

ΑΤΟΜΙΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΠΕΙΡΑΜΑΤΙΣΜΟΣ ΜΕ ΕΝΣΩΜΑΤΩΜΕΝΑ
ΣΥΣΤΗΜΑΤΑ**

Βασίλης Βασιλείου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2012

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΕΙΡΑΜΑΤΙΣΜΟΣ ΜΕ ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Βασίλης Βασιλείου

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ

Βάσος Βασιλείου

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

Μάιος 2012

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Επιβλέπων καθηγητή Δρ. Βάσο Βασιλείου για την βοήθεια και καθοδήγηση που μου παρείχε σε όλη τη διάρκεια της συνεργασίας μας καθώς και τον υποψήφιο διδάκτορα Ανδρέα Καμηλάρη για τον χρόνο που αφιέρωσε ενημερώνοντας με για το αντικείμενο που είχα να ασχοληθώ καθώς και για το υλικό που μου παρείχε μέσα από το οποίο κατάφερα να εκπληρώσω τον σκοπό της παρούσας εργασίας. Επίσης ιδιαίτερες ευχαριστίες στους 37 φίλους που έδωσαν λίγο από τον χρόνους τους για να είναι παρών στις παρουσιάσεις που διεξήχθησαν για την αξιολόγηση των λειτουργιών τους συστήματος που διεκπεραιώθηκε.

Περίληψη

Κύριος στόχος της εργασίας ήταν ο συσχετισμός ενσωματωμένων συστημάτων, στην προκειμένη περίπτωση Arduino boards, με δίκτυα πρωτοκόλλου ZigBee και 802.15.4 για την δημιουργία συστήματος βασισμένο στην ιδέα του Web of Things. Έχοντας στην κατοχή μας ένα σύνολο από συσκευές Plogg που έχουν την δυνατότητα καταγραφής της κατανάλωσης ενέργειας σε συσκευές και δημιουργίας ZigBee-based wireless mesh networks, λαμβάνουμε μετρήσεις από διάφορες διασκορπισμένες συσκευές και μέσα από ένα δίκτυο 802.15.4 από Arduino ενημερώνουμε μέσω του local router μια συγκεκριμένη ιστοσελίδα.

Με αυτό τον τρόπο μπορούμε να έχουμε μια πλήρως ελεγχόμενη και οικολογική οικία(Smart Home) όπου ένας ιδιοκτήτης θα μπορεί να ενημερώνεται για την κατανάλωση ενέργειας των συσκευών του ανά πάσα ώρα και στιγμή βοηθώντας τον στον περιορισμό άσκοπων ή υπερβολικών σπαταλών σε αυτό τον τομέα. Η κατανάλωση ενέργειας εμπερικλείει 2 ερμηνείες, την ενέργεια σε Watts που απαιτεί μια συσκευή για να βρίσκεται σε λειτουργία καθώς και την ενέργεια που καταναλώθηκε σε ένα δεδομένο χρονικό διάστημα. Ανάλογα με τις απαιτήσεις μπορούμε να ελέγχουμε είτε την μια είτε την άλλη παράμετρο δίνοντας στον ενδιαφερόμενο την δυνατότητα να εξάγει τα ανάλογα συμπεράσματα.

Περιεχόμενα

Κεφάλαιο 1 Εισαγωγή.....	1
1.1 Εισαγωγή	1
1.2 Στόχος Διπλωματικής Εργασίας	1
1.3 Μεθοδολογία	2
1.4 Προηγούμενη Σχετική Ερευνητική Εργασία	3
 Κεφάλαιο 2 Περιγραφή Αρχιτεκτονικών και Πρωτοκόλλων.....	4
2.1 Αρχιτεκτονική Συστήματος	4
2.2 ZigBee δίκτυο	5
2.2.1 Συσκευή Plogg – Telegesis USB stick	6
2.3 802.15.4 δίκτυο	7
2.3.1 Arduino – XBee module – Ethernet shield	9
2.4 Java drivers	11
2.5 Pachube	11
 Κεφάλαιο 3 Υλοποίηση.....	13
3.1 Συνοπτική Ανάλυση Υλοποίησης	13
3.2 Υλοποίηση Hardware	14
3.2.1 Σύνδεση Arduino – XBee module	15
3.2.2 Σύνδεση Arduino – XBee module – Ethernet shield	16
3.3 Υλοποίηση Software	18
3.3.1 Προγραμματισμός Arduino	18
3.3.1.1 Κώδικας Arduino sender	20
3.3.1.2 Κώδικας Arduino receiver	21
3.3.2 Προγραμματισμός XBee module	25
3.3.3 Αναβάθμιση Λειτουργιών Java drivers	28
3.4 Εναλλακτικές Υλοποιήσεις	33

Κεφάλαιο 4 Περιφερειακές λειτουργίες.....	39
4.1 Δομή Λειτουργιών.....	39
4.2 Ανάπτυξης Ιστοσελίδας καταγραφής αποτελεσμάτων.....	40
4.3 Triggers.....	43
4.3.1 SMS alert.....	44
4.3.2 Email alert.....	45
4.3.2.1 Επιθεώρηση HTTP POST.....	46
 Κεφάλαιο 5 Αξιολόγηση.....	48
5.1 Τρόπος λήψης μετρήσεων.....	48
5.2 Αποτελέσματα και Συμπεράσματα.....	49
 Κεφάλαιο 6 Τελικά Συμπεράσματα και Μελλοντική Εργασία.....	51
6.1 Γενικές Παρατηρήσεις.....	51
6.1.1 Γνώσεις και εμπειρίες που αποκτήθηκαν.....	53
6.2 Μελλοντική Εργασία.....	53
 Βιβλιογραφία.....	55
 Παράρτημα Α.....	A1
 Παράρτημα Β.....	B1

Κεφάλαιο 1

Εισαγωγή

1.1	Εισαγωγή	1
1.2	Σκοπός Διπλωματικής Εργασίας	1
1.3	Μεθοδολογία	2
1.4	Προηγούμενη Σχετική Ερευνητική Εργασία	3

1.1 Εισαγωγή

Η παρούσα Ατομική Διπλωματική Εργασία παρουσιάζει 2 πτυχές. Από την μια έχουμε το κομμάτι των 2 δικτύων που δημιουργούμε και όλα όσα έχουν να κάμουν με την wireless επικοινωνία των κόμβων που τα απαρτίζουν και από την άλλη τον προγραμματισμό ενσωματωμένων συστημάτων έτσι ώστε να εκτελούν τις απαιτούμενες λειτουργίες που ζητούμε από αυτά.

Μετά από μελέτη γύρω από ένα τέτοιου είδους σύστημα μπορούμε να κατανοήσουμε τις ιδιαιτερότητες των ZigBee-based δικτύων σε σχέση με άλλα δίκτυα, τον τρόπο λειτουργίας και αξιοποίησης ενσωματωμένων συστημάτων στον τομέα των αυτόνομων δικτύων αισθητήρων και τα διάφορα προβλήματα που προκύπτουν προσπαθώντας να αξιοποιήσουμε το υλικό που έχουμε στα χέρια μας για την κάλυψη των απαιτήσεων που έχουν τεθεί.

1.2 Σκοπός Διπλωματικής Εργασίας

Βλέποντας την αυξητική τάση στην κοστολόγηση του ηλεκτρικού ρεύματος απαιτείται από όλους μια εγκράτεια στην χρήση ηλεκτρικών συσκευών αφού απαιτείται ένα σημαντικό κονδύλι για την κάλυψη των αναγκών του εκάστοτε νοικοκυριού. Αυτή η εγκράτεια μπορεί να προκύψει όταν ο καθένας από εμάς αντιληφθεί την σημαντικότητα του περιορισμού φαινομένων άσκοπης σπατάλης και αυτό μπορεί να γίνει πιο εύκολα μετά από σωστή ενημέρωση για τις κύριες πηγές που τις συντελούν. Αναφερόμενοι σε κύριες πηγές

σπαταλών εννοούμε ξεχασμένες αναμμένες ηλεκτρικές συσκευές, υπερβολική χρήση συσκευών με υψηλές απαιτήσεις σε ηλεκτρικό ρεύμα κτλ.

Δημιουργώντας ένα σύστημα που έχει την δυνατότητα να καταγράφει την κατανάλωση ηλεκτρικού ρεύματος για ένα σύνολο από συσκευές και να παρουσιάζει μια πλήρη εικόνα μέσω web υπό μορφή γραφικών, υποβοηθούμε στην καλλιέργεια κλίματος περιορισμού και ελέγχου των ηλεκτρικών συσκευών που χρειαζόμαστε καθημερινά. Μέσα από έρευνες προκύπτει ότι η γνώση γύρω από την ενέργεια που καταναλώνεται κάνει τους ανθρώπους πιο προσεκτικούς με τα αποτελέσματα που καταγράφτηκαν να μιλούν για 5-15% μείωση στην κατανάλωση ηλεκτρικής ενέργειας που έχει ως άμεσο αντίκτυπο την μείωση και των λογαριασμών ρεύματος κατά ένα σημαντικό ποσοστό.

1.3 Μεθοδολογία

Για την όσο το δυνατό καλύτερη υλοποίηση του συστήματος χρειάστηκε η μελέτη των διαφόρων κομματιών που το απαρτίζουν. Ξεκινώντας την έρευνα έγινε μια προκαταρκτική μελέτη για γνωριμία με την τεχνολογία των ZigBee δικτύων έχοντας ως background την ύλη που καλύφθηκε στο μάθημα Κινητά Δίκτυα. Η τεχνολογία αυτή διέπεται από κάποιες βασικές αρχές και παραμέτρους που πρέπει να τηρούνται καθιστώντας την ως ένα αξιόπιστο μέσο για wireless επικοινωνία μεταξύ 2 κόμβων βασισμένη στο IEEE 802 standard. Μια τέτοια επικοινωνία εγκαθιδρύεται μεταξύ των συσκευών Plogg και ενός USB stick με τα αποτελέσματα να παρουσιάζονται στο αντίστοιχο Manager(Plogg drivers) ή στην περίπτωση την δική μας στα ήδη υλοποιημένα Java drivers.

Μέσα από διάφορα guides μπορεί κάποιος να κατανοήσει τον τρόπο που συνδέονται οι κόμβοι μεταξύ τους και με την χρήση packet analyzers όπως για παράδειγμα Wireshark και PortMon να αναλύσει τα πακέτα και τις εντολές που ανταλλάσσονται για εγκατάσταση και συντονισμό ενός τέτοιου δικτύου.

Όσον αφορά το Embedded κομμάτι και τα Arduino με τις παρεμφερή συσκευές, ξεκινώντας με απλά tutorials δίνεται η δυνατότητα κατανόησης του τρόπου προγραμματισμού και λειτουργίας τους η οποία διαμορφώνεται ανάλογα με τον εξοπλισμό και τον ρόλο που έχουν να διαδραματίσουν στο όλο σύστημα. Για την λήψη αποτελεσμάτων απαιτείται η καταρχήν

αλληλεπίδραση με τα Java drivers τα οποία για τους σκοπούς της εργασίας αφού κατανοήθηκαν τροποποιήθηκαν αναλόγως.

1.4 Προηγούμενη Σχετική Ερευνητική Εργασία

Οι συσκευές Plogg αποτελούν ένα βασικό εργαλείο ανά το παγκόσμιο και χρησιμοποιούνται ευρέως αφού δίνουν την δυνατότητα στο χρήστη να εκμεταλλευτεί τα αποτελέσματα που παράγουν με αρκετά εποικοδομητικό και απλό τρόπο. Τα διάφορα Plogg Projects είναι χτισμένα είτε με την βοήθεια ενσωματωμένων επεξεργαστών είτε συνδυάζουν κώδικα Java υπεύθυνο κατά πρώτη φάση να μπορεί να πράττει τις απαραίτητες λειτουργίες του Plogg Manager και ακολούθως οτιδήποτε άλλο μπορεί να ζητηθεί.

Στο Πανεπιστήμιο Κύπρου βρίσκεται σε εξέλιξη διατριβή για την δημιουργία Smart Home από mesh network με συσκευές Plogg, όπου κάθε συσκευή διαχειρίζεται από ένα thread που εξάγεται από το threadpool ενός server. Η εργασία αυτή προχωρά και ένα βήμα παρακάτω αφού όταν παρατηρηθεί υπερβολική σπατάλη σε κάποια σημεία η αντίστοιχη συσκευή μπορεί να απενεργοποιηθεί wireless για την όσο το δυνατή μεγαλύτερη εξοικονόμηση ενέργειας.

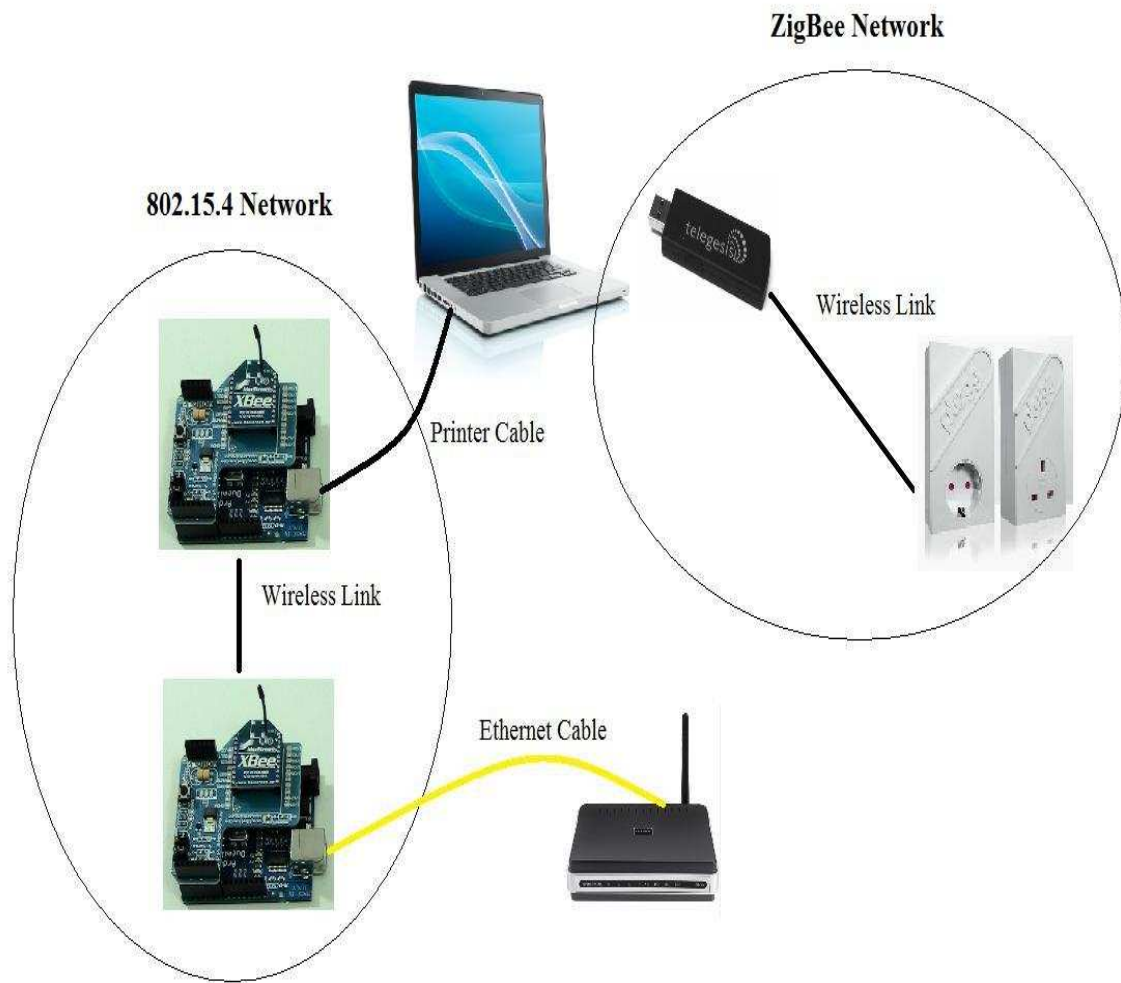
Κεφάλαιο 2

Περιγραφή Αρχιτεκτονικών και Πρωτοκόλλων

2.1	Αρχιτεκτονική Συστήματος	4
2.2	ZigBee δίκτυο	5
2.2.1	Συσκευή Plogg – Telegesis USB stick	6
2.3	802.15.4 δίκτυο	7
2.3.1	Arduino – XBee module – Ethernet shield	9
2.4	Java drivers	11
2.5	Pachube	11

2.1 Αρχιτεκτονική-Δομή Συστήματος

Το Σύστημα αποτελείται από το ZigBee-based δίκτυο που δημιουργείται μεταξύ του Telegesis USB stick και των Plogg καθώς και το 802.15.4 δίκτυο για επικοινωνία 2 Arduino. Ξεκινώντας με την λειτουργία του πρώτου αυτή επιτυγχάνεται με την χρήση υπάρχοντος Java κώδικα ο οποίος θα εξηγηθεί με λεπτομέρεια στην συνέχεια. Αφού γίνουν discover οι διαθέσιμες Plogg συσκευές και επικοινωνήσουν με το Telegesis USB stick στέλλοντας τα αποτελέσματα που συνέλλεξαν, γίνεται προώθηση των αποτελεσμάτων αυτών στο serial port του ενός εκ των 2 Arduino και αυτό με την σειρά του κάνει broadcast στο δεύτερο που είναι συνδεδεμένο με router μέσω καλωδίου Ethernet. Το δεύτερο Arduino είναι υπεύθυνο να συνδέεται με την ιστοσελίδα www.pachube.com και να ενημερώνει με HTTP POST το feed που έχει δημιουργηθεί ειδικά για τον σκοπό αυτό.



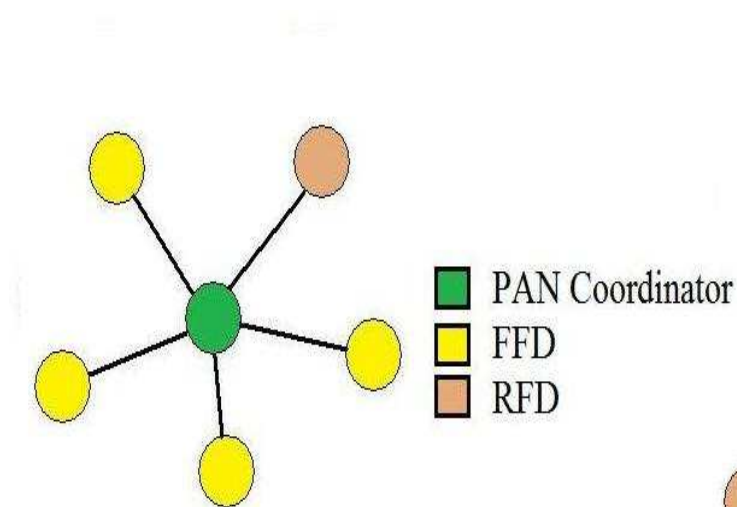
Σχήμα 2.1 Δομή Συστήματος. Διακρίνεται ο διαχωρισμός των 2 δικτύων και η σύνδεση των κόμβων μεταξύ τους.

2.2 ZigBee δίκτυο

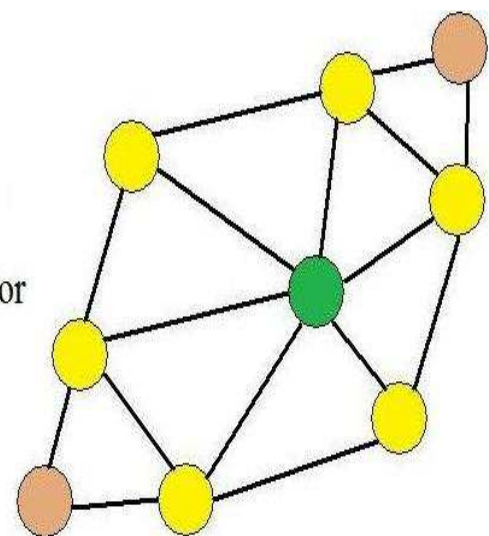
Η τεχνολογία ZigBee μπορεί να θεωρηθεί πανομοιότυπη με το Bluetooth αφού και αυτή προορίζεται για Wireless Personal Area Networks (WPAN). Οι συχνότητες στις οποίες κάνουν operate οι κόμβοι σε ένα ZigBee δίκτυο καθορίζονται στα 868 MHz στην Ευρώπη, στα 915 MHz σε ΗΠΑ και Αυστραλία και στα 2.4 GHz στην ISM ζώνη. Η τεχνολογία αυτή διαθέτει το πλεονέκτημα της πολύ χαμηλής ισχύος που την καθιστά ιδιαίτερα ελκυστική σε συστήματα που απαιτούν wireless επικοινωνία μεταξύ συσκευών-κόμβων αφού εγγυάται μεγαλύτερη διάρκεια ζωής σε σχέση με προηγούμενες από αυτή τεχνολογίες. Η μόνη καινοτομία που χρίζει αναφοράς είναι η παρουσία του ZigBee stack αφού τα πρώτα 2 layers

καθορίζονται από το 802.15.4 IEEE standard με το MAC layer να προσφέρει μέχρι και 16 διαθέσιμα κανάλια(από 11 μέχρι 26) με μέγιστο data rate 250 kbps.

Ο τρόπος δικτύωσης που προκύπτει σε τέτοιου είδους δίκτυα είναι συγκεκριμένος και καθορίζεται κατά την εγκατάσταση και δημιουργία τους. Οι συσκευές χωρίζονται σε Full Function Device(FFD) και Reduced Function Device(RFD), με τις πρώτες να έχουν πλήρεις λειτουργίες και τις άλλες περιορισμένες. Τα δεδομένα που έχουμε σε ένα ZigBee δίκτυο είναι η απαραίτητη ύπαρξη ενός κεντρικού διαχειριστή – Personal Area Network Coordinator, και ο πιθανός ορισμός κόμβων δρομολογητών – Router, και τερματικών συσκευών – End Devices που συνθέτουν είτε ένα star είτε ένα tree network. Ο PAN Coordinator είναι αυτός που ξεκινά την λειτουργία του δικτύου, επιλέγει το Personal Area Network Identifier(PAN ID), επιλέγει τις συσκευές που κάνουν join ή exit το δίκτυο και μπορεί να παίζει τον ρόλο του router. Σε ένα star network ο PAN Coordinator είναι ο κεντρικός κόμβος και επικοινωνεί απευθείας με όλους τους υπόλοιπους ενώ στο tree υποστηρίζεται η δυνατότητα επικοινωνίας FFD κόμβων μεταξύ τους.



Σχήμα 2.2 Star Topology Network



Σχήμα 2.3 Tree Topology Network

2.2.1 Συσκευή Plogg – Telegesis USB stick

Η συσκευή Plogg είναι ένας ηλεκτρικός μετρητής που αποτελεί προϊόν της εταιρείας Energy Optimizers Limited και δουλεύει με την τεχνολογία ZigBee. Οι συγκεκριμένες συσκευές

που έχουμε στην κατοχή μας είναι firmware v.1.70 και το USB stick που τες συνοδεύει ETRX2. Για την χρήση των 2 απαιτείται η ύπαρξη των Telegesis ETRX2 USB drivers καθώς και του Plogg Manager (Plogg drivers) ή εναλλακτικά του κώδικα Java που το αντικαθιστά. Τα drives για το Telegesis USB stick μπορούν να ανεβρεθούν στη επίσημη ιστοσελίδα του Plogg, www.plogg.co.uk, όμως όσον αφορά το Plogg Manager το διαθέσιμο application αφορά το νεότερο firmware και δεν είναι συμβατό με τες συσκευές που έχουμε εμείς. PAN Coordinator του ZigBee δικτύου ορίζεται το USB stick το οποίο δημιουργεί δίκτυο με PAN ID = 31f4(HEX) και επικοινωνεί με τα Plogg, by default, στο κανάλι B(HEX). Τα Plogg παίζουν το ρόλο του End Device αφού δουλειά τους είναι μόνο να συλλέγουν και να αποστέλλουν τα δεδομένα στο PAN Coordinator.



Σχήμα 2.4 Συσκευή Plogg

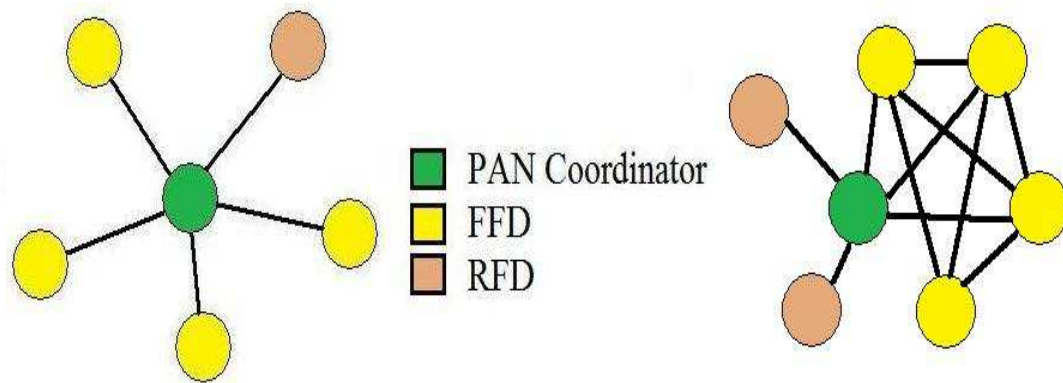


Σχήμα 2.5 Telegesis USB stick

2.3 802.15.4 δίκτυο

Το IEEE 802.15.4 είναι standard το οποίο προσδιορίζει τα πρώτα 2 layers(MAC και Physical) σε Wireless Personal Area Networks(WPAN) που λειτουργούν με transmissions χαμηλού data rate(μέχρι 250 kbps). Όπως αναφέραμε και πριν αποτελεί την βάση της τεχνολογίας ZigBee καθώς και άλλων τεχνολογιών όπως MiWi, WirelessHART, ISA100.11a με το καθένα από αυτά να αναπτύσσει τα πιο πάνω layers με τον δικό του τρόπο.

Επικεντρώνεται στην δημιουργία δικτύων χαμηλής ισχύος με τους κόμβους να βρίσκονται σε αποστάσεις μέχρι και 10 μέτρων ο ένας με τον άλλο και να επικοινωνούν χρησιμοποιώντας τεχνικές Carrier sense multiple access with collision avoidance (CSMA/CA). Υπάρχουν οι FFD και RFD κόμβοι όπως τους περιγράψαμε στο ZigBee που συνθέτουν star και peer-to-peer δίκτυα με την ιδιαιτερότητα ότι ακόμα και στα peer-to-peer, όπου οι κόμβοι δύναται να επικοινωνούν απευθείας μεταξύ τους, απαιτείται η ύπαρξη ενός FFD κόμβου που λειτουργεί ως PAN Coordinator. Αυτός ο κόμβος έχει την αρμοδιότητα κατά την δημιουργία ενός 802.15.4 δικτύου να επιλέγει το κανάλι που θα στέλλονται τα frames και αυτό γίνεται με την τεχνική Energy Detection(ED) scan που προσδιορίζει την αξιοπιστία των διαθέσιμων καναλιών.

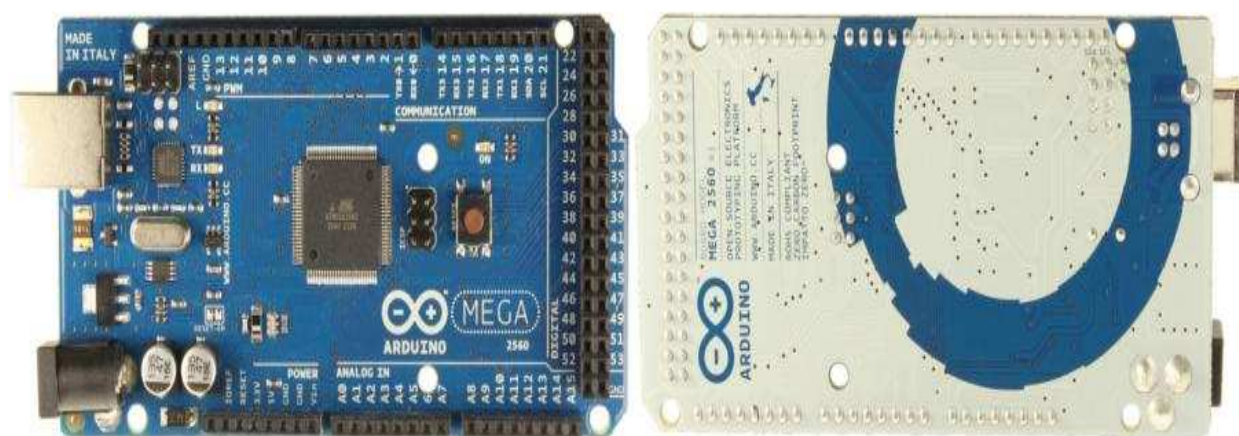


Σχήμα 2.6 Star Topology Network **Σχήμα 2.7** Peer-to-Peer Topology Network

Κάθε συσκευή προσδιορίζεται από ένα 64-bit identifier που κάτω από ορισμένες προϋποθέσεις μπορεί να μειωθεί στα 16 bits και κάνει operate στα 3 διαθέσιμα bands συχνοτήτων που αναφέραμε και για την τεχνολογία ZigBee. Για να μπορέσει μια συσκευή-κόμβος να κάνει join σε ένα υπάρχον δίκτυο κοιτάζει τι υπάρχει διαθέσιμο στην ακτίνα που έχει τη δυνατότητα να κάνει operate, επιλέγει ποιο δίκτυο θέλει και ξεκινά η διαδικασία εισαγωγής του. Το search σε μια συγκεκριμένη περιοχή μπορεί να γίνει με Passive scan, όπου οι FFD περιοδικά στέλλουν beacon frames για να δηλώσουν την ύπαρξη τους με τους “ενδιαφερόμενους” κόμβους να τα λαμβάνουν και να απαντούν ή με Active scan στο οποίο οι κόμβοι προς εισαγωγή στέλλουν beacon requests έτσι ώστε να ενημερωθούν για την τοπολογία που βρίσκεται γύρω τους.

2.3.1 Arduino – XBee module – Ethernet shield

Το Arduino Mega 2560 είναι ένα microcontroller με 54 ψηφιακά input και output pins, σύνδεση USB, βύσμα για ενεργοποίηση από μπαταρία ή καλώδιο ρεύματος, ένα ICSP header για ένωση και άλλων συσκευών, flash memory 256Kb εκ των οποίων τα 8KB χρησιμοποιούνται από το bootloader, SRAM 8KB, ταχύτητα ρολογιού 16 MHz, EEPROM(μη προσωρινή μνήμη) 4KB και ένα reset button. Για ενεργοποίηση του board απαιτούνται 5V με τα αποδεκτά όρια Input Voltage να είναι 6-20V.



Σχήμα 2.8 Arduino Mega 2560, μπροστινή και πίσω όψη

Το XBee module είναι modem type XB24 series 1 και παρέχει την δυνατότητα κάλυψης μέχρι 30 μ. σε κλειστό και 100 μ. σε ανοιχτό χώρο με Transmit Power Output 1 mW, RF Data Rate 250 Kbps, Receiver Sensivity -92dbm, Supply Voltage 2.8 – 3.4 V, Transmit Current 45 mA, Idle/Receive Current 50 mA, Power-down Current 10 uA, Frequency ISM 2.4 GHz και λειτουργεί σε θερμοκρασίες -40 με 85 C. Υποστηρίζει την δυνατότητα 16 Direct Sequence Channels κτίζοντας δίκτυα τύπου Point to Point και Star. Η χρήση του μπορεί να γίνει μέσω τοποθέτησης του σε shield ή με καλωδίωση σε ένα breadboard.



Σχήμα 2.9 XBee module series 1(modem XB24)

Το Ethernet shield επιτρέπει στο Arduino board να ενωθεί μέσω Ethernet cable σε ένα router και να έχει πρόσβαση στο διαδίκτυο χρησιμοποιώντας το Ethernet Library. Η βιβλιοθήκη αυτή επιτρέπει την δημιουργία ακόμα και server στο board που είναι σε θέση να δέχεται πάνω του εισερχόμενες συνδέσεις ή την μετατροπή του board σε client που δημιουργεί εκείνος τις συνδέσεις με μέγιστο αριθμό 4 συνδέσεων ανά χρονική στιγμή.



Σχήμα 2.10 Ethernet shield

Για να δημιουργήσουμε το δίκτυο 802.15.4 έχουμε στην κατοχή μας 2 Arduino Mega 2560, 2 XBee modules στερεωμένα σε shield και 1 Ethernet shield εγκατεστημένο στο 1 από τα 2 Arduino. Για τον προγραμματισμό των Arduino απαιτείται το Arduino Software(διαθέσιμο στην επίσημη σελίδα του Arduino, www.arduino.cc, με την τελευταία έκδοση να είναι το arduino-0023. Για την ρύθμιση των XBee modules υπάρχουν 2 μέθοδοι, ο “manual” τρόπος

που γίνεται μέσω terminal ή ο πιο αυτόματος με την χρήση application της εταιρείας Digi που σου δίνει την δυνατότητα να ρυθμίσεις με πιο high level τρόπο την συσκευή.

2.4 Java drivers

Τα Java drivers αποτελούν ένα κομμάτι της διατριβής του υποψήφιου διδάκτορα Ανδρέα Καμηλάρη και στην ουσία υποκαθιστούν το Plogg Manager. Βασικές λειτουργίες που ακολουθούνται είναι:

- A) Επιλογή από τον χρήστη για εκτύπωση των Watts που απαιτεί μια συσκευή για να τεθεί σε λειτουργία ή των Watts που κατανάλωσε η συσκευή σε ένα χρονικό διάστημα.
- B) Σύνδεση Telegesis USB stick με το COM port από το οποίο θα δέχεται και θα στέλλει τα δεδομένα.
- C) Σύνδεση Arduino με το COM port από το οποίο θα δέχεται και θα στέλλει τα δεδομένα.
- D) Δημιουργία thread που ακούει συνεχώς στα 2 ανοιχτά ports και κάνει ανάλυση των δεδομένων που λαμβάνει καταχωρώντας τα ταυτόχρονα σε αντικείμενο που δημιουργείται ειδικά για τον σκοπό αυτό.
- E) Δημιουργία thread που έχει τις πιο κάτω αρμοδιότητες
 - 1) Εύρεση ενεργοποιημένων συσκευών Plogg ανά προκαθορισμένο χρόνο(delay)
 - 2) Αποστολή αιτήματος μετρήσεων προς όλες τις ενεργοποιημένες συσκευές ανά προκαθορισμένο χρόνο(delay)
 - 3) Τύπωση αποτελεσμάτων στο terminal του Eclipse υπό μορφή .json.
 - 4) Αποστολή αποτελεσμάτων στο serial του Arduino μέσω του COM port που είναι συνδεδεμένο

Ακλουθώντας την διαδικασία αυτή πετυχαίνουμε να μεταφέρουμε στο serial του Arduino τα αποτελέσματα που καταγράφουν τα Java drivers υπό μορφή String το οποίο εύκολα γίνεται broadcast στο δεύτερο Arduino που βρίσκεται συνδεδεμένο στο router.

2.5 Pachube

Η ιστοσελίδα www.pachube.com αποτελεί μια υπηρεσία διαδικτύου που διαχειρίζεται real time δεδομένα ανά το παγκόσμιο. Οι χρήστες-μέλη έχουν την δυνατότητα να μοιράζονται με τον υπόλοιπο κόσμο πληροφορίες που συλλέγουν από πειραματικές εργασίες που αλληλεπιδρούν με το περιβάλλον και γενικά τον κόσμο γύρω μας. Για να μπορέσει κάποιος

να κάνει upload αποτελέσματα απαιτείται η δημιουργία account και ακολούθως η δημιουργία feed που αντιστοιχεί στο αντικείμενο των δεδομένων που θα αποστέλλονται. Δημιουργώντας το account ο χρήστης καλείται να δώσει ως είσοδο ένα τυχαίο αριθμό ο οποίος κάνει generate ένα API key που θα χρησιμοποιείται για verification κατά την διαδικασία της σύνδεσης με την ιστοσελίδα. Επίσης κάθε feed έχει δικό του identity number που χρησιμοποιείται στην φάση του upload των δεδομένων και access permissions(read,write,update) ανάλογα με τους σκοπούς που το θέλει ο χρήστης. Για αλληλεπίδραση με την ιστοσελίδα χρησιμοποιούνται οι HTTP request μέθοδοι GET και PUT για λήψη ή ενημέρωση της σελίδας με νέα δεδομένα. Και στις 2 περιπτώσεις τα δεδομένα αποστέλλονται σε non persistent σύνδεση HTTP 1.0 σε μορφή .json, .xml ή .csv.

Κεφάλαιο 3

Υλοποίηση

3.1	Συνοπτική Ανάλυση Υλοποίησης	13
3.2	Υλοποίηση Hardware	14
3.2.1	Σύνδεση Arduino – XBee module	15
3.2.2	Σύνδεση Arduino – XBee module – Ethernet shield	16
3.3	Υλοποίηση Software	18
3.3.1	Προγραμματισμός Arduino	18
3.3.1.1	Κώδικας Arduino sender	20
3.3.1.2	Κώδικας Arduino receiver	21
3.3.2	Προγραμματισμός XBee module	25
3.3.3	Αναβάθμιση Λειτουργιών Java drivers	28
3.4	Εναλλακτικές Υλοποιήσεις	33

3.1 Συνοπτική Ανάλυση Υλοποίησης

Η υλοποίηση της παρούσας Ατομικής Διπλωματικής Εργασίας αποτελείται από τις διεργασίες που απαιτούνται για να λειτουργεί σωστά το hardware και τον κώδικα που χτίζεται για επίτευξη του ρόλου των συσκευών στο κομμάτι του software. Ξεκινώντας με το hardware ακολουθεί ανάλυση για τον τρόπο συνδεσμολογίας των 2 Arduino με τις υπόλοιπες συσκευές, τις ιδιαιτερότητες που έχουν οι 2 περιπτώσεις και επεξήγηση των τεχνικών χαρακτηριστικών των συσκευών που βοηθούν τα Arduino να προσθέτουν λειτουργίες που πριν δεν ήταν ικανά να πράξουν. Από τούδε και στο εξής το Arduino που λαμβάνει τα δεδομένα στο serial από τα Java drivers και τα κάνει broadcast στο 802.15.4 δίκτυο θα αναφέρεται ως Arduino sender ενώ το δεύτερο που τα λαμβάνει Arduino receiver. Όσον αφορά το software γίνεται μια περιεκτική εξήγηση της δομής του κώδικα που γίνεται upload στα Arduino boards και παρατίθενται κομμάτια κώδικα που κάνουν σαφές πως η θεωρία γίνεται πράξη και τα Arduino αλληλεπιδρούν μεταξύ τους αλλά και με το ZigBee δίκτυο του Telegesis USB stick και των συσκευών Plogg. Για το 802.15.4 δίκτυο και τον τρόπο

δημιουργίας του υπάρχει ειδική ενότητα που έχει να κάνει με τον τρόπο προγραμματισμού και ρύθμισης των XBee modules μέσω AT commands και το πώς αυτές καθορίζουν τις παραμέτρους του δικτύου. Συνδετικός κρίκος του δικτύου 802.15.4 και ZigBee αποτελεί ο κώδικας Java όπου και παρουσιάζονται οι βασικές κλάσεις και μέθοδοι που αντικαθιστούν το Plogg Manager και δίνουν το σωστό input στο Arduino sender. Τέλος παρουσιάζονται κάποιες εναλλακτικές υλοποιήσεις που για διάφορους λόγους δεν προχώρησαν αφού παρουσιάστηκαν ανυπέρβλητες δυσκολίες που τις καθιστούσαν ανέφικτες.

3.2 Υλοποίηση Hardware

Για να μπορέσουμε να προγραμματίσουμε τα Arduino boards απαιτείται η σωστή διασύνδεση με το υπόλοιπο υλικό. Με βάση τους στόχους που έχουν τεθεί η μόνη λειτουργία που κάνει το ένα board είναι να δέχεται στο serial τα αποτελέσματα και να τα κάνει forward με ένα hop στο άλλο. Το δεύτερο board χρειάζεται το Ethernet shield για ένωση με τον router και ένα XBee module που θα το θέτει ως κόμβο του WPAN που δημιουργεί το πρώτο. Η ζωή μας γίνεται πιο εύκολη με την ύπαρξη του shield για τοποθέτηση του XBee module αλλά πιο κάτω θα δούμε και την εναλλακτική μέθοδο με το breadboard. Η υλοποίηση της εργασίας χωρίζεται σε 3 μέρη.

- A) Την λήψη μετρήσεων από μια μεμονωμένη Plogg συσκευή που αλληλεπιδρά με το Telegeisis USB stick και μέσω των Java drivers προωθεί τα αποτελέσματα στα Arduino.
- B) Την λήψη μετρήσεων από διάφορες Plogg συσκευές που βρίσκονται σε μια ακτίνα μεταξύ τους και καταμετρούν την κατανάλωση ρεύματος από διάφορες ενεργοποιημένες ηλεκτρικές συσκευές με την χρήση και πάλι του USB stick.
- Γ) Την απαλλαγή από το Telegeisis USB stick και την λήψη των μετρήσεων απευθείας από τα Arduino.

Με βάση την Μεθοδολογία(Κεφ.1 Ενот.1.3) που ακολουθήθηκε η υλοποίηση του Μέρους Γ' ήταν αδύνατη αφού τα Java drivers απαιτούν την χρήση USB stick που λειτουργεί σαν sink ανάμεσα στις συσκευές. Οι εναλλακτικές υλοποιήσεις που επιχειρήθηκαν και οι λόγοι που δεν προχώρησαν εξηγούνται με λεπτομέρεια στο τέλος του παρών Κεφαλαίου.

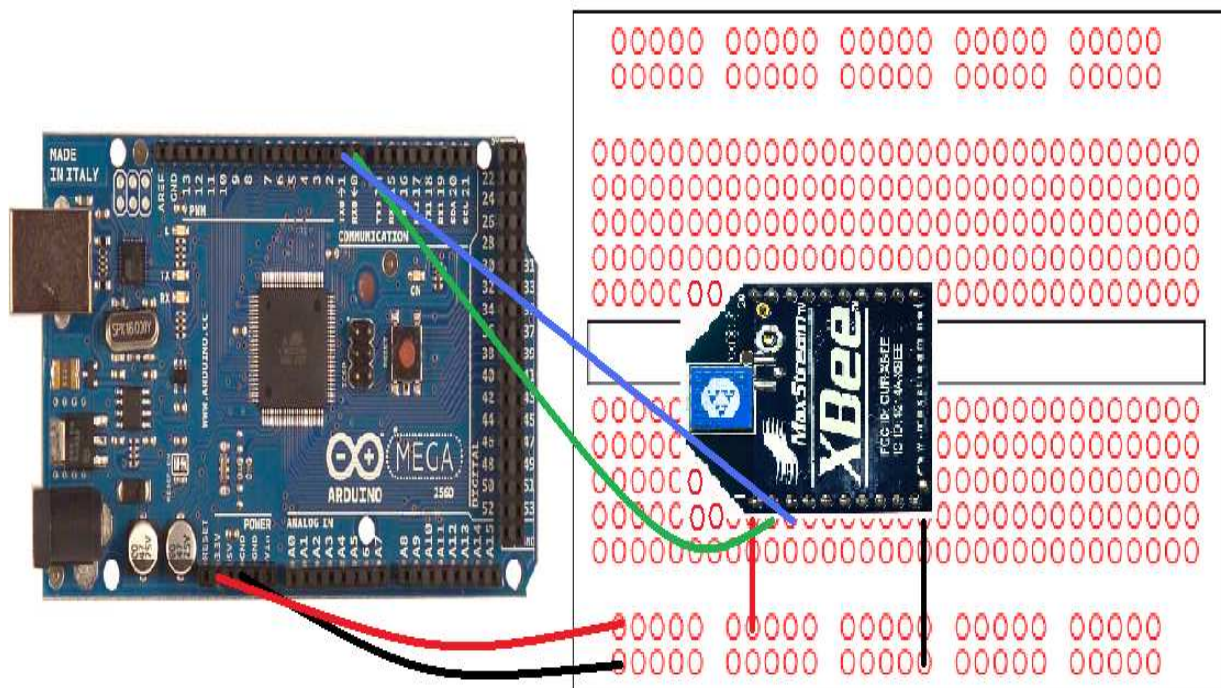
3.2.1 Σύνδεση Arduino – XBee module

Η πρώτη μέθοδος που εμφανίστηκε και θα μπορούσαμε να χρησιμοποιήσουμε αφορούσε την χρήση βοηθητικού breadboard στο οποίο εφαρμόζει το XBee module εφοδιασμένο από κάτω με μια βάση. Αν και κάποιος θα το έβλεπε σαν μια αρκετά περίπλοκη διαδικασία το μόνο που απαιτείται για την σύνδεση είναι η χρήση 4 συρμάτων και το Arduino μπορεί να επικοινωνεί wireless χρησιμοποιώντας το built-in serial communication protocol του. Αρχικά ενώνουμε 2 σύρματα που προσδίδουν τα απαραίτητα προς λειτουργία : power και electrical ground και ακολούθως τα άλλα 2 που κάνουν enable το transmit και receive του module.

Η συνδεσμολογία έχει ως εξής :

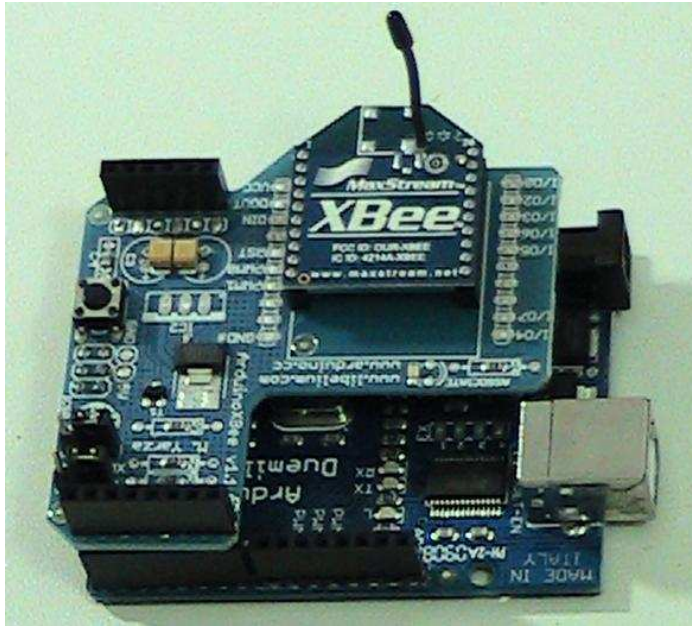
XBee module	Arduino board
VCC ή 3.3 V	3.3 V
TX ή DOUT	RX ή digital pin 0
RX ή DIN	TX ή digital pin 1
GND	GND

Πίνακας 3.1 Συνδεσμολογία Xbee module – Arduino Board



Σχήμα 3.2 Ένωση Arduino board – Xbee module

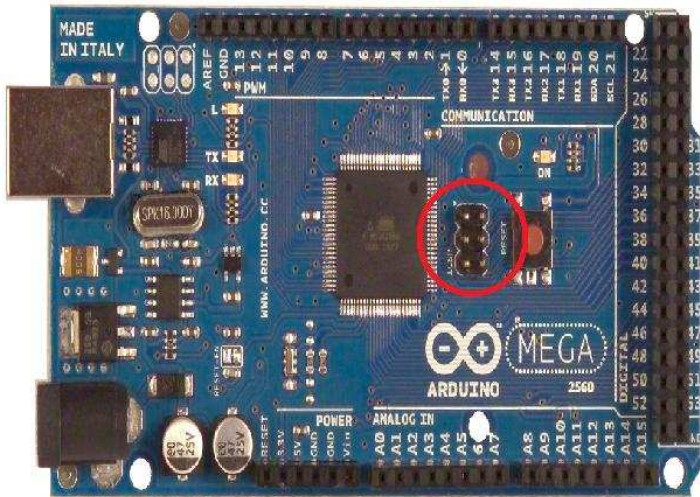
Η δεύτερη μέθοδος που θεωρείται και η πιο απλή απαιτεί την ύπαρξη XBee shield πάνω στο οποίο τοποθετείτε το XBee module και όλο το shield εφαρμόζει στο Arduino board. Το XBee shield καλύπτει τα digital pins RX(pin 0),TX(pin 1) και από το 2-7. Επίσης καλύπτει τα αντίστοιχα Analog pins. Το XBee shield παίρνει Power για να τίθεται σε λειτουργία , το GND signal όπως επίσης το transmit και το receive signal μέσω των pins του ICSP header που βρίσκεται στο κέντρο του board.



Σχήμα 3.3 XBee shield

3.2.2 Σύνδεση Arduino – XBee module – Ethernet shield

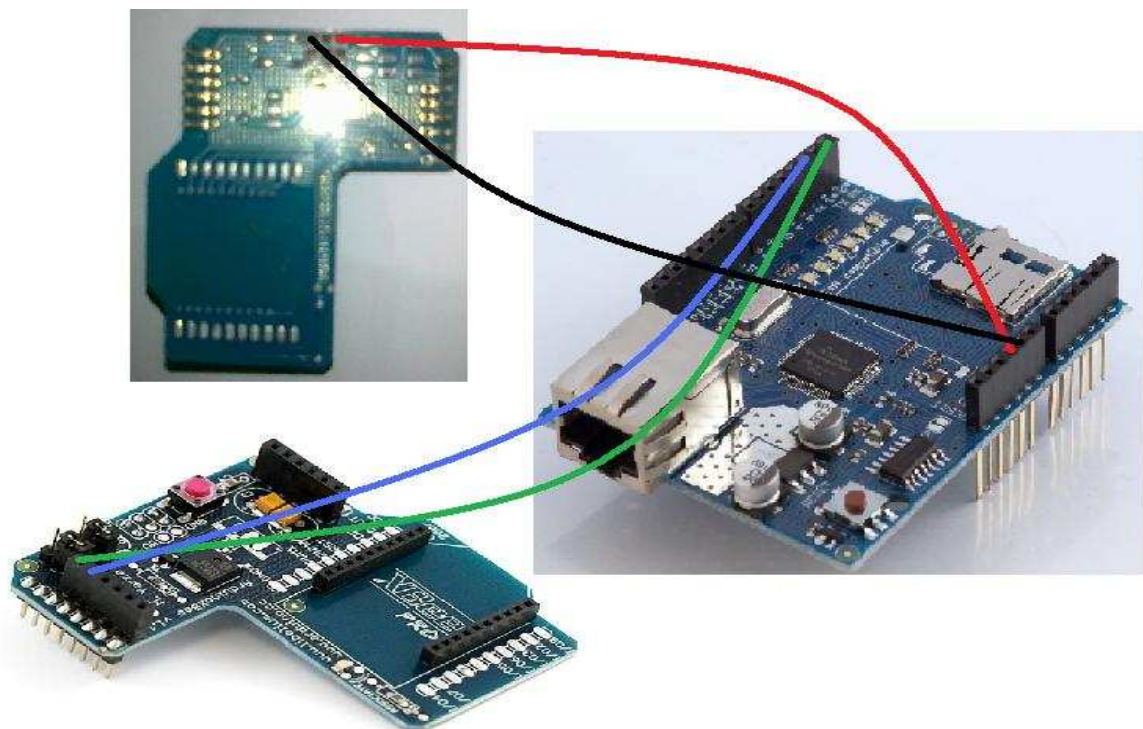
Περνώντας στην συνδεσμολογία του Arduino receiver ενώνουμε το Ethernet shield απευθείας με το board στα digital pins RX(pin 0),TX(pin 1) και από το 2-13 και στα Analog pins από 0 μέχρι 5. Επίσης εφαρμόζουν και τα pins του ICSP header παίρνοντας το Power και τα signals που περιγράψαμε για το XBee shield. Πάνω από το Ethernet shield θα πρέπει να ενωθεί το XBee shield όμως εδώ προκύπτει ένα πρόβλημα. Το XBee shield αναμένει να πάρει Power από το ICSP header του Arduino όμως το Ethernet shield δεν οδηγεί αυτά τα pins στην κορυφή του με αποτέλεσμα το XBee να μην μπορεί να ενεργοποιηθεί κάνοντας απλά την ένωση των shields. Για να λυθεί αυτό το πρόβλημα χρησιμοποιούμε όπως και πριν 4 σύρματα τα οποία συνδέουμε από το Ethernet shield απευθείας στο XBee shield.



Σχήμα 3.5 ICSP header

XBee shield	Ethernet shield
TX ή DOUT	RX ή pin 0
RX ή DIN	TX ή pin 1
ICSP Power pin	5V
ICSP GND pin	GND

Πίνακας 3.4 Συνδεσμολογία XBee shield – Ethernet shield



Σχήμα 3.6 Κάτοψη και πάνω όψη του XBee shield που συνδέεται με το Ethernet shield

Η ηλεκτρική τροφοδότηση των boards μπορεί να γίνει μέσω μπαταρίας που τοποθετείτε σε battery pack και ενώνεται με το βύσμα του Arduino, είτε με καλώδιο ρεύματος ή απλά ενώνοντας το Arduino με USB cable σε ένα υπολογιστή που είναι και ο τρόπος προγραμματισμού του. Με βάση την υλοποίηση που έχει σχεδιαστεί το Arduino sender θα πρέπει να είναι ενωμένο μέσω USB cable μέσω του οποίου ανοίγει το COM port που θα λάβει τις μετρήσεις ενώ το Arduino receiver μπορεί να αφηθεί σε απόσταση από υπολογιστή, να είναι καλωδιωμένο με σύρμα ρεύματος και να δέχεται τα αποτελέσματα wireless από το πρώτο.

3.3 Υλοποίηση Software

Στην ενότητα αυτή γίνεται ανάλυση του τρόπου προγραμματισμού των Arduino boards και XBee modules λαμβάνοντας υπόψη τις ιδιαιτερότητες της κάθε περίπτωσης και παρατίθεται κώδικας των βασικών λειτουργιών που αναμένουμε να υποστηρίζονται από το σύστημα. Επίσης επισημαίνονται οι μετατροπές στα Java drivers και εξηγείται στο που αποσκοπούν.

3.3.1 Προγραμματισμός Arduino

Αφού έχουμε συνδέσει τα Arduino με το υπόλοιπο hardware που χρειαζόμαστε προχωρούμε ένα βήμα παρακάτω με τον προγραμματισμό τους μέσω του arduino_0021 software. Προτού γράψουμε τον κώδικα θα πρέπει να επιλέξουμε το σωστό Arduino board από την λίστα που προσφέρεται στο Tools → Board tab, καθώς και το σωστό serial port στο Tools → Serial Port tab μέσω του οποίου θα γίνεται upload ο κώδικας που θέλουμε να εκτελεί το board. Η βασική δομή ενός Arduino προγράμματος αποτελείται από 2 συναρτήσεις και έχει την εξής δομή:

**** Πληροφορίες προγραμματιστή**

**** Δήλωση βιβλιοθηκών**

**** Δήλωση σταθερών**

**** Δήλωση global μεταβλητών και πινάκων**

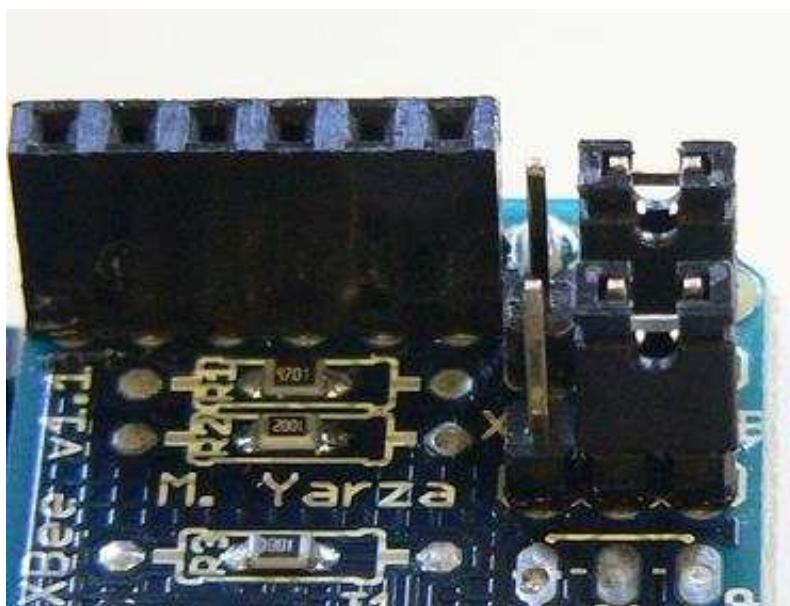
**** Συνάρτηση `setup` (void `setup`())**

**** Συνάρτηση `loop` (void `loop`())**

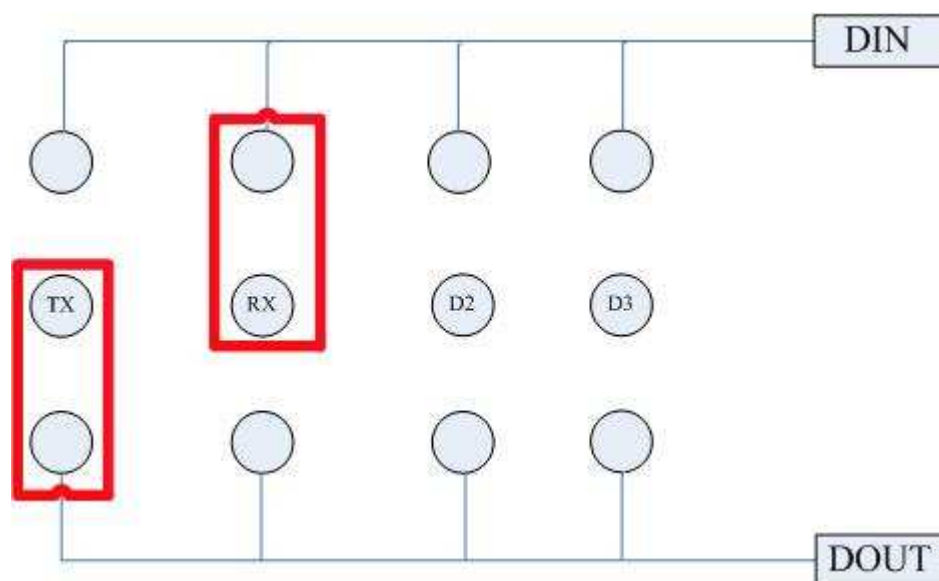
Η συνάρτηση `setup` χρησιμοποιείται για την αρχικοποίηση μεταβλητών, pin modes καθώς και για την έναρξη χρήσης βιβλιοθηκών και τρέχει μόνο μια φορά κατά την ενεργοποίηση ή το reset του Arduino. Η συνάρτηση `loop` όπως λέει και το όνομα τρέχει επ'άπειρον(μέχρι την απενεργοποίηση του board) και εκτελεί οτιδήποτε βρίσκεται στο body της. Πολλές φορές για καθυστέρηση του χρόνου μεταξύ των iterations γίνεται χρήση της συνάρτησης `delay()`.

Για να μπορέσουμε να προγραμματίσουμε ένα Arduino board που είναι εφοδιασμένο με XBee shield θα πρέπει να έχουμε υπόψη κάποιες σημαντικές παραμέτρους. Το XBee shield διαθέτει 3 pins με την ονομασία XBEE/USB στα οποία εφαρμόζουν 2 αφαιρούμενες πλαστικές λαβές(jumpers) που καθορίζουν ποιον από τα 2 mode θα ισχύει. Έχοντας τις λαβές στη θέση USB, δηλαδή στα ακριανά pins, έχουμε το DOUT pin του XBee να συνδέεται με το RX pin του FTDI chip και αντίστοιχα το DIN να τοποθετείτε στο RX του FTDI chip. Αυτό έχει σαν αποτέλεσμα το XBee module να επικοινωνεί απευθείας με το computer με ταυτόχρονη ύπαρξη του FTDI chip(σε παλαιότερες εκδόσεις όπου το FTDI chip δεν ήταν ενσωματωμένο στο chip υπήρχε η ανάγκη αφαίρεσης του αφού μπορούσε να κάνει transmit δεδομένα προς το computer αλλά όχι να ακούσει από αυτό ή από το microcontroller του Arduino). Για τον προγραμματισμό των Arduino μετακινούμε τις λαβές στην θέση USB και κάνουμε κανονικά upload των κώδικα.

Από την άλλη, έχοντας τις λαβές στα 2 εσωτερικά pins στην θέση XBEE το DOUT pin του XBee module αντιστοιχεί στο RX του microcontroller και το DIN στο TX του. Με αυτό τον τρόπο είναι ενωμένα και στα αντίστοιχα TX και RX pins του FTDI chip και είναι σε θέση κάνοντας transmit δεδομένα από το microcontroller μέσω USB να φτάνουν στο computer. Από την άλλη όμως, για την λήψη δεδομένων η μόνη δίοδος που υπάρχει είναι μέσω του XBee module του(δηλ. wireless).



Σχήμα 3.7 XBee jumpers, USB position



Σχήμα 3.8 XBee shield pins

3.3.1.1 Προγραμματισμός Arduino sender

Προχωρώντας στην ανάλυση του κώδικα που αναθέσαμε στα 2 board ξεκινάμε από το Arduino sender που κάνει την λειτουργία της προώθησης των αποτελεσμάτων. Οι λαβές στο XBee shield βρίσκονται στην θέση USB αφού αναμένουμε input από τα Java drivers μέσω του USB cable.

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

Στην συνάρτηση **setup()** η εντολή `Serial.begin(9600)` ανοίγει το serial port και θέτει στο baud rate, δηλαδή τα σύμβολα που στέλλονται ανά δευτερόλεπτο, την τιμή 9600. Το σύνολο των baud rate που είναι δυνατό να ανατεθούν είναι {300,1200,2400,4800,9600,14400,19200,28800,38400,57600,115200}.

```
void loop()
{
  if (Serial.available() > 0)
  {
    incomingByte = Serial.read();
    Serial.print(incomingByte);
  }
}
```

Ακολούθως στην συνάρτηση **loop()** με την χρήση της `Serial.available()` γίνεται έλεγχος κατά πόσον έχουν ληφθεί δεδομένα(bytes) στο serial port. Αυτά τα δεδομένα μέχρις ότου διαβαστούν βρίσκονται αποθηκευμένα στο serial receive buffer που έχει την δυνατότητα να κρατά μέχρι 128 bytes. Αν ο έλεγχος του βρόγχου είναι true, δηλαδή έχουν αποσταλεί μετρήσεις, τις διαβάζουμε byte-byte δηλαδή χαρακτήρα-χαρακτήρα με την `Serial.read()` και τις κάνουμε forward στο δίκτυο και πάλι byte ανά byte με την συνάρτηση `Serial.print(**1 character**)`.

3.3.1.2 Προγραμματισμός Arduino receiver

Το δεύτερο Arduino board που έχουμε στην κατοχή μας εκτελεί την ίδια λειτουργία με το πρώτο αφού περιμένει είσοδο στο serial port την οποία όμως αυτή την φορά την παίρνει wireless και την προωθεί μέσω Ethernet cable στο router και από εκεί στην ιστοσελίδα www.pachube.com.

```

void setup()
{
  Serial.begin(9600);
  Ethernet.begin(mac, ip, gateway, subnet);
}

```

Ξεκινώντας με την συνάρτηση **setup()** ανοίγουμε το serial port θέτοντας το baud rate στο 9600 σύμβολα ανά δευτερόλεπτο και αρχικοποιούμε τα settings της συσκευής Ethernet με 4 σταθερές που αντιστοιχούν στο Mac και IP address του Arduino, στο IP address του default gateway(δηλ. του local router που έχουμε συνδεδεμένο το Arduino receiver) καθώς και το IP address του subnet mask που βρισκόμαστε. Για τις 2 τελευταίες διευθύνσεις ανοίγουμε ένα Command Prompt παράθυρο και με την εντολή “ipconfig” έχουμε τις πιο κάτω πληροφορίες.

```

C:\Users\user>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 
Wireless LAN adapter Wireless Network Connection:

    Connection-specific DNS Suffix  . : lan
    Link-local IPv6 Address . . . . . : fe80::93b6:301a:8149:24e9%4
    IPv4 Address. . . . . : 192.168.1.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

Ethernet adapter Hamachi:

    Connection-specific DNS Suffix  . : 
    IPv6 Address. . . . . : fe80::93b6:301a:8149:24e9%4
    Link-local IPv6 Address . . . . . : fe80::93b6:301a:8149:24e9%4
    IPv4 Address. . . . . : 192.168.1.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

Tunnel adapter Local Area Connection* 10:

    Connection-specific DNS Suffix  . : 
    IPv6 Address. . . . . : fe80::93b6:301a:8149:24e9%4
    Link-local IPv6 Address . . . . . : fe80::93b6:301a:8149:24e9%4
    Default Gateway . . . . . : 

Tunnel adapter Local Area Connection* 11:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 
Tunnel adapter Local Area Connection* 24:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 
Tunnel adapter Local Area Connection* 16:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 
Tunnel adapter Local Area Connection* 25:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : lan
Tunnel adapter Local Area Connection* 26:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 
C:\Users\user>_

```

Σχήμα 3.9 Παράθυρο Command Prompt

Για IP address του Arduino αναθέτουμε μια διεύθυνση που ανήκει στο subnet που βρισκόμαστε και είναι ελεύθερη προς χρήση. Για να το ελέγξουμε αυτό γράφουμε την εντολή “ping ip_address_to_check” και αν δεν λάβουμε κάποια απάντηση τότε δεν υπάρχει κάποια συσκευή που να την έχει δεσμεύσει άρα η διεύθυνση είναι έγκυρη προς χρήση. Για το Mac address του Arduino ορίζουμε εμείς μια τυχαία διεύθυνση της μορφής 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED που δεν αντιστοιχεί σε κάποια άλλη συσκευή του subnet και είμαστε έτοιμοι για τα επόμενα βήματα στην συνάρτηση **loop()**. Πιο κάτω παρουσιάζονται σε συντομία οι κυριότερες λειτουργίες που εκτελεί το board.

```
if (incomingByte == '!')
{
    if (!client.connected())
    {
        client.connect();
    }
    if (client.connected())
    {
        sending();
        Serial.println(pachubeString);
        pachubeString="";
    }
    if (!client.connected())
    {
        client.stop();
    }
}
else
{
    pachubeString+=incomingByte;
}
```

Αυτή η συνάρτηση ελέγχει επ’άπειρον αν έχει ληφθεί κάποιο byte και αν αυτό έχει γίνει περνά από ένα έλεγχο. Αν ο χαρακτήρας αντιστοιχεί σε “!” είναι ένδειξη ότι η είσοδος έχει ολοκληρωθεί αλλιώς συνεχίζεται το χτίσιμο του string που θα αποσταλεί σαν μέτρηση στην

ιστοσελίδα. Ο χαρακτήρας “!” μπαίνει στο τέλος κάθε μέτρησης στα Java drivers και χρησιμοποιείται ως τερματικός χαρακτήρας των string υποκαθιστώντας κατά κάποιο τρόπο την τιμή null. Αν ληφθεί αυτός ο χαρακτήρας δηλ. έχουν καταφθάσει όλοι οι χαρακτήρες μιας μέτρησης, ο χρήστης δοκιμάζει να συνδεθεί με τον server της ιστοσελίδας. Να σημειώσουμε ότι ο χρήστης που λειτουργεί ως client έχει ανοίξει ήδη port και περιμένει από εκεί να λάβει τα δεδομένα από τον server. Η εντολή που χρησιμοποιείται για τον σκοπό αυτό γίνεται κατά την διάρκεια της αρχικοποίησης πριν την συνάρτηση **setup()**: Client client(server_ip, 80) όπου δηλώνεται το IP του web server που θα ενωθούμε και το port που προτίθεται να ανοίξει ο client. Προτού γίνει η σύνδεση (client.**connect()**) ελέγχεται ότι στην παρούσα κατάσταση αυτή είναι κλειστή με την συνάρτηση client.**connected()**.

Συνάρτηση **sending()**

```
client.print(PACHUBE_FEED_ID);  
client.println(".csv HTTP/1.0");  
client.println(PACHUBE_API_KEY);  
client.println(pachubeString);
```

Ακολούθως αφού ξέρουμε ότι ο χρήστης είναι συνδεδεμένος στέλλουμε τα δεδομένα με HTTP POST με την **sending()** και καθαρίζουμε το string για να είναι σε θέση να δεχτεί τα επόμενα αποτελέσματα. Η συνάρτηση αυτή μέσω non persistent σύνδεσης HTTP 1.0 και μορφής .csv κάνει PUT το String που δημιουργήσαμε έχοντας προηγουμένως κάνει verify στην ιστοσελίδα το σωστό feed id και API Key του feed.

Για τύπωση των μηνυμάτων που λαμβάνονται από τον server στο serial του board χρησιμοποιείται η συνάρτηση client.**available()** που είναι η αντίστοιχη της serial.available() και ελέγχει αν ο server που είναι ήδη συνδεδεμένος ο client έχει στείλει δεδομένα. Αν αυτό έχει γίνει, μέχρις ότου καθαρίσει το buffer του client τυπώνονται στο serial monitor byte-byte δηλ. χαρακτήρας-χαρακτήρας τα μηνύματα που έχουν μαζευτεί. Το serial monitor αποτελεί μια οθόνη που παρουσιάζει όλα τα μηνύματα που στέλλονται από το Arduino board έτσι τα τυπώνουμε εκεί για να δούμε τι έχει ληφθεί μέχρις εκείνο το σημείο.

```

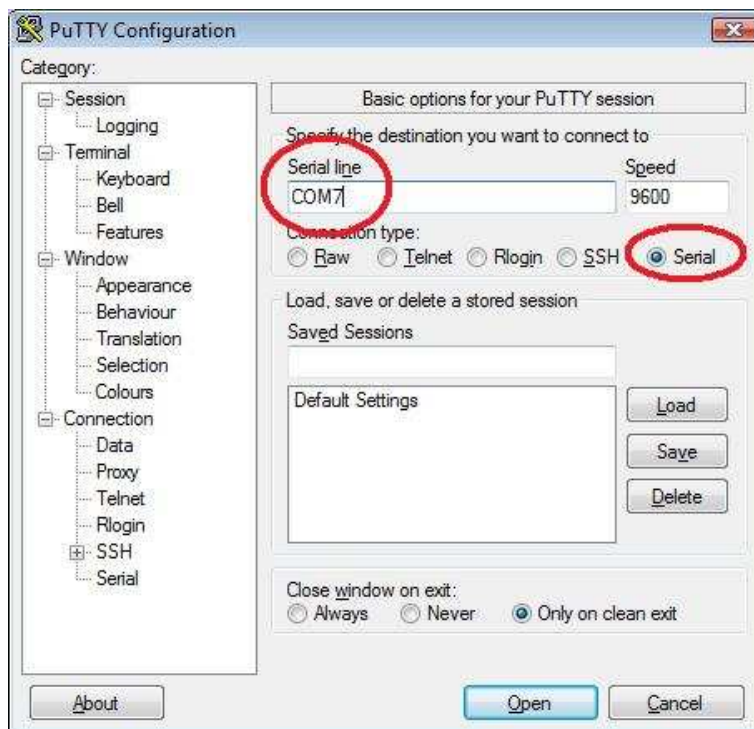
if (client.available())
{
    while (client.available())
    {
        char c = client.read();
        Serial.print(c);
    }
}

```

Τέλος ελέγχουμε αν η σύνδεση έχει κλείσει έτσι ώστε ο χρήστης να αποσυνδεθεί από τον server(`client.stop()`) και η διαδικασία να ξεκινήσει ξανά από την αρχή.

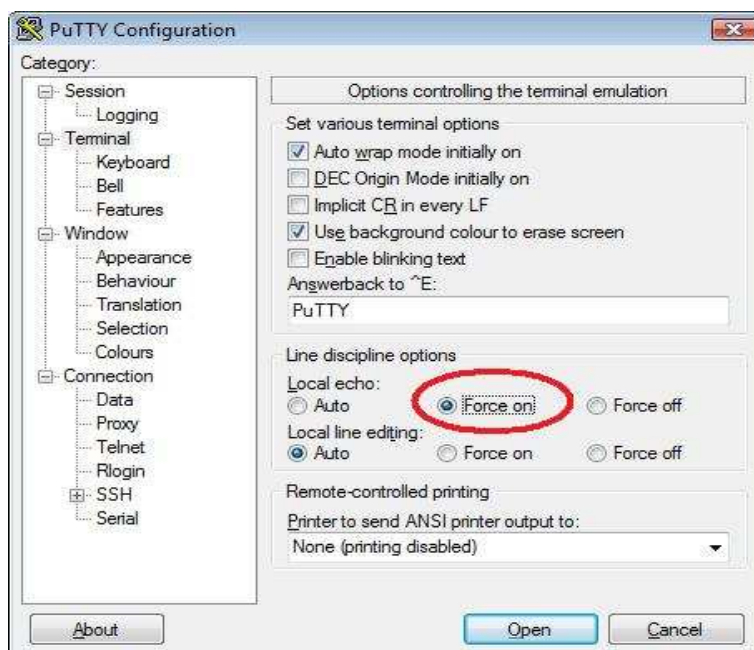
3.3.2 Προγραμματισμός XBee module

Τα XBee modules έρχονται εργοστασιακά με κάποιες συγκεκριμένες ρυθμίσεις που για τις ανάγκες μιας εργασίας που τα θέλει να δημιουργούν ένα δίκτυο και να επικοινωνούν wireless μεταξύ τους χρειάζεται να τροποποιηθούν. Οι ρυθμίσεις αυτές αφορούν το PAN ID το δικτύου στο οποίο θα μπορούν να εισαχθούν και άλλοι κόμβοι στην συνέχεια, το κανάλι(PAN CH) που ανταλλάσσονται τα δεδομένα, το εύρος των διευθύνσεων που στέλλονται τα δεδομένα στο κανάλι που ανατέθηκε και οι ρόλοι των κόμβων αν το δίκτυο το απαιτεί. Στην δική μας περίπτωση τα 2 Arduino θεωρούνται ισότιμοι κόμβοι ενός 802.15.4 WPAN άρα δεν υπάρχει η δυνατότητα ανάθεσης διαφορετικών αρμοδιοτήτων σε αυτά. Αυτό θα ίσχυε στην περίπτωση ενός ZigBee-based δικτύου που προϋποθέτει την ύπαρξη ενός PAN Coordinator και δίνει την δυνατότητα επιλογής στην ιδιότητα των υπόλοιπων κόμβων ανάμεσα σε Router και End Device. Μια τέτοια ανάθεση ρόλων είναι εύκολη να γίνει με το εργαλείο X-CTU της εταιρείας Digi ή εναλλακτικά υπάρχει η λύση του terminal που σου επιτρέπει απευθείας επικοινωνία με το module με λήψη και ανάθεση δεδομένων βάση εντολών. Και στις 2 περιπτώσεις το XBee module για να μπορεί να ανταποκρίνεται θα πρέπει να έχει τις 2 λαβές σε USB θέση αφού οι εντολές ανταλλάσσονται μέσω USB cable. Το terminal application που χρησιμοποιήθηκε είναι το PuTTY και η μέθοδος του configuration των XBee modules γίνεται αφού ενωθούμε με το serial port του Arduino.



Σχήμα 3.10 Εργαλείο PuTTY, Session Category

Για να είμαστε σίγουροι ότι οι εντολές που αποστέλλουμε γίνονται δεκτές από το module θα πρέπει να ρυθμίσουμε το PuTTY έτσι ώστε να τυπώνει τα responses στην οθόνη. Αυτό γίνεται μέσα από το Terminal Category και την επιλογή “Force on” στο option “Local echo”.



Σχήμα 3.11 Εργαλείο PuTTY, Terminal Category

Ακολουθώντας πατώντας “Open” έχουμε μπροστά μας το terminal και ξεκινούμε την αποστολή εντολών έχοντας υπόψη ότι οι αριθμοί που στέλλονται και λαμβάνονται είναι σε Hexadecimal μορφή και όχι σε Decimal.

Αποστολή : +++ ,για να μπούμε σε φάση Command mode

Λήψη : OK

Από εδώ και πέρα αν το terminal μείνει αδρανές για 3-4 δευτερόλεπτα βγαίνουμε από το Command mode και απαιτείται να ξεκινήσει η διαδικασία από την αρχή.

Αποστολή : ATID ,για να δούμε σε πιο δίκτυο κάνει operate το module

Λήψη : π.χ 1234

Αποστολή : ATID3332 ,θέτουμε ως PAN ID το 3332

Λήψη : OK

Αποστολή : ATMY ,ζητάμε την διεύθυνση του module

Λήψη : π.χ 1311

Αποστολή : ATDL ,παίρνουμε τον αριθμό του Destination Address Low που αντιστοιχεί στα πρώτα 16 bits της διεύθυνσης του κόμβου παραλήπτη από το υφιστάμενο module

Εδώ να πούμε ότι υπάρχει και η αντίστοιχη εντολή ATDH που αναφέρεται στα 16 high bits αλλά δύναται να μην χρησιμοποιηθεί αν οι διευθύνσεις των κόμβων δεν απαιτούν περισσότερα από 16 bits για να αναπαρασταθούν.

Αποστολή : ATDLFFFF ,τα δεδομένα γίνονται broadcast σε όλους τους κόμβους που βρίσκονται στο δίκτυο. Αν θέλαμε τα δεδομένα να αποστέλλονται σε ένα μόνο κόμβο θα έπρεπε η εντολή να περιλαμβάνει την διεύθυνση του κόμβου(την οποία παίρνουμε όταν επικοινωνήσουμε με το άλλο module και πάρουμε την διεύθυνση του από την εντολή ATMY)

Λήψη : OK

Αποστολή : ATCH ,για να λάβουμε το κανάλι που κάνει operate το module

Λήψη : 0 ,by default το κανάλι είναι το 0

Αποστολή : ATCH15 ,θέτουμε τυχαία ως κανάλι επικοινωνίας το κανάλι 15(HEX) που αντιπροσωπεύει το 21(DEC). Αν είχαμε ένα ZigBee δίκτυο θα μπορούσαμε να ορίσουμε κανάλια μόνο από το 11 μέχρι το 26

Λήψη : OK

Αποστολή : ATWR

,τα settings που θέσαμε μέχρι τώρα γράφονται στην non-volatile μνήμη του module έτσι ώστε όταν αυτό σβήσει να μην χαθούν. Η εντολή αυτή όπως και όλες οι εντολές μπορούν να χρησιμοποιηθούν σε οποιαδήποτε φάση της διαδικασίας του configuration.

Λήψη : OK

Αποστολή : ATRE

,για επαναφορά των factory settings

Λήψη : OK

Στις περιπτώσεις όπου μια εντολή δεν γράφεται στο σωστό format ή δεν είναι δυνατό να εκτελεστεί από το module λαμβάνεται το μήνυμα “ERROR”. Αφού έχουμε τελειώσει με το ένα XBee module προχωρούμε στο configuration του δεύτερου θέτοντας τις τιμές του PAN ID, PAN Channel και Destination Address Low παρόμοια με το πρώτο.

3.3.3 Αναβάθμιση Λειτουργιών Java drivers

Έχοντας λάβει τα αρχεία Java που είχαν ως βασική λειτουργία την αντικατάσταση του Plogg Manager και την δημιουργία ενός multithread συστήματος για εξυπηρέτηση πολλαπλών συσκευών Plogg υπήρχε η ανάγκη μετατροπής που θα τους επέτρεπε την ένταξη των λειτουργιών για τους σκοπούς της παρούσας Ατομικής Διπλωματικής Εργασίας. Ξεκινώντας με τις υπάρχουσες λειτουργίες ακολουθεί ανάλυση για τον τρόπο που διεκπεραιώθηκαν μέσω των κλάσεων και μεθόδων καθώς και οι υποβοηθητικές δομές που υλοποιήθηκαν. Στο Project που τρέχουμε υπάρχουν 5 διαφορετικά interfaces από τα οποία μελετήθηκαν και μετατράπηκαν μόνο τα 2. Αυτά είναι το interface **controlLayer** που περιλαμβάνει την βασική κλάση **Core()** με την **main()** και το **deviceLayer.smartMeters** με τις κλάσεις **PloggDevices()**, **ParseBuffer()**, **PrepString()**, **MediaCalc()** και **PloggMonitor()**.

Από τις 5 αυτές κλάσεις τροποποιήθηκε μόνο η **PloggMonitor()** αφού είναι αυτή που περιέχει, εκτός των άλλων, τον κώδικα του thread που διαχειρίζεται τα devices και χρησιμοποιεί τις υπόλοιπες για να δημιουργεί τα κατάλληλα αντικείμενα στις λειτουργίες που επιχειρεί. Ξεκινώντας όμως από την κλάση **Core()** δίνεται σαν παράμετρος στην **main()** η πρόταση “PloggHome Ucy localhost 8080 SmartMeter COM8” με το μόνο που μας ενδιαφέρει να είναι το **argv[5]** δηλ. το COM8 που αντιστοιχεί στο COM port που θα

επιχειρηθεί στην συνέχεια να δεσμευτεί για το Telegesis USB stick. Ο κώδικας στην **main()** έχει ως ακολούθως:

```
String comPort = args[5];  
PloggMonitor ploggMonitor = new PloggMonitor(Core.getInstance().getDevices(),  
comPort);  
String choice=menu();  
ploggMonitor.startDevice(choice);  
Core.getInstance().submitToThreadPool(ploggMonitor);
```

Μετά την ανάθεση στο comPort δημιουργείται ένα αντικείμενο ploggMonitor που παίρνει ως παράμετρο στο constructor ένα αντικείμενο κλάσης **Devices()** που το θέτει ως δική του υπερκλάση και το comPort. Η κλάση **Devices()** ανήκει στο interface **deviceLayer** και περιλαμβάνει δομή λίστας που αποθηκεύει τα ids των συσκευών που βρέθηκαν στην περιοχή μαζί με αντικείμενο που κρατά όλες τις τεχνικές πληροφορίες που τις διέπουν και έχουν να κάνουν με την κατάσταση στην οποία βρίσκονται.

Ακολούθως παρουσιάζεται menu που δίνει στο χρήστη την δυνατότητα να επιλέξει είτε την τρέχον κατανάλωση ενέργειας των συσκευών που πρόκειται να ανεβρεθούν είτε την κατανάλωση που καταγράφεται στα χρονικά σημεία που θα παίρνονται οι μετρήσεις και με βάση αυτή την επιλογή ενημερώνεται ανάλογα και το feed της σελίδας www.pachube.com. Στην συνέχεια καλείται η μέθοδος **startDevice()** του αντικειμένου ploggMonitor που καθορίζει την πρόσβαση στο COM port ανάλογα με το λειτουργικό που τρέχει και ενεργοποιεί τόσο το COM port του Telegesis USB stick όσο και αυτό του Arduino. Παράλληλα δημιουργείται thread υπεύθυνο λήψης, ανάλυσης και αποθήκευσης των δεδομένων που καταφθάνουν στα 2 ports το οποίο τρέχει επ' άπειρον.

Τέλος καλούμε το αντικείμενο της κλάσης **main()** και τρέχοντας την μέθοδο **submitToThreadPool()** παίρνουμε ένα thread από το thread pool που αναλαμβάνει τις εναπομείναντες λειτουργίες.

Ο κώδικας του νέου thread βρίσκεται στην κλάση **PloggMonitor()** στην μέθοδο **run()** και έχει ως εξής: Αρχικά το thread με την μέθοδο **discoverDevices()** ανακαλύπτει τις Plogg συσκευές που βρίσκονται στον περιβάλλοντα χώρο. Αυτό γίνεται με την αποστολή

συγκεκριμένης εντολής από το COM port του Telegesis USB stick, ανάλυση των αποτελεσμάτων που λαμβάνονται με την ανταπόκριση των ενεργοποιημένων συσκευών και έλεγχο ότι μια συσκευή δεν βρίσκεται καταχωρημένη ήδη στην δομή μιας λίστας που έχουμε δημιουργήσει.

Ξεκινώντας όμως από την πρώτη λειτουργία που αφορά τις εντολές που αναμένουν τα Ploggs για να ανταποκριθούν θα δούμε ένα σενάριο επικοινωνίας Telegesis USB stick με συσκευή Plogg. Στο πιο κάτω σενάριο το Telegesis USB stick αφού ανακαλύψει την ύπαρξη μιας συσκευής Plogg ζητά κάποιες λεπτομέρειες που αφορούν το version της συσκευής, το PAN ID στο οποίο ανήκει, το ρόλο της συσκευής στο PAN κτλ.

USB stick : AT+SN

Plogg Device : OK

Plogg Device : FFD:000D6F0000224328

USB stick : FFD:000D6F0000262B19

USB stick : AT+UCAST:000d6f0000224328=v

Plogg Device : +UCAST:00

Plogg Device : OK

Plogg Device : ACK:00

Plogg Device : UCAST:000D6F0000224328==SW Version 1.70~Unit friendly name = Scaldabagno~PCB Build R

**Plogg Device : UCAST:000D6F0000224328=esistors DIO9=1 , DIO10=0 , DIO11=0
~Unit is Zigbee Enabled~**

Plogg Device : UCAST:000D6F0000224328=- Unit is a configured as a router~- Current Pan = 31F4 unit

Plogg Device : UCAST:000D6F0000224328= is router~Zigbee echo is on!!!~TickCount = 9741087 ~~~~~>

Στα πλαίσια της υφιστάμενης εργασίας έγινε χρήση μόνο των πιο κάτω εντολών.

Εντολή	Αποτέλεσμα
AT+SN	Αναζήτηση συσκευών
AT+UCAST:<Plogg_id>=sv	Αίτημα λήψης μετρήσεων κατανάλωσης ρεύματος

Πίνακας 3.12 Εντολές συσκευών Plogg και αποτέλεσμα που προκύπτει από αυτές

Όταν στέλλεται μια εντολή γίνεται broadcast σε ολόκληρο το ZigBee δίκτυο άρα φτάνει σε όλες τις συσκευές και αυτές με την σειρά τους ανταποκρίνονται. Τα αποτελέσματα που λαμβάνονται όπως φαίνεται και πιο πάνω απαιτούν ανάλυση αφού περιλαμβάνουν πληροφορίες που δεν χρειαζόμαστε. Υπεύθυνη για τη μετατροπή των μετρήσεων είναι η κλάση **ParseBuffer()** που εκτελεί τις λειτουργίες τις μέσα από ένα αντικείμενο που δημιουργείται στην μέθοδο που ακούει συνεχώς στα ports και δέχεται τα ληφθέντα δεδομένα. Η μέθοδος αυτή είναι η **SerialReader()** και βρίσκεται και αυτή στην κλάση **PloggMonitor()**.

Μετά την ανάλυση γίνεται έλεγχος ότι οι συσκευές που ανεβρέθηκαν δεν βρίσκονται ήδη στη λίστα των Devices που αναφερθήκαμε προηγουμένως για να μην υπάρχουν διπλότυπα. Ακολούθως αναθέτουμε στο αντικείμενο outputstream, που είναι υπεύθυνο για την ακολουθία χαρακτήρων που εξάγεται από κάποιο port, ως port εξόδου το COM port που είναι συνδεδεμένο το Arduino και μπαίνουμε σε ένα ατέρμονο βρόγχο όπου το thread θα αλληλεπιδρά με τις συσκευές Plogg βάση προκαθορισμένων delays.

```
outputStream = serialPort1.getOutputStream();  
while (true)  
{  
    ** Ανακάλυψε νέες συσκευές.  
    ** Ετοίμασε Electricity Request.  
    ** Στείλε το αίτημα σε όλες τις συσκευές που έχουν ανεβρεθεί.  
    ** Ετοίμασε πίνακα από Strings που θα χτίσουν το τελικό String που θα αποσταλεί  
        στο serial port του Arduino.  
    ** Τύπωσε αποτελέσματα στην οθόνη  
    ** Στείλε το String στο Arduino.  
}
```

Αρχικά αναζητεί συσκευές που έχουν ενταχθεί στο δίκτυο μετά την τελευταία ανίχνευση και στην συνέχεια δημιουργεί αιτήματα για λήψη μετρήσεων ηλεκτρικής ενέργειας και τα αποστέλλει σε όλους με την βοήθεια ενός αντικειμένου που ανήκει στην κλάση **ParseBuffer()**. Το αντικείμενο αυτό περιλαμβάνει πεδίο Hashtable που καταχωρεί τα ids των συσκευών Plogg και στις αντίστοιχες θέσεις τα αντικείμενα Devices με τις πληροφορίες που τους αναλογούν. Στην συνέχεια χτίζουμε μια λίστα με όλα τα ids και ακολούθως περνάμε πάνω από την λίστα ελέγχοντας μια μια τις τιμές αυτές.

Οι συσκευές αφού λάβουν τα αιτήματα αποστέλλουν τις μετρήσεις που πρώτα περνούν από ανάλυση και μετά καταχωρούνται σε αντικείμενο τύπου **PloggDevices()** για να έρθουν να τυπωθούν στο terminal του Eclipse σε μορφή .json με την βοήθεια της μεθόδου **JSONprintAll()**. Σε αυτή την μέθοδο κάθε φορά που τυπώνονται οι μετρήσεις της εκάστοτε συσκευής αποθηκεύονται τα αποτελέσματα και σε ένα global πίνακα από Strings με την κάθε θέση του πίνακα να αντιπροσωπεύει και μια τιμή ακολουθούμενη με τον χαρακτήρα ‘,’ ή ‘!’. Ο χαρακτήρας ‘!’ μετά από σχετικό έλεγχο μπαίνει κάθε φορά στη τελευταία θέση του πίνακα ενώ στις υπόλοιπες τα Strings συνοδεύει ‘,’. Αυτό γίνεται διότι σκοπός μας είναι να ενώσουμε όλες αυτές τις τιμές φτιάχνοντας ένα String που θα διαβαστεί από το Arduino και θα περάσει στην ιστοσελίδα ενημερώνοντας τις αντίστοιχες γραφικές.

Μορφή .json : `{"KWH":["0.003"],"NAME":[" Plogg No Name"],"ID":["000D6F00007D2B29"],"KWHM":[0],"WATTS":["41.94"],"TIMESTAMP":["2001 JAN 01 01:05:32 "]}`

Η ιστοσελίδα λαμβάνοντας ένα HTTP POST της μορφής : “value 1,value 2,value 3,value 4” ενημερώνει την 1^η γραφική παράσταση με το value 1, την 2^η με το value 2 και ούτω καθεξής άρα αυτή η λειτουργία προϋποθέτει κάθε φορά μια συσκευή να αναθέτει τις τιμές της στην ίδια θέση του πίνακα έτσι ώστε να χτίζεται πάντα σωστά το τελικό String. Αυτό δεν αποτελεί πρόβλημα αφού το Hashtable του αντικειμένου κλάσης **ParseBuffer()** διαπερνάται πάντα με την ίδια σειρά άρα ο global πίνακας που χτίζεται θα είναι πάντα ο ίδιος άρα και το τελικό String δομείται με τον ίδιο ακριβώς τρόπο.

Παράδειγμα υλοποίησης:

Global Array

“3.15,”
“6.67,”
“2.45,”
“10.01!”

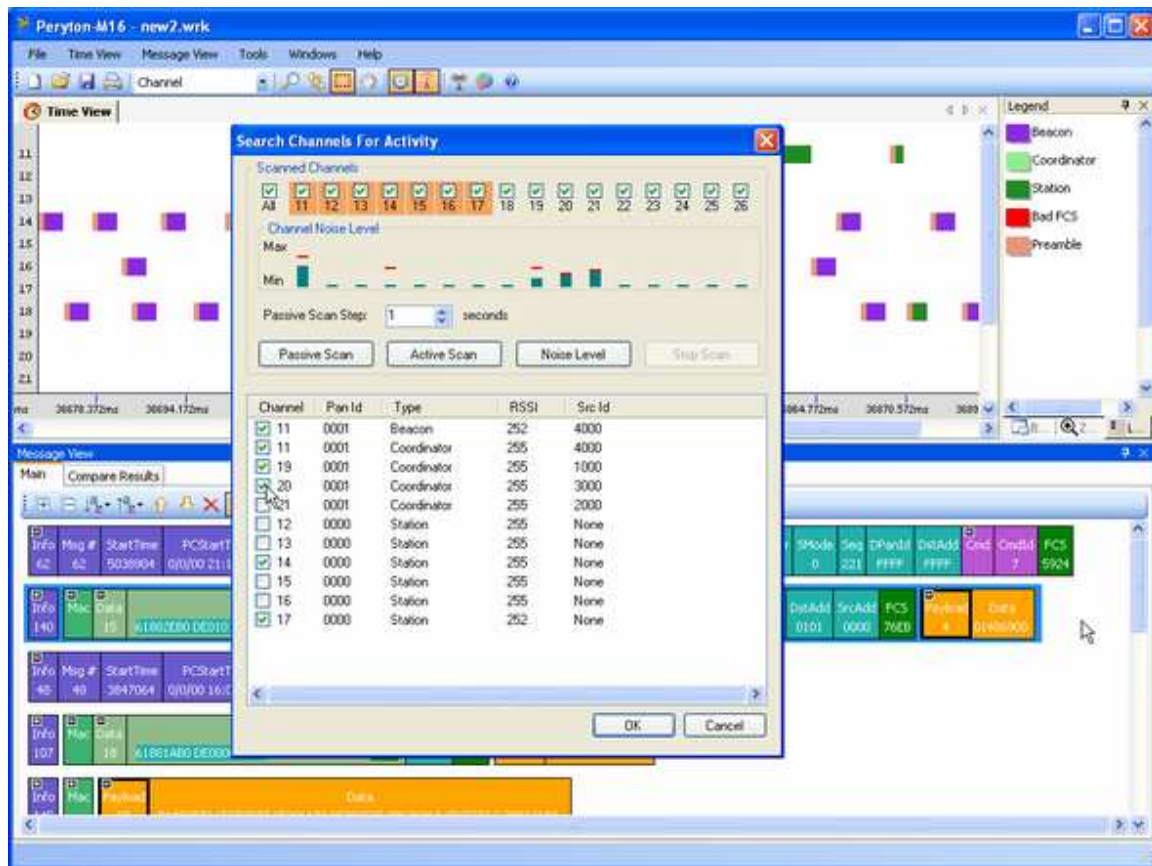
Final String

“3.15,6.67,2.45,10.01!”

3.4 Εναλλακτικές Υλοποιήσεις

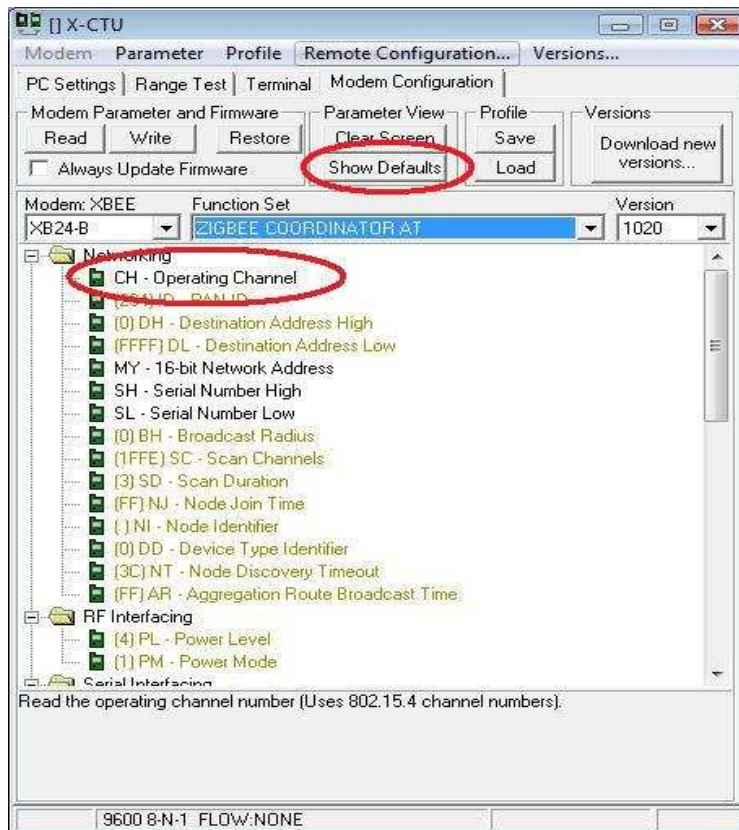
Πρώτη Υλοποίηση : Με βάση την λογική ότι σκοπός μας σε αυτή την εργασία ήταν η απεμπλοκή από το USB stick προσπαθήσαμε να δημιουργήσουμε 1 Arduino κόμβο που θα είχε το ρόλο του PAN Coordinator και αντικαθιστά το Telegesis USB stick σε ένα και μοναδικό δίκτυο που βασίζεται στην τεχνολογία ZigBee. Για την δημιουργία του ZigBee δικτύου έχουμε στην κατοχή μας XBee modules series 2 που τοποθετούνται στο XBee shield και ενώνονται απευθείας στα boards δίνοντας τους ZigBee δυνατότητες μέσα από το λογισμικό X-CTU. Έτσι είμαστε σε θέση να δημιουργήσουμε το ZigBee-based δίκτυο με συγκεκριμένο PAN ID, να θέσουμε τον ρόλο των κόμβων στο δίκτυο και αυτά να επικοινωνούν μεταξύ τους κανονικά έχοντας τις λαβές στα XBee shields στην θέση XBEE. Άρα το μόνο που απομένει είναι να ανακαλύψει ο Arduino κόμβος που έχει οριστεί σαν PAN Coordinator τις Plogg συσκευές που βρίσκονται στο γύρω περιβάλλον, με την εντολή “AT+SN”, και να αλληλεπιδρά απευθείας μεταξύ τους χωρίς την ύπαρξη άλλου software(βλέπε Java drivers) ή hardware(βλέπε Telegesis USB stick) βοηθήματος.

Το πρόβλημα που δημιουργείται είναι ότι εγκαθιδρύοντας το δίκτυο ο PAN Coordinator αναθέτει το κανάλι που θα κάνουν operate οι κόμβοι μέσα από δικές του διεργασίες και αυτό το πιο πιθανό είναι ότι δεν συμβαδίζει με αυτό που ακούει και στέλλει μια συσκευή Plogg. Για να είμαστε σε θέση να ξέρουμε ότι συμβαίνει κάτι τέτοιο και να ανακαλύψουμε τα κανάλια που γίνονται τα operations από τον εκάστοτε κόμβο χρησιμοποιούμε το λογισμικό Perytons. Το εργαλείο αυτό με την βοήθεια ενός ZigBee USB stick είναι σε θέση να κάνει analyse τα πακέτα που στέλλονται σε wireless δίκτυα πρωτοκόλλων 802.15.4, ZigBee, ZigBee RF4CE, 6LoWPAN/ZigBeeIP, 802.15.4a, DASH7 και PLC και να παρουσιάζει όλα τα δεδομένα που αφορούν το δίκτυο μέχρι και το επίπεδο του MAC Layer, δηλαδή το επίπεδο των καναλιών.



Σχήμα 3.13 Εργαλείο Perytons, ανίχνευση ενεργών καναλιών

Αρα για να μπορέσουμε να δούμε το κανάλι που κάνει τα operations μια συσκευή Plogg ανοίγουμε το Plogg Manager δημιουργούμε ένα απλό δίκτυο ZigBee μεταξύ Telegesis USB stick και Plogg και ταυτόχρονα κάνουμε capture τα πακέτα που ανταλλάσσονται έχοντας σε λειτουργία το Perytons. Κάνοντας την ίδια διαδικασία και για το δίκτυο των Arduino είδαμε αυτό που είχαμε κατά νου, ότι δηλαδή για να μην ανταποκρίνονται τα Ploggs στις εντολές που στέλλουμε από το Arduino PAN Coordinator είναι διότι λειτουργούν σε διαφορετικά κανάλια-συχνότητες. Έτσι αφού ξέρουμε το κανάλι των Ploggs αν με κάποιο τρόπο καταφέρναμε να μεταβάλουμε το κανάλι που επιλέγει ο PAN Coordinator θα μπορούσε να κάνει ότι και το Telegesis USB stick και η διαδικασία να ξεκινήσει, όμως είναι εδώ που παρουσιάζεται και το πρόβλημα. Η σχετική ρύθμιση του καναλιού γίνεται στο XBee module το οποίο ως series 2 firmware έχει ορισμένη ως read-only την κατάσταση του πεδίου που αφορά το κανάλι αφήνοντας τον κόμβο που ορίζεται ως PAN Coordinator να επιλέγει ο ίδιος κάθε φορά το ιδανικό κανάλι με βάση το energy scan που επιχειρεί. Έτσι είτε δοκιμάζοντας μέσω X-CTU είτε μέσω PuTTY να μεταβάλουμε αυτή την τιμή το αποτέλεσμα είναι αρνητικό.



Σχήμα 3.14 Εργαλείο X-CTU, παρουσιάζονται οι default τιμές των παραμέτρων όπου με κίτρινο είναι αυτές που μπορούν να γίνουν update και με μαύρο αυτές που είναι read-only

Παράδειγμα με Terminal PuTTY

Αποστολή : +++

,για να μπούμε σε φάση Command mode

Λήψη : OK

Αποστολή : ATCHB

,θέτουμε ως κανάλι επικοινωνίας το κανάλι B(HEX) που αντιπροσωπεύει το 11(DEC) και αντιστοιχεί σε αυτό των Ploggs

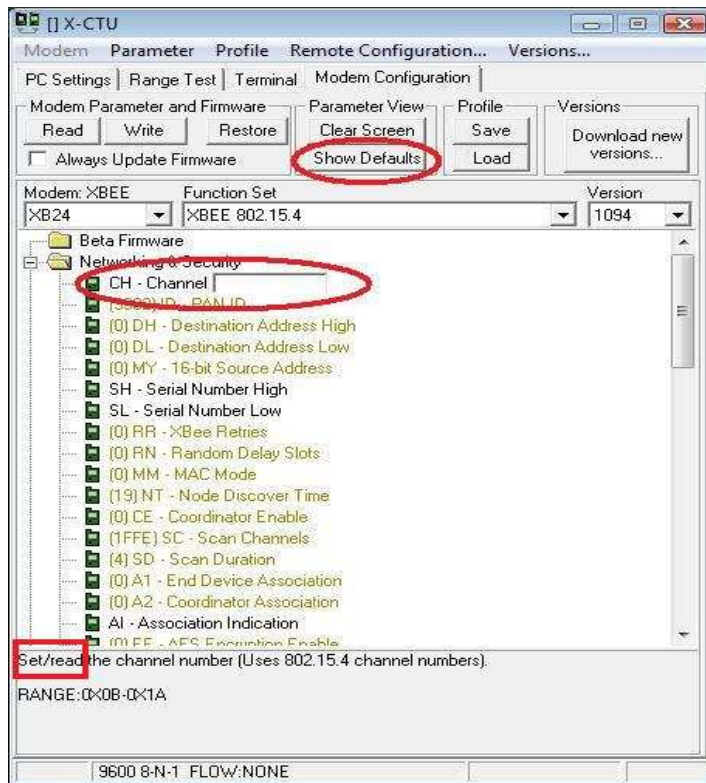
Λήψη : ERROR

Καταληκτικά, αφού η επικοινωνία μεταξύ Arduino PAN Coordinator και Plogg δεν δύναται να επιτευχθεί με την μεταβολή του καναλιού του XBee module series 2 η διαδικασία λήψης μετρήσεων από μια συσκευή δεν μπορεί καν να ξεκινήσει άρα προσφεύγουμε σε άλλες λύσεις.

Δεύτερη Υλοποίηση : Έχοντας ως δεδομένο την αδυναμία μεταβολής του καναλιού στο δίκτυο ZigBee των Arduino χρειαζόταν να εξευρεθεί μια λύση που δεν απαιτούσε μια τέτοια

προσπάθεια. Βασική σκέψη για την δεύτερη υλοποίηση αποτέλεσε μια βασική αρχή της τεχνολογίας ZigBee που θέλει τους κόμβους που είναι προς ένταξη να κάνουν scan για ενεργά κανάλια και ακολούθως να επιλέγουν ποιος θα είναι ο κόμβος πατέρας τους, στην περίπτωση την δική μας αφού δεν ορίζουμε PAN Router ο πατέρας όλων είναι ο PAN Coordinator, με βασικό συνεπακόλουθο τον ορισμό ίδιου καναλιού με αυτόν. Με αυτό τον τρόπο θα μπορούσαμε να έχουμε το Telegesis USB stick να επικοινωνεί με τις συσκευές Plogg και να υπάρχει στο δίκτυο το Arduino σαν End Device ή Router που με την σειρά του θα μπορούσε να επικοινωνεί με τα Ploggs που και αυτά θεωρούνται End Devices. Στην περίπτωση ορισμού ως End Device προκύπτει πρόβλημα αφού τέτοιοι κόμβοι αποτελούν RFD κόμβους που προϋποθέτουν την μεσολάβηση FFD, όπως PAN Router ή PAN Coordinator, για να μιλήσουν μεταξύ τους. Άρα η μόνη λύση είναι ο ορισμός σαν PAN Router που θα είναι ορισμένο από πριν στο PAN ID του δικτύου που δημιουργεί το Telegesis USB stick, θα το βρίσκει, θα εντάσσεται στο δίκτυο και θα αποστέλλει στα Ploggs τα AT commands. Ξεκινώντας από το πρώτο βήμα και τρέχοντας παράλληλα το εργαλείο Peryons παρατηρήσαμε ότι το Arduino παρόλο το scan που επιχειρεί φαίνεται να μην επικοινωνεί με το USB stick αφού δεν λαμβάνει τίποτα στο Serial Monitor και παραμένει στο κανάλι που έρχεται εργοστασιακά που είναι το κανάλι 0. Η λύση αυτή δεν προχώρησε αφού παρόλο που είναι πραγματοποιήσιμη προϋποθέτει την ύπαρξη του Telegesis USB stick που στην ουσία δεν θα μας διευκόλυνε καθόλου αφού παρά την ύπαρξη του όλες οι λειτουργίες θα έπρεπε να γίνονται στο Arduino receiver κάτι που αποφεύγεται με την χρήση των βοηθητικών Java drivers.

Τρίτη Υλοποίηση : Αφού έγινε προσπάθεια με τα XBee modules series 2 και δεν καρποφόρησε λόγω αδυναμίας εξεύρεσης του σωστού καναλιού του ZigBee δικτύου η επόμενη προσπάθεια αφορούσε την χρήση XBee modules series 1. Αυτά τα modules είναι σε θέση να μεταβάλουν το κανάλι που κάνουν operate άρα ορίζοντας στον Arduino κόμβο το PAN ID του ZigBee δικτύου είμαστε σε θέση να “ακούμε” στο κανάλι που ανταλλάζουν τα δεδομένα το Telegesis USB stick με τις συσκευές Plogg και να εκτελούμε λειτουργία παρόμοια με αυτή του εργαλείου Perytons.



Σχήμα 3.15 Εργαλείο X-CTU, το πεδίο CH-Channel είναι σε θέση να γίνει update

Επειδή όμως το Arduino αποτελεί 802.15.4 κόμβο και όχι ZigBee τα δεδομένα παρουσιάζονται στο Serial Monitor σε Hexadecimal μορφή γεγονός που καθιστά την κατανόηση και μετατροπή τους πολύ δύσκολη. Επίσης στα δεδομένα που λαμβάνονται εμπερικλείονται και όλα τα headers του ZigBee stack με πολλά fields να είναι κρυπτογραφημένα έτσι η προσπάθεια εγκαταλείφθηκε. Τρέχοντας παράλληλα με τον Arduino κόμβο και το Perytons μπορούσαμε να δούμε ότι λάμβαναν τα ίδια δεδομένα, με το Perytons να τα χωρίζει και να τα αναλύει σε fields όμως και πάλι αυτό δεν στάθηκε ιδιαίτερα βοηθητικό.

Τέταρτη Υλοποίηση : Σαν γενικό συμπέρασμα από τις προηγούμενες υλοποιήσεις εξάγαμε ότι χρειαζόμαστε ένα XBee module που να μπορεί να θέτει το κανάλι που θέλουμε και να έχει δυνατότητες ZigBee. Στην περίπτωση του XBee module series 2 δημιουργούμε το ZigBee δίκτυο αλλά το κανάλι που γίνονται τα operations δεν είναι αυτό που ζητάμε. Από την άλλη τα XBee modules series 1 που μπορούν να μεταβάλουν το κανάλι τους δεν έχουν το ZigBee stack που να τους επιτρέπει να επικοινωνούν απευθείας με άλλους ZigBee κόμβους έτσι το πρόβλημα υφίσταται. Τι θα γινόταν όμως αν υπήρχε τρόπος μέσω κάποιου αναβαθμισμένου firmware ένα XBee module series 1 να μπορούσε να έχει ZigBee δυνατότητες; Ψάχνοντας

περισσότερο αυτή την ιδέα έγινε αντιληπτή η ύπαρξη θέματος σε διάφορα Arduino forum που έκαναν λόγο για κάποιο ZigBee beta firmware για XBee series 1 που είχε εκδοθεί για κάποιο διάστημα από την εταιρεία Maxstream (εταιρεία που αργότερα αγοράστηκε από την Digi) και σου έδινε την δυνατότητα κάνοντας το upload σε ένα XBee series 1 να παρουσιάζονται τα function options router AT, router API, coordinator AT και coordinator API. Το version αυτό αποτελούσε ένα δοκιμαστικό στάδιο πριν την έκδοση των XBee series 2 και για αυτό στην συνέχεια αποσύρθηκε και δεν παρουσιάστηκε ξανά προς διάθεση. Κάνοντας τις απαραίτητες ενέργειες και επικοινωνώντας με άλλους χρήστες τα αρχεία ανεβρέθηκαν και ξεκίνησε η διαδικασία φόρτωσης τους στο application X-CTU για να είμαστε σε θέση να τα κάνουμε upload στα modules. Καταρχήν, κατεβάσαμε το version 5.1.0.0 του X-CTU που αποτελεί μια παλαιότερη έκδοση του εργαλείου και επιλέξαμε τα updates να μην γίνονται αυτόματα για να το κάνουμε εμείς manually. Στην συνέχεια ανοίγοντας το application, ανοίξαμε το παράθυρο “Modem Configuration”, επιλέξαμε το Modem XB24 που αντιστοιχεί στα modules που έχουμε διαθέσιμα και πατώντας στο κουμπί “Download new versions” ζητήσαμε αυτό να γίνει από το “File” και όχι το “Web”. Ενημερώνοντας το X-CTU με τα αρχεία αυτά έχουμε πλέον διαθέσιμα versions με ZigBee ιδιότητες για XBee modules series 1 όμως δυστυχώς για εμάς τα συγκεκριμένα modules σε αντίθεση με τα series 2 δεν αποδέχονται αλλαγή του version τους και οι παράμετροι που επιλέγουμε δεν είναι εφικτό να γίνουν upload. Από την μεριά της η Digi υποστηρίζει με ανακοίνωση της ότι τα τελευταία XBee series 1 που έχουν δοθεί στην αγορά δεν μπορούν να κάνουν load το ZigBee beta firmware έχοντας προφανώς κίνητρο την προώθηση των XBee series 2 έτσι η προσπάθεια και σε αυτή την υλοποίηση σταματά πρόωρα.

Κεφάλαιο 4

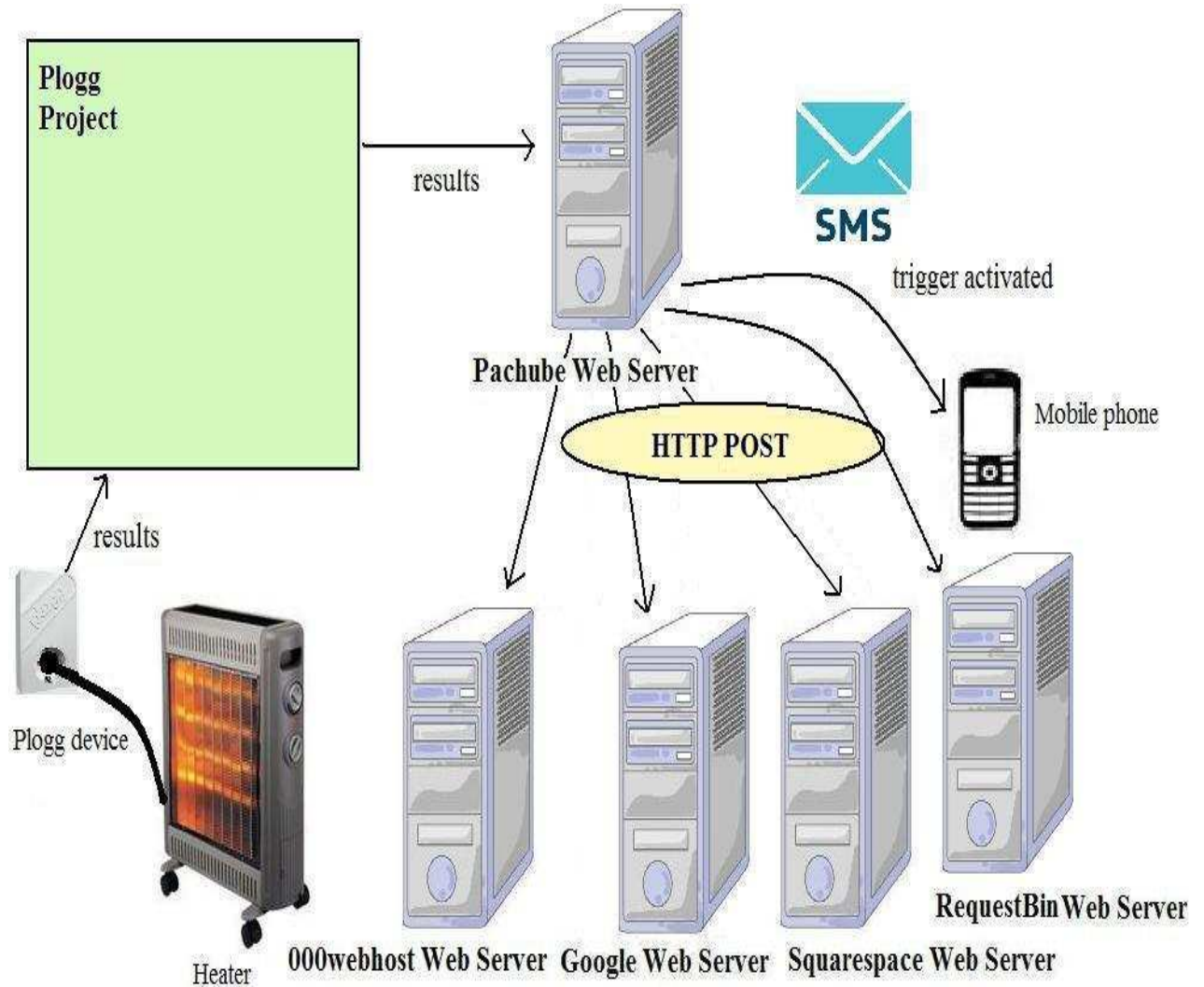
Περιφερειακές Λειτουργίες

4.1	Δομή λειτουργιών	39
4.2	Ανάπτυξη Ιστοσελίδας καταγραφής αποτελεσμάτων	40
4.3	Triggers	43
4.3.1	SMS alert	44
4.3.2	Email alert	45
4.3.2.1	Επιθεώρηση HTTP POST	46

4.1 Δομή Λειτουργιών

Έχοντας πλέον υλοποιημένο ένα σύστημα καταγραφής ενέργειας προχωρήσαμε ένα βήμα περισσότερο προσδίδοντας του δυνατότητες ιδιαίτερα χρήσιμες και λειτουργικές προς το κοινό. Με την ιστοσελίδα Pachube είμαστε σε θέση να χτίζουμε γραφικές παραστάσεις και να τις έχουμε φυλαγμένες στα περιορισμένα όρια της συγκεκριμένης ιστοσελίδας. Αν αυτό που ζητάμε είναι η δική μας ενημέρωση και όσον τύχει να δουν κάποιο από τα feed μας είμαστε ικανοποιημένοι, αν όμως θέλουμε να έχουμε απήχηση σε ένα μεγαλύτερο κοινό και να μπορούμε να του παρουσιάσουμε κάτι πιο επίσημο και ουσιώδες δημιουργούμε μια ιστοσελίδα σε ένα public server κάνοντας POST γραφικές παραστάσεις σε διάφορες μορφές που είναι ενημερωμένες up-to-date με το τελευταίο version των μετρήσεων που καταγράψαμε. Στην ιστοσελίδα που έχει δημιουργηθεί μπορεί κάποιος να ενημερωθεί για τις μετρήσεις όλων των συσκευών Plogg που είναι σε λειτουργία και να αφήσει το δικό του σχόλιο αφού περιλαμβάνει blog απόψεων κατάλληλο για διεξαγωγή συζητήσεων. Για μια πιο πλήρη και άμεση ενημέρωση του χρήστη που έχει στην διάθεση του το σύστημα η ιστοσελίδα Pachube παρέχει την δυνατότητα μέσα από triggers να μαθαίνει τυχόν απρόβλεπτες μετρήσεις μέσα από την αποστολή γραπτών μηνυμάτων SMS στο κινητό του τηλέφωνο, αποστολή Email ή ενημέρωση του iGoogle gadget στο παράθυρο του browser. Η λειτουργία των triggers προσδίδει εξυπνάδα στο σύστημα και είναι ιδιαίτερα χρήσιμη στις περιπτώσεις όπου π.χ έχουμε ένα αισθητήρα φωτιάς που έχει καταμετρήσει θερμοκρασία μεγαλύτερη από κάποιους συγκεκριμένους βαθμούς κελσίου αδικαιολόγητα. Τότε ο χρήστης

μπορεί να ξέρει άμεσα τον κίνδυνο και να μην χρειάζεται κάθε φορά να ελέγχει ο ίδιος ότι οι μετρήσεις κινούνται σε φυσιολογικά επίπεδα. Στην δική μας περίπτωση αφού είμαστε σε θέση να μετρήσουμε την κατανάλωση ενέργειας θέτουμε ένα ανώτατο όριο που αν το υπερβεί κάποια από τις ενεργοποιημένες ηλεκτρικές συσκευές, όπως π.χ μια θερμάστρα των 7500 ή 15000W που είναι ιδιαίτερα ακριβές στη χρήση τους, ενημερώνεται ο χρήστης και είναι σε θέση να την κλείσει εξοικονομώντας σημαντική ποσότητα ενέργειας.



Σχήμα 4.1 Περιφερειακές λειτουργίες

4.2 Ανάπτυξη Ιστοσελίδας καταγραφής αποτελεσμάτων

Η ιστοσελίδα που δημιουργήθηκε αποτελεί κάτι εντελώς πρόχειρο και μπορεί κάποιος να έχει πρόσβαση σε αυτή στην διεύθυνση <http://myelectricityusage.squarespace.com/>.

Αποτελεί προϊόν της σελίδας <http://www.squarespace.com/> που σου δίνει τον χώρο και όλα

τα εργαλεία για να χτίσεις γραφικά, χωρίς κάποια ανάπτυξη κώδικα, και δωρεάν την δική σου ιστοσελίδα με ενσωματωμένο προσωπικό blog για κατάθεση απόψεων και ανάπτυξη, σχετικών προς το αντικείμενο, συζητήσεων. Στην ιστοσελίδα μπορεί κάποιος να βρει, εκτός από φωτογραφίες και βίντεο, τις τελευταίες μετρήσεις που έχουν καταγραφεί από την σελίδα Pachube σε μορφή διάφορων γραφικών παραστάσεων. Αυτό γίνεται με την εισαγωγή HTTP κώδικα που ενσωματώνει απευθείας τα αποτελέσματα από το Pachube μόλις αυτά καταγραφούν. Ο κώδικας αυτός δημιουργείται πολύ εύκολα και αυτόματα από τα applications του Pachube Internet of Things repository που χρειάζονται input μόνο το feed id και ένα datastream number από τα οποία απορροφά τις μετρήσεις παρουσιάζοντας τες στην μορφή που εμείς θέλουμε.

Οι μορφές των γραφικών είναι οι ακόλουθες:

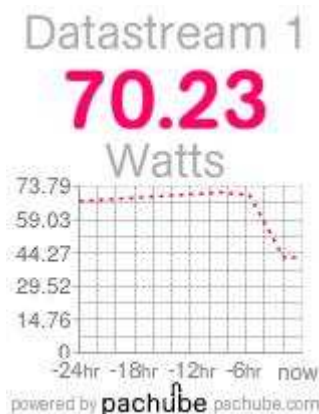
A) PachuDial



Σχήμα 4.2

Το PachuDial με τον μαύρο δείκτη καταδεικνύει την τρέχον μέτρηση και τον κόκκινο την μέγιστη που έχει σημειωθεί μέχρι τώρα.

B) PachuBlog

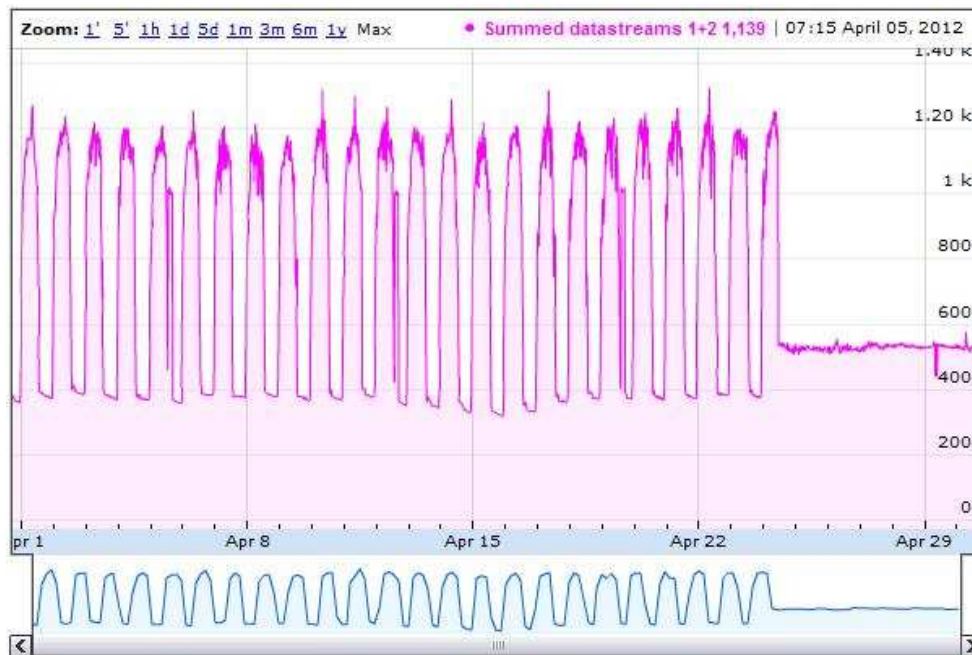


Σχήμα 4.3

Το PachuBlog είναι ένα δυναμικό gadget που παρουσιάζει υπό μορφή τιμής την τελευταία μέτρηση και την γραφική που έχει χτιστεί μέχρις εκείνο το σημείο.

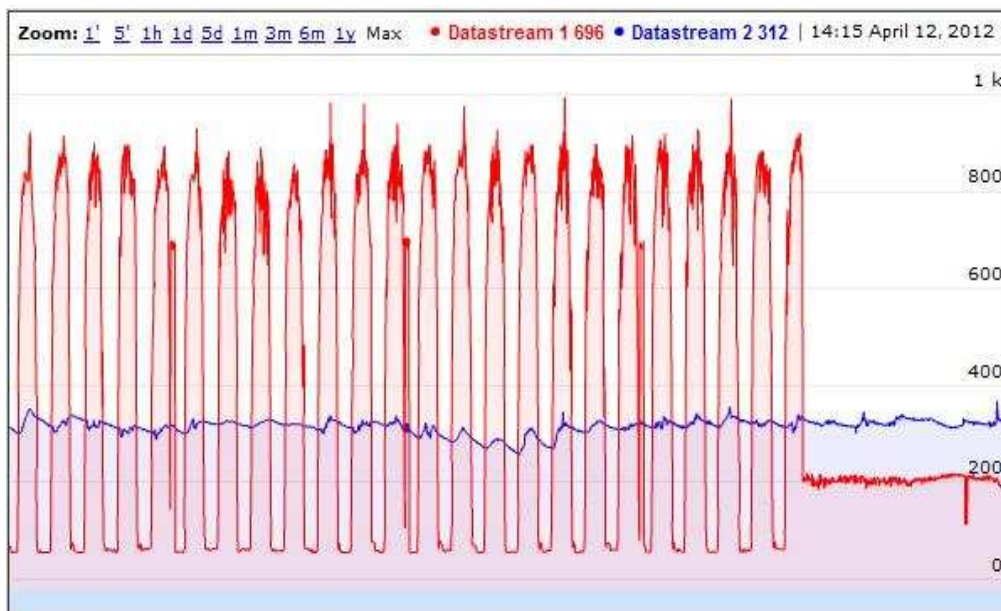
C) Zoomable 30-day graph

i) Summed graph



Σχήμα 4.4

ii) Normal graph



Σχήμα 4.5

Με αυτές τις παραστάσεις μπορούμε να συγκρίνουμε τις τιμές των datastream για περίοδο 30 ημερών κάνοντας zoom στα επίμαχα σημεία με τις γραφικές να κρατάνε και να παρουσιάζουν τις τιμές όλων των datastream που έχουν γίνει import.

Τέλος να πούμε ότι για την παρουσίαση του gadget στο iGoogle panel ακολουθείται η διαδικασία που περιγράψαμε με τα plug-ins του feed στην ιστοσελίδα και έχουμε σαν αποτέλεσμα μπροστά μας ένα όμορφο περιβάλλον με τις δραστηριότητες που είχαμε επιλεγμένες στο panel από πριν και επιπρόσθετα ενημέρωση με gadget για την εξέλιξη των μετρήσεων που παίρνουμε στο feed του Pachube.

4.3 Triggers

Με την χρήση των triggers στα datastream του Pachube feed που δημιουργήσαμε προσδίδουμε λειτουργίες άμεσης καταγραφής απρόβλεπτων τιμών που θέλει ο χρήστης να ξέρει την ώρα που συμβαίνουν. Αυτές οι λειτουργίες αποτελούν κάτι το αρκετά συνηθισμένο για συστήματα παρόμοια με αυτό που επιχειρούμε γι' αυτό και η ιστοσελίδα έχει τους μηχανισμούς για να υλοποιούνται σχετικά γρήγορα και απλά. Τα triggers ενεργοποιούνται στις εξής καταστάσεις:

- A) Καταγραφή τιμής μεγαλύτερης από αυτή που θέσαμε ως τιμή αναφοράς.
- B) Καταγραφή τιμής μεγαλύτερης ή ίσης από αυτή που θέσαμε ως τιμή αναφοράς.
- Γ) Καταγραφή τιμής ίσης με αυτή που θέσαμε ως τιμή αναφοράς.
- Δ) Καταγραφή τιμής μικρότερης από αυτή που θέσαμε ως τιμή αναφοράς.
- E) Καταγραφή τιμής μικρότερης ή ίσης από αυτή που θέσαμε ως τιμή αναφοράς.
- Z) Καταγραφή οποιασδήποτε τιμής.
- H) Αλλαγή κατάστασης του feed σε live. Αυτό γίνεται όταν γίνει κάποια μετατροπή σε οποιοδήποτε από τα datastreams.
- Θ) Αλλαγή κατάστασης του feed σε frozen. Αυτό γίνεται όταν το feed μείνει idle για κάποιο χρονικό διάστημα.

Αν θέλουμε να ενημερωνόμαστε όταν καταγραφεί μέτρηση μεγαλύτερη από μια τιμή αλλά μικρότερη από κάποια άλλη χρειάζεται απαραίτητα η δημιουργία ενός επιπλέον trigger που να διεξάγει το δεύτερο έλεγχο αλλά αυτό δεν είναι πρόβλημα αφού για ένα datastream μπορούμε να έχουμε απεριόριστο αριθμό από triggers.

Για τους σκοπούς της τρέχον Ατομικής Διπλωματικής Εργασίας έχουμε χρησιμοποιήσει την πρώτη περίπτωση λήψης μιας τιμής έτσι ώστε να ενημερωνόμαστε όταν έχουμε αυξημένα επίπεδα κατανάλωσης ηλεκτρικού ρεύματος για να μπορούμε να κάνουμε τις ενέργειες αποτροπής περεταίρω κατανάλωσης. Τα triggers ενημερώνουν το κινητό τηλέφωνο καθώς και το Email του χρήστη έτσι ώστε να φαίνονται καταγεγραμμένες όλες οι τιμές που έχουν ξεπεράσει το όριο που έχει τεθεί.

4.3.1 SMS alert

Στην γενιά που ζούμε κάθε άνθρωπος είναι συνυφασμένος με το κινητό του τηλέφωνο. Πλέον οι συσκευές αυτές αποτελούν μέρος της ζωής και του εαυτού μας αφού η επικοινωνία με τον γύρω κόσμο περνά σε ένα πολύ μεγάλο βαθμό από την δράση τους. Λόγω αυτής της κουλτούρας έχουν αναπτυχθεί από διάφορα συστήματα και τα SMS alerts που παρέχουν την πιο άμεση ενημέρωση από οποιαδήποτε άλλη πηγή. Το Pachube είναι σε θέση να παρέχει αυτή την λειτουργία με την δημιουργία trigger και την χρήση έτοιμου application που ζητώντας μόνο το Country Code(για την Κύπρο 00), το Area Code(για την Κύπρο 357) και τον αριθμό του κινητού αποστέλλει το μήνυμα SMS στον προορισμό του. Να σημειώσουμε ότι το συγκεκριμένο alert δεν περιορίζει τους χρήστες μόνο στα δικά τους feeds αφού κάποιος μπορεί να ενημερώνεται και για άλλες μετρήσεις άλλων χρηστών που τον ενδιαφέρουν δίνοντας απλά το ανάλογο feed και datastream.

Το προκαθορισμένο μήνυμα SMS που στέλλεται έχει ως εξής:

From: Pachube

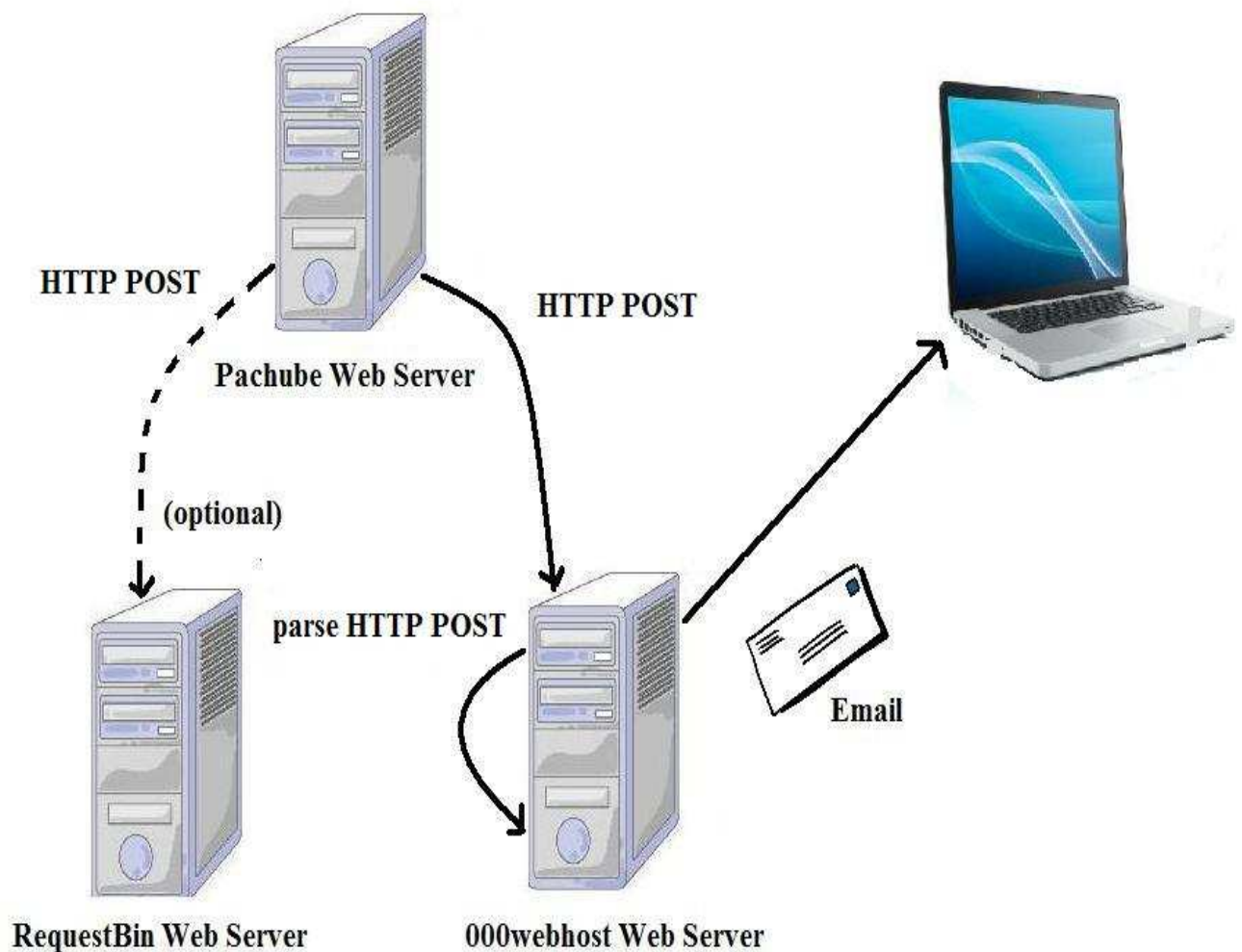
[Subject of the feed] alert: f [feed number], ds [datastream number] val [τιμή που καταγράφηκε και ήταν η αιτία να ενεργοποιηθεί το trigger] [κατάσταση trigger που παραβιάστηκε π.χ gt 12] @ [timestamp]. (αύξων αριθμός μηνύματος/20)

Το τελευταίο πεδίο έχει να κάνει με τον αριθμό μηνυμάτων που έχουν σταλεί μέχρι τώρα.

Κάθε κινητό τηλέφωνο μπορεί να δεχτεί τον μέγιστο αριθμό των 20 μηνυμάτων SMS και αυτό σύμφωνα με το support team της ιστοσελίδας Pachube έγινε λόγω της υψηλής ζήτησης που είχε ως επακόλουθο το μεγάλο κόστος που ήταν αναγκασμένοι να επιφορτιστούν για την αποστολή των μηνυμάτων. Για να μπορεί κάποιος να απολαμβάνει αυξημένα ωφέληματα που συμπεριλαμβάνουν απεριόριστο διαθέσιμο αριθμό μηνυμάτων απαιτείται η αναβάθμιση του λογαριασμού του σε Premium μέσω χρηματικής αποζημίωσης.

4.3.2 Email alert

Σε συνέχεια της προηγούμενης λειτουργίας και της ανάγκης λήψης ενημερωτικών μηνυμάτων SMS προχωρούμε ένα βήμα παρακάτω και εισαγάγουμε την λειτουργία αποστολής ενημερωτικών email. Όπως και πριν δημιουργούμε ένα trigger που ενεργοποιείται όταν καταγραφεί μια μη αναμενόμενη τιμή και με το κάλεσμα ενός php script που έχουμε ανεβάσει σε δικό μας server το email αποστέλλεται στον προορισμό που επιζητούμε. Στην συγκεκριμένη περίπτωση δεν υπάρχει κάποιος περιορισμός στον αριθμό των email που είναι δυνατό να σταλούν και οι χρήστες έχουν την δυνατότητα ενημέρωσης ηλεκτρονικά τόσο των δικών τους μετρήσεων όσο και των άλλων χρηστών. Το php script που χρησιμοποιούμε παίρνει σαν input ένα HTTP POST σε μορφή .json το αναλύει και στέλλει το email στην μορφή που εμείς θέλουμε. Για το πώς γίνεται ο έλεγχος του HTTP POST που στέλλεται από την ιστοσελίδα και λαμβάνεται σαν input από το script θα δούμε στην επόμενη ενότητα.



Σχήμα 4.6 Διάγραμμα λειτουργιών για αποστολή Email

Προχωρώντας ένα βήμα παρακάτω αναλύουμε τον κώδικα του php script.

```
$trigger = rawurldecode(substr(stripslashes(file_get_contents('php://input')),5));
```

Ξεκινώντας από τις εσωτερικές παρενθέσεις με την χρήση της συνάρτησης **file_get_contents()** διαβάζουμε τα raw POST δεδομένα αφού η παράμετρος 'php://input' αντιστοιχεί στο input stream που λαμβάνει το php script, αφαιρούμε τα διπλά backslash και κάνουμε decode το substring που μας ενδιαφέρει.

```
$json = json_decode($trigger);  
$environment_id = $json->{'environment'}->{'id'};  
$datastream_id = $json->{'triggering_datastream'}->{'id'};  
$value = $json->{'triggering_datastream'}->{'value'}->{'current_value'};  
$timestamp = $json->{'timestamp'};
```

Ακολούθως παίρνουμε το \$trigger που είναι ένα encoded JSON string και το κάνουμε decode δημιουργώντας μια php μεταβλητή json που θα μας δώσει τα πεδία που αναζητούμε.

```
$message = "Trigger activated by feed $environment_id, datastream $datastream_id, at  
$timestamp.\n\n";  
$message .= "Complete trigger message: \n $trigger";  
$message = wordwrap($message, 70);
```

Στην συνέχεια κτίζουμε το μήνυμα που θέλουμε να γράφεται στο Email και το αποστέλλουμε στην ηλεκτρονική διεύθυνση με την χρήση της συνάρτησης **mail()**. Η δεύτερη παράμετρος της συνάρτησης καθορίζει τον τίτλο και η τρίτη το περιεχόμενο του email.

```
mail("example@domain.com", "trigger activated", $message)
```

4.3.2.1 Επιθεώρηση HTTP POST

Για σκοπούς ελέγχου των HTTP requests χρησιμοποιούμε την εξειδικευμένη ιστοσελίδα <http://requestb.in/> που επιτρέπει στους χρήστες την δημιουργία URL που συλλέγει τα requests

που γίνονται προς αυτό παρουσιάζοντας τα σε user friendly περιβάλλον. Μέσα από την ιστοσελίδα αυτή μπορεί κάποιος να δει τι στέλλει ο προσωπικός του HTTP client ή να έχει ενώπιον του τι έχει ληφθεί από άλλους διαδικτυακούς ιστότοπους. Δημιουργώντας το URL και αναθέτοντας το ως link του trigger του Pachube datastream μπορούμε να λαμβάνουμε τα POST για τις επόμενες 48 ώρες. Τα POST εμφανίζονται σε μορφή .json και καθώς συλλέγονται παρουσιάζονται μπροστά μας σαν λίστα με το πρώτο να αντιστοιχεί στην τελευταία “πυροδότηση” του trigger.

POST /w5gzkew5



```
body { "debug": true, "environment": { "description": "Measuring electrical power consumption of a variety of Plogg Devices in order to build a Web Service Information panel of graphs available to everyone.\r\n\r\nThe experiments are held as a Project of University of Cyprus.\r\n\r\nComputer Science Department.\r\n\r\nAcademic Year: 2011-2012\r\n\r\nSemester: Spring 2012", "feed": "http://api.pachube.com/v2/feeds/43522", "id": "43522", "location": { "lat": "35.1002487433244", "lon": "33.3723449707031", "name": "UCY-Cyprus", "private": false, "title": "Plogg Measurements", "id": "7427", "threshold_value": "42", "timestamp": "2012-05-08T11:04:10Z", "triggering_datastream": { "id": "1", "units": { "label": "laptop Watts", "symbol": null, "type": null }, "uri": "http://api.pachube.com/v2/feeds/43522/datastreams/1", "value": { "max_value": 119.23, "min_value": 0.0, "value": "82.1", "type": "gt", "uri": "http://api.pachube.com/v2/triggers/7427" }
```

Από το πιο πάνω POST μπορούμε να ξεχωρίσουμε το threshold value που έχει τεθεί στα 42W, το value που αναφέρεται στην τιμή που ενεργοποίησε το trigger και στην συνθήκη ενεργοποίησης “gt”, δηλαδή greater than. Τα υπόλοιπα πεδία αντιπροσωπεύουν τις εναπομείναντες πληροφορίες του feed καθώς και αναρτημένα σχόλια που περιγράφουν τους σκοπούς της εργασίας.

Κεφάλαιο 5

Αξιολόγηση

5.1 Τρόπος λήψης μετρήσεων	48
5.2 Αποτελέσματα και Συμπεράσματα	49

5.1 Τρόπος λήψης μετρήσεων

Για την αξιολόγηση της παρούσας Ατομικής Διπλωματικής Εργασίας έγιναν διάφορες παρουσιάσεις σε φοιτητές πληροφορικής αλλά και σε κόσμο που δεν έχει κάποια σχέση με το αντικείμενο έτσι ώστε να συλλέξουμε μια πιο αντικειμενική και σφαιρική εικόνα για το πώς βλέπει κάποιος ουδέτερος τα πλεονεκτήματα-μειονεκτήματα του συστήματος αλλά και την χρησιμότητα του ως προς την κοινωνία.

Οι παρουσιάσεις ήταν πολύ σύντομες αφού οι συσκευές ήταν τοποθετημένες και έτοιμες από πριν έτσι γινόταν μια σύντομη επεξήγηση για το πώς εξάγονταν τα αποτελέσματα που αναρτιόνταν στο διαδίκτυο και κάποια λόγια γύρω από την θεωρία των χρησιμοποιούμενων τεχνολογιών. Ακολούθως οι επισκέπτες συμπλήρωναν σχετικό online ερωτηματολόγιο που περιελάμβανε ερωτήσεις επιλογής, ερωτήσεις με βαθμολογία από 1 μέχρι 5 και τέλος ζητείτο η άποψη των ερωτηθέντων για να συμπληρώσουν σε μορφή κειμένου. Οι χρήστες απαντούσαν σε ερωτήσεις που είχαν να κάνουν με τις παραμέτρους και την απόδοση του συστήματος αλλά και σε πιο προσωπικά ερωτήματα που στηρίζονταν στο πως οι ίδιοι το βλέπουν σαν υποτιθέμενοι πιθανοί χρήστες του.

Να σημειώσουμε ότι κατά την διάρκεια των παρουσιάσεων δεν γινόταν καμία αναφορά για τα προβλήματα που διαπιστώθηκαν και για τα κενά του συστήματος αφού θα επηρεαζόταν η άποψη των ερωτηθέντων και πιθανός να είχαμε απαντήσεις βασιζόμενες γύρω από τα συμπεράσματα που εξήγαγε ο υποφαινόμενος και όχι οι ίδιοι. Βλέποντας τις απαντήσεις φαίνεται ότι πολλοί αντιλήφθηκαν και είχαν σωστές απόψεις γύρω από αυτό που είδαν και κλίθηκαν να κρίνουν.

5.2 Αποτελέσματα και Συμπεράσματα

Η έρευνα διεξήχθη με μοιρασμένο αριθμό αγοριών και κοριτσιών που κατά το πλείστον περιλαμβάνονταν στην τάξη ηλικιών 23-29 και έφτασε τον συνολικό αριθμό των 37 ατόμων. Το μεγαλύτερο ποσοστό από αυτούς ήταν φοιτητές που πρώτη φορά έβλεπαν τις χρησιμοποιούμενες συσκευές (στην έρευνα η σχετική ερώτηση αναφέρεται στις συσκευές Plogg). Αν και οι περισσότεροι είδαν θερμά την ιδέα της ελεγχόμενης κατανάλωσης ενέργειας που αναπτύξαμε φαίνεται ότι ο κόπος του να θέτουν σε λειτουργία και να ενημερώνονται για τις μετρήσεις τους ώθησε να απαντήσουν με μεγαλύτερο ποσοστό ότι σε περίπτωση που είχαν στην διάθεση τους το σύστημα θα το χρησιμοποιούσαν περίπου 1 φορά την εβδομάδα. Οι περισσότεροι ενθουσιασμένοι απάντησαν από 2 μέχρι 3 φορές, οι λιγότεροι θερμοί επέλεξαν την επιλογή του “σπάνια” ενώ κανένας δεν θα το χρησιμοποιούσε κάθε μέρα γεγονός που προβληματίζει.

Μια πρώτη απάντηση στον πιο πάνω προβληματισμό παίρνουμε βλέποντας την βαθμολογία στο ερώτημα για τον βαθμό που τους ενοχλεί να υπάρχει ένας μεγάλος αριθμός από ενεργοποιημένες συσκευές Plogg στο σπίτι όπου 2 στους 3 επέλεξαν από 4 αστεράκια και πάνω φαινόμενο αρκετά αναμενόμενο και φυσιολογικό. Προχωρώντας στις ερωτήσεις για τις δυνατότητες του συστήματος τα αποτελέσματα ήταν ενθαρρυντικά αφού αν εξαιρέσουμε την επιλογή “ευκολία εγκατάστασης” σε όλες τις άλλες οι χρήστες έμειναν αρκετά ευχαριστημένοι. Οι επιλογές αυτές είχαν να κάνουν με την αξιοπιστία, την ταχύτητα-ανταπόκριση, την ευκολία χρήσης καθώς και τις διεπιφάνειες που προσφέρονταν για παρουσίαση των τελικών αποτελεσμάτων. Οι γραφικές παραστάσεις χαρακτηρίστηκαν αρκετά κατανοητικές και κατατοπιστικές, το σύστημα φαίνεται να άρεσε και να φάνηκε χρήσιμο στο κοινό αλλά από την άλλη πολλοί λίγοι θα πλήρωναν για να το έχουν και αυτό είναι το δεύτερο πρόβλημα που πρέπει να σημειωθεί.

Όσον αφορά τα τελευταία ερωτήματα επιλογής φαίνεται ότι η ενημέρωση για τις σπατάλες ενέργειας θα έκανε πολλούς να αλλάξουν συμπεριφορά και εκφράζεται η πίστη ότι με την βοήθεια του συστήματος οι επόμενοι λογαριασμοί ρεύματος θα έρχονταν πολλοί χαμηλότεροι. Την συμπεριφορά αυτή την είχαμε τονίσει επανειλημμένα και προηγουμένως αφού όπως είχαμε αναφερθεί είχε αποδειχτεί μέσω έρευνας η συγκεκριμένη αλλαγή στάσης και είναι ένα ιδιαίτερα θετικό στοιχείο για την ωφελιμότητα χρήσης του υφιστάμενου αλλά και γενικά παρόμοιων συστημάτων ενημέρωσης.

Τα τελευταία και σημαντικότερα ερωτήματα ζητούσαν από το κοινό να δώσει τις δικές του εισηγήσεις για α) μια πιθανή πρόσθεση περισσότερων λειτουργικοτήτων στο σύστημα και β) γενικά για την βελτίωση του συστήματος σε θέματα απόδοσης, δομής κτλ. Επίσης μπορούσαν να επισημάνουν προβλήματα που εντόπισαν αν και πολλά από αυτά απαντούντων έμμεσα σε προηγούμενα ερωτήματα. Όσον αφορά το πρώτο κομμάτι δεν υπήρχαν ισχυρές απόψεις ή σύγκλιση απόψεων για κάτι συγκεκριμένο αφού ο ρόλος του συστήματος κατά γενική ομολογία έχει κατά ένα μεγάλο ποσοστό επιτευχθεί και σε συνάρτηση με τις λειτουργίες SMS και Email alert άφηναν άπαντες ικανοποιημένους. Τα προβλήματα που καταγράφηκαν συνοπτικά αφορούν το κόστος εγκατάστασης ενός τέτοιου συστήματος, το πλήθος των συσκευών που απαιτούνται και την απαραίτητη σύνδεση στο διαδίκτυο. Αυτά αναλύονται σε περισσότερο βάθος στο επόμενο κεφάλαιο(Κεφ. 6) και αποτυπώνεται το γιατί οι ανησυχίες του κοινού κρίνονται σε γενικές γραμμές δικαιολογημένες.

Σαν τελικό σχόλιο από την συμπλήρωση του ερωτηματολογίου μπορούμε να αναφέρουμε την ικανοποίηση του κοινού για την ιδέα και υλοποίηση που του παρουσιάστηκε, για πολλούς χαρακτηρίστηκε κάτι το πρωτοποριακό που θα το πρότειναν σε γνωστούς και φίλους αλλά το γενικό συμπέρασμα είναι ότι κάτι τέτοιο δεν θα μπορούσε να επιβιώσει εύκολα στις σκληρές συνθήκες της αγοράς αφού για τους ερωτηθέντες πάσχει σε σοβαρούς τομείς.

Κεφάλαιο 6

Τελικά Συμπεράσματα και Μελλοντική Εργασία

6.1 Γενικές Παρατηρήσεις	51
6.1.1 Γνώσεις και εμπειρίες που αποκτήθηκαν	53
6.2 Μελλοντική Εργασία	53

6.1 Γενικές Παρατηρήσεις

Το ολοκληρωμένο σύστημα καταγραφής κατανάλωσης ενέργειας αποτελεί ένα αξιόπιστο μέσο ενημέρωσης του κοινού σε ένα τομέα που ποτέ προηγουμένως δεν μπορούσε να έχει πρόσβαση. Η εργασία που εκπονήθηκε είχε ως στόχο από την μια την δημιουργία ενός λειτουργικού συστήματος που θα μπορούσε να χρησιμοποιηθεί και στην καθημερινή ζωή αλλά από την άλλη περιελάμβανε λειτουργίες που σκοπό είχαν την γνωριμία με ενσωματωμένα συστήματα όπως το Arduino και την αλληλεπίδραση τους με τις υπόλοιπες συσκευές που χρησιμοποιήθηκαν. Σαν γενικό σχόλιο θα λέγαμε ότι η όλη ιδέα για εκμαθησιακούς σκοπούς μπορεί να χαρακτηριστεί ιδανική αφού όπως θα εξηγηθεί και στην επόμενη ενότητα ήταν η αφορμή για να έχουμε μια πολύπλευρη μόρφωση γύρω από προσωπικά δίκτυα, αισθητήρες(ρεύματος στην προκειμένη περίπτωση), λειτουργίες ενσωματωμένων συστημάτων και συνδυασμού όλων αυτών. Κρίνοντας όμως με την λογική της αγοράς θα λέγαμε ότι δύσκολα θα μπορούσε να πλασαριστεί ως προϊόν για εμπορικούς σκοπούς αφού πάσχει σε κάποιους βασικούς τομείς που έχουν να κάνουν με τον τρόπο λειτουργίας του. Καταρχάς το κόστος, που είναι και το πρώτο που θα δει κάποιος πιθανός αγοραστής, αποτελεί αποτρεπτικό παράγοντα:

Υλικό(ανά κομμάτι)	Τιμή(euro)
Arduino board	37
Συσκευές Plogg	75
Telegesis USB stick	29
Xbee module	18
Ethernet shield	36

Πίνακας 6.1 Κόστος υλικών

Προχωρώντας σε κάποιους πρόχειρους υπολογισμούς, για την δημιουργία δικτύου για καταμέτρηση κατανάλωσης σε 5-10 συσκευές σε ένα σπίτι χρειαζόμαστε 375-750 euro μόνο για τα Ploggs + 29 euro για το Telegesis USB stick. Το σύστημα μας αφού προνοεί 2 Arduino boards έχουμε κόστος ακόμα 74 euro για αυτά + 36 euro για τα 2 XBee modules + 36 euro για το Ethernet shield που χρειάζεται το 1 από τα 2 = 146 euro. Άρα κάποιος θα πρέπει να επιβαρυνθεί ένα κόστος από 550 μέχρι 925 euro που στις μέρες μας ίσως να αντιστοιχεί σε ένα ολόκληρο μηνιαίο μισθό. Εδώ να σημειώσουμε ότι δεν λάβαμε υπόψη το κόστος ηλεκτροδότησης που απαιτεί ένα τέτοιο σύστημα για να τίθεται σε λειτουργία καθώς και μικρότερα κόστη που αφορούν σύρματα Ethernet και USB, battery packs για τα ηλεκτροδότηση των Arduino κτλ.

Επόμενο εμπόδιο στην αγορά και εγκατάσταση ενός τέτοιου συστήματος είναι η μεγάλη ποσότητα υλικών που απαιτούνται αφού αν θα θέλαμε να το έχουμε σε μια κατοικία θα ήταν δύσκολο και πολύ άκομφο να υπάρχουν 10 με 15 συσκευές διασκορπισμένες στο χώρο από την στιγμή που ήδη τα περισσότερα σπίτια έχουν εκατοντάδες άλλες συσκευές εγκατεστημένες. Άρα καταλήγουμε στο συμπέρασμα ότι θα ήταν δύσκολο να πειστεί κάποιος να γεμίσει το σπίτι του συσκευές και αν αυτό γινόταν ίσως να υπήρχε κάποια ανασφάλεια αφού πρόκειται για ηλεκτρικές συσκευές ενεργοποιημένες στον χώρο που ζει καθημερινά αυτός και τα παιδιά του. Εδώ να πούμε ότι ανάλογο σύστημα θα μπορούσε να χτιστεί χωρίς την χρήση των Arduino αλλά και πάλι αυτό δεν παύει να το καθιστά αρκετά βαρετό από συσκευές και μη ευέλικτο. Άρα αν αποκλείσουμε την χρήση του σε κατοικίες το μόνο που απομένει είναι μεγάλες επιχειρήσεις, εργοστάσια, εμπορικοί χώροι που η έκταση των υποστατικών είναι αρκετά μεγάλη και η παρουσία ενός τέτοιου αριθμού συσκευών μπορεί να περάσει απαρατήρητη.

Για την χρήση του συστήματος απαιτούνται να πληρούνται κάποιες συνθήκες που το καθιστούν δύσκολο στην εγκατάσταση και στην λειτουργία. Καταρχάς για να συλλέγονται τα δεδομένα από τις συσκευές Plogg θα πρέπει να τρέχει ο κώδικας Java σε μια μηχανή ή φορητό υπολογιστή. Αν αυτό ακούγεται εφικτό για όλους αφού όλοι έχουν έναν υπολογιστή στο σπίτι, η απαίτηση σύνδεσης στο διαδίκτυο είναι ένα σημαντικό πρόβλημα αφού σε πολλές περιπτώσεις αυτό δεν είναι πραγματοποιήσιμο ή αν είναι έρχεται να προστεθεί στα κόστη που έχουμε ήδη υπολογίσει. Επίσης θα λέγαμε ότι η εικόνα ενός υπολογιστή που είναι ενεργοποιημένος και στέλλει πληροφορίες στο διαδίκτυο 24 ώρες δεν μπορεί να ικανοποιεί

ούτε να προσελκύσει εύκολα ενδιαφερόμενους αγοραστές και αυτό σίγουρα αποτελεί ένα μεγάλο μείον.

6.1.1 Γνώσεις και εμπειρίες που αποκτήθηκαν

Στα πλαίσια της παρούσας Ατομικής Διπλωματικής Εργασίας μελετήθηκαν οι βασικές αρχές των ZigBee-based και 802.15.4-based δικτύων που έχουν να κάνουν με τον τρόπο λειτουργίας, τις παραμέτρους που τα διέπουν, τις τοπολογίες δικτύων που δημιουργούν και αποκρυσταλλώθηκε η εικόνα δημιουργίας προσωπικών δικτύων χαμηλής ισχύος σε περιορισμένο χώρο. Για να γίνει αυτό εφικτό μελετήθηκε εις βάθος ο τρόπος λειτουργίας ενσωματωμένων συστημάτων (Arduino boards) που απαιτούσαν καταρχάς σωστή συνδεσμολογία ανάμεσα σε αυτά και τις βοηθητικές τους συσκευές και ακολούθως τον κατάλληλο κώδικα που κάνει πράξη τις απαιτήσεις που είχαν οριστεί. Έτσι για την ολοκλήρωση του όλου project είχαμε την ευκαιρία διατριβής τόσο σε hardware όσο και σε software υλοποίηση γεγονός που βοήθησε σε μια πιο σφαιρική εκμάθηση των συσκευών που είχαμε στην κατοχή μας όπως επίσης και του τρόπου συνεργασίας μεταξύ των συσκευών αυτών για εξαγωγή του τελικού αποτελέσματος. Η δυσκολία σε πολλές φάσεις ήταν ο συνδυασμός hardware με software κομματιού όπου όταν είχαμε κάποιες αλλαγές στο ένα από τα δύο αυτό προϋπέθετε αλλαγές και στο άλλο. Εκτός από τα Arduino boards είχαμε την ευκαιρία χρησιμοποίησης των συσκευών Plogg που μέσα από τη λειτουργία τους δίνουν στον χρήστη την δυνατότητα δημιουργίας διαφόρων συστημάτων βασισμένων στην ιδέα της ενημέρωσης για την κατανάλωση ενέργειας που με την προσθήκη περιφερειακών και άλλων λειτουργιών μπορούν να συγκαταλέγονται στα συστήματα που υλοποιούν την ιδέα του Web of Things.

6.2 Μελλοντική Εργασία

Η ενότητα αυτή παρουσιάζει 2 πτυχές που θα μπορούσαν να αναπτυχθούν καλύτερα στο μέλλον. Από την μια έχουμε την ανάγκη ενός καλύτερου, πιο ευέλικτου και προσαρμόσιμου συστήματος που θα στοιχίζει λιγότερα και τους τρόπους που μπορεί να γίνει αυτό κατορθωτό και από την άλλη τον ορισμό παρόμοιας εργασίας που θα μπορεί να εμπεριέχει ακόμα περισσότερες πηγές λήψης μετρήσεων και θα δίνει στον χρήστη την δυνατότητα εκμάθησης και τριβής με άλλους αισθητήρες. Άρα από την μια έχουμε είτε την καλύτερευση του υπάρχοντος συστήματος είτε τον ορισμό πανομοιότυπου με αυτό που θα καταγράφει

αποτελέσματα που δεν έχουν να κάνουν μόνο με το ρεύμα αλλά και με άλλες μορφές ενέργειας(θερμότητα, υγρασία κτλ) και έχει ως περισσότερο βάση στο να μάθει ο χρήστης-δημιουργός και να εμπλακεί περισσότερο στον χώρο των δικτύων αισθητήρων.

Ξεκινώντας από το πρώτο θα λέγαμε ότι με βάση τα προβλήματα που αναλύθηκαν πιο πάνω το μόνο που φαντάζει βελτιώσιμο είναι η αποφυγή χρήσης της μηχανής που τρέχει τα Java drivers και η ταυτόχρονη μη χρησιμοποίηση του Telegesis USB stick. Αυτό προϋποθέτει την απευθείας επικοινωνία των Arduino boards με τις συσκευές Plogg σε ένα κοινό ZigBee-based δίκτυο και απαιτεί μια λύση όπως έχει αναλυθεί στην Δεύτερη Εναλλακτική Υλοποίηση(Κεφ.3 Ενот. 3.4) όπου η αδυναμία εξεύρεσης του σωστού καναλιού που κάνουν operate οι συσκευές Plogg μας εμποδίζει στο να προχωρήσουμε. Σε μια μελλοντική εργασία θα ήταν καλό να αναλυθεί το γιατί το scan που επιχειρούν τα Arduino boards στα διαθέσιμα κανάλια του δικτύου δεν τα κάνει να επιλέγουν το κανάλι που θέλουμε αλλά επιλέγουν διαφορετικά κανάλια είτε μένουν στο default κανάλι 0.

Όσον αφορά το δεύτερο κομμάτι που έχουμε αναφερθεί θα λέγαμε ότι για εκμαθησιακούς σκοπούς περισσότερο θα μπορούσαμε να δημιουργήσουμε ένα πολύ πιο μεγάλο σύστημα καταγραφής μετρήσεων από πολλαπλές πηγές που θα συνδύαζε αισθητήρες με Arduino boards και θα παρουσίαζε σε διάφορες παραστάσεις και σχέδια ένα φάσμα αποτελεσμάτων. Αυτό θα έδινε την δυνατότητα ολοκληρωμένης ενημέρωσης και θα κινούσε το ενδιαφέρον σε ένα μεγαλύτερο κομμάτι του κοινού αφού θα περιελάμβανε ποικιλία πληροφοριών. Φυσικά όμως κάτι τέτοιο δεν θα ήταν προς πώληση αλλά θα μπορούσε να αξιοποιηθεί στα πλαίσια της έρευνας και ανάπτυξης που επιχειρεί ένα Πανεπιστήμιο.

Βιβλιογραφία

Βιβλία

- [1] J.Schiller, Mobile Communications, Second Edition, Addison-Wesley 2003
- [2] W.Stallings, Wireless Communications and Networks, Prentice Hall, Second Edition, 2002
- [3] Computer Networking: A Top-Down Approach Featuring the Internet, 5th Edition, 2010 by Kurose and Ross

Ιστοσελίδες

- [4] Wikipedia the free encyclopedia
<http://www.wikipedia.org/>
- [5] Arduino Website
<http://www.arduino.cc/>
- [6] Plogg Website
<http://www.plogg.co.uk/>
- [7] Digi Website
<http://www.digi.com/>
- [8] The Java tutorials
<http://docs.oracle.com/javase/tutorial/>
- [9] Ploggs Commands
<http://www.webofthings.org/2008/12/23/are-you-energy-efficient/>
- [10] Ethernet shield – Xbee shield connection
<http://www.sindono.com/stacking-an-arduino-ethernet-shield-and-an-xbee-shield/>

Παράρτημα Α

Ερωτηματολόγιο που δόθηκε για αξιολόγηση του συστήματος Evaluation of Energy Consumption System

Vasiliou Vasilis

Undergraduate student

Department of Computer Science

University of Cyprus

1. Gender

- ☐ Male
- ☐ Female

2. Age

- ☐ 12-16
- ☐ 17-22
- ☐ 23-29
- ☐ 30-36
- ☐ 37-45
- ☐ Over 45

3. Academic Status

- ☐ None
- ☐ Secondary Education
- ☐ Higher Education
- ☐ Master
- ☐ Ph.D

4. Previous working experience with a Plogg device

- ☐ None
- ☐ Only heard about it
- ☐ Bought it and have been using it since then
- ☐ Using it to accomplish an assigned project

5. Frequency which would you use the system

- ☐ Not interested
- ☐ Rarely

- Once a week
- 2-3 times a week
- Every day

6. Evaluation of the idea of getting informed about energy consumption of your house by a website



7. Level of discomfort to have multiple electrical appliances plugged-in with Plogg devices



8. Website's graph understanding



9. System's parameters

--- Reliability	
--- Quickness/Responsiveness	
--- Safety	
--- Ease of setup	
--- Ease of use	
--- Interfaces	

10. Your opinions and impressions about the system

--- Did you like it?	
--- Did you find it usefull?	
--- Would you pay to have it?	

--- Would you change attribute about energy consumption if you were informed about the main sources tha cause it?



--- Would you call it a “completed” system?



--- Do you believe after using the system you could have electricity bills with a lower cost?



11. Recommend the system to others

- ☐ No
- ☐ Maybe
- ☐ Yes
- ☐ Certainly Yes

12. The system except from the information collected in the website offers functionalities as SMS & Email alert. Would you consider any alternative functionalities adding?

13. If you had the power what would you improve to make system better?

14. Did you notify any problems of the system that you want to mention?

Thanks for completing our Survey!

<Finish Survey>

Παράρτημα Β

Κώδικας Arduino

a) Arduino sender

```
/*  
Name : Vasiliou Vasilis  
ID : 1011971  
Semester : 8th  
Year : 2011-2012  
Project : Plogg Measurements  
*/  
  
byte incomingByte;  
  
void setup()  
{  
  
Serial.begin(9600);  
  
}  
  
void loop()  
{  
if (Serial.available() > 0)  
{  
    // read the incoming byte:  
  
    incomingByte = Serial.read();  
  
    Serial.print(incomingByte);  
}
```

```
}
```

```
}
```

b) Arduino receiver

```
/*
```

```
Name : Vasiliou Vasilis
```

```
ID : 1011971
```

```
Semester : 8th
```

```
Year : 2011-2012
```

```
Project : Plogg Measurements
```

```
*/
```

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
#define PACHUBE_FEED_ID 43522    /* Feed Id */
```

```
#define PACHUBE_API_KEY
```

```
"KzMNlS3IljmgIHv5vFZgwKBmflLuGdueI_yP3lpAOjyWEImYdka8nCiWytyhwC_UNir  
CQ4OH8tDxYA6drlq2uAe5_LkUY6yGe0DxsIOExJrcesajL1Gv-tCSwxCUe9Eb" /* API  
Key */
```

```
#define PACHUBE_UPDATE_INTERVAL 1000    /* Delay between updates,  
milliseconds */
```

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };    /* Controller's MAC  
Address */
```

```
byte ip[] = { 192, 168, 10, 1 };    /* IP Address for Arduino */
```

```
byte gateway[] = { 192, 168, 10, 254 };    /* IP Address of the Gateway or Router */
```

```
byte subnet[] = { 255, 255, 255, 0 };    /* IP Address of the Subnet */
```

```
byte server[] = { 173, 203, 98, 29 };    /* www.pachube.com */
```

```
byte incomingByte; /* Byte coming wirelessly */
```

```

Client client(server, 80);    /* Client Listening on Port 80 */

String pachubeString=""; /* Construct String Using Test Data */

/* Setup Function */

void setup()
{
  Serial.begin(9600);          /* Initialise Serial Line on Arduino */
  Ethernet.begin(mac, ip, gateway, subnet); /* Initialise Ethernet Settings */
}

/* Loop Function */

void loop()
{
  if (Serial.available() > 0)
  {
    // read the incoming byte:

    incomingByte = Serial.read(); /* Read */

    if (incomingByte == '!') /* Check if string is completed */
    {
      Serial.println("WAITING...");
      //delay(PACHUBE_UPDATE_INTERVAL); /* Provide Delay Between Updates
      */

      if (!client.connected())
      {
        Serial.println("CONNECTING...");
        client.connect(); /* Connect to Server */
      }
    }
  }
}

```

```

if (client.connected())
{
    Serial.println("SENDING...");
    sending();      /* Do PUT if Connection is Present */
    Serial.print("DATA SENT: ");
    Serial.println(pachubeString);
    pachubeString=""; /* Clean String */
}
else
{
    Serial.println("CONNECTION FAILED...");
}

if (client.available())
{
    Serial.println("READING...");
    while (client.available())    /* When Buffer has Data */
    {
        char c = client.read();    /* Read byte */
        Serial.print(c);          /* Output Byte */
    }
}

if (!client.connected())
{
    Serial.println();
    Serial.println("DISCONNECTING...");
    client.stop();                /* Stop Client */
}

else{
    pachubeString+=incomingByte; /* Create String */
}

```

```

    }
}

/* Build PUT Request and Send Data */

void sending()
{
    client.print("PUT /api/feeds/");
    client.print(PACHUBE_FEED_ID);
    client.println(".csv HTTP/1.0");    /* Update Using csv Format */
    client.println("Host: www.pachube.com");
    client.print("X-PachubeApiKey: ");
    client.println(PACHUBE_API_KEY);
    client.print("Content-Length: ");
    client.println(pachubeString.length(), DEC);    /* Calculate length of Request */
    client.println("Content-Type: text/csv");
    client.println("Connection: close");
    client.println();
    client.println(pachubeString);    /* Output Test Data */
}

```