτήσει παλιότερες μετρήσεις και να τις εξάγει σε αρχείο.

Η επιλογή Settings → Database settings καθορίζει τη βάση δεδομένων που θα χρησιμοποιηθεί και είναι δυνατή η διαγραφή όλων των δεδομένων.



**Εικόνα 36.** Ρυθμίσεις βάσης δεδομένων

Η επιλογή Settings → Node settings επιτρέπει στο χρήστη να δηλώσει τη θύρα επικοινωνίας με το XBee module αλλά και να δώσει φιλικά ονόματα στους αισθητήριους κόμβους.



**Εικόνα 37.** Ρυθμίσεις κόμβων

Τέλος, η επιλογή About επιστρέφει στοιχεία της εφαρμογής και η επιλογή File→ Exit τερματίζει τη λειτουργία.

## 5.5 Έλεγχος Λειτουργίας Συστήματος

Σε πραγματικές συνθήκες το σύστημα θα αποστέλλει μετρήσεις 4 φορές την ώρα. Έτσι κάθε μέρα θα λαμβάνονται 96 μετρήσεις.

Για να ελεγχθεί η αξιοπιστία και η απόδοση σε ένα αρχικό στάδιο, είναι δυνατή η υλοποίηση ενός ελέγχου υπό τη μορφή stress test όπου ο κόμβος θα ρυθμιστεί να αποστέλλει δεδομένα ανά 6 δευτερόλεπτα και η εφαρμογή – δέκτης να ελέγχει ομοίως την είσοδο αν 6 δευτερόλεπτα. Άρα θα πρέπει να καταγράφονται 600 μετρήσεις κάθε ώρα, που αντιστοιχούν στις μετρήσεις που σε κανονική λειτουργία θα αποστέλλονταν σε 6 μέρες και 6 ώρες.

Η δοκιμή του συστήματος έγινε τοποθετώντας τον αισθητήριο κόμβο σε απόσταση 5m, χωρίς άμεση οπτική επαφή, και καταγράφηκαν 880 εγγραφές στον πίνακα CONDITIONS με 1<sup>η</sup> εγγραφή: τη χρονική στιγμή με timestamp 1441425980493 (07:06:20) και τελευταία τη στιγμή 1441431282297 (08:34:42). Η διαφορά των δύο χρονικών στιγμών είναι 5301804ms = 5301sec.

Καθώς ο κόμβος είχε ρυθμιστεί να στέλνει 1 πλαίσιο δεδομένων ανά 6sec, θα έπρεπε να έχουν σταλεί  $5301/6=883$  πλαίσια, άρα να υπάρχουν στον πίνακα CONDITIONS 883 εγγραφείς. Εκτελώντας το ερώτημα:

```
SELECT COUNT(*) FROM CONDITIONS:
```

Το σύστημα επιστέφει 880. Άρα έχουν καταγραφεί 880 από τις 883 αποστολές, ποσοστό που πλησιάζει το 100%. Άρα βάση αυτής της αρχικής δοκιμής το σύστημα αποδεικνύεται πως λειτουργεί αξιόπιστα.

# Κεφάλαιο 6

## Επίλογος

Τα Ασύρματα Δίκτυα Αισθητήρων έχουν δώσει σημαντική ώθηση σε πολλούς τομείς της καθημερινότητας. Παράμετροι πλαισίου είναι δυνατό να ανιχνεύονται και να μεταδίδονται προς επεξεργασία, αποθήκευση και λήψη αποφάσεων. Ένας από τους τομείς ο οποίος παραδοσιακά στη χώρα μας δε φημίζεται ως πεδίο τεχνολογικής έρευνας κι ανάπτυξης, είναι αυτός της γεωργίας. Η εισαγωγή όμως των ΤΠΕ, με βασική τεχνολογία αυτή των ασυρμάτων δικτύων αισθητήρων, εισήγαγε νέους τρόπους αντιμετώπισης προβλημάτων και βελτίωσης των συνθηκών και της παραγωγής, με αποτέλεσμα την ανάπτυξη της γεωργίας ακριβείας.

Στα προηγούμενα κεφάλαια, μελετήθηκε κι αναπτύχθηκε ένα σύστημα συλλογής κι αποστολής δεδομένων πλαισίου, συγκεκριμένα θερμοκρασίας και υγρασίας, με σκοπό τον εντοπισμό σχηματισμού δρόσου σε καλλιέργειες θερμοκηπίων. Για την υλοποίηση του συστήματος χρησιμοποιήθηκε ως ασύρματος κόμβος Ασύρματος Δικτύου Αισθητήρων η ανοικτή πλατφόρμα Arduino. Ο κάθε κόμβος προγραμματίστηκε κατάλληλα και εξοπλίστηκε με ψηφιακό αισθητήρα για τον υπολογισμό της θερμοκρασίας και της σχετικής υγρασίας του χώρου και διατάξεις θερμίστορες οι οποίες καταγράφουν στις θερμοκρασίες στα φυτά. Τα δεδομένα αυτά μεταδίδονται ασύρματα από modules που προστέθηκαν στο Arduino, με χρήση του πρωτοκόλλου ZigBee και λαμβάνονται από εφαρμογή που αναπτύχθηκε σε γλώσσα Java. Η εφαρμογή αυτή αφού λάβει τα δεδομένα υπολογίζει το σημείο δρόσου και προειδοποιεί σε περίπτωση σχηματισμού ή προσέγγισης θερμοκρασίας σχηματισμού δρόσου.

Το σύστημα, στην παρούσα του μορφή αποτελεί πρότυπο του οποίου η μελλοντική εξέλιξη μπορεί να αποτελέσει ιδιαίτερα αποτελεσματικό αρωγό στη διατήρηση ευνοϊκών συνθηκών εντός των θερμοκηπίων με στόχο την προστασία των φυτών και την αύξηση της απόδοσης της καλλιέργειας. Πέρα από την ανίχνευση περίπτωσης δρόσου, η ανοικτότητα της πλατφόρμας και η δυνατότητα επεκτασιμότητας του Arduino επιτρέπει την επέκταση και καθιστά δυνατό τον έλεγχο περισσότερων παραμέτρων πλαισίου καθώς η πλατφόρμα είναι προγραμματιζόμενη μέσω λογισμικού κι επιτρέπει τη σύνδεση και χρήση αρκετών και διαφορετικών αισθητήρων. Επίσης η αντικατάσταση του οικονομικού αλλά μη ακριβούς στις μετρήσεις αισθητήρα DHT11 με τον πιο ακριβή (αλλά όχι ιδιαίτερα οικονομικό) SHT11 αλλά και η περαιτέρω μείωση της

κατανάλωσης ενέργειας μπορούν να καταστήσουν τη λύση ακόμη πιο αποδοτική ιδίως για πιο απαιτητικά περιβάλλοντα.

Τέλος η εφαρμογή είναι εύκολα επεκτάσιμη με χρήση τεχνολογιών όπως RESTful web services. Έτσι οι πληροφορίες μπορούν να γίνουν διαθέσιμες σε συσκευές όπως smartphones, smart watches κ.ο.κ, για την αποδοτικότερη παρακολούθηση, ή να συνδεθεί και με κόμβους οι οποίοι θα λάμβαναν κι εκτελούσαν εντολές όπως το άνοιγμα παραθύρων, εξαερισμού ή όποια άλλη ενέργεια μπορούσε να αντιμετωπίσει την κατάσταση με βέλτιστο τρόπο. Οι κόμβοι αυτοί θα μπορούσαν ακόμη και να ενσωματωθούν στους υπάρχοντες, που έχουν αναπτυχθεί στο πλαίσιο της παρούσας εργασίας, ώστε να είναι δυνατή η τοπική αντιμετώπιση φαινομένων.

Οι τεχνολογίες διάχυτου υπολογισμού έχουν στις μέρες μας καταφέρει να αποτελούν αναπόσπαστο κομμάτι της καθημερινότητας διευκολύνοντας και βελτιώνοντας τις συνθήκες της ανθρώπινης ζωής και δραστηριότητας. Επιδίωξη όλων των εμπλεκόμενων σε θέματα ΤΠΕ είναι η βέλτιστη χρήση τους έτσι ώστε να αποτελέσουν αρωγό στην πρόοδο, την ανάπτυξη και την ευημερία του ανθρώπου και του κοινωνικού συνόλου.

# Παράρτημα Α

## Πηγαίος Κώδικας εφαρμογών

### A.1 Arduino

```
#include <dht.h>
#include <math.h>

// Create DHT object
dht DHT;

//Constant Definition
#define DHT11_PIN 2
#define THERMISTOR_IMPEDANCE 10000

// Get thermistor temperature
double getTemp(int V) {
    double T, R, Rt;
    Rt = THERMISTOR_IMPEDANCE*((1023.0-V)/V);
    R = log(Rt/THERMISTOR_IMPEDANCE);
    T=1/(0.003354016+0.0002569850*R+0.000002620131*R*R+0.0000000383091*R*R*R);
    T = T - 273.15;
    return T;
}

// Set communication Speed
void setup(){
    Serial.begin(9600);
}

void loop(){
    // Read data from DHT pin
    int chk = DHT.read11(DHT11_PIN);
    // Create String
    Serial.print(DHT.humidity, 1);
    Serial.print(":");
    Serial.print(DHT.temperature, 1);
    Serial.print(":");
    Serial.print(getTemp(analogRead(0)));
    Serial.print(":");
    Serial.print(getTemp(analogRead(1)));
    Serial.print(":");
    Serial.print(getTemp(analogRead(2)));
    Serial.print(":");
    Serial.print(getTemp(analogRead(3)));
    Serial.print(":");
    Serial.print(getTemp(analogRead(4)));
    Serial.print(":");
    Serial.print(getTemp(analogRead(5)));
    Serial.println("");
    delay(900000);
}
```

## A.2 JAVA

### ΚΛΑΣΗ PlantMonitor

```
package plantmonitor;

import db.Database;
import db.SQLite;
import gui.MainGui;
/**
 *
 * @author caterina
 */
/*
Main Class
*/
public class PlantMonitor {

    /*
    Start Application
    */
    public static void main(String[] args) {
        /*
        Start and show GUI
        */
        MainGui myGUI = new MainGui();
        myGUI.setVisible(true);
        /*
        Init db
        */
        Database db = new SQLite();
    }
}
```



## ΚΛΑΣΗ Dew

```
package plantmonitor;

import java.text.DecimalFormat;

/**
 *
 * @author caterina
 */
public class Dew {

    private double temperature; // Temperature
    private double humidity;    // Relative humidity

    private final double A=17.625; //Constant A of Magnus Formula
    private final double B=243.04; //Constant B of Magnus Formula

    public Dew(double temperature, double humidity){
        this.temperature = temperature;
        this.humidity = humidity;
    }

    public double getTemperature() {
        return temperature;
    }

    public void setTemperature(double temperature) {
        this.temperature = temperature;
    }

    public double getHumidity() {
        return humidity;
    }

    public void setHumidity(double humidity) {
        this.humidity = humidity;
    }

    /**
     Calculate and Return dew point
    */
    public double getDewPoint(){
        double dewPoint = 0.0;
        DecimalFormat df = new DecimalFormat("##.##");
        dewPoint = B*(Math.log(humidity/100)+(A*temperature/(B+temperature)))/(A-
(Math.log(humidity/100)-(A*temperature/(B+temperature))));
        return Double.valueOf(df.format(dewPoint));
    }
}
```

## **ΚΛΑΣΗ Settings**

```
package plantmonitor;

import java.util.prefs.Preferences;

/**
 *
 * @author caterina
 */
public class Settings {

    Preferences prefs = Preferences.userNodeForPackage(Settings.class);
    final String DBTYPE = "DatabaseType";
    final String DBUSER = "DatabaseUser";
    final String DBPASS = "DatabasePass";
    final String DBHOST = "DatabaseHost";
    final String DBSCHEMA = "DatabaseSchema";
    final String COMPORT = "ComPort";
    final String COMBAUD = "ComBaud";
}
```

## **XBee.java**

```
package xbee;

import com.rapplogic.xbee.api.XBeeException;
import java.util.Map;
import java.util.prefs.Preferences;
import plantmonitor.Settings;

public class Xbee {

    private Preferences prefs = Preferences.userNodeForPackage(Settings.class);
    private XbeeThread xbeeThread;

    // Constructor that reads from settings
    public Xbee() throws XBeeException{
        // Load rxtx library
        System.loadLibrary("rxtxSerial");
        // Create thread
        xbeeThread = new XbeeThread(prefs.get("ComPort", "/dev/ttyUSB0"),
Integer.parseInt(prefs.get("ComBaud", "9600")));
        // Start thread
        xbeeThread.start();
    }

    // Constructor takes comm parameters
    public Xbee(String port, int baud) throws XBeeException{
        System.loadLibrary("rxtxSerial");
        xbeeThread = new XbeeThread(port, baud);
        xbeeThread.start();
    }

    public Map getData() throws XBeeException{
        return xbeeThread.getData();
    }
}
```

## XBeeThread.java

```
package xbee;

import com.rapplogic.xbee.api.ApiId;
import com.rapplogic.xbee.api.XBee;
import com.rapplogic.xbee.api.XBeeException;
import com.rapplogic.xbee.api.XBeeResponse;
import com.rapplogic.xbee.api.zigbee.ZNetRxResponse;
import com.rapplogic.xbee.util.ByteUtils;
import db.Database;
import gui.MainGui;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;

public class XbeeThread extends Thread {

    String Port;
    int baudRate;
    XBee xbee;
    private long datetime;
    private String serial;
    private String data;
    Map readingData;

    public XbeeThread(String Port, int baudRate) throws XBeeException{
        this.Port = Port;
        this.baudRate = baudRate;
        xbee = new XBee();
        readingData = new HashMap();
    }

    public String getPort() {
        return Port;
    }

    public void setPort(String Port) {
        this.Port = Port;
    }

    public int getBaudRate() {
        return baudRate;
    }

    public void setBaudRate(int baudRate) {
        this.baudRate = baudRate;
    }

    @Override
    public void run(){
        try {
            // Open port to XBee module
            xbee.open(Port, baudRate);
            while (true) {
                try {
                    XBeeResponse response=xbee.getResponse();
                    // Inform user that XBee is connected
```

```

        MainGui.setStatus("Connected");

        if (response.getApiId() == ApiId.ZNET_RX_RESPONSE) {
            // Get data from XBee
            ZNetRxResponse rx=(ZNetRxResponse)response;
            // Get human-readable datetime and timestamp for DB
            Date date = new Date();
            SimpleDateFormat dt = new SimpleDateFormat("dd/MM/yyyy
HH:mm:ss");

            datetime = date.getTime();

            // Extract from string
            serial =
ByteUtils.toBase16(rx.getRemoteAddress64().getAddress()).replace("0x",
"".replace(", ", ""));

            data = ByteUtils.toString(rx.getData()).replace(";", "");
            readingData.put(serial, data);

            String[] recString = data.split(":", -1);
            String[] conditions = {recString[0], recString[1]};
            String[] temperatures = new String[recString.length-2];

            // Create new array with thermistor readings
            System.arraycopy(recString, 2, temperatures, 0,
recString.length-2);

            // Store to database

Database.insertTemperatureReadings(Database.insertEnvironmentStatus(serial,
conditions, datetime), temperatures);

                } else {
                    System.out.println("received unexpected packet " +
response.toString());
                    MainGui.setStatus("Problem reading packet");
                }
            } catch (XBeeException | SQLException e) {
                MainGui.setStatus("Database or data error");
            }
        }
    } catch (XBeeException ex) {
        Logger.getLogger(XbeeThread.class.getName()).log(Level.SEVERE, null,
ex);
        MainGui.setStatus("Unable to open port");
    } finally {
        if (xbee.isConnected()) {
            xbee.close();
            MainGui.setStatus("Disconnected");
        }
    }
}

public Map getData(){
    return readingData;
}
}

```

## Database.java

```
package db;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

/**
 *
 * @author caterina
 */
public abstract class Database {
    protected static Connection conn = null;

    /**
     * Method to execute query
     */
    static public void dbExecute(String sql) throws SQLException{
        Statement stmt;
        stmt = conn.createStatement();
        stmt.executeUpdate(sql);
    }

    /**
     * Insert method that returns primary key value
     */
    static public int dbInsert(String sql) throws SQLException{
        Statement stmt;
        stmt = conn.createStatement();
        stmt.executeUpdate(sql);
        return stmt.getGeneratedKeys().getInt("last_insert_rowid()");
    }

    /**
     * Insert greenhouse temperature and relative humidity to database
     */
    static public int insertEnvironmentStatus(String Sensor, String[] data, long
datetime) throws SQLException{
        String sql = "INSERT INTO CONDITIONS (SERIAL, DATETIME, HUMIDITY,
TEMPERATURE) VALUES ('"+Sensor+"', '"+datetime+"', '"+data[0]+"', '"+data[1]+"')";
        return dbInsert(sql);
    }

    /**
     * Insert temperature readings from themistors
     */
    static public void insertTemperatureReadings(int record, String[]
temperatures) throws SQLException{
        int sensorNumber = temperatures.length;
        String sql = "INSERT INTO THERMISTORS (REFERRAL, TEMPERATURE, SENSOR)
VALUES ";
        for(int i=0; i<sensorNumber; i++){
            sql = sql+"('"+record+"', '"+temperatures[i]+"', '"+i+"')";
            if(i<sensorNumber-1){
                sql = sql+", ";
            }
        }
        dbExecute(sql);
    }
}
```

```
    static public double getTemperature(String serial, long datetime) throws
SQLException{
        double val=0.0;
        String sql = "SELECT temperature FROM temperatures WHERE
serial='"+serial+"' AND datetime='"+datetime;
        Statement stmt;
        stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(sql);
        while(rs.next()){
            val= rs.getDouble("temperature");
        }
        return val;
    }

    static public void truncatedB() throws SQLException{
        dbExecute("DELETE FROM THERMISTORS");
        dbExecute("DELETE FROM CONDITIONS");
    }
}
```

## SQLite.java

```
package db;

import gui.MainGui;
import java.sql.*;
/**
 *
 * @author caterina
 */
public class SQLite extends Database{

    public SQLite(){
        Statement stmt = null;
        String sql = null;
        try {
            Class.forName("org.sqlite.JDBC");
            conn = DriverManager.getConnection("jdbc:sqlite:PlantMonitor.db");

            stmt = conn.createStatement();
            sql = "CREATE TABLE IF NOT EXISTS NODES " +
                "(ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
                " NAME TEXT NOT NULL, " +
                " SERIAL CHAR(16), " +
                " T0 TEXT, T1 TEXT, T2 TEXT, T3 TEXT, T4 TEXT, T5 TEXT)";
            stmt.executeUpdate(sql);
            stmt.close();

            stmt = conn.createStatement();
            sql = "CREATE TABLE IF NOT EXISTS CONDITIONS " +
                "(ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
                " SERIAL CHAR(16) NOT NULL, " +
                " DATETIME BIGINT NOT NULL, " +
                " HUMIDITY REAL, " +
                " TEMPERATURE REAL)";
            stmt.executeUpdate(sql);
            stmt.close();

            stmt = conn.createStatement();
            sql = "CREATE TABLE IF NOT EXISTS THERMISTORS " +
                "(ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
                " REFERRAL INTEGER NOT NULL, " +
                " SENSOR INTEGER, " +
                " STATUS INTEGER, " +
                " TEMPERATURE REAL)";
            stmt.executeUpdate(sql);
            stmt.close();
        } catch ( ClassNotFoundException | SQLException e ) {
            System.err.println( "Unable to open database: "+e.getClass().getName() +
                ": " + e.getMessage() );
            System.exit(0);
        }
        MainGui.setStatus("Database opened successfully");
    }
}
```



# Βιβλιογραφία

1. Abowd, Gregory D., et al. "Towards a better understanding of context and context-awareness." *Handheld and ubiquitous computing*. Springer Berlin Heidelberg, 1999.
2. Akyildiz, Ian F., Tommaso Melodia, and Kaushik R. Chowdhury. "A survey on wireless multimedia sensor networks." *Computer networks* 51.4 (2007): 921-960.
3. Akyildiz, Ian F, Weilian Su Yogesh Sankarasubramaniam and Erdal Cayirci. «A Survey on Sensor Network». *IEEE Communications Magazine*, August 2002.
4. Alemdar, Hande, and Cem Ersoy. "Wireless sensor networks for healthcare: A survey." *Computer Networks* 54.15 (2010): 2688-2710.
5. Amir Hoshang Kioumars, Dr. Liqiong Tang, «Wireless Network for Health Monitoring: Heart Rate and Temperature Sensor». *Fifth International Conference on Sensing Technology*, 2011
6. Aosong (Guangzhou) Electronics, DHT11, <http://www.aosong.com/en/products/details.asp?id=109>. (Πρόσβαση: 01/09/2015)
7. Arduino, "http://www.arduino.cc", (Πρόσβαση 31/08/2015)
8. Atzori, Luigi, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey." *Computer networks* 54.15 (2010): 2787-2805.
9. Baggio, Aline. "Wireless sensor networks in precision agriculture." *ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2005)*, Stockholm, Sweden. 2005..
10. Baldauf, Matthias, Schahram Dustdar, and Florian Rosenberg. "A survey on context-aware systems." *International Journal of Ad Hoc and Ubiquitous Computing* 2.4 (2007): 263-277.
11. Baronti, Paolo, et al. "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards." *Computer communications* 30.7 (2007): 1655-1695.
12. Chaudhary, D. D., S. P. Nayse, and L. M. Waghmare. "Application of wireless sensor networks for greenhouse parameter control in precision agriculture." *International Journal of Wireless & Mobile Networks (IJWMN)* Vol 3.1 (2011): 140-149.
13. Chong, Chee-Yee, and Srikanta P. Kumar. "Sensor networks: evolution, opportunities, and challenges." *Proceedings of the IEEE* 91.8 (2003): 1247-1256.

14. Corke, Peter, et al. "Environmental wireless sensor networks." *Proceedings of the IEEE* 98.11 (2010): 1903-1917.
15. Crow, Brian P., et al. "IEEE 802.11 wireless local area networks." *Communications Magazine*, IEEE 35.9 (1997): 116-126.
16. Damas, Miguel, et al. "HidroBus® system: fieldbus for integrated management of extensive areas of irrigated land." *Microprocessors and Microsystems* 25.3 (2001): 177-184.
17. Digi International, XBee Products, "<http://www.digi.com/products/XBee>" (Πρόσβαση: 31/08/2015)
18. El Emary, Ibrahim MM, and S. Ramakrishnan, eds. *Wireless Sensor Networks: From Theory to Applications*. CRC Press, 2013.
19. García-Hernández, Carlos F., et al. "Wireless sensor networks and applications: a survey." *IJCSNS International Journal of Computer Science and Network Security* 7.3 (2007): 264-273.
20. Haartsen, Jaap C. "The Bluetooth radio system." *Personal Communications*, IEEE 7.1 (2000): 28-36.
21. Hussain, Sajid, Scott Schaffner, and Dyllon Moseychuck. "Applications of wireless sensor networks and RFID in a smart home environment." *Communication Networks and Services Research Conference, 2009. CNSR'09. Seventh Annual. IEEE, 2009*.
22. Jensen, Allan Leck, et al. "Pl@ nteInfo®—a web-based system for personalised decision support in crop management." *Computers and Electronics in Agriculture* 25.3 (2000): 271-293.
23. Kushner, David. "The making of arduino." *IEEE Spectrum* 26 (2011).
24. Lawrence, Mark G. "The relationship between relative humidity and the dewpoint temperature in moist air: A simple conversion and applications." *Bulletin of the American Meteorological Society* 86.2 (2005): 225-233.
25. Lee, Jin-Shyan, Yu-Wei Su, and Chung-Chou Shen. "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi." *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE. IEEE, 2007*.
26. Lee, Sang Hyuk, et al. "Wireless sensor network design for tactical military applications: remote large-scale environments." *Military Communications Conference, 2009. MILCOM 2009. IEEE. IEEE, 2009*.
27. Lee, Won Suk, Thomas F. Burks, and John K. Schueller. "Silage yield monitoring system." *ASABE paper 021165* (2002).

28. Mulligan, Geoff. "The 6LoWPAN architecture." Proceedings of the 4th workshop on Embedded networked sensors. ACM, 2007.
29. Noble, Joshua. Programming Interactivity: A Designer's Guide to Processing, Arduino, and Openframeworks. " O'Reilly Media, Inc.", 2009.
30. Ramanathan, Kumaran, and Bengt Danielsson. "Principles and applications of thermal biosensors." Biosensors and Bioelectronics 16.6 (2001): 417-423.
31. Ruiz-Garcia, Luis, et al. "A review of wireless sensor technologies and applications in agriculture and food industry: state of the art and current trends." Sensors 9.6 (2009): 4728-4750.
32. Sensirion, SHT1x - Digital Humidity & Temperature Sensor (RH/T), Online at: "<http://www.sensirion.com/en/products/humidity-temperature/humidity-temperature-sensor-sht1x/>" (Πρόσβαση 01/09/2015)
33. Sohraby, Kazem, Daniel Minoli, and Taieb Znati. Wireless sensor networks: technology, protocols, and applications. John Wiley & Sons, 2007.
34. Song, Jianping, et al. "WirelessHART: Applying wireless technology in real-time industrial process control." Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS'08. IEEE. IEEE, 2008.
35. Tong, Alan. "Improving the accuracy of temperature measurements." Sensor Review 21.3 (2001): 193-198.
36. Tseng, Yi-Hsien, Eric Hsiao-kuang Wu, and Gen-Huey Chen. "Maximum traffic scheduling and capacity analysis for IEEE 802.15. 3 high data rate MAC protocol." Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th. Vol. 3. IEEE, 2003.
37. Van Herwaarden, A. W., and P. M. Sarro. "Thermal sensors based on the Seebeck effect." Sensors and Actuators 10.3 (1986): 321-346.
38. Warneke, Brett, and Kristofer SJ Pister. "MEMS for distributed wireless sensor networks." Electronics, Circuits and Systems, 2002. 9th International Conference on. Vol. 1. IEEE, 2002.
39. Weiser, Mark. "The computer for the 21st century." Scientific american 265.3 (1991): 94-104.
40. Yick Jennifer, Mukherjee Biswanath and Ghosal Dipak "Wireless sensor network survey". Computer Networks 52, 2008. 2292-2330.
41. Zhang, Naiqian, Maohua Wang, and Ning Wang. "Precision agriculture - a worldwide overview." Computers and electronics in agriculture 36.2 (2002): 113-132.