

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΕΠΙΣΤΗΜΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Πτυχιακή Εργασία

Μελέτη Παραμέτρων Λειτουργίας Ζεύξεων Πειραματικού
Δικτύου Ιονίου Πανεπιστημίου

Σοφρονία Φουτσιτζή Α.Μ.: Π2017063

Επιβλέπων: Κωνσταντίνος Οικονόμου

18 Ιουνίου 2021

Επιβλέπων

Κωνσταντίνος Οικονόμου, Καθηγητής,
Ιόνιο Πανεπιστήμιο

Τριμελής Επιτροπή

Κωνσταντίνος Οικονόμου, Καθηγητής,
Ιόνιο Πανεπιστήμιο

Θεόδωρος Ανδρόνικος, Αναπληρωτής Καθηγητής,
Ιόνιο Πανεπιστήμιο

Μάρκος Αυλωνίτης, Αναπληρωτής Καθηγητής,
Ιόνιο Πανεπιστήμιο

Η προσπάθεια αυτή είναι αφιερωμένη στον αδερφό μου Φουτσιτζή Ευστράτιο
Χρήστο, την οικογένεια, τους φίλους και τους καθηγητές μου.

Περίληψη

Τα Ασύρματα Δίκτυα Αισθητήρων WSN είναι μία καινοτομία, η οποία αρχίζει και καταλαμβάνει όλο και περισσότερο σημαντικό ρόλο στον τομέα της επιστήμης της πληροφορικής, αλλά και διάφορων άλλων κλάδων.

Σκοπός της συγκεκριμένης πτυχιακής εργασίας αποτέλεσε η μελέτη του πειραματικού δικτύου που έχει εγκατασταθεί στο Ιόνιο Πανεπιστήμιο. Έτσι, παρατηρήθηκε και αναλύθηκε η επιρροή διαφόρων παραγόντων φυσικών, προγραμματιστικών και περιβαλλοντολογικών των οποίων η επιρροή στην ποιότητα μεταξύ των ζεύξεων του δικτύου ήταν πιθανή.

Οι έντεκα κόμβοι που αποτελούνται από ένα Arduino Uno, μία κεραία XBee S2C Pro Zig-Bee, έναν αισθητήρα θερμοκρασίας και υγρασίας DHT22 και μία κάρτα SD, τοποθετήθηκαν σε διαφορετικά μέρη και δωμάτια του ίδιου κτιρίου. Έπειτα, αναζητήθηκαν οι ζεύξεις μεταξύ των κόμβων, ώστε στην συνέχεια να μελετηθούν. Υλοποιώντας έναν αλγόριθμο τύπου Flooding κάθε κόμβος αναγνώρισε και κατέγραψε τους γείτονές του, ενώ δημιούργησε, παράλληλα και σχέσεις πατέρα-παιδιού μεταξύ των κόμβων. Το αποτέλεσμα της δημιουργίας αυτών των σχέσεων ήταν η υλοποίηση ενός Spanning Tree, που χρησιμοποιήθηκε στην συνέχεια για την συλλογή και την συγκέντρωση των δεδομένων στον κόμβο coordinator, καθώς τα δεδομένα επέστρεφαν μέσω των συγκεκριμένων καναλιών από τους κόμβους φύλλα στη ρίζα του δέντρου.

Μελετήθηκαν και δόθηκε έμφαση σε διάφορους παράγοντες για την αξιολόγηση της ποιότητας των ζεύξεων ενός πειραματικού δικτύου WSN. Αρχικά, μελετήθηκε η λήψη ή όχι επιτυχημένων πακέτων μεταξύ δύο κόμβων. Οι επιτυχημένες μεταδόσεις μεταξύ δύο κόμβων προβάλλουν μία ποιοτική ζεύξη μεταξύ τους. Έπειτα, για κάθε μία επιτυχημένη μετάδοση πακέτου, υπολογίστηκαν διάφορες τιμές, όπως το Received Strength Signal, η τιμή της θερμοκρασίας της κεραίας, του περιβάλλοντος και η υγρασία αυτού. Οι τιμές των θερμοκρασιών και της υγρασίας δεν είχαν ιδιαίτερη απόκλιση τόσο μεταξύ του ίδιου του κόμβου, όσο και με τις καταγραφές των υπολοίπων. Για τον λόγο αυτό, υλοποιήθηκε μία τεχνητή μεταβολή των περιβαλλοντολογικών

συνθηκών, σκοπεύοντας στην αναζήτηση ή όχι επιρροής αυτών στον δίκτυο. Αντίθετα, η τιμή του RSS αυξομειωνόταν ανάλογα με την ζεύξη και εξαρτώταν, κυρίως, από την απόσταση των κόμβων. Τέλος, ανακτήθηκαν και μέσω του προγράμματος XCTU οι τιμές του Link Quality Indicator, που αποδίδει το ίδιο το πρόγραμμα μέσω της κεραίας και αφορά στην ποιότητα των συνδέσεων μεταξύ των κόμβων. Τα δεδομένα από το πρόγραμμα XCTU και από τον κώδικα μελετήθηκαν και απέδωσαν τα δεδομένα, προβάλλοντας την ποιότητα της εκάστοτε ζεύξης.

Abstract

Wireless sensor networks (WSN) are an innovation that plays an increasingly important role not only in the information science field but also in multiple others. The purpose of this dissertation was to study the experimental network installed in the Ionian University. Thus, the affect of various natural, coding and environmental factors that may affect the link quality of the network were observed and analysed. Eleven nodes consisting of an Arduino Uno, an XBee S2C Pro ZigBee antenna, a DHT22 temperature and humidity sensor and a micro SD card were installed in different parts and rooms of the same building. Afterwards, the links between the nodes were identified and studied. By applying a flooding algorithm, each node identified and registered its neighbours while, in parallel, creating father-child relationships between the nodes. The outcome of the creation of those relationships was the application of a spanning tree, which was used to collect and gather data at the coordinator node, as the data was returning through specific channels from the leaf nodes to the root of the tree. In order to evaluate the link quality of the WSN, various factors were studied and emphasised. Initially, the reception or not of successful packages between two nodes was observed. Successful transmissions between two nodes demonstrate a quality link among them. Then, for each successful package transmission, various values like the received strength signal (RSS), the temperature of the antenna and the environment along with the humidity were calculated. The temperature and humidity values did not have significant variance, both for the same node and among different nodes. For this reason, the environmental conditions were altered artificially with the aim to check whether they affect the network. On the contrary, the RSS values varied depending on the link and were mainly affected by the distance between the nodes and , as observed, the rapid increase of temperature. Finally, the link quality indicator values were also retrieved using the XCTU software. These values are assigned by the software via the antenna and demonstrate the quality of the links among the nodes. The data from

the XCTU software and the code was studied and produced the data to demonstrate the quality of each link.

Πρόλογος και Ευχαριστίες

Η προσπάθεια υλοποίησης της συγκεκριμένης πτυχιακής εργασίας αποτέλεσε μία προσπάθεια με αρκετές δυσκολίες, αλλά και ένα όμορφο ταξίδι που ολοκλήρωσε το κεφάλαιο των σπουδών μου με τον καλύτερο δυνατό τρόπο. Όπως σε κάθε σημαντικό στάδιο της ζωής μας έτσι και σε αυτό δεν έλειπαν οι δυσκολίες και οι απογοητεύσεις, οι οποίες σου δίνουν, με τον καιρό, θέληση και πείσμα για να συνεχίσεις και να δώσεις τον καλύτερό σου εαυτό, για τους ανθρώπους που πιστεύουν σε εσένα αλλά και για τον ίδιο σου τον εαυτό, γνωρίζοντας πως οι κόποι σου αυτοί θα ανταμειφθούν και θα σε βοηθήσουν στο να δημιουργήσεις κάτι όμορφο και ουσιαστικό. Έτσι ήταν το ταξίδι της εκπόνησης της πτυχιακής μου εργασίας και ελπίζω ότι έδωσα τον καλύτερο μου εαυτό για ένα αποτέλεσμα που να με αντιπροσωπεύει.

Ωστόσο, το αποτέλεσμα δεν ήταν αποκλειστικά δική μου υπόθεση. Αντίθετα ήταν ένα αποτέλεσμα συλλογικής προσπάθειας όλων εκείνων που πίστεψαν και συνέβαλαν με τον δικό τους τρόπο στην υλοποίηση του. Το αποτέλεσμα δεν θα ήταν το ίδιο χωρίς την βοήθεια και την στήριξη της οικογένειας και των φίλων μου. Ένα ιδιαίτερο ευχαριστώ θα ήθελα να πω στους συμφοιτητές, πλέον συναδέλφους, και φίλους που με βοήθησαν και με στήριξαν σε όλη την διάρκεια της υλοποίησης της πτυχιακής εργασίας με τον δικό τους τρόπο, στον Νέμανια Γιέβτις, τον Γεώργιο Σιδερά, την Ευτυχία Φαρμάκη, την Ανθή Καπετάνιου, την Ελευθερία Ασλανίδου, την Μαρία Κεσίδου, τον Αλέξανδρο Πολίτη και τον Σπύρο Βέργη.

Για τη στήριξη και τη βοήθεια που μου προσέφερε όλο αυτόν τον καιρό, θα ήθελα να ευχαριστήσω τον καθηγητή μου κ.Κωνσταντίνο Οικονόμου, αλλά και όλους τους καθηγητές που γνώρισα και συνάντησα σε όλη την διάρκεια των σπουδών μου, για όλες τις γνώσεις, τις πληροφορίες αλλά και τις εμπειρίες που μου χάρισαν. Επίσης, θα ήθελα να ευχαριστήσω όλους όσων η στήριξη και η πίστη σε εμένα με έκαναν να συνεχίσω, να προσπαθήσω και να ανταπεξέλθω σε κάθε δυσκολία που αντιμετώπισα.

Κέρκυρα, 18 Ιουνίου 2021

Σοφρονία Φουτσιτζή

Περιεχόμενα

A´	Εισαγωγή	1
A´.1	ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΡΕΥΝΑ	1
A´.2	ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΝΕΙΣΦΟΡΑ	3
A´.3	ΑΝΤΙΣΤΟΙΧΟΥΜΕΝΑ ΚΕΦΑΛΑΙΑ	3
B´	Ασύρματα Δίκτυα Αισθητήρων	4
B´.1	ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΕΝΟΣ ΔΙΚΤΥΟΥ ΑΣΥΡΜΑΤΩΝ ΑΙΣΘΗΤΗΡΩΝ	5
B´.2	ΜΟΝΤΕΛΟ OSI	6
B´.3	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΔΙΚΤΥΩΝ ΑΣΥΡΜΑΤΩΝ ΑΙΣΘΗΤΗΡΩΝ	7
B´.4	ΕΦΑΡΜΟΓΕΣ	7
B´.5	ΤΟΠΟΛΟΓΙΑ ΚΑΙ ΠΕΡΙΓΡΑΦΗ ΠΕΙΡΑΜΑΤΙΚΟΥ ΔΙΚΤΥΟΥ	9
Γ´	Υλικός Εξοπλισμός	11
Γ´.1	ARDUINO UNO	11
Γ´.2	XBEE S2C PRO ZIGBEE RF MODULE	13
Γ´.2.1	Πρωτόκολλο ZigBee	17
Γ´.3	ΑΙΣΘΗΤΗΡΑΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΑΙ ΥΓΡΑΣΙΑΣ DHT22	18
Δ´	Υλοποίηση του δικτύου	20
Δ´.1	ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΤΡΟΠΟΥ ΣΤΑΛΟΓΗΣ ΤΩΝ ΔΕΔΟΜΕΝΩΝ	20
Δ´.2	ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ ΣΥΛΛΟΓΗΣ	22
Δ´.3	ΔΥΣΚΟΛΙΕΣ ΚΑΙ ΠΡΟΒΛΗΜΑΤΑ ΚΑΤΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ	23

Ε΄	Αποτελέσματα της υλοποίησης	25
Ε΄.1	ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΣΩ ΤΗΣ ΣΥΝΕΧΟΜΕΝΗΣ ΑΠΟΣΤΟΛΗΣ ΠΑΚΕΤΩΝ	25
Ε΄.2	RECEIVED STRENGTH SIGNAL	28
Ε΄.3	ΑΠΟΤΕΛΕΣΜΑΤΑ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΑΙ ΥΓΡΑΣΙΑΣ	29
Ε΄.4	ΑΠΟΤΕΛΕΣΜΑ ΜΕΤΑΒΟΛΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΑΙ ΥΓΡΑΣΙΑΣ	31
Ε΄.4.1	Απώλεια Μηνυμάτων	31
Ε΄.4.2	Αποτελέσματα Μεταβολής Θερμοκρασίας και Υγρασίας στην τιμή RSS	32
Γ΄	Σύνοψη και Συμπεράσμα	34
Γ΄.1	ΣΥΝΟΨΗ ΤΗΣ ΕΡΓΑΣΙΑΣ	34
Γ΄.2	ΣΥΜΠΕΡΑΣΜΑΤΑ	34
Γ΄.3	ΣΥΓΝΕΙΣΦΟΡΑ	35
	Βιβλιογραφία	36
	Παράρτημα Α΄	40
	Συντμήσεις	56
	Γλωσσάρι Ξενικών Όρων	57

Κατάλογος Σχημάτων

B'.1	Το επίπεδα του μοντέλου OSI.	6
B'.2	Η τοπολογία των κόμβων στο κτίριο.	9
B'.3	Το Arduino Uno πάνω στο οποίο έχει τοποθετηθεί η κεραία XBee S2C Pro ZigBee και η SD.	10
Γ'.4	Arduino Uno.	13
Γ'.5	Το εσωτερικό περιβάλλον του προγράμματος XCTU.	14
Γ'.6	Η κεραία XBee S2C Pro ZigBee.	15
Γ'.7	Ο αισθητήρας DHT22 και μία αντίσταση 10Kohm.	18
Δ'.8	Το δέντρο που προκύπτει από το πρώτο μέρος του κώδικα.	21
Ε'.9	Οι ζεύξεις μεταξύ των κόμβων και η τιμή LQI, όπως παρουσιάζεται μέσα από το πρόγραμμα XCTU.	28
Ε'.10	Ο μέσος όρος της απόλυτης τιμής του RSS μεταξύ των κόμβων του δικτύου. .	29
Ε'.11	Ο μέσος όρος θερμοκρασίας των κόμβων.	30
Ε'.12	Ο μέσος όρος θερμοκρασίας των κόμβων.	30
Ε'.13	Οι απόλυτες τιμές του RSS για κάθε πακέτο σε δύο διαφορετικά πεδία θερμοκρασίας του κόμβου παραλήπτη.	32
Ε'.14	Έντονη μεταβολή της τιμής RSS μεταξύ μίας ζεύξης κατά την μείωση της υγρασίας.	33
Ε'.15	Έντονη μεταβολή της τιμής RSS μεταξύ μίας ζεύξης κατά την αύξηση της θερμοκρασίας.	33

Κατάλογος Πινάκων

<i>Γ'.1 Τα pins μία κεραίας XBee S2C Pro ZigBee.</i>	16
<i>Γ'.2 Τα χαρακτηριστικά του αισθητήρα DHT22.</i>	19
<i>Ε'.1 Πίνακας Γειννίαςης μεταξύ των κόμβων του δικτύου.</i>	26
<i>Ε'.2 Το ποσοστό των επιτυχημένων μεταδόσεων μεταξύ των κόμβων.</i>	27
<i>Ε'.3 Ο μέσος όρος της RSS και της απώλειας πακέτων σε σχέση με τον μέσο όρο των τιμών της θερμοκρασίας και υγρασίας του περιβάλλοντος.</i>	32

Κεφάλαιο Α΄

Εισαγωγή

ΤΑ Δίκτυα Ασύρματων Αισθητήρων Wireless Sensor Networks αρχίζουν και διεισδύουν όλο και περισσότερο σε διάφορους τομείς της καθημερινότητας των ανθρώπων. Η χρήση τους δεν περιορίζεται μόνο στον τομέα της πληροφορικής και της τεχνολογίας, αντιθέτως συναντάται σε διάφορους κλάδους όπως η ιατρική, η βιομηχανία, η ασφάλεια αλλά και για περιβαλλοντολογικούς σκοπούς. Το χαμηλό τους κόστος, η ελάχιστη κατανάλωση ενέργειας, αλλά, κυρίως το εξαιρετικά μικρό μέγεθος των κόμβων και των αισθητήρων είναι βασικοί λόγοι αξιοποίησης τους, σε περιπτώσεις και περιοχές όπου είναι δυσπρόσιτες για τον άνθρωπο. Στην συγκεκριμένη πτυχιακή εργασία μελετάται, αρχικά, η λειτουργία και η εφαρμογή των δικτύων WSN, ενώ στην συνέχεια ειδικεύεται στην μελέτη του πειραματικού δικτύου που έχει εγκατασταθεί στο Ιόνιο Πανεπιστήμιο [30].

Α΄.1 Βιβλιογραφική Έρευνα

Η εκπόνηση της εργασίας υλοποιήθηκε για την μελέτη των παραμέτρων λειτουργίας των ζεύξεων μεταξύ των κόμβων του πειραματικού δικτύου, επομένως, αρχικά αναζητήθηκαν από παρόμοιες μελέτες και έρευνες παράγοντες που είναι πιθανόν να επηρεάζουν την επικοινωνία και την ποιότητα αυτής μεταξύ των κεραιών (XBee S2C Pro RF Module), αλλά και γενικότερα των ασύρματων δικτύων αισθητήρων. Στόχος της βιβλιογραφικής έρευνας ήταν η γενικότερη κατανόηση των δικτύων αυτών, αλλά και των παραγόντων που είναι πιθανό να ασκούν επιρροή σε αυτά. Οι παρατηρήσεις και οι πειραματισμοί των αντίστοιχων εργασιών αποτέλεσαν ένα έναυσμα στην εύρεση νέων παραγόντων που θα μπορούσαν να επηρεάζουν

μία ζεύξη, αλλά και στην επιβεβαίωση των καταγεγραμμένων επάνω στο πειραματικό δίκτυο του Ιονίου Πανεπιστημίου. Παρατηρήθηκε ότι σε πολλές έρευνες ο βασικός παράγοντας που επηρεάζει την επικοινωνία μεταξύ των κόμβων του δικτύου είναι κυρίως η απόσταση [19]. Σε πολλές εργασίες παρατηρήθηκε, όπως ήταν άλλωστε και πιθανόν, αλλαγή στην ποιότητα των ζεύξεων με την αυξομείωση της απόστασης μεταξύ τους [31]. Επίσης, πολλές έρευνες ασχολήθηκαν και με την επιρροή του περιβάλλοντος στην ποιότητα των ζεύξεων. Συγκεκριμένα, πολλές έρευνες διεξήγαγαν τα ίδια πειράματα τόσο σε εξωτερικό όσο και σε εσωτερικό χώρο. Τα αποτελέσματα που προέκυψαν έδειξαν ότι το RSS (Received Strength Signal) κυμαινόταν σε χαμηλότερα, συνεπώς και καλύτερα επίπεδα όταν οι κόμβοι τοποθετούνταν σε εξωτερικό χώρο, με αντίστοιχες αποστάσεις [14] [24]. Ενώ παράλληλα, παρατηρήθηκε ότι στο εξωτερικό περιβάλλον η σύνδεση μεταξύ των δύο κόμβων παρέμενε ενεργή σε μεγαλύτερη απόσταση από ό,τι στο εσωτερικό περιβάλλον. Ενώ, άλλες έρευνες τα arduino με τις κεραίες τοποθετήθηκαν ακόμα και πάνω στο γρασίδι, ώστε να παρατηρηθούν τυχόν αλλαγές που μπορούν να προκληθούν. Συγκεκριμένα, στην προαναφερθείσα έρευνα έγιναν δύο τύπου πειράματα, εκ των οποίων το ένα ήταν στο ίδιο επίπεδο με την επιφάνεια της γης, στο γρασίδι όπως χαρακτηριστικά αναφέρεται, ενώ, στο δεύτερο πείραμα οι κόμβοι τοποθετήθηκαν σε λίγο υψηλότερες επιφάνειες αλλά επίσης σε εξωτερικό χώρο, ενώ οι περιβαλλοντικές συνθήκες ήταν αντίστοιχες, καθώς η θερμοκρασία και η υγρασία κυμαίνονταν σε αντίστοιχα επίπεδα. Για την μελέτη της ποιότητας του σήματος και την καταμέτρηση του RSSI χρησιμοποιήθηκε και η απόσταση μεταξύ των κόμβων, η οποία προσεγγίστηκε από την συνάρτηση `pulseIn()` [26], που χρησιμοποιήθηκε για την αποτύπωση της διάρκειας του παλμού ενός σήματος από τον κόμβο αποστολέα προς τον δέκτη. Στην πρώτη εκδοχή του πειράματος, δηλαδή στην περίπτωση που οι δύο κόμβοι βρίσκονταν στο έδαφος, η χαμηλότερη τιμή με την οποία χαρακτηρίστηκε το RSSI αποτυπώθηκε ως -85dBm, ενώ η ακτίνα επικοινωνίας των κόμβων περιορίστηκε στα εννέα μέτρα. Αντίθετα, παρατηρήθηκε κατά το δεύτερο πείραμα μία ραγδαία αύξηση της εμβέλειας του σήματος μεταξύ των δύο κόμβων, καθώς η σύνδεση διακόπηκε στα 26 μέτρα, σχεδόν τρεις φορές μεγαλύτερη εμβέλεια και η ισχύς του σήματος δεν έφτασε χαμηλότερα από τα -76dBm. Ωστόσο, στην δεύτερη εκδοχή του πειράματος, παρά την σύνδεση των κόμβων με μεγάλη απόσταση, παρατηρήθηκε ότι η ποιότητά της ήταν σε πολύ χαμηλά επίπεδα όταν η απόσταση έφτανε και ξεπερνούσε τα δεκαπέντε μέτρα [12].

Α'.2 Αποτελέσματα και Συνεισφορά

Η ποιότητα των ζεύξεων του πειραματικού δικτύου στο Ιόνιο Πανεπιστήμιο κρίθηκε πιθανόν να επηρεάζεται από διάφορους παράγοντες περιβαλλοντολογικούς και μη. Για τον λόγο αυτό διερευνήθηκαν διάφοροι παράμετροι των οποίων η επίδραση κρίθηκε πιθανή. Τα δεδομένα που καταγράφηκαν για κάθε ζεύξη είναι το ποσοστό των επιτυχημένων μεταδόσεων μεταξύ των κόμβων αλλά και η τιμή του Received Strength Signal που αποτυπώνει την ισχύ του τελευταίου hop του πακέτου που λήφθηκε. Επιπλέον, καταγράφηκαν και η θερμοκρασία και η υγρασία του περιβάλλοντος στο οποίο βρισκόταν κάθε κόμβος όταν λάμβανε κάποιο μήνυμα. Τα αποτελέσματα που προέκυψαν μέσω αυτών των καταγραφών πρόβαλαν την βασική σημασία της απόστασης μεταξύ των δύο κόμβων, καθώς αυτή ήταν ο βασικός παράγοντας που επηρέασε τις ζεύξεις. Η θερμοκρασία και η υγρασία παρέμειναν σε σταθερά επίπεδα, καθώς οι κόμβοι βρίσκονταν στο εσωτερικό περιβάλλον του κτηρίου Αρεταίος του Ιονίου Πανεπιστημίου. Με βάση τις τιμές των επιτυχημένων αποστολών των πακέτων αλλά και της τιμής RSS και LQI προέκυψε και η ποιότητα μεταξύ των ζεύξεων, καθώς αυτοί ήταν και οι βασικοί παράγοντες που την επηρέασαν.

Α'.3 Αντιστοιχούμενα Κεφάλαια

Η δομή της εργασίας είναι τέτοια ώστε να εισάγει ομαλά τον αναγνώστη στο κύριο θέμα. Αρχικά, στο Κεφάλαιο Β' αναλύεται ο ορισμός των Δικτύων Ασύρματων Αισθητήρων, τα χαρακτηριστικά και οι εφαρμογές τους σε διάφορους κλάδους της καθημερινότητας των ανθρώπων. Επιπλέον, περιγράφεται και το Πειραματικό δίκτυο του Ιονίου Πανεπιστημίου [30], το οποίο αποτέλεσε τη βάση της συγκεκριμένης πτυχιακής εργασίας, καθώς σε αυτό υλοποιήθηκαν τα πειράματα και η μελέτη αυτών.

Στην συνέχεια, στο Κεφάλαιο Γ' αναλύεται και περιγράφεται όλος ο εξοπλισμός που χρησιμοποιήθηκε στο δίκτυο. Συγκεκριμένα, «διασπάστηκε» ο κάθε κόμβος στα τρία βασικά στοιχεία από τα οποία αποτελείται: στον μικροελεγκτή, το Arduino Uno στην συγκεκριμένη περίπτωση, την κεραία XBee S2C Pro ZigBee, αλλά και τον αισθητήρα θερμοκρασίας και υγρασίας DHT22. Ο εξοπλισμός αυτός αξιοποιήθηκε, όπως παρουσιάζεται στο Κεφάλαιο Δ', ώστε να προκύψουν τα αποτελέσματα και τα συμπεράσματα που παρουσιάζονται στο Κεφάλαιο Ε'.

Κεφάλαιο Β΄

Ασύρματα Δίκτυα Αισθητήρων

ΤΑ ασύρματα δίκτυα αισθητήρων Wireless Sensor Networks αποτελούν μία καινοτομία, η οποία αρχίζει και εξελίσσεται όλο και περισσότερο έχοντας τεράστια εφαρμογή, πλέον, σε πολλά πεδία της τεχνολογίας και γενικότερα της επιστήμης των ηλεκτρονικών υπολογιστών. Η ραγδαία ανάπτυξη των τηλεπικοινωνιών αλλά και οι συνεχείς βελτιώσεις του υλικού συνέβαλαν στην επέκταση της χρήσης τους [21]. Η λειτουργία των ασύρματων δικτύων αισθητήρων ορίζεται ως η χρήση υπολογιστικών συστημάτων για την επίτευξη ενός συγκεκριμένου στόχου [6]. Ένα ασύρματο δίκτυο αισθητήρων αποτελείται από κόμβους, οι οποίοι μπορούν να είναι Arduino Uno όπως στην περίπτωση της συγκεκριμένης πτυχιακής εργασίας, οι οποίοι συνδέονται με έναν ή και περισσότερους αισθητήρες. Σκοπός των κόμβων σε ένα ασύρματο δίκτυο αισθητήρων είναι κυρίως η παρακολούθηση φυσικών ή περιβαλλοντολογικών συνθηκών που επιτυγχάνεται με την συνεργασία των κόμβων και μεταφορά των δεδομένων σε μία συγκεκριμένη τοποθεσία/ κόμβο. Σε ένα δίκτυο, ο αριθμός τους ποικίλει, καθώς αυτοί μπορεί να είναι ελάχιστοι ή ακόμα και χιλιάδες. Στα δίκτυα αισθητήρων η μετάδοση των δεδομένων γίνεται με τη χρήση ηλεκτρομαγνητικών κυμάτων (RF). Έτσι σε κάθε κόμβο υπάρχει ένας πομποδέκτης (transceiver) που λειτουργεί συνήθως στην περιοχή 433MHz μέχρι 2.4GHz και μετατρέπει τα ψηφιακά δεδομένα σε ραδιοκύματα και το αντίστροφο.

Β'.1 Βασικά χαρακτηριστικά ενός δικτύου ασύρματων αισθητήρων

Η εξέλιξη και η ραγδαία ανάπτυξη της τεχνολογίας οδήγησαν στην δημιουργία αισθητήρων με χαμηλό κόστος, μικρό μέγεθος, υψηλή ισχύ και ενεργειακή αυτονομία, που να είναι σε θέση να επικοινωνούν μεταξύ τους [3]. Τα δίκτυα ασύρματων αισθητήρων όπως, προαναφέρθηκε, αποτελούνται από πολλούς κόμβους αισθητήρες, οι οποίοι καταγράφουν διάφορα φαινόμενα και στην συνέχεια τα συγκεντρώνουν σε έναν από τους κόμβους του δικτύου, τον κόμβο συλλέκτη (snik). Η θέση των κόμβων και κατ' επέκταση των αισθητήρων του δικτύου είναι πιθανόν να είναι είτε προκαθορισμένη είτε όχι. Το γεγονός αυτό επιτρέπει την τυχαία ανάπτυξή τους σε μη προσπελάσιμα περιβάλλοντα από τον άνθρωπο, προϋποθέτοντας ότι ο αλγόριθμος του δικτύου έχει την δυνατότητα να το οργανώσει. Ωστόσο, τα ασύρματα δίκτυα αισθητήρων χαρακτηρίζονται, επίσης, και από την περιορισμένη ενεργειακή κάλυψη των κόμβων, με αποτέλεσμα την διαρκή μεταβολή του δικτύου. Οι βασικές ιδιότητες των ασύρματων δικτύων αισθητήρων συνοψίζονται στα παρακάτω σημεία [3] [33]:

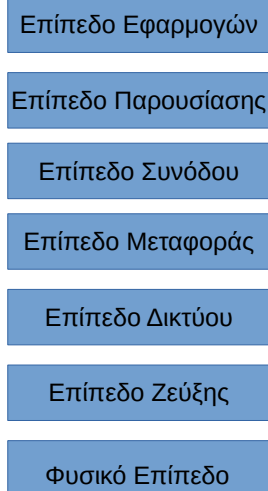
- Δυνατότητα αυτό-οργάνωσης
- επικοινωνία περιορισμένου βεληνεκούς και δρομολόγηση πολλαπλών αλμάτων (multi-hop)
- πυκνή τοποθέτηση των κόμβων και συνεργατική προσπάθεια
- συχνά μεταβαλλόμενη τοπολογία λόγω εξασθένησης του σήματος και αποτυχίας των κόμβων
- περιορισμοί στην ενέργεια, την ισχύ εκπομπής, την μνήμη και την υπολογιστική δυνατότητα
- πιθανότατα έλλειψη γενικής αναγνώρισης (identification) των κόμβων λόγω υψηλού overhead και μεγάλου αριθμού κόμβων.

Η κατασκευή και ο σχεδιασμός ενός δικτύου ασύρματων αισθητήρων πρέπει να υπόκειται σε κάποιους βασικούς κανόνες, καθώς τέτοιου είδους δίκτυα επηρεάζονται από ποικίλους παράγοντες. Αρχικά, είναι πολύ σύνηθες φαινόμενο η ύπαρξη σφαλμάτων, καθώς ένα τέτοιο δίκτυο είναι αφενώς ιδιαίτερα ευαίσθητο σε φυσικές παρεμβολές και αφετέρου συχνή είναι η έλλειψη ενέργειας από τους κόμβους. Συνεπώς, κρίνεται βασικό κάθε κόμβος να είναι

ενεργειακά αυτόνομος, να καταναλώνει ελάχιστη ενέργεια και να είναι ευπροσάρμοστος από θέμα περιβάλλοντος. Επιπλέον, ένα τέτοιο δίκτυο αποτελείται από, όπως προαναφέρθηκε, ελάχιστους έως και χιλιάδες κόμβους, γεγονός που πρέπει να ληφθεί υπόψη τόσο σε θέμα επεκτασιμότητας του δικτύου όσο και σε θέμα κόστους, καθώς αυτό δεν θα πρέπει να είναι υπέρογκο. Τέλος, σημαντικός παράγοντας χρήσης τέτοιων δικτύων είναι το μέγεθος των αισθητήρων, αλλά και γενικότερα του εκάστοτε κόμβου, καθώς κάθε κόμβος είναι επιθυμητό να είναι ιδιαίτερα μικρός σε μέγεθος.

Β'.2 Μοντέλο OSI

Τα WSN ακολουθούν το μοντέλο του Διεθνούς Οργανισμού Τυποποίησης, το μοντέλο ISO επτά επιπέδων. Στο συγκεκριμένο μοντέλο κάθε επίπεδο επιτελεί μία λειτουργία την οποία μπορεί να αξιοποιήσει κάθε ανώτερο επίπεδο. Στα WSN, ωστόσο, απουσιάζουν τα επίπεδα Παρουσίασης και Συνόδου, ενώ ενσωματώνονται παράλληλα τρία άλλα επίπεδα, το επίπεδο διαχείρισης ενέργειας, φορητότητας και διαχείρισης λειτουργιών, που συμβάλλουν στην αύξηση της ποιότητας του δικτύου. Το μοντέλο OSI αποτελείται από επτά επίπεδα, όπως φαίνεται και στο σχήμα Β'.1 [38] [18] [13].



Σχήμα Β'.1: Το επίπεδα του μοντέλου OSI.

Ουσιαστικά, όπως προαναφέρθηκε, στο μοντέλο αυτό οι λειτουργίες ξεκινούν από το χαμηλό-

τερο επίπεδο και απολήγουν στο ανώτερο, καθώς λειτουργίες που υλοποιούν τα χαμηλότερα επίπεδα αξιοποιούν τα υψηλότερα. Το φυσικό επίπεδο αφορά στο υλικό, στο φυσικό μέσο και στις λειτουργίες και προδιαγραφές αυτού. Στο επίπεδο ζεύξης υποστηρίζεται η υπηρεσία της διευθυνσιοδότησης, ουσιαστικά, παρέχεται η φυσική διεύθυνση του κόμβου προορισμού MAC [11]. Ενώ, στο παραπάνω επίπεδο, το Δικτύου, καθορίζεται ο τρόπος και ο αλγόριθμος δρομολόγησης των πακέτων. Έτσι, δρούν τα τρία χαμηλότερα επίπεδα του μοντέλου που αναφέρονται στο υλικό και ακολουθούν τα τέσσερα επόμενα, που ο ρόλος τους αναφέρεται στο λογισμικό. Το επίπεδο μεταφοράς διασφαλίζει την επικοινωνία μεταξύ των κόμβων. Τα δύο επόμενα επίπεδα που ουσιαστικά δεν χρησιμοποιούνται στα WSN αναφέρονται, αρχικά το επίπεδο συνόδου στον έλεγχο του διαλόγου μεταξύ των άκρων της επικοινωνίας, ενώ το επίπεδο παρουσίασης στην αναπαράσταση/ παρουσίαση των δεδομένων. Το ανώτερο επίπεδο Εφαρμογών είναι ο βασικός τρόπος διάδρασης του χρήστη με την εφαρμογή, η οποία του παρέχει τις πληροφορίες του δικτύου.

Β'.3 Αρχιτεκτονική Δικτύων Ασύρματων Αισθητήρων

Τα WSN βασίζονται, κυρίως, στην λειτουργία που θα υλοποιήσει το δίκτυο, καθώς ανάλογα με τις συνθήκες, τις απαιτήσεις και τις λειτουργίες θα σχεδιαστεί και η δομή του. Ωστόσο, σε κάθε δίκτυο αισθητήρων υπάρχουν κάποια βασικά στοιχεία τα οποία παραμένουν ίδια. Ένα από αυτά είναι οι κόμβοι και η λειτουργία τους σε ευρύτερη έννοια. Τα είδη των κόμβων παραμένουν σταθερά και χωρίζονται σε δύο κατηγορίες. Στην πρώτη κατηγορία κόμβων είναι οι κόμβοι που λειτουργούν ως πηγή των δεδομένων και είναι οι κόμβοι οι οποίοι συλλέγουν τις πληροφορίες, τις οποίες διαβιβάζουν στους δρομολογητές κόμβους που είναι και αυτοί που μεταφέρουν την πληροφορία από κόμβο σε κόμβο, με τελικό απολήκτη τον κόμβο, sink, όπου θα γίνει η συλλογή και ίσως επεξεργασία και ανάλυση των δεδομένων [4]. Ωστόσο, στα WSN ο κάθε κόμβος μπορεί να επεξεργαστεί τα δεδομένα πριν τα αποστείλει. Ο sink μπορεί ουσιαστικά να αποτελεί μέρος του δικτύου ως κόμβος αλλά είναι πιθανόν και να είναι μία εξωτερική συσκευή με την οποία το δίκτυο αλληλεπιδρά.

Β'.4 Εφαρμογές

Τα δίκτυα αισθητήρων έχουν διάφορες εφαρμογές και λειτουργίες ανάλογα με τον αισθητήρα που χρησιμοποιούν. Οι αισθητήρες χρησιμοποιούνται για την παρακολούθηση κυρίως φυσι-

κών φαινομένων αλλά και μη. Συγκεκριμένα, υπάρχουν αισθητήρες που παρακολουθούν την θερμοκρασία, την υγρασία, την κίνηση οχημάτων, την πίεση, τον ήχο/ θόρυβο, ανιχνεύουν το φως, την παρουσία συγκεκριμένων αντικειμένων και διάφορα άλλα φαινόμενα, καθώς και συνδυασμούς αυτών [2]. Στην συγκεκριμένη πτυχιακή εργασία, καταγράφεται τόσο η θερμοκρασία όσο και η υγρασία στην οποία βρίσκεται κάθε κόμβος, χωρίς ωστόσο την χρήση διαφορετικών αισθητήρων. Το γεγονός αυτό είναι εξαιρετικά σημαντικό τόσο από θέμα χώρου και τοποθέτησης, όσο και από οικονομικής πλευράς. Γενικότερα, όμως, τα ασύρματα δίκτυα αισθητήρων έχουν εφαρμογές κυρίως σε περιβάλλοντα που είναι δύσκολο να διερευνηθούν ή δεν είναι προσβάσιμα από άνθρωπο [15] και μπορούν να κατηγοριοποιηθούν ως δίκτυα ανίχνευσης ή επίβλεψης [3]. Ειδικότερα, τα συγκεκριμένα δίκτυα χρησιμοποιούνται, κυρίως, για περιβαλλοντικούς σκοπούς, όπως την πρόβλεψη καταστροφικών φαινομένων για παράδειγμα πλημμύρες και γενικότερα για την πρόγνωση των καιρικών συνθηκών αλλά και άλλων φαινομένων όπως κατολισθήσεις, τσουνάμι ή και έκρηξη κάποιου ηφαιστείου χάρη στους αισθητήρες που χρησιμοποιούν [1]. Ωστόσο, η χρήση των δικτύων ασύρματων αισθητήρων δεν περιορίζεται αποκλειστικά στα φυσικά φαινόμενα, καθώς η ακρίβεια και οι δυνατότητές τους αυξάνονται, κατακτούν θέση σε διάφορους τομείς της καθημερινότητας των ανθρώπων, από πράγματα πολύ σημαντικά έως και πιο απλά, όπως για παράδειγμα στην υλοποίηση σηματοδοτών για τη ρύθμιση της κυκλοφορίας [35]. Ένας «χώρος» που όλο και περισσότερο ενσωματώνει τα τελευταία χρόνια νέα τεχνολογικά επιτεύγματα είναι αυτός της ιατρικής επιστήμης. Η παραγωγή αισθητήρων εξαιρετικά μικρού μεγέθους που θα μπορούν να ενσωματωθούν στο σώμα του ασθενούς είναι μία πρόκληση και ένας μελλοντικός στόχος, καθώς θα μπορούσε να αποτελέσει σημαντικό παράγοντα στην παρακολούθηση ασθενών με διάφορα νοσήματα τόσο στα πλαίσια του νοσοκομείου όσο και σε όλη την διάρκεια της ημέρας του ασθενούς [34]. Το γεγονός αυτό, όμως, είναι διφορούμενο, καθώς προκύπτουν διάφορα νομικά ζητήματα [10] [9]. Ταυτόχρονα, σημαντική θέση καταλαμβάνουν και στον τομέα της βιομηχανικής παραγωγής και γενικότερα της βιομηχανίας. Η χρήση των WSN στην βιομηχανία γίνεται κυρίως για την επίβλεψη της παραγωγής, αλλά και για την ασφάλεια του προσωπικού, καθώς λειτουργούν διαρκώς, με αποτέλεσμα να ανιχνεύουν τυχόν επιπλοκές που προκύπτουν στην παραγωγή, όπως βλάβη σε κάποιο μηχάνημα ή και επιπλοκές που μπορούν να θέσουν σε κίνδυνο την υγεία των εργαζομένων, όπως ραγδαία αύξηση θερμοκρασίας ή πίεσης [3]. Ενώ, ένας κλάδος που αξιοποιεί πλήρως τις δυνατότητες των ασύρματων δικτύων αισθητήρων είναι αυτός των υδατοκαλλιεργειών, καθώς τα προϊόντα των καλλιεργειών έχουν εξαιρετικά μεγάλη ζήτηση. Το γεγονός αυτό δημιουργεί την επιτακτική ανάγκη μεγαλύτερης παραγωγής, βελτιώνοντας όσο το δυνατόν περισσότερο την διαδικασία και το περιβάλλον της [8].

Β'.5 Τοπολογία και Περιγραφή Πειραματικού Δικτύου

Το δίκτυο που υλοποιήθηκε στα πλαίσια εκπόνησης της πτυχιακής εργασίας αποτέλεσε ένα μέρος ενός προϋπάρχοντος δικτύου που βρίσκεται στο Ιόνιο Πανεπιστήμιο (7, Πλατεία Τσιριγώτη, Κέρκυρα) [30]. Το πειραματικό Δίκτυο του Ιονίου Πανεπιστημίου απαρτίζεται συνολικά από τριάντα κόμβους, οι οποίοι βρίσκονται σε διάφορα σημεία και κτίρια της πλατείας. Ωστόσο, στην συγκεκριμένη πτυχιακή μελετήθηκε ένα μέρος του συνολικού δικτύου που είχε υλοποιηθεί αρχικά. Συγκεκριμένα, χρησιμοποιήθηκαν έντεκα arduino uno ως κόμβοι του δικτύου. Οι κόμβοι αυτοί βρίσκονται σε διάφορα σημεία ενός από τα κτίρια της πλατείας Τσιριγώτη, το κτίριο Αρεταίος (Σχήμα Β'.2). Ειδικότερα, στον δεύτερο όροφο του κτιρίου έχουν τοποθετηθεί εννέα κόμβοι, ενώ στον τρίτο όροφο υπάρχουν τρεις κόμβοι. Στον πρώτο όροφο είναι τοποθετημένος και ο κόμβος που έχει οριστεί ως coordinator στο συγκεκριμένο δίκτυο, ώστε να είναι στην εμβέλειά του όσο το δυνατόν μεγαλύτερο πλήθος κόμβων, ώστε να περιοριστεί και το πλήθος των hop μεταξύ αυτού και των πιο απομακρυσμένων κόμβων. Ωστόσο, στα συγκεκριμένα πειράματα, με τη χρήση του συγκεκριμένου αριθμού κόμβων αλλά και τη τοποθέτησή τους, υπήρξαν διάφοροι κόμβοι, όπως ο κόμβος με αναγνωριστικό «2» ή με αναγνωριστικό «3», οι οποίοι προέκυψε να έχουν περισσότερους κόμβους ως γείτονες, στην εμβέλειά τους.



Σχήμα Β'.2: Η τοπολογία των κόμβων στο κτίριο.

Οι κόμβοι με μαύρο χρώμα είναι τοποθετημένοι στον Δεύτερο Όροφο, ενώ με κόκκινο είναι τοποθετημένοι στον Τρίτο.

Κάθε κόμβος του δικτύου χρησιμοποιεί μία μπαταρία (Power Bank) ως πηγή ρεύματος, ενώ σε κάθε Arduino Uno, είναι τοποθετημένη μία κεραία Xbee S2C Pro ZigBee, όπως φαίνεται και στο σχήμα Β'.3, ώστε να επιτυγχάνεται η επικοινωνία μεταξύ των κόμβων. Επιπλέον, σε κάθε κόμβο υπάρχει ένας αισθητήρας θερμοκρασίας και υγρασίας DHT22, αλλά και μία κάρτα micro SD με χωρητικότητα 8GB. Η SD χρησιμοποιήθηκε για την αποθήκευση της πληροφορίας σε κάθε κόμβο πριν την συλλογή της από τον coordinator. Αυτό υλοποιήθηκε, καθώς η πληροφορία, που πρέπει να καταγράφει κάθε κόμβος, είναι σχεδόν βέβαιο ότι θα ξεπερνάει τον διαθέσιμο αποθηκευτικό χώρο που προσφέρει ένα Arduino Uno μετά την φόρτωση του προγράμματος, αλλά και το μέγεθος του μηνύματος που μπορεί να αποστείλει και να λάβει ένας κόμβος του δικτύου. Επομένως, κάθε κόμβος αποθηκεύει τις πληροφορίες σε μία κάρτα SD μέχρι να λάβει το αίτημα/μήνυμα να προωθήσει όσες πληροφορίες έχει καταγράψει στον coordinator.



Σχήμα Β'.3: Το Arduino Uno πάνω στο οποίο έχει τοποθετηθεί η κεραία XBee S2C Pro ZigBee και η SD.

Κεφάλαιο Γ΄

Υλικός Εξοπλισμός

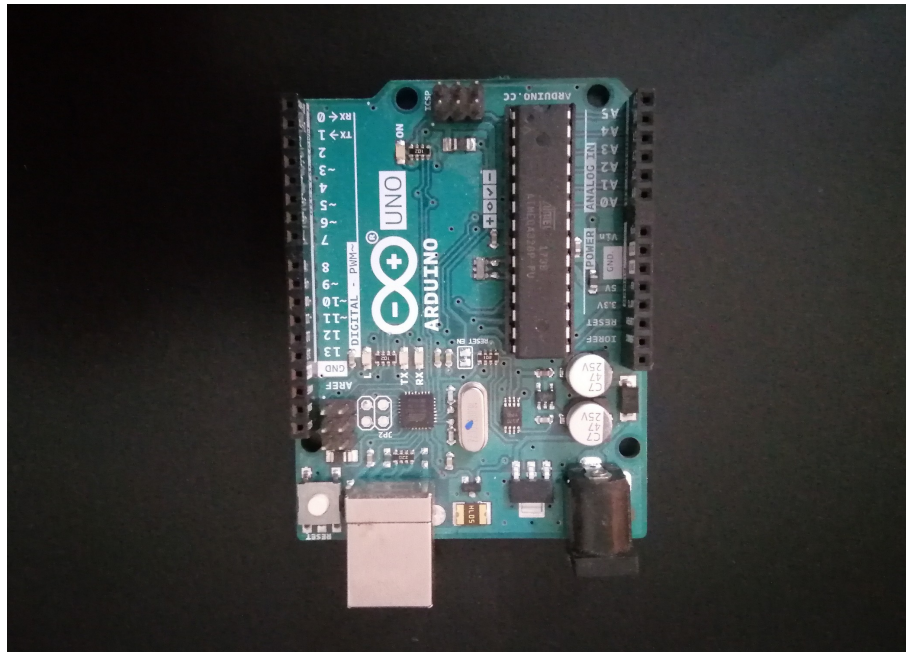
Η υλοποίηση της συγκεκριμένης πτυχιακής εργασίας βασίστηκε τόσο στο θεωρητικό σκέλος, δηλαδή στην μελέτη σχετικών βιβλιογραφιών και μελετών, αλλά και στην πρακτική εφαρμογή τους στο πειραματικό δίκτυο στο Ιόνιο Πανεπιστήμιο. Συνεπώς, δεν θα ήταν ορθό να μην αναφερθεί και ο υλικός εξοπλισμός που χρησιμοποιήθηκε.

Γ΄.1 Arduino Uno

Το Arduino είναι μία πλατφόρμα ανοιχτού κώδικα [7], η οποία έχει την δυνατότητα να λαμβάνει και να στέλνει πληροφορίες σε διάφορες συσκευές και για τον λόγο αυτό χρησιμοποιείται σε διάφορες εφαρμογές και τομείς της καθημερινότητας, με στόχο την βελτίωση και την διευκόλυνσή της [21]. Το Arduino αποτελείται από δύο βασικά μέρη. Το πρώτο αποτελεί το μηχανικό μέρος, δηλαδή την συσκευή και το δεύτερο αποτελεί το προγραμματιστικό, το λογισμικό μέρος. Αρχικά, ένα Arduino Uno αποτελείται από διάφορα εσωτερικά κυκλώματα και διεπαφές, ο συνδυασμός των οποίων το καθιστά λειτουργικό [21]. Η βασική προϋπόθεση στην λειτουργία ενός arduino uno αλλά και γενικότερα όλων των ηλεκτρικών συσκευών είναι η παροχή ρεύματος. Στο arduino αυτό επιτυγχάνεται με δύο τρόπους, δηλαδή με δύο εισόδους. Η πρώτη πηγή εισόδου ηλεκτρικού ρεύματος αναφέρεται ως Power Connector ή και ως External Power Supply και η λειτουργία της είναι αποκλειστικά η διοχέτευση ρεύματος στον μικροελεγκτή, με την τάση του οποίου να κυμαίνεται μεταξύ 7-12V [25]. Ο δεύτερος τρόπος παροχής ρεύματος του μικροελεγκτή αποτελεί η θύρα USB, η οποία, ωστόσο, έχει διπλό ρόλο στην λειτουργία ενός arduino. Εκτός από την διοχέτευση ρεύματος, ο ρόλος της είναι και

προγραμματιστικός, καθώς μέσω αυτής ο μικροελεγκτής προγραμματίζεται. Επιπλέον, επάνω στην πλακέτα ενός arduino υπάρχει και μία διεπαφή με τον χρήστη, ένα κουμπί, το λεγόμενο Reset Button, η λειτουργία του οποίου είναι η επαναφορά του μικροελεγκτή και προγραμματιστικά μεταφράζεται ως επανεκκίνηση του προγράμματος (software) από το hardware. Στην πλακέτα του Arduino Uno υπάρχουν και διάφορα λαμπάκια τα οποία χρησιμοποιούνται για την κατανόηση της λειτουργίας και την επίλυση προγραμματιστικών αλλά και τεχνικών σφαλμάτων. Ένα τέτοιο λαμπάκι είναι αυτό που υποδηλώνει την ύπαρξη ή όχι ηλεκτρικού ρεύματος στην πλακέτα (Power Led), αλλά και τα Tx και Rx Led τα οποία υποδηλώνουν την επικοινωνία της πλακέτας με μία άλλη πλακέτα, τον υπολογιστή ή γενικότερα μία άλλη συσκευή. Ένα μεγάλο μέρος της πλακέτας καταλαμβάνεται από ακίδες (pins), οι οποίες εκτός από τις πιο βασικές όπως αυτές που λειτουργούν ως πηγή με τάση 5V και γείωση για τα κυκλώματα που θα υλοποιηθούν, υπάρχουν και αυτές που χωρίζονται σε αναλογικές (analog pins) και ψηφιακές (digital pins). Η καρδιά, όμως, της πλακέτας του Arduino Uno είναι ο μικροελεγκτής «ATmega», ουσιαστικά είναι ένας μικροελεγκτής Advanced Virtual RISC (AVR), ο οποίος υποστηρίζει δεδομένα έως και οκτώ (8) bit, έχοντας εσωτερική ενσωματωμένη μνήμη 32KB. Το δεύτερο βασικό μέρος από το οποίο αποτελείται ένα Arduino Uno είναι το λογισμικό μέρος. Ο προγραμματισμός του Arduino υλοποιείται μέσω του προγράμματος Arduino IDE, το οποίο αποτελεί ένα περιβάλλον ανάπτυξης ανοικτού κώδικα για κάθε είδος arduino. Ο κάθε κώδικας αναφέρεται, κοινώς, ως σκίτσο (sketch) και είναι γραμμένος σε μία απλοποιημένη μορφή της Cpp [36]. Το περιβάλλον ανάπτυξης του κώδικα απαρτίζεται από διάφορα ενδιάμεσα στοιχεία, ώστε να είναι λειτουργικό και εύχρηστο. Ξεκινώντας από πάνω προς τα κάτω το περιβάλλον απαρτίζεται από τη γραμμή εργαλείων, η οποία περιλαμβάνει διάφορες διεπαφές/κουμπιά που σχετίζονται με την εκτέλεση, την αποθήκευση αλλά και το «ανέβασμα» του κώδικα στην εκάστοτε πλακέτα arduino που είναι συνδεδεμένη με τον ηλεκτρονικό υπολογιστή μέσω της θύρας USB που προαναφέρθηκε. Ταυτόχρονα, στο δεξί μέρος της γραμμής εργαλείων εμφανίζεται και μία διεπαφή που «ανοίγει» την κονσόλα (serial monitor). Η κονσόλα αυτή εμφανίζει τυχόν μηνύματα, σφάλματα και άλλες πληροφορίες που εξάγονται από τον μικροελεγκτή στο serial monitor. Το μεγαλύτερο μέρος του περιβάλλοντος του προγράμματος αποτελείται από την περιοχή στην οποία γράφεται ο κώδικας. Σε κάθε σκίτσο, αρχικά, εμφανίζονται οι δύο βασικές συναρτήσεις του προγράμματος, η συνάρτηση «SetUp()» που του εκτελείται μία φορά κατά την εκκίνηση του προγράμματος και η συνάρτηση «loop()» που εκτελείται διαρκώς. Στη συνέχεια ο κώδικας μπορεί να επεκταθεί με την ενσωμάτωση βιβλιοθηκών, αλλά και άλλων συναρτήσεων που θα δημιουργηθούν από τον εκάστοτε χρήστη για την υλοποίηση των λειτουργιών που επιθυμεί. Τέλος, υπάρχει η περιοχή των μηνυμάτων, στην οποία αναγράφεται

αν υλοποιήθηκε σωστά η εκτέλεση του κώδικα, αναφέροντας σχόλια και προειδοποιήσεις για πιθανές βελτιώσεις του κώδικα, αλλά και μηνύματα σφάλματος εάν αυτά υπάρχουν στο αντίστοιχο sketch. Στο κάτω δεξιά μέρος εμφανίζεται η θύρα, αλλά και το είδος της πλακέτας που χρησιμοποιείται [36].

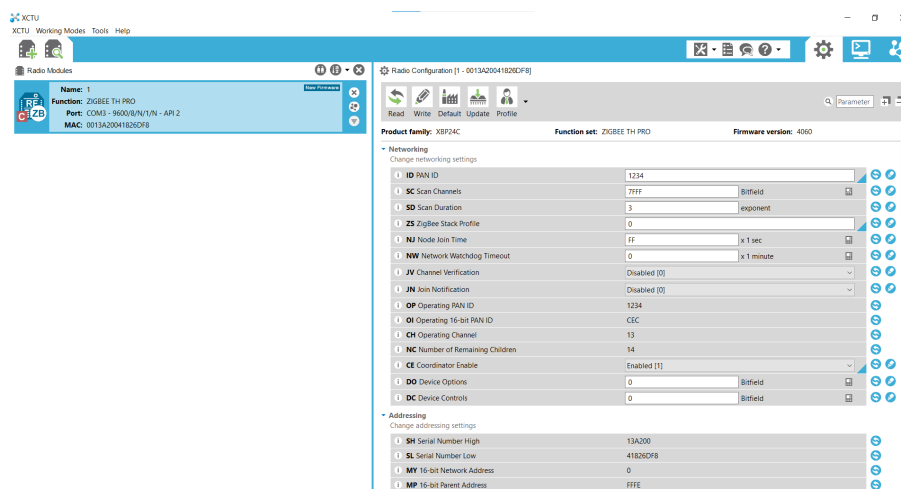


Σχήμα Γ'.4: *Arduino Uno*.

Γ'.2 XBee S2C Pro ZigBee RF Module

Η επικοινωνία μεταξύ των κόμβων του δικτύου επιτεύχθηκε μέσω των κεραίων XBee S2C ZigBee. Οι κεραίες αυτές, όπως και τα arduino uno παραπάνω, απαρτίζονται από δύο μέρη, τα οποία τις καθιστούν λειτουργικές, από το μηχανικό/ υλικό μέρος και από το λογισμικό προγραμματιστικό. Το πρώτο μέρος, δηλαδή το υλικό μέρος απαρτίζεται από την κεραία που είναι τοποθετημένη πάνω σε μία πλακέτα XBee ZigBee Through hole technology modules (S2C). Η πλακέτα αποτελείται από είκοσι «ακίδες» (pins) κάθε μία από τις οποίες έχει μία διαφορετική λειτουργία και χρησιμότητα. Αρχικά, υπάρχουν οι ακίδες η λειτουργία των οποίων είναι η παροχή ρεύματος (pin 1) και η γείωση (pin 10). Ο πίνακας παρακάτω παρουσιάζει και τις είκοσι ακίδες ονομαστικά αλλά και με βάση την λειτουργία τους. Ο προγραμματισμός της

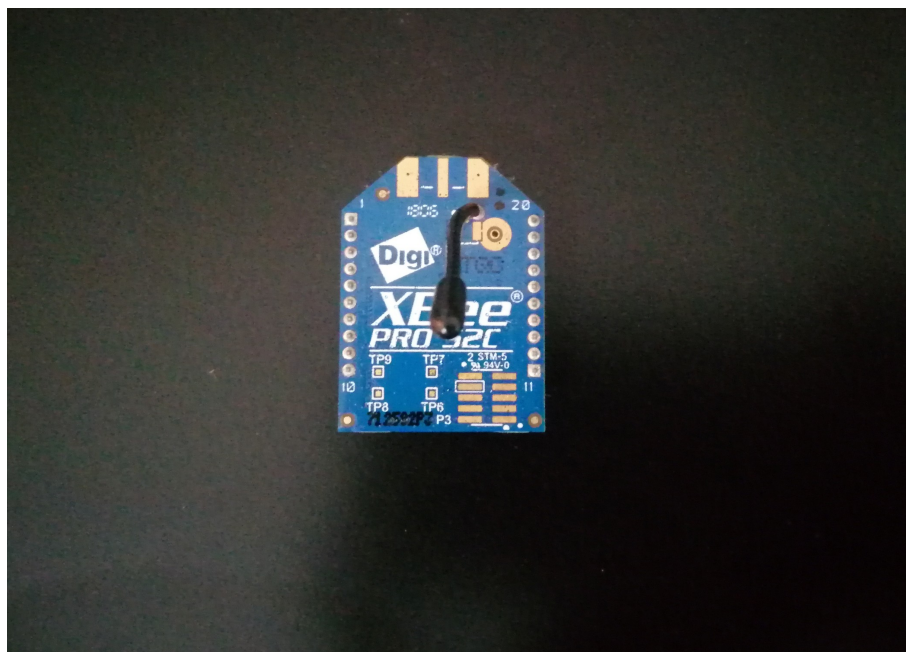
κεραίας επιτυγχάνεται μέσω του προγράμματος XCTU, το γραφικό περιβάλλον του οποίου είναι ορατό στο Σχήμα Γ'.5. Μέσω του προγράμματος αυτού ο χρήστης μπορεί να ανιχνεύσει τους κόμβους που είναι συνδεδεμένοι στο ίδιο δίκτυο αλλά και τις διασυνδέσεις μεταξύ των κόμβων (βασική προϋπόθεση όλοι οι κόμβοι του δικτύου να έχουν το ίδιο PAN ID. Επιπλέον, μέσω του προγράμματος αυτού ο χρήστης μπορεί να επεξεργαστεί και άλλα χαρακτηριστικά της εκάστοτε κεραίας, όπως για παράδειγμα το εάν κάποιος κόμβος θα είναι ο coordinator του δικτύου, τον μέγιστο αριθμό των hop, την ακτίνα, ρυθμίσεις που αφορούν δηλαδή γενικότερα το δίκτυο, τις διευθύνσεις, την ασφάλεια, τις εισόδους και τις εξόδους αλλά και τους τρόπους διεπαφής.



Σχήμα Γ'.5: Το εσωτερικό περιβάλλον του προγράμματος XCTU.

Συγκεκριμένα, στο πειραματικό δίκτυο που έλαβε χώρα στο Ιόνιο Πανεπιστήμιο οι ρυθμίσεις που επεξεργάστηκαν από τις προκαθορισμένες δεν ήταν πολλές. Οι βασικές αλλαγές που υλοποιήθηκαν είναι στην ιδιομορφία του PAN ID, ώστε να είναι ίδιο σε όλους τους κόμβους, για να επιτυγχάνεται η επικοινωνία μεταξύ τους. Επίσης, σε έναν από τους κόμβους του δικτύου ενεργοποιήθηκε το «Coordinator Enable», ενώ σε όλους τους άλλους κόμβους ενεργοποιήθηκε το «Channel Verification». Ο κόμβος που ορίζεται ως Coordinator έχει κάποια προνόμια στο δίκτυο, αλλά και υποχρεώσεις. Συγκεκριμένα, σε κάθε δίκτυο, δηλαδή ανά κάθε PAN ID είναι υποχρεωτική η ύπαρξη ενός μόνο coordinator, ο οποίος είναι ο «εκκινητής» του δικτύου ορίζοντας το κανάλι και το PAN ID, ελέγχει ποιοι κόμβοι βρίσκονται στο δίκτυο και εξασφαλίζει την ασφάλεια και γενικότερα την ποιότητα του δικτύου. Αντίθετα, οι κόμβοι του δικτύου που ορίζονται ως routers μπορεί να είναι παραπάνω από έναν σε κάθε δίκτυο

ενώ συμμετέχουν σε ένα ήδη υπάρχον δίκτυο και μπορούν να στέλνουν και να λαμβάνουν ή να προωθούν πληροφορίες. Οι κόμβοι που ορίζονται ως Router αλλά και ο Coordinator του δικτύου δεν μπορούν να «μπουν» σε λειτουργία «sleep», αντίθετα, πρέπει να είναι ενεργοί συνεχώς. Παράλληλα, σε κάθε κόμβο αποδόθηκε ένας ακεραίος μοναδικός αριθμός ως Node Identifier, ώστε να χρησιμοποιηθεί και το αναγνωριστικό αυτό στον κώδικα, για την αναγνώριση του κόμβου με βάση το ID του και όχι την διεύθυνσή του, κυρίως για λόγους ευκολίας και εξοικονόμησης χώρου. Αξίζει να αναφερθεί ότι η ακτίνα εμβέλειας κάθε κόμβου παρέμεινε στο υψηλότερο επίπεδο, καθώς οι αποστάσεις στο πειραματικό δίκτυο ήταν αρκετά μεγάλες. Τέλος, υλοποιήθηκαν δύο αλλαγές στο Serial Interfacing μέρος της κεραίας. Συγκεκριμένα, μεταβλήθηκε το «API Enable» από την επιλογή «Transparent Mode», στην επιλογή «API enable with escaping» και το «API Output Mode» σε «Explicit». Οι αλλαγές αυτές έγιναν διότι στο δίκτυο το οποίο υλοποιήθηκε ήταν απαραίτητη η επικοινωνία μεταξύ πολλαπλών κόμβων αλλά και η αναγνώριση της διεύθυνσης του αποστολέα, ώστε να επιτυγχάνεται η βελτιστοποίηση και γενικότερα, η επικοινωνία μεταξύ των κόμβων του δικτύου. Οι απαιτήσεις αυτές του δικτύου οδήγησαν στις προαναφερθείσες αλλαγές.



Σχήμα Γ'.6: Η κεραία XBee S2C Pro ZigBee.

Πίνακας Γ'.1: Τα pins μία κεράλας XBee S2C Pro ZigBee.

Pin	Όνομα	Περιγραφή
1	VCC	Power Supply
2	DOUT/DIO13	UART sata out
3	DIN/CONFIG/DIO14	UART data in
4	DIO12/SPI MISO	GPIO/SPI slave out
5	RESET	Reset Module
6	RSSI PWM/PWMO/- DIO10	Rx signal Strength indi- cator/GPIO
7	PWM1/DIO11	GPIO
8	[reserved]	Do not connect
9	DTR/SLEEP RQ/DIO8	Pin sleep control line/G- PIO
10	GND	Ground
11	SPI MOSI/DIO4	GPIO/SPI slave in
12	CTS/DIO7	Clear to send flow con- trol/ GPIO
13	ON SLEEP/DIO9	Device status indica- tor/GPIO
14	VRED	Not connected
15	ASSOCIATE/DIO5	Associate indicator / GPIO
16	RTS/DIO6	Request to send flow con- trol/GPIO
17	AD3/DIO3/SPI SSEL	Analog input/GPIO/SPI slave select
18	AD2/DIO2/SPI CLK	Analog input/GPIO/SPI attention
19	AD1/DIO1/SPI ATTN	Analog input/GPIO/SPI attention
20	AD0/DIO0/CB	Analog input / GPIO / Commissioning button

Γ'.2.1 Πρωτόκολλο ZigBee

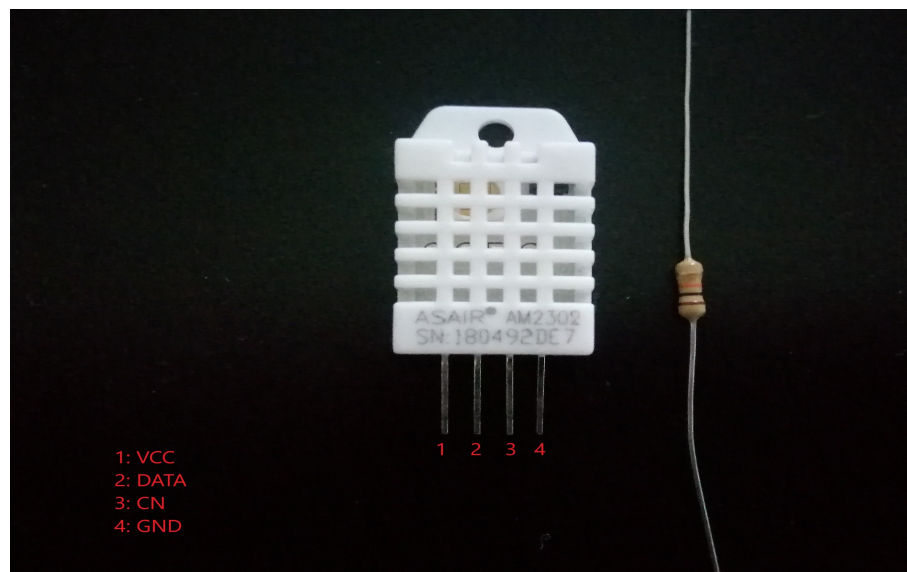
Το ZigBee είναι ένα βασικό πρωτόκολλο δικτύου που σχεδιάστηκε με σκοπό την εύκολη διαχείριση ασύρματης πληροφορίας και την επικοινωνία σε Noisy RF περιβάλλοντα που είναι συνήθη σε εμπορικές εφαρμογές [28]. Η ισχύς που απαιτείται για την υλοποίησή του είναι ιδιαίτερα μικρή με «κόστος» 1mW ή και λιγότερο, ωστόσο η εμβέλειά του σε εξωτερικό χώρο καλύπτει και τα 150 μέτρα εξαιτίας της τεχνικής «Direct Sequence Spread Spectrum». Η συχνότητα που λειτουργεί είναι 868 MHz (Ευρώπη), 915 MHz (North America and Australia) και 2.4 GHz (παγκοσμίως) [29] [32].

Το πρωτόκολλο ZigBee αποτελείται από τέσσερα επιμέρους επίπεδα, κάθε ένα εκ των οποίων έχει καθορισμένες λειτουργίες και παράγει υπηρεσίες στα παραπάνω επίπεδα μέσω μιας διεπαφής (service access point, SAP). Αρχικά, το φυσικό επίπεδο (Physical Layer) υλοποιεί κάποιες ιδιαίτερα βασικές διεργασίες για την λειτουργικότητα του δικτύου, όπως η ενεργοποίηση ή όχι του πομπού ή/και του δέκτη, μετάδοση και λήψη δεδομένων, ανίχνευση ενέργειας στο κανάλι με στόχο την αποφυγή των αποτυχημένων μεταδόσεων και των συγκρούσεων (CSMA-CA) [17], καταμέτρηση της ποιότητας των πακέτων που λαμβάνει ο δέκτης. Στην συνέχεια, το επίπεδο ελέγχου πρόσβασης (Medium Access Control Layer), το οποίο αποτελεί ένα «εσωτερικό» επίπεδο του επιπέδου ζεύξης δεδομένων (Data Link Layer) του μοντέλου αναφοράς Open Systems Interconnection (OSI) προσφέρει υπηρεσίες σχετικά με την μετάδοση και αξιοποίηση της πληροφορίας, στοχεύοντας σε μία αξιόπιστη και ασφαλή σύνδεση. Το επίπεδο δικτύου (Network Layer) το οποίο δημιουργεί το δίκτυο, είναι υπεύθυνο για την δρομολόγηση των πακέτων, για την είσοδο και έξοδο των κόμβων από το δίκτυο αλλά και για την ασφάλεια αυτού. Τέλος, το επίπεδο εφαρμογών (Application Layer), το ανώτερο επίπεδο του μοντέλου αναφοράς OSI, αποτελείται από υποεπίπεδο υποστήριξης εφαρμογών (Application support sublayer (APS)) αναφέρεται σε ένα πλαίσιο εφαρμογών (Application framework, AF) και έχει στη διάθεσή του τα αντικείμενα συσκευής ZigBee (ZigBee Device Objects, ZDO) και τις καθορισμένες από τον κατασκευαστή εφαρμογές. Το APS είναι υπεύθυνο για τη σύνδεση δύο συσκευών βάσει των αναγκών και των υπηρεσιών τους, καθώς επίσης και για την αμφίδρομη αποστολή δεδομένων ανάμεσα στις υπηρεσίες. Τα ZDO μπορούν να ανιχνεύσουν τις συσκευές που βρίσκονται σε ένα δίκτυο αλλά και να προσδιορίσουν κάθε υπηρεσία που αυτές παρέχουν ενώ, παράλληλα, είναι υπεύθυνα για τον καθορισμό του ρόλου των συσκευών στο δίκτυο, δημιουργώντας και καθορίζοντας έτσι τους κόμβους που λειτουργούν ως συντονιστής (coordinator), δρομολογητές (router) και τελικών συσκευών (end-devices) [5]. Οι κόμβοι ενός δικτύου ZigBee έχουν συγκεκριμένους ρόλους όπως προαναφέρθηκε και

παραπάνω. Ένας κόμβος σε όλο το δίκτυο είναι ο συντονιστής κόμβος (coordinator). Ο κόμβος αυτός πρέπει να είναι αποκλειστικά ένας σε κάθε δίκτυο και είναι αυτός που δημιουργεί το δίκτυο και καθορίζει το PAN ID του εκάστοτε δικτύου, συνεπώς είναι ένας coordinator ανά ένα PAN ID. Προαιρετικά στο δίκτυο υπάρχουν και οι κόμβοι αναφερόμενοι ως δρομολογητές (routers). Οι κόμβοι αυτοί δεν είναι απαραίτητο να υπάρχουν σε κάθε δίκτυο, ωστόσο η λειτουργία τους είναι ιδιαίτερα χρήσιμη, καθώς είναι αυτοί που δρομολογούν και λειτουργούν ως μεσάζοντες μεταφέροντας τα πακέτα. [1]

Γ'.3 Αισθητήρας θερμοκρασίας και υγρασίας DHT22

Για την καταμέτρηση και καταγραφή της θερμοκρασίας και της υγρασίας στα πειράματα που διεξήχθησαν χρησιμοποιήθηκε ο αισθητήρας θερμοκρασίας και υγρασίας DHT22.



Σχήμα Γ'.7: Ο αισθητήρας DHT22 και μία αντίσταση 10Kohm.

Η θερμοκρασία και η υγρασία του εξωτερικού περιβάλλοντος θεωρήθηκαν ως πιθανοί συντελεστές στην ποιότητα των ζεύξεων μεταξύ των arduino uno. Γενικότερα, στη βιβλιογραφική έρευνα που υλοποιήθηκε παρατηρήθηκε η έντονη επιρροή του εξωτερικού περιβάλλοντος μεταξύ την επικοινωνίας των Arduino, για τον λόγο αυτό στην συγκεκριμένη πτυχιακή εργασία μελετήθηκαν και αυτοί οι παράγοντες. Ο DHT22 είναι ένας ψηφιακός αισθητήρας που εσωτερικά αποτελείται από έναν αισθητήρα υγρασίας και ένα θερμίστορ για την καταμέτρηση της

Πίνακας Γ'.2: Τα χαρακτηριστικά του αισθητήρα DHT22.

	Temperature	Humidity
Power Supply Range	3.3V to 6V	3.3V to 6V
Range	-40 έως +80°C	0 έως 100%
Accuracy	+/- 0.5°C	2-5%
Sensitivity	0.1°C	0.1%

θερμοκρασίας [27] [37]. Η θερμοκρασία που μπορεί να ανιχνευθεί με την χρήση του συγκεκριμένου αισθητήρα κυμαίνεται από -40° C έως και 80° C . Οι μετρήσεις της θερμοκρασίας με τον συγκεκριμένο αισθητήρα μπορεί να επιτευχθούν με διαφορά μεταξύ τους δύο δευτερολέπτων, γεγονός που αποτελεί το βασικό του μειονέκτημα. Ωστόσο, προσφέρει ακρίβεια με διαφορά 0.5° C και ευαισθησία 0.1° C [20] [23]. Η υγρασία, αντίστοιχα, αποτυπώνεται ως θετικό ποσοστό επί τοις εκατό, οι τιμές του οποίου διακυμαίνονται από το μηδέν έως το εκατό, με ακρίβεια 2-5% και ευαισθησία αντίστοιχη της θερμοκρασίας. Η ιδανική παροχή τάσης που απαιτείται για την λειτουργία του είναι τα 5V, χωρίς ωστόσο να είναι η απαραίτητη, καθώς μπορεί και να λειτουργήσει με τάση που ισούται με 3.3V έως και 6V [22]. Η σύνδεση του αισθητήρα με το arduino, όπως παρουσιάζεται και στο σχήμα Γ'.7, είναι ιδιαίτερα απλή, καθώς από τις τέσσερις εξόδους του θα αξιοποιηθούν οι τρεις, με την πρώτη και την τελευταία να αποτελούν την πηγή και την γείωση, αντίστοιχα, και τη δεύτερη να αποδίδει τα δεδομένα στο πρόγραμμα.

Κεφάλαιο Δ΄

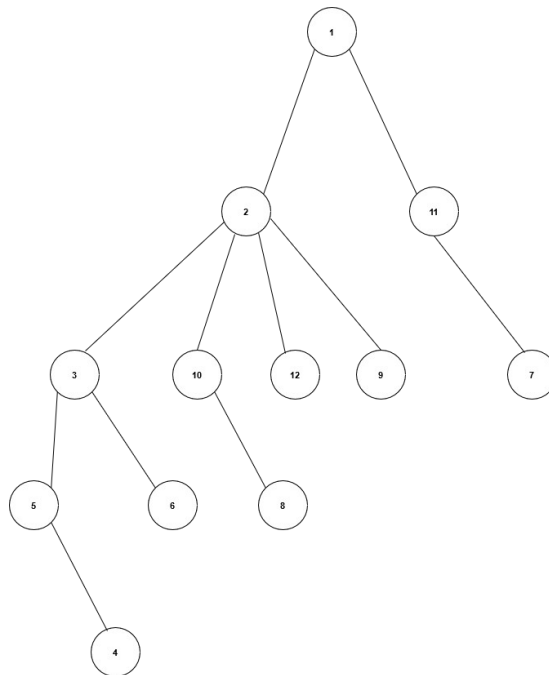
Υλοποίηση του δικτύου

Η υλοποίηση του δικτύου για την άντληση των μετρήσεων αποτέλεσε μία μακροσκελή διαδικασία. Αρχικά, υλοποιήθηκε ο κώδικας με την χρήση λιγότερων κόμβων και μελετήθηκε ως προς την λειτουργικότητά του. Στη συνέχεια, έγιναν αρκετοί πειραματισμοί στα πλαίσια του πειραματικού δικτύου του Ιονίου Πανεπιστημίου, με σκοπό την βέλτιστη συλλογή των δεδομένων και την αντιμετώπιση όλων των προκληθέντων προβλημάτων.

Δ΄.1 Περιγραφή του τρόπου συλλογής των δεδομένων

Ο κώδικας που υλοποιήθηκε απαρτίζεται από τρία σημαντικά μέρη. Το πρώτο μέρος του κώδικα είναι αναγνωριστικό για το δίκτυο. Η υλοποίησή του βασίζεται, κυρίως στην λογική των αλγορίθμων Flooding. Ο κόμβος που έχει οριστεί ως coordinator ξεκινάει με την αποστολή ενός μηνύματος 'Hi' προς όλους τους κόμβους που βρίσκονται σε ακτίνα ενός hop από αυτόν. Το μήνυμα αυτό ουσιαστικά λειτουργεί ενημερωτικά για την ανακοίνωση της ύπαρξης του εκάστοτε κόμβου στην εμβέλεια των άλλων. Όποιος κόμβος λάβει αυτό το μήνυμα ορίζει τον αποστολέα ως «γείτονα» (Πίνακας Ε΄.1) και εάν το μήνυμα το έλαβε πρώτη φορά τον ορίζει και ως πατέρα κόμβο και του αποστέλλει ένα μήνυμα ώστε να ορίσει και ο αποστολέας του μηνύματος «Hi» τον δέκτη ως κόμβο παιδί. Το γεγονός αυτό δημιουργεί έμμεσα ένα Spanning Tree [16], όπως φαίνεται και στο Σχήμα Δ΄.8. Με αυτό τον τρόπο εδραιώνονται κάποια κανάλια μεταξύ των κόμβων τα οποία θα συμβάλλουν στην αποστολή λιγότερων μηνυμάτων στα υπόλοιπα μέρη του κώδικα, καθώς ο coordinator θα συλλέγει τις πληροφορίες μόνο μέσω

αυτών των ζεύξεων. Έπειτα, ο κόμβος που έλαβε το μήνυμα «Hi», θα το αποστείλει και ο ίδιος σε όλους τους κόμβους που βρίσκονται στην εμβέλειά του.



Σχήμα Δ'.8: Το δέντρο που προκύπτει από το πρώτο μέρος του κώδικα.

Το δεύτερο και σημαντικότερο μέρος του κώδικα αποτέλεσε η αποστολή μηνυμάτων και η καταγραφή των πληροφοριών και των δεδομένων που είναι πιθανόν να επηρεάζουν την ποιότητα των ζεύξεων του δικτύου. Ο εκκινητής κόμβος για ακόμα μία φορά είναι ο coordinator. Ξεκινάει να αποστέλλει σε κάθε κόμβο παιδί ένα μήνυμα συλλογής δεδομένων. Ακολουθεί μία λογική αλγορίθμου Depth First [16], καθώς στέλνει ξεκινώντας από τον πρώτον κόμβο παιδί. Ο κόμβος που λαμβάνει το μήνυμα το προωθεί με την σειρά του στον πρώτο κόμβο που έχει ορίσει ως παιδί. Εάν ένας κόμβος δεν έχει κανένα παιδί, ξεκινάει να αποστέλλει εκατό μηνύματα.

Τα εκατό αυτά μηνύματα εμπεριέχουν έναν μετρητή και το αναγνωριστικό του αποστολέα, ώστε να γνωρίζει κάθε δέκτης ποιο από τα εκατό μηνύματα έλαβε αλλά και τον κόμβο αποστολέα. Έτσι, για κάθε ένα από τα μηνύματα που λαμβάνει, καταγράφει σε ένα αρχείο τα

δεδομένα που έλαβε αλλά και το δικό του αναγνωριστικό, την θερμοκρασία της κεραίας του XBee S2C Pro ZigBee, του περιβάλλοντος και την υγρασία αυτού. Όταν η αποστολή των εκατό μηνυμάτων φθάσει στο τέλος της, ο κόμβος αποστολεάς προωθεί ένα μήνυμα στον πατέρα του, μόνο σε περίπτωση που δεν έχει οριστεί ως coordinator, διότι ο κόμβος αυτός αποτελεί την ρίζα του δέντρου που υλοποιήθηκε, ώστε αυτός να συνεχίσει με τα επόμενα παιδιά ή να ξεκινήσει ο ίδιος την αποστολή των εκατό μηνυμάτων. Η διαδικασία αυτή επαναλαμβάνεται διαρκώς μέχρι την εκκίνηση του τρίτου κύριου μέρους του προγράμματος.

Όταν περάσει αρκετό διάστημα, ώστε να συλλεχθούν δεδομένα και ο τελευταίος που απέστειλε τα εκατό μηνύματα που προαναφέρθηκαν είναι ο κόμβος coordinator, ξεκινάει το τρίτο μέρος του κώδικα, που είναι η συγκέντρωση όλων των δεδομένων στον κόμβο coordinator. Με την ίδια λογική του δεύτερου μέρους ξεκινά και λαμβάνει χώρα η διαδικασία, με μοναδική διαφορά το ότι όταν φτάσουμε στα φύλλα, αντί ο κόμβος να στείλει τα εκατό μηνύματα, ανοίγει και αποστέλει γραμμή γραμμή τις πληροφορίες που κατέγραψε στο αρχείο του. Κάθε κόμβος που λαμβάνει ένα τέτοιο μήνυμα το προωθεί στον πατέρα κόμβο έως να φτάσουν οι πληροφορίες στην ρίζα του δέντρου. Επίσης, η διαδικασία αυτή επαναλαμβάνεται για κάθε κόμβο του δικτύου εκτός από τον συλλέκτη coordinator. Με την λήψη της διαδικασίας αυτής, ο κόμβος ρίζα του δέντρου επανεκκινεί το δεύτερο μέρος του κώδικα, επαναλαμβάνοντας την ίδια διαδικασία, συνεχόμενα.

Δ'.2 Περιγραφή των δεδομένων συλλογής

Η επιλογή των δεδομένων τα οποία συλλέχθηκαν στη συγκεκριμένη πτυχιακή εργασία για την ποιότητα των ζεύξεων βασίστηκε, κυρίως, στην βιβλιογραφική έρευνα που υλοποιήθηκε. Αρχικά, αναζητήθηκαν όλοι οι παράμετροι που αξιολογούν με ποικίλους τρόπους μία ζεύξη μεταξύ δύο κεραίων XBee S2C Pro ZigBee. Η αναζήτηση αυτή οδήγησε στην εύρεση της τιμής RSS, η οποία μπορεί να ανακτηθεί εύκολα από τον μικροεπεξεργαστή με την αποστολή ενός AT-command στην κεραία, που είναι τοποθετημένη σε αυτόν. Η τιμή αυτή προβάλλει την ισχύ του τελευταίου hop του ληφθέντος πακέτου και αποτελεί έναν αρνητικό αριθμό. Όσο πιο κοντά στο μηδέν φτάνει η τιμή τόσο πιο ποιοτική εμφανίζεται η ζεύξη. Επιπλέον, θεωρήθηκε σημαντικό δεδομένο για την αξιολόγηση μίας ζεύξης το ποσοστό απώλειας μηνυμάτων μεταξύ των κόμβων. Η πληροφορία αυτή αποδόθηκε μέσω του κώδικα.

Τέλος, τα δεδομένα θεωρήθηκε σημαντική η συγκέντρωση πληροφοριών που δεν αφορούσαν άμεσα τον κόμβο, αλλά το περιβάλλον γύρω από αυτόν. Τα δεδομένα αυτά αποτέλεσαν η

θερμοκρασία και η υγρασία. Ωστόσο, καθώς τα πειράματα έλαβαν χώρα στο ίδιο κτίριο, την ίδια χρονική περίοδο, η μεταβολή των δύο αυτών τιμών ήταν ελάχιστη και δεν φανέρωνε καμία επίδραση στην ποιότητα των ζεύξεων. Το γεγονός, όμως, της μη επιρροής του εξωτερικού περιβάλλοντος στο δίκτυο δεν φαινόταν πιθανό και, έτσι μεταβλήθηκε η θερμοκρασία με τεχνητούς τρόπους.

Δ'.3 Δυσκολίες και προβλήματα κατά την υλοποίηση

Η συλλογή και η καταγραφή δεδομένων σε ένα πραγματικό δίκτυο με αισθητήρες απαιτεί χρόνο, συγχέντρωση και εμφανίζει προβλήματα τα οποία δεν συναντώνται στον εικονικό κόσμο. Οι δυσκολίες προκύπτουν, καθώς στο πραγματικό περιβάλλον η εξέλιξη ενός κώδικα δεν εξαρτάται αποκλειστικά από τον ίδιο, αλλά και από εξωτερικούς παράγοντες που πρέπει να ληφθούν υπόψη ώστε αυτό που θα προκύψει να είναι ευέλικτο προς αυτούς και, ταυτόχρονα, λειτουργικό.

Στην υλοποίηση, αρχικά, προέκυψε ως πρόβλημα ο μεγάλος όγκος δεδομένων που πρέπει να καταγράφει από κάθε κόμβο. Το πρόβλημα, αυτό επιλύθηκε με την χρήση των καρτών SD, οι οποίες και τοποθετήθηκαν επάνω σε κάθε κόμβο του δικτύου. Έτσι, τα δεδομένα αποθηκεύονται σε αυτή μέχρι να ξεκινήσει η συλλογή των δεδομένων κάθε κόμβου από τον coordinator. Επίσης, ο μεγάλος όγκος δεδομένων δημιουργούσε πρόβλημα στην σωστή αποστολή και συγχέντρωση των δεδομένων στον κόμβο coordinator. Το πρόβλημα αυτό επιλύθηκε με την αύξηση του μεγέθους του rx buffer και tx buffer από τα αρχεία εγκατάστασης του Arduino και συγκεκριμένα το αρχείο HardwareSerial.h.

Ταυτόχρονα, ένα ιδιαίτερα σημαντικό πρόβλημα που απαιτήθηκε να διεκπεραιωθεί ήταν το γεγονός της χαμηλής απαίτησης ενέργειας του κόμβου με μόνο έναν αισθητήρα επάνω σε αυτόν. Το αποτέλεσμα αυτού, κατά τα άλλα θετικού χαρακτηριστικού του δικτύου, ήταν η απενεργοποίηση των μπαταριών και κατ'επέκταση του ίδιου του κόμβου. Η μπαταρία, δηλαδή, δεν ανίχνευε ότι δίνει ρεύμα σε κάποιο αντικείμενο και για αυτό τον λόγο απενεργοποιούνταν. Ο τρόπος που επιλύθηκε το συγκεκριμένο πρόβλημα είναι η προσθήκη παραπάνω από ενός κόμβου σε κάθε μπαταρία. Με αυτό τον τρόπο η μπαταρία παρέμεινε ενεργή σε όλη την διάρκεια του προγράμματος. Επίσης, παρατηρήθηκε καλύτερα η άμεση σχέση της απόστασης δύο κόμβων με την ποιότητα της ζεύξης μεταξύ τους.

Το παραπάνω πρόβλημα, ωστόσο, οδήγησε και στην δημιουργία ενός κώδικα ο οποίος δεν θα είχε τις «αλληλο-εξαρτήσεις» μεταξύ των κόμβων, δηλαδή δεν θα ορίζονταν οι κόμβοι ως

πατέρας και παιδί. Το γεγονός αυτό υλοποιούνταν με τη χρήση ενός προγράμματος με πολλά μηνύματα τύπου broadcast. Ωστόσο, αυτό δημιουργούσε πολλές συγκρούσεις και δεν θεωρήθηκε η βέλτιστη λύση. Αντίθετα, ακολουθήθηκε η λύση με την προσθήκη επιπλέον κόμβων σε κάθε μπαταρία και βελτιώθηκε ο κώδικας, ώστε να είναι πλήρως λειτουργικός, όπως παρουσιάστηκε και παραπάνω.

Τέλος, δημιουργήθηκε ένα μικρό πρόβλημα όταν υλοποιήθηκαν κάποιες διαφοροποιήσεις στον κώδικα. Το πρόβλημα αυτό ήταν ο περιορισμός των οκτώ broadcast μηνυμάτων ανά οκτώ δευτερόλεπτα. Αυτό δημιουργούσε ένα πρόβλημα, αρχικά, καθώς περιόριζε την συνεχόμενη αποστολή μηνυμάτων χωρίς καθυστέρηση. Για τον λόγο αυτό χρησιμοποιήθηκε ένα delay τριών δευτερολέπτων μεταξύ την συνεχόμενη αποστολή εκατό μηνυμάτων. Ωστόσο, το γεγονός αυτό παρείχε την δυνατότητα ο δέκτης να έχει τον χρόνο να αποστείλει τα AT-commands που αφορούν την ανάκτηση του RSS και της θερμοκρασίας της κεραίας και να τα καταγράψει σε ένα αρχείο για κάθε ένα από τα εκατό μηνύματα που λάμβανε.

Κεφάλαιο Ε΄

Αποτελέσματα της υλοποίησης

Η εκπόνηση της συγκεκριμένης πτυχιακής εργασίας βασίστηκε στο πειραματικό δίκτυο που έχει δημιουργηθεί στα πλαίσια του Ιονίου Πανεπιστημίου [30]. Η μελέτη της ποιότητας των ζεύξεων κρίθηκε πιθανόν να επηρεάζεται από διάφορους παράγοντες τόσο περιβαλλοντολογικούς όσο και παράγοντες που αφορούν στο ίδιο το δίκτυο, όπως η απόσταση μεταξύ των κόμβων και την ύπαρξη ή όχι εμποδίων ανάμεσα σε αυτούς. Παράλληλα, ωστόσο, χρησιμοποιήθηκαν και τιμές όπως η τιμή LQI αλλά και η τιμή του RSS που επιβεβαίωσαν τα αποτελέσματα.

Ε΄.1 Αποτελέσματα μέσω της συνεχόμενης αποστολής πακέτων

Για να υλοποιηθεί η μελέτη της ποιότητας μεταξύ των ζεύξεων του πειραματικού δικτύου, αρχικά, είναι απαραίτητη η εύρεση των κόμβων που είναι «γείτονες», δηλαδή όσων βρίσκονται στην εμβέλεια του σήματος του άλλου. Ο Πίνακας Ε΄.1 εμφανίζει την ύπαρξη ή όχι σύνδεσης μεταξύ των κόμβων, αλλά και το πόσο ισχυρή είναι η σύνδεση με βάση την τιμή Link Quality Indicator (LQI)[19], που αποτυπώνεται μέσω του προγράμματος XCTU και είναι ορατή και στο Σχήμα Ε΄.9. Επίσης, αξίζει να σημειωθεί ότι οι γείτονες που προέκυψαν από τα μηνύματα διέφεραν ελάχιστα με τον πίνακα που προέκυψε μέσω του XCTU. Το γεγονός αυτό συνέβη στις περιπτώσεις όπου μία ζεύξη έχει χαρακτηριστεί αδύναμη, με το χρώμα γκρι στον Πίνακα Ε΄.1. Μέσω αυτών των πολύ αδύναμων ζεύξεων δεν ήταν εφικτή η ανταλλαγή μηνυμάτων και συνεπώς δεν αναγνωρίστηκαν και οι κόμβοι στον κώδικα ως γείτονες. Αυτό επιβεβαιώνεται

Πίνακας Ε'.1: Πίνακας Γειτνίασης μεταξύ των κόμβων του δικτύου.

Node Id	1	2	3	4	5	6	7	8	9	10	11	12
1		1	0	0	0	0	0	0	0	0	1	0
2	1		1	0	0	1	0	0	1	1	1	1
3	0	1		1	1	1	0	0	1	1	1	1
4	0	0	1		1	1	0	0	0	0	0	0
5	0	0	1	1		1	0	0	0	1	0	0
6	0	1	1	1	1		0	0	0	0	0	0
7	0	0	1	0	0	0		1	1	1	1	1
8	0	0	0	0	0	0	1		1	1	0	1
9	0	1	1	0	0	0	1	1		1	0	1
10	0	1	1	0	1	0	1	1	1		0	1
11	1	1	1	0	0	0	1	0	0	0		0
12	0	1	1	0	0	0	1	1	1	1	0	

Τα χρώματα που αναπαριστούν τον αριθμό ένα (True), δηλαδή την ύπαρξη σύνδεσης, επιλέχθηκαν με βάση το LQI. Το πράσινο υποδεικνύει μία βέλτιστη σύνδεση, ακολουθεί το χρώμα μπλε, έπειτα το κόκκινο και τέλος το γκρι.

και στον Πίνακα Ε'.2, η επιτυχημένη παράδοση των μηνυμάτων μεταξύ των αντίστοιχων αυτών κόμβων ήταν ελάχιστη και πολλές φορές δεν υπήρξε καμία επιτυχημένη προσπάθεια.

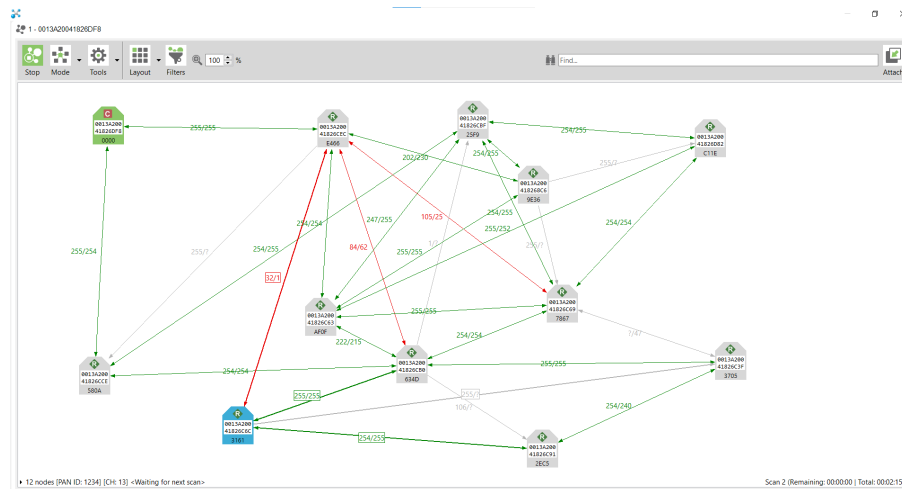
Η ποιότητα μεταξύ δύο ζεύξεων εκ των πραγμάτων είναι εξαρτώμενη από την αποστολή επιτυχημένων ή μη πακέτων μεταξύ των κόμβων. Ειδικότερα, η ύπαρξη επιτυχημένων αποστολών πακέτων μεταξύ δύο κόμβων είναι ένας ακράδαντος παράγοντας, ο οποίος υποδηλώνει την ποιότητα μεταξύ της συγκεκριμένης ζεύξης. Επομένως, κρίθηκε αναγκαία η διερεύνηση του ποσοστού λήψης μηνυμάτων μεταξύ των κόμβων του πειραματικού δικτύου.

Κάθε κόμβος απέστειλε έναν συγκεκριμένο αριθμό μηνυμάτων και όσοι από τους υπόλοιπους λάμβαναν ένα τέτοιο μήνυμα το κατέγραφαν. Ο Πίνακας Ε'.2 παρουσιάζει το πλήθος των επιτυχημένων μεταδόσεων μηνυμάτων που έλαβαν χώρα μεταξύ δύο ζεύξεων. Έτσι, παρατηρήθηκε ότι βασικός παράγοντας που επηρέασε την επιτυχημένη λήψη μηνυμάτων ήταν η απόσταση δύο κόμβων. Οι κόμβοι, οι οποίοι είχαν μικρότερη απόσταση μεταξύ τους εμφάνισαν μεγαλύτερη επιτυχία. Ωστόσο, παρατηρήθηκε και το γεγονός της ύπαρξης κόμβων που δεν ακολουθούσαν

Πίνακας Ε'.2: Το ποσοστό των επιτυχημένων μεταδόσεων μεταξύ των κόμβων.

Node Id	1	2	3	4	5	6	7	8	9	10	11	12
1		0.5%	-	-	-	-	-	-	98%	92%	0.5%	-
2	0.5%		0,5%	-	-	46%	-	-	1%	1%	0%	2%
3	-	0%		0%	0%	0%	4%	-	4%	0%	0.5%	-
4	-	-	-		0%	0%	-	-	-	-	-	-
5	-	-	0%	0%		0%	-	-	-	13%	-	-
6	-	86%	0%	0%	0%		-	-	-		98%	-
7	-	-	-	-	-	-		2%	0%	0%	8%	0%
8	-	-	-	-	-	-	1%		1%	0%	-	0%
9	-	1%	2%	-	-	-	0%	0%		0%	0%	0%
10	85%	9%	3%	-	80%	-	2%	0%	0%		0%	0%
11	0%	0%	0%	-	-	-	6%	-	78%	93%		98%
12	-	0%	99%	-	-	-	0%	0%	0%	0%	66%	

Η πρώτη στήλη αναπαριστά τους κόμβους που έλαβαν τα πακέτα ενώ η πρώτη γραμμή αναπαριστά τους αποστολείς.



Σχήμα Ε'.9: Οι ζεύξεις μεταξύ των κόμβων και η τιμή LQI , όπως παρουσιάζεται μέσα από το πρόγραμμα *XCTU*.

τον κανόνα αυτό. Συγκεκριμένα, ο κόμβος με αναγνωριστικό «4» δεν έλαβε κανένα επιτυχημένο μήνυμα από τον κόμβο «3», παρά το γεγονός της μικρής απόστασης μεταξύ των δύο κόμβων και της ύπαρξης πολλών κοινών γειτόνων. Επίσης και στο πρώτο μέρος του κώδικα που αναφέρθηκε στο Κεφάλαιο Δ' ο κόμβος «4» δεν αναγνώρισε τον κόμβο «3», ενώ στον Πίνακα Ε'.1 εμφανίζεται να υπάρχει μία εξαιρετικά χαμηλής ποιότητας ζεύξη από την πλευρά του κόμβου «4».

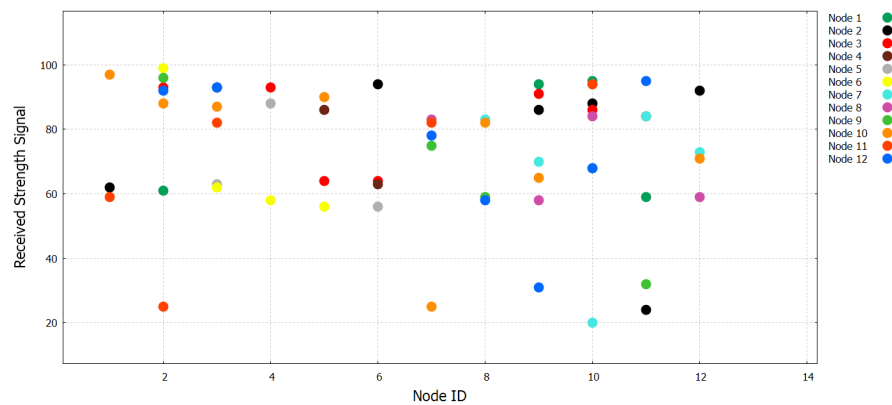
Ταυτόχρονα, παρατηρήθηκε και το αντίθετο γεγονός, καθώς για παράδειγμα ο κόμβος με αναγνωριστικό «1», δηλαδή ο coordinator, παρά το γεγονός ότι είχε μόνο δύο γείτονες, παρατηρήθηκε ότι έλαβε και άλλα ελάχιστα δεδομένα από άλλους κόμβους ορισμένες χρονικές στιγμές.

Ε'.2 Received Strength Signal

Η τιμή RSS υποδηλώνει την ισχύ του σήματος του τελευταίου hop του πακέτου, που έλαβε ο κόμβος και αναρτήθηκε με την χρήση μίας εντολής AT-command. Η τιμή του RSS είχε μια αρκετά μεγάλη διακύμανση, καθώς κυμαινόταν από -96dBm έως και -30dBm, όπως εμφανίζεται και στο Σχήμα Ε'.10. Παρατηρήθηκε ότι ο βασικός παράγοντας που επηρέασε αυτή την τιμή, όπως και την τιμή LQI ήταν η απόσταση μεταξύ των κόμβων. Ειδικότερα, η αύξηση

της απόστασης είχε ως αποτέλεσμα την αύξηση της απόλυτης τιμής του RSS, την μείωση αντίστροφα της τιμής LQI, αλλά και την ύπαρξη επιπλέον αποτυχημένων αποστολών πακέτων, όπως αναφέρεται και παραπάνω.

Επιπλέον, παρατηρήθηκε και επιβεβαιώθηκε η παρατήρηση από την βιβλιογραφική έρευνα ότι το RSS σε εσωτερικό χώρο, όπου βρίσκονται και οι συγκεκριμένοι κόμβοι δεν έφτανε σε τιμές μεγαλύτερες από περίπου -96dBm [14]. Όταν η τιμή του RSS πλησίαζε την «ακραία» για το συγκεκριμένο δίκτυο τιμή των -96dBm, ξεκινούσε και η απώλεια μηνυμάτων μεταξύ των κόμβων. Το γεγονός αυτό προβάλλεται ιδιαίτερα μεταξύ των κόμβων «6» και «7», όπου το RSS, όπως αποτυπώθηκε «άγγιζε» τις τιμές αυτές και το ποσοστό των αποτυχημένων μεταδόσεων, όπως παρουσιάζεται και στον Πίνακα Ε'.2, ήταν ιδιαίτερα αυξημένο.

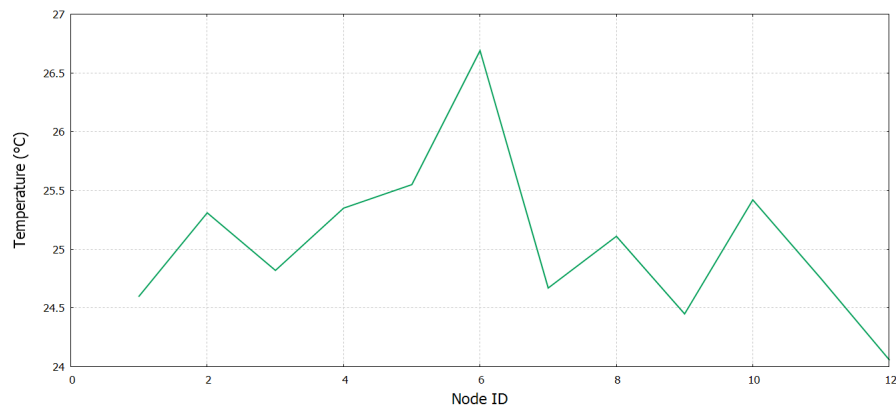


Σχήμα Ε'.10: Ο μέσος όρος της απόλυτης τιμής του RSS μεταξύ των κόμβων του δικτύου.

Ε'.3 Αποτελέσματα Θερμοκρασίας και Υγρασίας

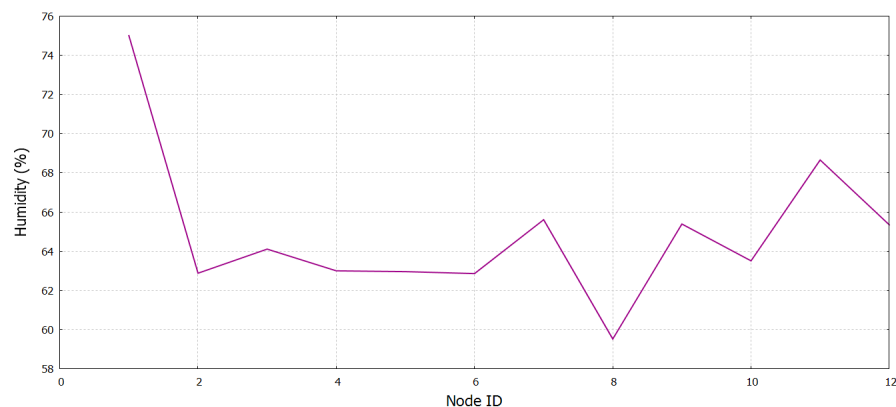
Παράλληλα, με την αποστολή μηνυμάτων παρακολουθήθηκε και η τιμή της θερμοκρασίας και της υγρασίας όπως παρουσιάζονται κατά μέσο όρο στα Σχήματα Ε'.11 και Ε'.12, αντίστοιχα. Η καταγραφή της θερμοκρασίας επιτυγχάνονταν κάθε φορά που κάποιος λάμβανε ένα επιτυχημένο μήνυμα από τα εκατό μηνύματα του δεύτερου σκέλους του προγράμματος (Δ'), τα οποία αποστέλλονταν με χρονική καθυστέρηση τριών δευτερολέπτων, οπότε σε κάθε νέο μήνυμα γινόταν εκ νέου καταμέτρηση της θερμοκρασίας (Κεφάλαιο Γ', ο αισθητήρας DHT22 υλοποιεί μετρήσεις με διαφορά δύο δευτερολέπτων). Η μεταβολή της θερμοκρασίας από κόμβο σε κόμβο (Σχήμα Ε'.11) αλλά και της υγρασίας παρατηρείται ότι δεν μεταβάλλονται ιδιαίτερα.

Το γεγονός ότι οι κόμβοι του δικτύου και οι αισθητήρες ήταν στο ίδιο εσωτερικό περιβάλλον την ίδια χρονική περίοδο οδήγησε στην ύπαρξη και αποτύπωση παρόμοιων τιμών μεταξύ της θερμοκρασίας και της υγρασίας. Ειδικότερα, το εύρος του μέσου όρου της θερμοκρασίας των κόμβων «φτάνει» οριακά τους 3(° C).



Σχήμα Ε'.11: Ο μέσος όρος θερμοκρασίας των κόμβων.

Η μεγαλύτερη τιμή του μέσου όρου παρατηρείται ότι εμφανίζεται στον κόμβο με αναγνωριστικό «6» που βρίσκεται στον τρίτο και τελευταίο όροφο του κτιρίου. Αντίστοιχα, και η υγρασία (Σχήμα Ε'.12) παρατηρείται ότι δεν μεταβάλλεται ιδιαίτερα στους περισσότερους κόμβους του δικτύου. Μόνο ο κόμβος «1», δηλαδή ο coordinator του δικτύου που βρίσκεται ακριβώς δίπλα σε παράθυρο του κτιρίου, εμφανίζει μία περισσότερο αυξημένη τιμή.



Σχήμα Ε'.12: Ο μέσος όρος θερμοκρασίας των κόμβων.

Ε'.4 Αποτέλεσμα Μεταβολής Θερμοκρασίας και Υγρασίας

Το γεγονός της σταθερότητας των τιμών της θερμοκρασίας και της υγρασίας μεταξύ των κόμβων κατά την διάρκεια των πειραμάτων, καθώς όλοι οι κόμβοι βρισκόντουσαν σε αντίστοιχες συνθήκες, οδήγησαν στην δημιουργία αποριών σχετικά με την ύπαρξη ή όχι σύνδεσης της θερμοκρασίας και της υγρασίας με την ποιότητα των ζεύξεων μεταξύ των κόμβων του δικτύου.

Επομένως, καθώς θεωρήθηκε πιθανή η επιρροή τους και καθώς είναι βέβαιο ότι κατά την διάρκεια του έτους και ανάλογα με την εποχή η θερμοκρασία και η υγρασία θα μεταβάλλονται, κρίθηκε αναγκαία η τεχνητή μεταβολή τους για την καταγραφή των αποτελεσμάτων των μεταβολών αυτών. Έτσι, υλοποιήθηκε τεχνητά μεταβολή της θερμοκρασίας με τη χρήση θερμοπομπών σε διάφορους κόμβους του δικτύου.

Ε'.4.1 Απώλεια Μηνυμάτων

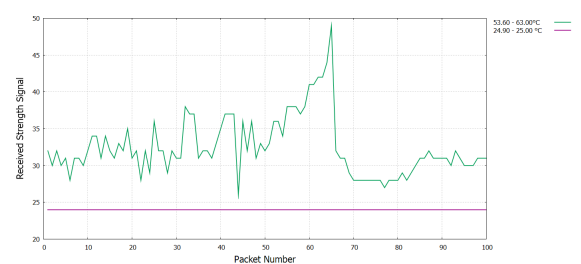
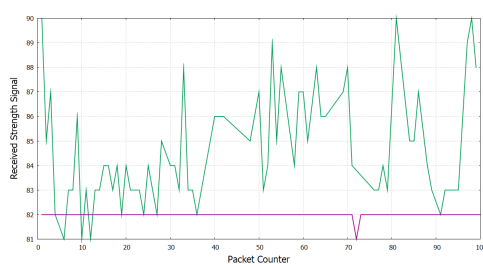
Τα αποτελέσματα που αποδόθηκαν με την αύξηση της θερμοκρασίας και αντίστοιχα την μείωση της υγρασίας ήταν ποικιλόμορφα. Αρχικά, παρατηρήθηκε το γεγονός της απώλειας πακέτων μεταξύ ζεύξεων του δικτύου. Στους κόμβους που αυξήθηκε η θερμοκρασία, όπως παρουσιάζεται και στον Πίνακα Ε'.3, παρατηρήθηκε αύξηση των αποτυχημένων μεταδόσεων. Συγκεκριμένα, ο κόμβος με αναγνωριστικό «2» του οποίου η θερμοκρασία έφτασε έως και τους 53.80 ° C κατά την διάρκεια επικοινωνίας του με τον κόμβο «3», παρατηρήθηκε ραγδαία μείωση των επιτυχημένων μεταδόσεων που «άγγιζε» έως και -76%, χωρίς να υλοποιηθεί καμία άλλη αλλαγή μεταξύ των δύο κόμβων, ούτε στο λογισμικό μέρος, αλλά ούτε και στο υλικό ή την τοποθεσία των κόμβων. Το γεγονός αυτό προβάλλει την μεγάλη επίδραση του περιβάλλοντος και συγκεκριμένα της υγρασίας και της θερμοκρασίας στις μεταδόσεις του δικτύου. Αντίστοιχα, και στον κόμβο «10» η λήψη επιτυχημένων πακέτων από τον κόμβο «9» μειώθηκε εξαιτίας της αύξησης της θερμοκρασίας και την μείωση της υγρασίας. Κατά την διάρκεια της συγκεκριμένης μετάδοσης ωστόσο παρατηρήθηκε και μείωση της τιμής του RSS.

Πίνακας Ε'.3: Ο μέσος όρος της RSS και της απώλειας πακέτων σε σχέση με τον μέσο όρο των τιμών της θερμοκρασίας και υγρασίας του περιβάλλοντος.

Sender (ID)	Receiver (ID)	RSS (dBm)	Packet Loss	Temperature (° C)	Humidity(%)
9	10	-79.5	78	24.40	72.4
9	10	-93.27	89	38.42	31.30
3	2	93	0	25	62.7
3	2	89.59	76	48.87	31.19

Ε'.4.2 Αποτελέσματα Μεταβολής Θερμοκρασίας και Υγρασίας στην τιμή RSS

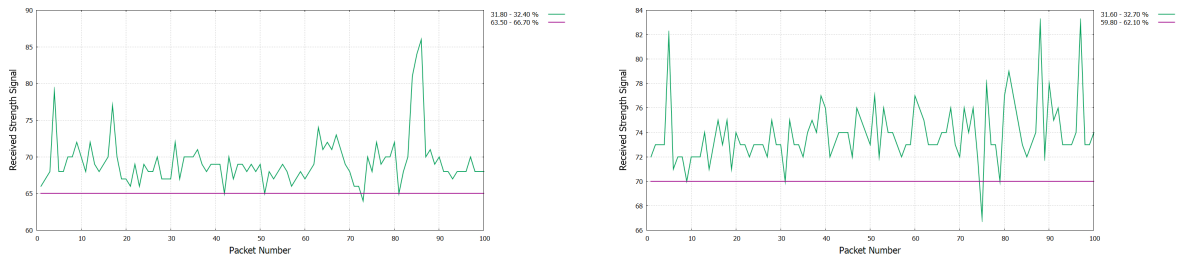
Ταυτόχρονα, παρατηρήθηκε αύξηση της τιμής RSS και σε άλλες μεταδόσεις στις οποίες αυξήθηκε η θερμοκρασία με τεχνητό τρόπο. Στο Σχήμα Ε'.13 είναι ιδιαίτερα ορατή η αυξημένη απόλυτη τιμή του RSS κατά την διάρκεια λήψης μηνυμάτων σε κόμβους όπου η θερμοκρασία τους αυξήθηκε αισθητά. Ουσιαστικά, η τιμή του RSS στην πραγματικότητα μικραίνει (καθώς έχει αρνητικό πρόσημο), υποδηλώνοντας ότι η ποιότητα μεταξύ της ζεύξης επηρεάστηκε από την αλλαγή της θερμοκρασίας (αντίστοιχα με την αύξηση της θερμοκρασίας μειώνεται η τιμή της υγρασίας του περιβάλλοντος). Ωστόσο, σε αντίθεση με τις ζεύξεις που εμφανίζονται στον Πίνακα Ε'.3, δεν υπήρξε απώλεια πακέτων.



α. Μεταβίβαση μηνυμάτων από τον «3» στον «11». β. Μεταβίβαση μηνυμάτων από τον «11» στον «2».

Σχήμα Ε'.13: Οι απόλυτες τιμές του RSS για κάθε πακέτο σε δύο διαφορετικά πεδία θερμοκρασίας του κόμβου παραλήπτη.

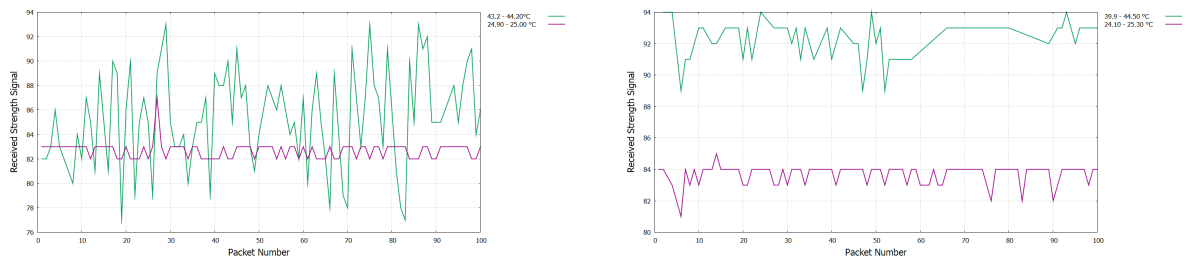
Επιπλέον, παρατηρήθηκε ότι η τιμή RSS ήταν εντελώς σταθερή ή με πολύ μικρές διακυμάνσεις (Σχήμα Ε'.13.α) όταν η θερμοκρασία δεν εμφάνιζε έντονες μεταβολές σε σχετικά χαμηλές θερμοκρασίες. Αντίθετα, η τιμή του RSS άρχισε να μεταβάλλεται διαρκώς χωρίς να σταθεροποιείται, όταν αυξανόταν η θερμοκρασία (Σχήμα Ε'.13.β) και μειωνόταν η υγρασία κατά συνέπεια (Σχήμα Ε'.14).



α. Μεταβίβαση μηνυμάτων από τον «9» στον «7». β. Μεταβίβαση μηνυμάτων από τον «9» στον «10».

Σχήμα Ε'.14: Έντονη μεταβολή της τιμής RSS μεταξύ μίας ζεύξης κατά την μείωση της υγρασίας.

Η μεγάλη και συνεχόμενη μεταβολή του RSS είναι ιδιαίτερα ορατή στο Σχήμα Ε'.15.α καθώς στο συγκεκριμένο σχήμα η τιμή αυτή έχει μία διακύμανση και στις χαμηλότερες θερμοκρασίες. Ωστόσο, το εύρος της τιμής όταν η θερμοκρασία είναι σε ιδιαίτερα υψηλά επίπεδα είναι σχεδόν δέκα φορές μεγαλύτερο από το εύρος αυτής στους περίπου 25° C.



α. Μεταβίβαση μηνυμάτων από τον «8» στον «10». β. Μεταβίβαση μηνυμάτων από τον «11» στον «7».

Σχήμα Ε'.15: Έντονη μεταβολή της τιμής RSS μεταξύ μίας ζεύξης κατά την αύξηση της θερμοκρασίας.

Κεφάλαιο Γ΄

Σύνοψη και Συμπεράσμα

Η μελέτη των δικτύων ασύρματων αισθητήρων και συγκεκριμένα του πειραματικού δικτύου του Ιονίου Πανεπιστημίου για τη συγκεκριμένη πτυχιακή εργασία κάπου εδώ ολοκληρώνεται.

Γ΄.1 Σύνοψη της εργασίας

Η συγκεκριμένη πτυχιακή εργασία μελετά τις παραμέτρους λειτουργίας των ζεύξεων του πειραματικού δικτύου του Ιονίου Πανεπιστημίου. Αρχικά, αναφέρονται και αναλύονται τα Ασύρματα Δίκτυα Αισθητήρων αλλά και οι εφαρμογές αυτών σε διάφορους κλάδους και επαγγέλματα. Στην συνέχεια, περιγράφεται το συγκεκριμένο δίκτυο τόσο από την πλευρά της τοπολογίας όσο και από τον εξοπλισμό που χρησιμοποιείται για την διεξαγωγή των πειραμάτων. Οι μετρήσεις που λήφθηκαν και καταγράφηκαν από τα διάφορα πειράματα αλλά και από το πρόγραμμα XCTU, προσέφεραν μία αξιολόγηση της ποιότητας των ζεύξεων.

Γ΄.2 Συμπεράσματα

Το Πειραματικό Δίκτυο που έχει εγκατασταθεί στο Ιόνιο Πανεπιστήμιο αποτελεί ένα μικρό σε αριθμό κόμβων Ασύρματο Δίκτυο Αισθητήρων. Ωστόσο, διευκολύνει τη διεξαγωγή πειραμάτων και μελετών σε τέτοιου είδους δίκτυα. Η ποιότητα των ζεύξεων μεταξύ των κόμβων του δικτύου είναι μία πολυπαραγοντική έννοια, καθώς επηρεάζεται από πολλές συνθήκες, περιβαλλοντολογικές, φυσικές, προγραμματιστικές. Για τον λόγο αυτό συλλέχθηκαν διάφορες

πληροφορίες. Αρχικά, μελετήθηκε το ποσοστό των επιτυχημένων αποστολών πακέτων μεταξύ δύο κόμβων, καθώς αυτό αντικατοπτρίζει και προβάλλει την ποιότητα μίας ζεύξης. Με κάθε επιτυχημένη αποστολή πακέτου συλλέχθηκαν και δεδομένα που αφορούσαν στο εξωτερικό περιβάλλον κάθε κόμβου, την υγρασία και την θερμοκρασία, αλλά και μετρήσεις, οι τιμές των οποίων προβάλλουν μία αξιολόγηση της ποιότητας των ζεύξεων. Τα δεδομένα που αφορούσαν στο εξωτερικό περιβάλλον δεν εμφάνισαν καμία έντονη και ακραία απόκλιση. Αντίθετα, οι τιμές RSS και LQI κυμαίνονται σε ένα μεγάλο εύρος τιμών η κάθε μία αντίστοιχα, προβάλλοντας κυρίως την επιρροή της απόστασης μεταξύ των κόμβων στην ποιότητα των ζεύξεων. Παράλληλα, οι τιμές αυτές είναι δυνατόν να λειτουργήσουν και αντίστροφα, καθώς γνωρίζοντάς τις είναι εφικτή η προσεγγιστική εύρεση της απόστασης μεταξύ των δύο κόμβων. Ωστόσο, υπήρξαν και πολλές εξαιρέσεις που επιβεβαίωσαν τον κανόνα και έδειξαν ότι κάποιες μεταβολές στον εσωτερικό χώρο μπορεί να δημιουργήσουν στιγμιαία ή να διακόψουν μία ζεύξη μεταξύ δύο κόμβων.

Γ'.3 Συνεισφορά

Η μελέτη των παραμέτρων λειτουργίας των ζεύξεων του πειραματικού δικτύου του Ιονίου Πανεπιστημίου, οδήγησε τόσο στην εξοικείωση με τα Ασύρματα Δίκτυα Αισθητήρων, όσο και με την λειτουργία αυτών. Ο έλεγχος όλων των παραγόντων που ήταν εφικτό να επηρεάζουν μία ζεύξη επιβεβαίωσαν όλα όσα μελετήθηκαν στην βιβλιογραφική έρευνα αλλά οδήγησαν και στην εύρεση νέων αποτελεσμάτων. Συγκεκριμένα, μέσα από τα πειράματα και τις συνεχείς αλλαγές αυτών, εντοπίστηκε η επιρροή της απόστασης μεταξύ δύο κόμβων στην ποιότητα της μεταξύ τους ζεύξης, αλλά και μία συσχέτιση αυτής με την ανόδο της θερμοκρασίας και της πτώσης της υγρασίας. Η συσχέτιση αυτών των περιβαλλοντολογικών παραγόντων επιβεβαιώθηκε μέσω των αποτελεσμάτων. Το αντίκτυπο της ραγδαίας αύξησης της θερμοκρασίας αποτυπώθηκε τόσο στην τιμή RSS όσο και στην απώλεια πακέτων μεταξύ των κόμβων.

Βιβλιογραφία

- [1] Σαφάκια Ι. *Μελέτη, ανάπτυξη και αξιολόγηση επικοινωνιακών πρωτοκόλλων ασύρματων δικτύων αισθητήρων για περιβαλλοντικές εφαρμογές ευρείας κλίμακας (Doctoral dissertation)*. PhD thesis, 2009.
- [2] Παύλου, Β. *Αλγόριθμοι δρομολόγησης για εξοικονόμηση ενέργειας στα ασύρματα δίκτυα αισθητήρων*. Master's thesis, 2010.
- [3] Σπένδας, Α. *Ασύρματα δίκτυα αισθητήρων και ζητήματα ασφάλειας*. PhD thesis, 2012.
- [4] Στάμος Π. *Σχεδιασμός και υλοποίηση δικτύου αισθητήρων για εφαρμογή σε έξυπνο δωμάτιο με την χρήση του opnet*. Master's thesis, 2013.
- [5] Στεφανίδης, Μ. *Ασύρματα δικτυακά αισθητήρια Zigbee*. PhD thesis, 2016.
- [6] Μπούντου, Ε. *Ασύρματα δίκτυα αισθητήρων*. PhD thesis, 2018.
- [7] Κετσεμενίδης, Ε. *Ανάπτυξη εφαρμογών με Αρδουινο*. Master's thesis, 2018.
- [8] Μποϊντά, Α. *Εφαρμογή συστημάτων ασύρματων δικτύων αισθητήρων στις Υδατοκαλλιέργειες*. Master's thesis, 2020.
- [9] Moshaddique Al Ameen, Jingwei Liu, and Kyungsup Kwak. Security and privacy issues in wireless sensor networks for healthcare applications. *Journal of medical systems*, 36(1):93–101, 2012.
- [10] Hande Alemdar and Cem Ersoy. Wireless sensor networks for healthcare: A survey. *Computer networks*, 54(15):2688–2710, 2010.
- [11] Monika Asija. Mac address. *IRA-International Journal of Technology & Engineering*, 3(1):41–60, 2016.

- [12] Bezerra, T and Silva, S and Silva, E and Sousa, M and Cavalcante, M. Performance evaluation of zigbee transmissions on the grass environment. In *UBICOMM 2014, The Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 287–290, 2014.
- [13] Neil Briscoe. Understanding the osi 7-layer model. *PC Network Advisor*, 120(2):13–15, 2000.
- [14] Cama-Pinto, Alejandro and Pineres-Espitia, Gabriel and Caicedo-Ortiz, Jose and Ramírez-Cerpa, Elkin and Betancur-Agudelo, Leonardo and Gómez-Mula, Francisco. received strength signal intensity performance analysis in wireless sensor network using arduino platform and xbee wireless modules. *International Journal of Distributed Sensor Networks*, 13(7):1550147717722691, 2017.
- [15] S Christodoulou, Agathoklis Agathokleous, A Kounoudes, and M Milis. Wireless sensor networks for water loss detection. *European Water*, 30:41–48, 2010.
- [16] Zeev Collin and Shlomi Dolev. Self-stabilizing depth-first search. *Information Processing Letters*, 49(6):297–301, 1994.
- [17] André Cunha, Anis Koubaa, Ricardo Severino, and Mário Alves. Open-zb: an open-source implementation of the ieee 802.15. 4/zigbee protocol stack on tinys. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–12. IEEE, 2007.
- [18] John D Day and Hubert Zimmermann. The osi reference model. *Proceedings of the IEEE*, 71(12):1334–1340, 1983.
- [19] Prashant V Mane Deshmukh, BP Ladgaonkar, and SK Tilekar. Investigation of wsn parameters for realization of quality of service. *Journal of Science and Technology*, 4(6):37–42, 2019.
- [20] Jonas Auråen. low-cost ctd instrument-arduino based ctd for autonomous measurement platform. Master’s thesis, 2019.
- [21] Yusuf Abdullahi Badamasi. the working principle of an arduino. In *2014 11th international conference on electronics, computer and computation (ICECCO)*, pages 1–4. IEEE, 2014.

- [22] Mihai Bogdan. how to use the dht22 sensor for measuring temperature and humidity with the arduino board. *ACTA Universitatis Cibiniensis*, 68(1):22–25, 2016.
- [23] Ali Çaylı, Adil Akyüz, A Nafi Baytorun, Sait Üstün, and Ali Selçuk Mercanlı. the feasibility of a cloud-based low-cost environmental monitoring system via open source hardware in greenhouses. 2018.
- [24] Ahmed Bashir Fakhri, Sadik Kamel Gharghan, and Saleem Latteef Mohammed. path-loss modelling for wsn deployment in indoor and outdoor environments for medical applications. *International Journal of Engineering & Technology*, 7(3):1666–1671, 2018.
- [25] Scott Fitzgerald and Michael Shiloh. *Arduino Projects Book*. Arduino, 2021.
- [26] V Kalyani. enhancing localization accuracy in wireless sensor networks using range-free methods and rss measurements. In *2018 International Conference on Recent Trends in Advance Computing (ICRTAC)*, pages 136–142. IEEE, 2018.
- [27] Medilla Kusriyanto and Agusti Anggara Putra. weather station design using iot platform based on arduino mega. In *2018 International Symposium on Electronics and Smart Devices (ISESD)*, pages 1–4. IEEE, 2018.
- [28] Pengfei Li, Jiakun Li, Luhua Nie, and Bo Wang. research and application of zigbee protocol stack. In *2010 International Conference on Measuring Technology and Mechatronics Automation*, volume 2, pages 1031–1034. IEEE, 2010.
- [29] Lili Liang, Lianfen Huang, Xueyuan Jiang, and Yan Yao. design and implementation of wireless smart-home sensor network based on zigbee protocol. In *2008 International Conference on Communications, Circuits and Systems*, pages 434–438. IEEE, 2008.
- [30] Asterios Papamichail, Aikaterini Georgia Alvanou, Alexandros Zervopoulos, Konstantinos Bezas, Spiridon Vergis, George Koufoudakis, Konstantinos Oikonomou, and Georgios Tsoumanis. description of the ionian university’s campus wireless network testbed infrastructure. In *2019 4th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pages 1–4. IEEE, 2019.

- [31] Rajeev Piyare and Seong-ro Lee. performance analysis of xbee zb module based wireless sensor networks. *International Journal of Scientific & Engineering Research*, 4(4):1615–1621, 2013.
- [32] C Muthu Ramya, M Shanmugaraj, and R Prabakaran. study on zigbee technology. In *2011 3rd International Conference on Electronics Computer Technology*, volume 6, pages 297–301. IEEE, 2011.
- [33] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.
- [34] Óscar Gama, Paulo Carvalho, José A Afonso, and PM Mendes. Quality of service support in wireless sensor networks for emergency healthcare services. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1296–1299. IEEE, 2008.
- [35] Mohamed Amine Kafi, Yacine Challal, Djamel Djenouri, Abdelmadjid Bouabdallah, Lyes Khelladi, and Nadjib Badache. A study of wireless sensor network architectures and projects for traffic light monitoring. *Procedia computer science*, 10:543–552, 2012.
- [36] Leo Louis. working principle of arduino and u sing it. *International Journal of Control, Automation, Communication and Systems (IJCACS)*, 1(2):21–29, 2016.
- [37] B Reshika, Nagalakshmi Naik, et al. Automated management of food grain warehouse condition. *International Journal of Research in Engineering, Science and Management*, 3(8):84–87, 2020.
- [38] Hubert Zimmermann. Osi reference model-the iso model of architecture for open systems interconnection. *IEEE Transactions on communications*, 28(4):425–432, 1980.

Παράρτημα Α΄

Στο Παράρτημα παρατίθεται το αρχείο του κώδικα που υλοποιήθηκε στα πλαίσια εκπόνησης της πτυχιακής εργασίας και συλλογής δεδομένων για την αξιολόγηση της ποιότητας ζεύξεων μεταξύ δύο κόμβων.

```
#include <DHT.h>
#include <MessageList.h>
#include <SPI.h>
#include <SD.h>
#include <XBee.h>
// create the XBee object
XBee xbee = XBee();
//contains messages information
MessageList *msg = new MessageList();
uint8_t payload[] = { ' ', ' ', ' ', ' ', ' ', ' ', ' ' };
/*
* 72: "H" - Hello message
* 67: "C" - I am your child
* 83: "S" - Send 100 messages
* 80: "P" - Ping
* 66: "B" - Bring me your data
* 68: "D" - Data
*/
//broadcast address & my father address
XBeeAddress64 broadcast = XBeeAddress64(0x00000000, 0x0000ffff);
XBeeAddress64 father = XBeeAddress64();
```

```

XBeeAddress64 *toSendAddress = new XBeeAddress64();
//Here we create a frame of type ZBTxRequest
ZBTxRequest *zbTx = new ZBTxRequest(broadcast, payload, sizeof(payload));
ZBTxStatusResponse txStatus = ZBTxStatusResponse();
//This is the type of message a node receives
//after another node has sent a ZBTxRequest
ZBExplicitRxResponse *explicitResponse = new ZBExplicitRxResponse();
//messages, neighbours and children counters and arrays
byte neighboursCount = 0;
byte childCount = 0;
uint32_t neighboursLow[10] = {0x0, 0x0, 0x0, 0x0, 0x0,
    0x0, 0x0, 0x0, 0x0, 0x0};
uint32_t neighboursHigh;
uint32_t childrenLow[10] = {0x0, 0x0, 0x0, 0x0, 0x0,
    0x0, 0x0, 0x0, 0x0, 0x0};
uint32_t childrenHigh;
bool isCoordinatorEnabled = false;
bool sentOutNeighboursBroadcast = false;
bool neighbor;
bool changed = false;
byte neig = 0;
bool started = false;
//Counters
int pingCounter = 0;
byte responseCounter = 0;
byte collectCounter = 0;
//timers
unsigned long timeout;
unsigned long timeToCollect;
unsigned long timer;
//temperature and humidity value
float t;
float h;

```

```

AtCommandResponse nodeID;
uint8_t* id;
File myFile;
const char *filename = "myD.txt";
const char *otherFile = "oDt.txt";
const char *childF = "child.txt";
const char *neighF = "neigh.txt";
#define DHTPIN 3      // what pin we're connected to
#define DHTTYPE DHT22 // DHT 22
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  randomSeed(analogRead(0));
  delay(1000);
  sendATCommand("CE"); //Am I coordinator?
  delay(300);
  nodeID = sendATCMD("NI"); //My Node Indedifier
  //Store AT-Command Response's Value
  id = nodeID.getValue();
  payload[1] = id[0];
  Serial.print(F("Initializing SD card..."));
  if (!SD.begin(4)) {
    Serial.println(F("initialization failed!"));
    while (1); }
  if (isCoordinatorEnabled){ //if I am the Coordinator
    delay(7000);
    payload[0] = 72;
    //First Transmition
    delete zbTx;
    zbTx = new ZBTxRequest(broadcast, payload, sizeof(payload));
    sendTheMessage(0x1);
    timeout = millis(); //start countdown
    timeToCollect = millis(); // start countdown to collect data

```

```

}
dht.begin();
timer = millis();
}
void loop(){
changed = false;
//if there is a message -> "changed" becomes true
checkForPacketDelivery();
//After 80sec the algorithm begins!
if(isCoordinatorEnabled && (millis() - timeout > 80000) && !started){
if(millis()-timeToCollect > 25000000){
//Time to collect Data
payload[0] = 66; //Bring me your Data
collectCounter = 1;
timeToCollect = millis(); //Restart Counter
}
else{
//First part of the algorithm
//Start sending 100 messages
payload[0] = 83;
}
neig = 0;
delete toSendAddress;
//toSendAddress is my first child's address high and low
toSendAddress = new XBeeAddress64(childrenHigh, childrenLow[neig]);
delete zbTx;
//Send the message ("83" or "66") to my first child
//Start either the first part of the algorithm
//either the data's collection
zbTx = new ZBTxRequest(*toSendAddress, payload, sizeof(payload));
sendTheMessage(0x1);
neig ++;
started = true; }

```

```

if(changed && msg->getMessageValue() == 72){
started = false;
//Check if sender address exists in neighbours array
neighbor = false;
for(int i = 0; i < neighboursCount; i++){
if (msg->getSenderAddressLow() == neighboursLow[i]){
neighbor = true;
break; } }
if(!neighbor){ //if the sender there is not in my neig array
//Add sender address to neighbours array
delay(random(80, 400));
if(neighboursCount == 0 && !isCoordinatorEnabled){
//if it is the first neighbour establish the father-child relationship
payload[0] = 67;
father = XBeeAddress64(msg->getSenderAddressHigh(),
    msg->getSenderAddressLow());
delete zbTx;
zbTx = new ZBTxRequest(father, payload, sizeof(payload));
sendTheMessage(0x1);
//the high address is the same at every node in the network
//so I store it in a variable the first time the node receives a message
neighboursHigh = msg->getSenderAddressHigh();
childrenHigh = msg->getSenderAddressHigh(); }
//the low address differs from node to node
//that's why, low addresses stored in an array
neighboursLow[neighboursCount] = msg->getSenderAddressLow();
neighboursCount++;
//Print my neighbours' low addresses in a file
myFile = SD.open(neighF, FILE_WRITE);
myFile.print(neighboursCount);
myFile.print(", ");
myFile.println(msg->getSenderAddressLow(), HEX);
myFile.close(); }

```

```

delay(random(100, 400));
//If the node is not a coordinator
//and he has not sent out neighbours broadcast
if (!isCoordinatorEnabled && !sentOutNeighboursBroadcast) {
payload[0] = 72;
delete zbTx;
//Send the message 72 - "Hi", broadcast (Radius = 1 hop)
zbTx = new ZBTxRequest(broadcast, payload, sizeof(payload));
sendMessage(0x1);
sentOutNeighboursBroadcast = true; } }
else if(changed && msg->getMessageValue() == 67) {
//Check if the sender's address there is in my children array
neighbor = false;
for(int i = 0; i < childCount; i++){
if (msg->getSenderAddressLow() == childrenLow[i]){
neighbor = true;
break; } }
if(!neighbor){ //if there isn't
//Add the sender address to the children array
childrenLow[childCount] = msg->getSenderAddressLow();
childCount += 1;
//Print my children's addresses in a file
myFile = SD.open(childF, FILE_WRITE);
myFile.print(childCount);
myFile.print(", ");
myFile.println(msg->getSenderAddressLow(), HEX);
myFile.close(); } }
else if(changed && msg->getMessageValue() == 83) {
if(childCount > 0 && neig < childCount){
//if i've got child(children)
//send "Send" message first to my child, firstly
payload[0] = 83;
delete toSendAddress;

```

```

toSendAddress = new XBeeAddress64(childrenHigh, childrenLow[neig]);
delete zbTx;
zbTx = new ZBTxRequest(*toSendAddress, payload, sizeof(payload));
sendTheMessage(0x1);
neig++; }
else if(neig == childCount){
//if I haven't children or if I've send to all of them before
payload[0] = 80;
pingCounter = 0;
delay(random(50,200));
//Start sending 100 messages
for(int k = 0; k <100; k++) {
//insert pingCounter into the message
pingCounter++;
uint8_t msgCnt[2] = { highByte(pingCounter), lowByte(pingCounter) };
payload[2] = msgCnt[0];
payload[3] = msgCnt[1];
delay(3000);
delete zbTx;
zbTx = new ZBTxRequest(broadcast, payload, sizeof(payload));
sendTheMessage(0x1); }
delay(4000);
if(!isCoordinatorEnabled) {
//Send to my "father" to continue the algorithm
payload[0] = 83;
delete zbTx;
zbTx = new ZBTxRequest(father, payload, sizeof(payload));
sendTheMessage(0x1); }
else if(isCoordinatorEnabled) {
//if I am the coordinator - Restart the algorithm
delay(3000);
timeout = millis(); //Restart the timer
started = false; }

```

```

neig = 0; } }
else if(changed && msg->getMessageValue() == 80){
if(millis() - timer > 2000){
timer = millis();
h = dht.readHumidity();
// Read temperature as Celsius
t = dht.readTemperature();
//Print the data into the file
myFile = SD.open(filename, FILE_WRITE);
//counter
myFile.print(msg->getData()[2]);
myFile.print(msg->getData()[3]);
myFile.print(", ");
//sender id
myFile.print(msg->getData()[1]);
myFile.print(", ");
//my id
myFile.print(payload[1]);
myFile.print(", ");
//rss Value
AtCommandResponse rss = sendATCMD("DB");
for (int i = 0; i < rss.getValueLength(); i++){
myFile.print(rss.getValue()[i]); }
myFile.print(", ");
// Xbee S2C Pro Zigbee temp value
AtCommandResponse tempAT = sendATCMD("TP");
uint8_t* tem = tempAT.getValue();
myFile.print(tem[1]);
myFile.print(", ");
//sensor's values t-temperature h-humidity
myFile.print(t);
myFile.print(", ");
myFile.print(h);

```

```

myFile.print(" , ");
myFile.print(millis());
myFile.println(" , ");
myFile.close(); }
else if(changed && msg->getMessageValue() == 66) {
if(collectCounter < childCount) {
//if i've got child(children)
//send "Collect" message first to my child, firstly
payload[0] = 66;
delete toSendAddress;
toSendAddress = new XBeeAddress64(childrenHigh ,
childrenLow[collectCounter]);
delete zbTx;
zbTx = new ZBTxRequest(*toSendAddress, payload, sizeof(payload));
sendTheMessage(0x1);
collectCounter++; }
else if(collectCounter == childCount && !isCoordinatorEnabled){
//Send the stored data line by line
myFile = SD.open(filename, FILE_READ); //file open
uint8_t* sd_data = new uint8_t[150];
int i = 1;
while(myFile.available()){
sd_data[i] = myFile.read();
if (sd_data[i] == '\n'){ //new line character
delete zbTx;
sd_data[0] = 68;
//send each line to my father
zbTx = new ZBTxRequest(father, sd_data, i);
sendTheMessage(0x1);
i = 0;
delete[] sd_data;
sd_data = new uint8_t[150];
delay(random(100, 150)); }

```

```

i++; }
//close and remove file
myFile.close();
SD.remove(filename);
if(sd_data){
delete[] sd_data; }
delay(3000);
//if I am not the coordinator
//Send to my father to continue the data's collection
payload[0] = 66;
delete zbTx;
zbTx = new ZBTxRequest(father , payload , sizeof(payload));
sendTheMessage(0x1);
collectCounter = 0;
} //else error opening
else if(collectCounter == childCount && isCoordinatorEnabled) {
//else I am Coordinator and I've finish with the collection
//Restart the algorithm (The first part)
delay(3000);
timeout = millis();
started = false;
collectCounter = 0; } }
else if(changed && msg->getMessageValue() == 68) {
//If i receive a message with data
if(!isCoordinatorEnabled) {
//if I am not the coordinator
//I send the message to my father
delete zbTx;
zbTx = new ZBTxRequest(father , msg->getData() , msg->getDataSize());
sendTheMessage(0x1); }
else if(isCoordinatorEnabled) {
//if i receive a data message and I am the coordinator
//Print the received data into a file

```

```

myFile = SD.open(otherFile , FILE_WRITE);
for(int i = 1; (i<msg->getDataSize()-1) ;i++) {
myFile.print(msg->getData()[i]); }
myFile.println(msg->getData()[msg->getDataSize()-1]);
myFile.close(); } } }
void sendTheMessage(uint8_t BH) {
zbTx->setBroadcastRadius( BH ); //BH = Radius
xbee.send(*zbTx); }
void checkForPacketDelivery() {
xbee.readPacket();
if (xbee.getResponse().isAvailable()) {
// got something
Serial.println(F("Got The Message"));
if (xbee.getResponse().getApiId() == ZB_EXPLICIT_RX_RESPONSE) {
xbee.getResponse().getZBExplicitRxResponse(*explicitResponse);
msg->newMessage(explicitResponse->getRemoteAddress64().getLsb(),
explicitResponse->getRemoteAddress64().getMsb(),
explicitResponse->getFrameData(),
explicitResponse->getDataOffset(),
explicitResponse->getDataLength());
changed = true; //Got a message
uint8_t *messageData = msg->getData();
delete explicitResponse;
explicitResponse = new ZBExplicitRxResponse(); }
} else if (xbee.getResponse().getApiId() == ZB_TX_STATUS_RESPONSE) {
xbee.getResponse().getZBTxStatusResponse(txStatus);
// get the delivery status, the fifth byte
if (txStatus.getDeliveryStatus() == SUCCESS) { //successful transmission
} else { //Unsuccessful
}
} else if (xbee.getResponse().getApiId() == ZB_RX_RESPONSE) {
} else if (xbee.getResponse().isError()) { //Error Reading Packet
}}

```

```

void sendATCommand(String command) {
//This is the Coordinator Command that we need to read upon
//the initialization of the network so that we can tell
//the coordinator to start looking for neighbours.
uint8_t CoEnCmd[] = { 'C', 'E' };
//This Command is nessessary for writing the command
// we previously gave to the XBee module.
uint8_t writeCmd[] = { 'W', 'R' };
//Below we have the request for the AT Command
AtCommandRequest atRequest = AtCommandRequest(CoEnCmd);
//And here we have the response of the AT Command Request.
AtCommandResponse atResponse = AtCommandResponse();
if (command == "CE") {
//Here we tell the request which command to send to the XBee module.
atRequest.setCommand(CoEnCmd);
} else if (command == "WR") {
atRequest.setCommand(writeCmd);
}
xbee.send(atRequest); // send the command
// wait up to 5 seconds for the status response
if (xbee.readPacket(250)) { // got a response!
// should be an AT command response
if (xbee.getResponse().getApiId() == AT_COMMAND_RESPONSE) {
xbee.getResponse().getAtCommandResponse(atResponse);
if (atResponse.isOk()) {
if (atResponse.getValueLength() > 0) {
for (int i = 0; i < atResponse.getValueLength(); i++) {
Serial.print(atResponse.getValue()[i], HEX);
if (atResponse.getValue()[i] == 0x1) {
isCoordinatorEnabled = true;
Serial.println(F(" Coordinator Is Enabled")); }
}
Serial.println(F("")); }
}
}
}
}

```

```

    } else {
        Serial.print(F("Command return error code: "));
        Serial.println(atResponse.getStatus(), HEX); }
    } else {
        Serial.print(F("Expected AT response but got "));
        Serial.print(xbee.getResponse().getApiId(), HEX); }
    } else {
        // at command failed
        if (xbee.getResponse().isError()) {
            Serial.print(F("Error reading packet. Error code: "));
            Serial.println(xbee.getResponse().getErrorCode());
        } else {
            Serial.print(F("No response from radio")); } } }
    AtCommandResponse sendATCMD(String command) {
        uint8_t CoEnCmd[] = { 'N', 'I' }; //Node Identifier
        uint8_t writeCmd[] = { 'D', 'B' }; //RSS
        uint8_t tempCmd[] = { 'T', 'P' }; //Xbee's Temperature
        //Below we have the request for the AT Command
        AtCommandRequest atRequest = AtCommandRequest(CoEnCmd);
        //And here we have the response of the AT Command Request.
        AtCommandResponse atResponse; //= AtCommandResponse();
        if (command == "NI") {
            //Here we tell the request which command to send to the XBee module.
            atRequest.setCommand(CoEnCmd);
        } else if (command == "DB") {
            atRequest.setCommand(writeCmd);
        } else if (command == "TP"){
            atRequest.setCommand(tempCmd); }
        xbee.send(atRequest); // send the command
        if (xbee.readPacket(250)) {
            // got a response!
            if (xbee.getResponse().getApiId() == AT.COMMAND_RESPONSE) {
                xbee.getResponse().getAtCommandResponse(atResponse);
            }
        }
    }
}

```

```
if (atResponse.isOk()) {  
return atResponse; }}}
```

Στην συνέχεια ακολουθεί ένα μέρος της κλάσης MessageList που υλοποιήθηκε, επίσης, στα πλαίσια εκπόνησης της εργασίας. Στον τελικό κώδικα η κλάση χρησιμοποιήθηκε για την προσωρινή αποθήκευση της διεύθυνσης του αποστολέα των μηνυμάτων, αλλά και σε πολλές περιπτώσεις για την απευθείας προώθηση ενός μηνύματος που μόλις έχει ληφθεί από έναν κόμβο του δικτύου (κυρίως κατά την συγκέντρωση των δεδομένων από τους κόμβους στον coordinator).

```
//MessageList.cpp

#include "MessageList.h"
#include "Arduino.h"
MessageList::MessageList(){
    this->addressLow = 0;
    this->addressHigh = 0;
    this->dataSize = 0;
}
MessageList::~~MessageList(){
//The Destructor
    this->deletePacketData();
}
void MessageList::newMessage(uint32_t addL, uint32_t addH,
    uint8_t *frameData, int offset, int dataSize){
//if there is a new message - delete the previous packetData
    this->deletePacketData();
    int j = 0;
//create a new packet data with the received data's size
    this->packetData = new uint8_t[dataSize];
    for (int i = offset; i < offset + dataSize; i++){
//store the recieved data into the packet data
        this->packetData[j] = frameData[i];
        j++;
    }
//Store the addresses High and Low
```

```

this->addressLow = addL;
this->addressHigh = addH;
this->dataSize = dataSize;
}
//delete the data
void MessageList::deletePacketData(){
if (this->dataSize>0){
delete [] this->packetData;
}}
//Return the first value of the recieved Data
uint8_t MessageList::getMessageValue(){
return this->packetData[0];
}
//The fuctions getSenderAddressLow() and getSenderAddressHigh()
//return the High and Low Sender's Address
uint32_t MessageList::getSenderAddressLow(){
return this->addressLow;
}
uint32_t MessageList::getSenderAddressHigh(){
return this->addressHigh;
}
//Return the received Data
uint8_t *MessageList::getData(){
return this->packetData;
}
//Return the size of the received data
int MessageList::getDataSize(){
return this->dataSize;
}

```

Συντμήσεις

API	Application Programming Interface
APS	Application Support Sublayer
LQI	Link Quality Indicator
MAC	Media Access Control
OSI	Open Systems Interconnection
PAN	Permanent Account Number
RSS	Received Strength Signal
SD	Security Digital
USB	Universal Serial Bus
WSN	Wireless Sensor Networks
ZDO	ZigBee Device Object

Γλωσσάρι Ξενικών Όρων

Analog	Αναλογικός
Broadcast	Αναμετάδοση
Coordinator	Συντονιστής
Digital	Ψηφιακός
Hardware	Υλικό
Hop	Άλμα
Network	Δίκτυο
Physical Layer	Φυσικό Επίπεδο
Pin	Απόληξη
Router	Δρομολογητής
Sensor	Αισθητήρας

Signal

Σήμα

Software

Λογισμικό

Tranciever

Πομποδέκτης

Wireless

Ασύρματος