

Ασύρματη επικοινωνία πομπών Xbee μέσω Arduino



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ασύρματη επικοινωνία πομπών Xbee μέσω Arduino

Ολέξι Ο. Νοβίτσκι

Εισηγητής: Κύριος Ιωάννης Έλληνας, Καθηγητής

Ασύρματη επικοινωνία πομπών Xbee μέσω Arduino

**ΑΘΗΝΑ
ΣΕΠΤΕΜΒΡΙΟΣ 2018**

(Κενό φύλλο)

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ασύρματη επικοινωνία πομπών Xbee μέσω Arduino
Ολέξι Ο. Νοβίτσκι
Α.Μ. 43341**

Εισηγητής:

Κύριος Ιωάννης Έλληνας, Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης:

(Κενό φύλλο)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Νοβίτσκι Ολέξι, του Όλεγκ, με αριθμό μητρώου 43341 φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού δμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Πρωτίστως θα ήθελα να ευχαριστήσω την οικογένεια μου για την συμπαράσταση σε όλα μου τα βήματα καθώς και την μεγάλη υποστήριξη και κατανόηση που έδειξαν τα χρόνια των σπουδών μου.

Θα θελα επίσης να ευχαριστήσω θερμά τον καθηγητή μου κύριο Ιωάννη Έλληνα κυρίως για την εμπιστοσύνη που μου έδειξε, και την υπομονή που έκανε κατά τη διάρκεια υλοποίησης της πτυχιακής εργασίας. Όπως επίσης και για την πολύτιμη βοήθεια και καθοδήγηση του, για την επίλυση διαφόρων θεμάτων.

Ακόμα θέλω να ευχαριστήσω τους φίλους και συναδέλφους μου που ήταν κοντά μου κάθε φορά που χρειάστηκε να λείπω από την εργασία μου για θέματα της σχολής.

Τέλος θέλω να ευχαριστήσω όλους εκείνους τους υπέροχους ανθρώπους που είχα την ευκαιρία να γνωρίσω κατά την διάρκεια της φοίτησής μου, που με στήριξαν και με βοήθησαν να φτάσω μέχρι το τέλος και να μην τα παρατήσω.

ΠΕΡΙΛΗΨΗ

Η ασύρματη επικοινωνία έχει γίνει πλέον μέρος της καθημερινότητας μας εξυπηρετώντας ολοένα και περισσότερο την κάθε μας ανάγκη. Όσο όμως αυξάνεται η τεχνολογική εξέλιξη τόσο αυξάνονται και οι ανάγκες μας. Στην πτυχιακή μας εργασία παρουσιάζεται ένας απλός τρόπος επικοινωνίας μεταξύ οθόνης και πληκτρολογίου με χρήση ασύρματης επικοινωνίας. Χάρη στην δυνατότητα που προσφέρει το Xbee που χρησιμοποιήθηκε στην εργασία αυτή και το Arduino καταφέραμε να επιτύχουμε το σκοπό αυτό. Η εργασία έχει ως στόχο να αναδείξει την χρησιμότητα της ασύρματης επικοινωνίας, τις απεριόριστες χρήσεις της και φυσικά τον τρόπο με τον οποίο μπορεί ο οποιοσδήποτε να κάνει μια κατασκευή ανάλογα με τις ανάγκες του είτε είναι για παράδειγμα μια κατασκευή αυτοματισμού είτε ασύρματη λήψη πληροφοριών εξ αποστάσεως. Αναλύονται σε βάθος οι έννοιες όπως είναι το Arduino το Rf module και τα πρωτοκόλλα ασύρματης επικοινωνίας γενικότερα. Γίνεται και μια πρώτη επαφή με το λογισμικό Xctu καθώς είναι απαραίτητο για τον προγραμματισμό των πομπών που προανέφερα. Τέλος να προσθέσω ότι στην εργασία αυτή γίνεται εκτενής αναφορά στον εξοπλισμό που θα χρειαστεί κάποιος για την κατασκευή, την αξία του κατά προσέγγιση και ενδεχομένως τον χρόνο που θα πρέπει να αφιερώσει κάποιος για να την επιτύχει.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Ασύρματη επικοινωνία

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Πομπός, πλακέτα, οθόνη ,πληκτρολόγιο, προγραμματισμός

ABSTRACT

The wireless communication has now become a vital part of our daily routine supporting more and more every kind of our needs. However as the technological development progresses our needs are growing as well. Our graduate thesis presents a simple way of communication between a monitor and a keyboard using wireless connection. Thanks to the capability that Xbee offers that has been utilized in this thesis plus the Arduino managed to accomplish that goal. This thesis intends to highlight the usefulness of the wireless communication, its unlimited sorts of use and definitely the way in which everyone can make a creation depending on their needs either it is for example an automation system or a wireless mass download of information from a certain distance. Terms like Arduino, the Rf module and the protocols of the wireless connection in general are being analyzed in depth. There is also an initial contact with the Xctu software as it is obligatory for the programming of the transmitters that have been already mentioned. Lastly, it is remarkable to mention that an extended report of the appropriate equipment is being carried out about the construction, the total value approximately and possibly the time that someone has to devote in order to achieve such a task.

Scientific Field: Wireless communication

Keywords: Arduino, Xbee, Rf module ,lcd, keyboard, Xctu

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ	12
1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας	12
1.2 Ιστορική αναδρομή	12
2. Modules/ πρωτόκολλα ασύρματης επικοινωνίας	14
2.1 Πρώτη επαφή με τα RF modules.....	14
2.2 Πρωτόκολλα ασύρματης επικοινωνίας.....	15
2.3 Τα RF modules και τα χαρακτηριστικά τους.....	22
3. ARDUINO	29
3.1 Γενικά	29
3.2 Μέρη ενός Arduino.....	31
3.3 Σειριακή Θύρα.....	32
3.4 Χαρακτηριστικά του Arduino.....	32
3.5 Βασικές μνήμες.....	33
3.6 Τροφοδοσία.....	33
3.7 Επικοινωνία.....	34
3.8 Γλώσσα Προγραμματισμού/εντολές.....	34
4. ΕΡΓΑΣΙΑ	39
4.1 Το κομμάτι της κατασκευής.....	39
4.2 Το κομμάτι του προγραμματισμού.....	41
5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ	52

4.1 Σύνοψη της πτυχιακής εργασίας.....	52
4.2 Προοπτικές.....	52
6. ΒΙΒΛΙΟΓΡΑΦΙΑ.....	53

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.1: Ένα RF module.....	14
Εικόνα 2.2: Το module XBee 802.15.4 (Series 1).....	25
Εικόνα 2.3: Το module XBee 802.15.4 (Series 2).....	27
Εικόνα 2.4: Το module Xbee-PRO XSC.....	28
Εικόνα 2.5: Το module Xbee-(PRO) ZB SMT.....	29
Εικόνα 3.1: Το λογότυπο Arduino.....	31
Εικόνα 3.2: Μέρη της πλατφόρμας Arduino Uno.....	33
Εικόνα 4.1: Arduino Wireless Proto Shield.....	40
Εικόνα 4.2: Keyboard(τύπου ps2).....	40
Εικόνα 4.3: Arduino LCD Display Module(blue).....	41
Εικόνα 4.4: Funduino Uno.....	41
Εικόνα 4.5: Jumper wires.....	41
Εικόνα 4.6: Αρχική οθόνη Xctu.....	42
Εικόνα 4.7: Βίντεο για την αρχικοποίηση του Xbee.....	43
Εικόνα 4.8: Σύνδεση πλακέτας Arduino και Xbee.....	43
Εικόνα 4.9: Εμφάνιση των 2 Xbee στο λογισμικό Xctu.....	44
Εικόνα 4.10: Παραμετροποίηση του Xbee.....	44
Εικόνα 4.11: Εμφάνιση των 2 Xbee σε ξεχωριστά παράθυρα.....	45
Εικόνα 4.12: Αρχική οθόνη του προγράμματος Arduino.....	46
Εικόνα 4.13: Τρόπος σύνδεσης του πληκτρολογίου και Arduino.....	47
Εικόνα 4.14: Τρόπος σύνδεσης της οθόνης Icd και Arduino.....	48

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

RF --> Radio Frequency

BPSK --> Binary Phase Shift Keying

QPSK --> Quadrature Phase Shift Keying

FFD --> Function Devices

RFD --> Reduced Function Devices

CAP --> Contention Access Period

PAN --> Private Area Network

WPAN --> Wide Private Area Network

ZDO --> Zigbee Device Objects

IDE --> Integrated Development Environment

GNU --> General Public License (GNU GPL or GPL)

LCD --> Liquid Crystal Display

USB --> Universal Serial Bus

LED --> Light Emitting Diode

DC--> Direct Current

GND --> Ground

TXD --> Transmit Data

RXD --> Receive Data

M/F --> Male/Female

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Η εξέλιξη της τεχνολογίας παρέχει την δυνατότητα για ασύρματη επικοινωνία και δικτύωση πολλών συσκευών. Μια δυνατότητα που πια έχει γίνει απαραίτητη για την ευκολία της καθημερινότητας των χρηστών και της εγκατάστασης συστημάτων. Για αυτούς τους λόγους όταν αναπτύσσετε ένα ηλεκτρονικό σύστημα από έναν ηλεκτρονικό μηχανικό για να κάνει κάποια διαδικασία είτε απλή είτε πολυσύνθετη συχνά συμπεριλαμβάνεται στη σχεδίαση και ένα κομμάτι ασύρματης επικοινωνίας. Για να σχεδιαστεί όμως και να υλοποιηθεί σαν ηλεκτρονικό σύστημα το κομμάτι της ασύρματης επικοινωνίας δεν είναι τόσο απλό. Απαιτείται μεγάλη εξειδίκευση και εμπειρία από κάποιον για να φέρει εις πέρας το κομμάτι αυτό. Επίσης υπάρχουν πολλοί παράγοντες που μπορούν να κάνουν την σχεδίαση – υλοποίηση του ακόμα και από εξειδικευμένο σχεδιαστή μη αποδοτική και να μην επιφέρει το ζητούμενο αποτέλεσμα. Άλλο ένα πρόβλημα είναι ότι σε κάποιες εφαρμογές το τηλεπικοινωνιακό κομμάτι των ηλεκτρονικών συστημάτων απαιτείται να είναι διαπιστευμένο από τους ανάλογους φορείς.

Στα προβλήματα αυτά έχουν καταφέρει να δώσουν λύσεις πληθώρα εταιρειών ανάπτυξης ηλεκτρονικών συστημάτων. Οι εταιρίες αυτές παράγουν έτοιμες τηλεπικοινωνιακές μονάδες (modules), τις οποίες οι σχεδιαστές συστημάτων μπορούν να τις αφομοιώσουν πολύ εύκολα στα συστήματά τους και να ξεπεράσουν έτσι όλα τα προβλήματα που θα είχαν σε περίπτωση που σχεδίαζαν και υλοποιούσαν εξ' αρχής μια τέτοια μονάδα.

1.1 Περιγραφή του αντικείμενου της πτυχιακής εργασίας

Στα πλαίσια της πτυχιακής αυτής εργασίας έγινε προσπάθεια επικοινωνίας δυο πλακετών Arduino με την βοήθεια ασύρματης επικοινωνίας δυο πομπών Xbee. Με σκοπό την ασύρματη ανταλλαγή μηνυμάτων που στέλνονται με την βοήθεια κοινού πληκτρολογίου από την μια πλακέτα στην άλλη και εμφάνιση των μηνυμάτων αυτών σε μια οθόνη lcd.

1.2 Ιστορική αναδρομή

Οι ασύρματες τηλεπικοινωνίες χαρακτηρίζονται από μια μακρόβια και αδιάκοπη ιστορική εξέλιξη, ενώ στις μέρες μας ο τομέας αυτός υφίσταται ιστορικά, τη ταχύτερη ανάπτυξη του. Η αλήθεια όμως είναι πως, η εξέλιξη των ασυρμάτων τηλεπικοινωνιών συνέβη με αργό ρυθμό και ήταν πάντοτε εξαρτημένη με τις τεχνολογικές εξελίξεις της κάθε εποχής. Συναντάται πιθανώς μια δυσκολία όσον αφορά στην ακριβή περιγραφή αυτής της εξέλιξης, ωστόσο θα ασχοληθούμε με μια σειρά από γεγονότα που αποτέλεσαν σταθμό σε αυτή τη διαδικασία. Αρχικά, ο Maxwell διατύπωσε μαθηματικά, τα ηλεκτρομαγνητικά κύματα, ενώ στη συνέχεια σειρά ο Hertz επιβεβαίωσε πειραματικά την ύπαρξη τους. Για τον λόγο αυτό ο Hertz θεωρείται ο θεμελιωτής των ασυρμάτων επικοινωνιών ενώ είναι χαρακτηριστικό ότι τα ραδιοκύματα ονομαστήκαν προς τιμήν του, Hertztian κύματα. Η συνέχεια αυτής της εξέλιξης σηματοδοτείται από τον Guglielmo Marconi, ο οποίος περίπου το 1894, μετά από μια σειρά πειραμάτων και προσπαθειών, καταφέρνει να μεταδώσει σήματα σε απόσταση 2 χιλιομέτρων. Είναι δε αξιοθαύμαστο πως, συνεχίζοντας τις προσπάθειες του ο Ιταλός, επιτυγχάνει,

6 χρόνια αργότερα, ασύρματη επικοινωνία σε πολύ μεγαλύτερη απόσταση και συγκεκριμένα στα 130 χιλιόμετρα. Μικρά βήματα προόδου παρατηρούνται πλέον όλο και συχνότερα και το ενδιαφέρον για ανάπτυξη απασχολεί όλο και συχνότερα την επιστημονική κοινότητα. Στις ΗΠΑ το 1934, 194 ραδιοσυστήματα της δημοτικής αστυνομίας και 58 αστυνομικοί σταθμοί είχαν υιοθετήσει κινητά επικοινωνιακά συστήματα με διαμόρφωση πλάτους (AM) για το καλό της δημόσιας ασφάλειας. Το 1935 για πρώτη φορά έχουμε εμφάνιση της διαμόρφωσης συχνότητας (FM) από τον Edwin Armstrong, μιας τεχνικής η οποία θα αποτελούσε τη βασική τεχνική διαμόρφωσης στο εξής. Ο δεύτερος παγκόσμιος πόλεμος επιτάχυνε τις εξελίξεις και την εμφάνιση καινοτομιών και έτσι καταλήξαμε σε ένα σημείο όπου ο αριθμός των κινούμενων χρηστών αυξήθηκε από μερικές χιλιάδες το 1940 σε περίπου 1,2 εκατομμύρια το 1962, πάντα έχοντας ως σημείο αναφοράς τις ΗΠΑ. Το 1946 ενσωματώθηκε για πρώτη φορά σε 25 πόλεις των ΗΠΑ, η δημόσια υπηρεσία κινητής τηλεφωνίας (public mobile telephone service). Τα αρχικά συστήματα χρησιμοποιούσαν έναν κεντρικό πομπό για τη κάλυψη μιας ολόκληρης μητροπολιτικής περιοχής. Η ανεπαρκής χρήση του ραδιοφάσματος περιόριζε τη χωρητικότητα του συστήματος, ενώ είναι χαρακτηριστικό το γεγονός ότι, 30 χρόνια μετά την εισαγωγή της υπηρεσίας της κινητής τηλεφωνίας, το σύστημα της Νέας Υόρκης (με εν δυνάμει αγορά 10 εκατομμυρίων χρηστών) διέθετε μόνο 12 κανάλια και μπορούσε να εξυπηρετήσει μόνο 543 πελάτες (με ουρά αναμονής, για την υπηρεσία, 3.700 πελατών το 1976). Στην πραγματικότητα, η δυνατότητα παροχής ασυρμάτων επικοινωνιών σε μια ευρύτατη μερίδα πληθυσμού δε θα γινόταν εφικτή πριν τις δεκαετίες 1960-1970. Κατά την διάρκεια αυτών των δεκαετιών, τα εργαστήρια BELL καταφέρνουν να αναπτύξουν την ιδέα της κυψελωτής δομής, σύμφωνα με την οποία μια περιοχή κάλυψης διασπάται σε μικρά κελιά με σκοπό την επαναχρησιμοποίηση των συχνοτήτων και τη βελτίωση της χρήσης του ραδιοφάσματος. Η ιδέα αυτή έπαιξε σημαντικό ρόλο στην λύση των προβλημάτων χωρητικότητας και αν και σαν ιδέα ήταν γενικώς ολοκληρωμένη στα τέλη του 1960, πιστοποιήθηκε εν τέλει από την FCC (Federal Communications Commission-Ομοσπονδιακή Επιτροπή Επικοινωνιών των ΗΠΑ) το 1982. Το 1983 τοποθετήθηκε το πρώτο αναλογικό κυψελωτό σύστημα στο Σικάγο, το οποίο μέχρι το 1984 καταστάθηκε πλήρες. Η τρομακτική ζήτηση επέφερε γρήγορες εξελίξεις, εκπλήσσοντας τους πάντες (χαρακτηριστικά, η AT&T είχε διεξάγει έρευνα αγοράς προτού εισαχθεί αυτή η νέα τεχνολογία και είχε εσφαλμένα προβλέψει πως μόνο προνομιούχοι οικονομικά πολίτες θα έδειχναν ενδιαφέρον). Αποτέλεσμα αυτών των απαιτήσεων της αγοράς ήταν και η δημιουργία της «ψηφιακής κυψελωτής τεχνολογίας» (digital cellular system hardware) με όφελος την αυξημένη χωρητικότητα, η οποία εισήχθη για πρώτη φορά το 1991 σε κύριες μητροπόλεις των ΗΠΑ. Η δεύτερη γενιά των κυψελωτών συστημάτων αναπτύχθηκε συνεπώς στις αρχές του 1990 και βασίστηκε στις ψηφιακές επικοινωνίες. Η μετάβαση από την αναλογική στην ψηφιακή εποχή είχε ως αποτέλεσμα υψηλότερη χωρητικότητα, σαφώς χαμηλότερο κόστος και καλύτερη ταχύτητα. Παρόλο που, αρχικά, η δεύτερη γενιά των κυψελωτών συστημάτων προσέφερε ως επί το πλείστον υπηρεσίες φωνής, αυτά τα συστήματα κατάφεραν σταδιακά να υποστηρίξουν και υπηρεσίες δεδομένων όπως e-mail, πρόσβαση στο internet και ανταλλαγή μικρών μηνυμάτων. Δυστυχώς, η ταχεία ανάπτυξη της αγοράς είχε ως αποτέλεσμα την ταχεία ανάπτυξη προτύπων για τα συστήματα δεύτερης γενιάς, όλα διαφορετικά μεταξύ τους. Διαφορετικά πρότυπα χρησιμοποιούνταν στην Ευρώπη, στην Ιαπωνία, ακόμη και στις ΗΠΑ, όλα ασύμβατα μεταξύ τους. Επιπλέον μερικές χώρες είχαν προχωρήσει σε υιοθέτηση συστημάτων 3ης γενιάς επίσης με πολλαπλά πρότυπα.

ΚΕΦΑΛΑΙΟ 2

Modules/ πρωτόκολλα ασύρματης επικοινωνίας

2.1 Πρώτη επαφή με τα RF modules

Ένα RF module (radio frequency module, μονάδα ραδιοσυχνοτήτων) είναι μια μικρή ηλεκτρονική μονάδα που χρησιμοποιείται για την επικοινωνία δυο ηλεκτρονικών συσκευών μέσω ραδιοσυχνοτήτων. Εκτός από τις ραδιοσυχνότητες υπάρχει και ο τρόπος επικοινωνίας μέσω οπτικών σημάτων αλλά στη περίπτωση αυτή πρέπει να υπάρχει οπτική επαφή μεταξύ των 2 συσκευών και γι' αυτό τον λόγο είναι περισσότερο διαδεδομένη η επικοινωνία μέσω ραδιοσημάτων. Τα RF modules χρησιμοποιούνται κυρίως στις ηλεκτρονικές συσκευές μιας και παρέχουν ευκολίες κατά την σχεδίαση, όσον αφορά το κομμάτι της επικοινωνίας.



Εικόνα 2.1: Ένα RF module

Το να σχεδιαστεί ένα σύστημα επικοινωνίας (module) είναι κάτι που δεν είναι τόσο απλό, χρειάζεται αρκετή προσοχή ώστε να λειτουργεί σε συγκεκριμένες συχνότητες, να μην επηρεάζεται και να μην παρεμποδίζει άλλα συστήματα και να είναι πιστοποιημένο από τους αρμόδιους φορείς επικοινωνιών. Όλα τα παραπάνω κάνουν την σχεδίαση του τμήματος επικοινωνίας της ηλεκτρονικής συσκευής αρκετά δύσκολη, χρονοβόρα και με μεγάλο κόστος, οπότε σε αρκετές περιπτώσεις μπορεί να προτιμηθεί μια έτοιμη λύση από κάποιο RF module που υπάρχει ήδη στην αγορά και σε αρκετά χαμηλή τιμή. Ηλεκτρονικές συσκευές όπου χρησιμοποιούν τέτοιες μονάδες επικοινωνίας είναι αρκετές, κάποιες από αυτές είναι: ασύρματος έλεγχος για γκαραζόπορτες, ασύρματα συστήματα ασφαλείας, ασύρματος έλεγχος διαφόρων συσκευών μηχανών που χρησιμοποιούνται στη βιομηχανία, δίκτυα αισθητήρων, αυτοματισμοί σπιτιών. Στα RF modules συναντώνται τα εξής είδη, οι πομποί (transmitter modules), οι δέκτες (receiver modules) και οι πομποδέκτες (transceiver modules), καθώς και υπάρχουν και τα ολοκληρωμένα modules (system on a chip modules). Οι πομποί παίρνουν τα δεδομένα εσωτερικά από τον μικροελεγκτή της συσκευής (η μονάδα που είναι υπεύθυνη για την όλη λειτουργία της συσκευής) και αφού τα διαμορφώσουν τα εκπέμπουν στην απαιτούμενη συχνότητα. Οι δέκτες λαμβάνουν τα δεδομένα, τα αποδιαμορφώνουν και τα στέλνουν εσωτερικά στον μικροελεγκτή της μονάδας. Οι πομποδέκτες έχουν την δυνατότητα να εκπέμπουν αλλά και να λαμβάνουν διαμορφωμένα σήματα – δεδομένα. Τέλος, τα ολοκληρωμένα modules έχουν την ικανότητα, χάρη στον ενσωματωμένο μικροελεγκτή που έχουν και

των ειδικών λειτουργιών που μπορεί να εκτελεί αυτός, να κάνουν και διαχείριση των δεδομένων ακλουθώντας κάποιο πρωτόκολλο επικοινωνίας, ούτως ώστε ο κεντρικός μικροελεγκτής του συστήματος να μην επιβαρύνεται και με αυτή την εργασία. Οι πιο διαδεδομένες συχνότητες λειτουργίας των RF modules είναι 433.92 MHz, 315 MHz, 868 MHz, 915 MHz και 2400 MHz. Οι συχνότητες αυτές ανήκουν στην ζώνη industrial, scientific and medical radio bands (ISM), η οποία είναι μια ζώνη συχνοτήτων ελεύθερη για επικοινωνίες που δεν χρειάζεται κάποια άδεια από τον υπεύθυνο φορέα. Τα RF modules συχνά επικοινωνούν με κάποιο μικροελεγκτή ή μικροεπεξεργαστή που είναι στη συσκευή. Τα πρωτόκολλα επικοινωνίας που χρησιμοποιούν για να επικοινωνήσουν με τον μικροελεγκτή – μικροεπεξεργαστή είναι αυτό της σειριακής επικοινωνίας (UART), της επικοινωνίας για X-Bee modules, της επικοινωνίας serial peripheral interface (SPI) και της universal serial bus (USB) επικοινωνίας. Τέλος υπάρχει περίπτωση το RF module να διαθέτει το δικό του πρωτόκολλο επικοινωνίας. Τα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται για την ασύρματη επικοινωνία των RF modules μεταξύ τους, είναι το Zigbee, το Bluetooth low energy, το Wi-Fi και το IEEE 802.15.4. Η απόδοση ενός RF module εξαρτάται από κάποιους παράγοντες. Ορισμένοι από αυτούς έχουν να κάνουν με την ισχύ εκπομπής σε σχέση με την κατανάλωση ενέργειας και τις πιθανές παρεμβολές σε άλλα συστήματα που μπορεί να προκύψουν, όταν πρόκειται για πομπό. Όταν πρόκειται για δέκτη, αυξάνοντας την ευαισθησία του μπορεί να υπάρξει καλύτερη επικοινωνία αλλά μπορεί να προκύψουν παρεμβολές από άλλες συσκευές. Για να περιοριστούν αυτοί οι παράγοντες που δημιουργούν προβλήματα στην επικοινωνία, χρησιμοποιούνται διάφορες τεχνικές όπως είναι σύμφωνες κεραίες σε πομπό και δέκτη. Εξαιτίας αυτών των παραγόντων, κυρίως στη πράξη, η απόδοση ενός RF module θα είναι πάντα χαμηλότερη σε σχέση με την εργοστασιακά αναφερόμενη.

Η σύνδεση ενός RF module με την κεντρική πλακέτα της συσκευής μπορεί να επιτευχθεί είτε με τεχνολογία ακροδεκτών (through hole) είτε με τεχνολογία επιφανειακής στήριξης (surface mount). Η τεχνολογία ακροδεκτών παρέχει την δυνατότητα εύκολης σύνδεσης και αποσύνδεσης του RF module από την κεντρική πλακέτα αλλά τις περισσότερες φορές αυτά τα modules είναι μεγαλύτερα σε μέγεθος σε σχέση με τα modules επιφανειακής στήριξης, γεγονός το οποίο κάποιες φορές είναι αρνητικό αφού ζητούμενο είναι η συσκευή να έχει το ελάχιστο δυνατό μέγεθος.

2.2 Πρωτόκολλα ασύρματης επικοινωνίας

2.2.1 Το πρωτόκολλο επικοινωνίας IEEE 802.15.4

Η ZigBee Alliance είναι μια ομοσπονδία εταιριών που συνεργάζονται για να αναπτύξουν πρότυπα (και προϊόντα) για αξιόπιστη, μικρού κόστους, χαμηλής ισχύος ασύρματη δικτύωση. Η τεχνολογία ZigBee ίσως θα εμπεριέχεται σε ένα μεγάλο εύρος από προϊόντα και εφαρμογές σε καταναλωτικές, εμπορικές και βιομηχανικές αγορές σε όλο τον κόσμο. Το ZigBee χτίζεται πάνω στο πρότυπο IEEE 802.15.4 που καθορίζει το φυσικό και το MAC επίπεδο, το οποίο είναι υπεύθυνο για την παροχή διευθύνσεων και καναλιών επικοινωνίας, προσωπικών δικτύων, χαμηλού κόστους και χαμηλού ρυθμού μετάδοσης. Το ZigBee καθορίζει τις προδιαγραφές του επιπέδου δικτύου για τοπολογίες δικτύου αστέρα, δέντρου και σημείου προς σημείο (peer – to – peer - τοπολογία σύμφωνα με την οποία όλοι οι κόμβοι του δικτύου έχουν ίδια δικαιώματα) και παρέχει ένα πλαίσιο για προγραμματισμό εφαρμογών στο επίπεδο εφαρμογών. Το πρωτόκολλο IEEE 802.15.4 προσδιορίζει τα χαρακτηριστικά του φυσικού και του MAC επιπέδου για Χαμηλού Ρυθμού Ασύρματα Προσωπικά Δίκτυα (LRWPAN). Τα πλεονεκτήματα ενός LR-WPAN είναι ευκολία στην εγκατάσταση,

ασφαλής μεταφορά δεδομένων, μικρής εμβέλειας λειτουργία, πολύ χαμηλό κόστος και μεγάλη διάρκεια ζωής μπαταρίας, ενώ ταυτόχρονα διατηρεί μία απλή και ευέλικτη στοίβα πρωτοκόλλων. Δίνεται μεγάλη προσοχή στην χαμηλή κόστους επικοινωνία μεταξύ κοντινών συσκευών με καμία ή ελάχιστη σχετική υποδομή, έχοντας στόχο την εκμετάλλευση της χαμηλής κατανάλωσης πολύ περισσότερο. Το φυσικό επίπεδο προσφέρει την υπηρεσία μετάδοσης δεδομένων καθώς και τη διεπαφή για την διαχείριση του επιπέδου αυτού, η οποία δίνει πρόσβαση σε όλες τις λειτουργίες διαχείρισης του και κρατάει μία βάση δεδομένων με πληροφορίες που έχουν σχέση με δίκτυα προσωπικής περιοχής. Διαθέτει τρεις μπάντες συχνοτήτων: μία μπάντα στα 2450 MHz (με 16 κανάλια), μία μπάντα στα 915 MHz (με 10 κανάλια) και μία μπάντα στα 868 MHz (1 κανάλι), με όλα να χρησιμοποιούν τον τρόπο εισαγωγής, Απευθείας Ακολουθίας Διασποράς Φάσματος (Direct Sequence Spread Spectrum – DSSS). Η μπάντα των 2450 MHz χρησιμοποιεί Offset Quadrature Phase Shift Keying (O-QPSK) για κωδικοποίηση ενώ οι 868 - 915 MHz μπάντες βασίζονται σε Binary Phase Shift Keying (BPSK). Εκτός από την λειτουργία εκπομπής ή μη εκπομπής, το φυσικό επίπεδο υποστηρίζει λειτουργίες για επιλογή καναλιών, υπολογισμό της ποιότητας της σύνδεσης, ανίχνευση μετρήσεων ενέργειας και εκτίμηση ελευθέρου καναλιού. Το MAC επίπεδο ξεκινά την μετάδοση των πλαισίων MAC μέσω της χρήσης ενός φυσικού καναλιού. Εκτός από την υπηρεσία δεδομένων, προσφέρει μία διασύνδεση χειρισμού και το ίδιο το επίπεδο χειρίζεται την πρόσβαση στο φυσικό κανάλι. Επίσης, διαθέτει έλεγχο πλαισίων, εγγυάται φορτοθυρίδες και διαχειρίζεται συσχετίσεις κόμβων. Δύο είναι οι τύποι των κόμβων: Συσκευές Μειωμένης Λειτουργίας (Reduced Function Devices – RFDs) και Συσκευές Πλήρους Λειτουργίας (Full Function Devices – FFDs). Οι FFD είναι διαθέτουν ένα πλήρες σετ από λειτουργίες επιπέδου MAC, το οποίο τους προσφέρει την ικανότητα να συμπεριφέρονται σαν συντονιστές δικτύου ή σαν τελικές συσκευές δικτύου. Όταν λειτουργούν σαν συντονιστές δικτύου, οι FFD στέλνουν φάρους – σήματα (beacons) που προσφέρουν συγχρονισμό και υπηρεσίες επικοινωνίας και συνδέσμου των δικτύων. Οι RFD λειτουργούν μόνο σαν τερματικές συσκευές και είναι εξοπλισμένες με αισθητήρες - ενεργοποιητές όπως μεταδότες, διακόπτες φωτός, λάμπες κτλ. Μπορούν να αλληλεπιδρούν μόνο με μία μοναδική FFD. Δύο κύριοι τύποι από τοπολογίες δικτύου μπορούν να υλοποιηθούν στο IEEE802.15.4, η τοπολογία αστέρα και η τοπολογία peer-to-peer. Ωστόσο, κάθε δίκτυο χρειάζεται έστω και έναν κόμβο FFD για να δουλεύει ως ο συντονιστής του δικτύου. Έτσι τα δίκτυα δημιουργούνται από ομάδες συσκευών που χωρίζονται από κατάλληλες αποστάσεις. Κάθε συσκευή έχει ένα συγκεκριμένο 64-bit αναγνωριστικό και αν τηρούνται ορισμένες προδιαγραφές, μπορεί να χρησιμοποιηθεί και αναγνωριστικό των 16-bit. Όπως για παράδειγμα, σε κάθε τομέα PAN, χρησιμοποιείται αναγνωριστικό των 16-bit. Στην τοπολογία αστέρα υιοθετείται ένα μοντέλο δικτύου αφέντη – σκλάβου (master – slave). Ένας FFD αναλαμβάνει τον ρόλο του συντονιστή και οι υπόλοιποι κόμβοι μπορούν να είναι RFDs ή FFDs και θα επικοινωνούν μόνο με τον συντονιστή. Στην τοπολογία peer-to-peer, ένας FFD μπορεί να μιλάει σε άλλους FFD μέσα στην εμβέλεια εκπομπής του και μπορεί να αναμεταδίδει μηνύματα σε άλλους FFDs εκτός της εμβέλειας εκπομπής του μέσω ενός ενδιάμεσου FFD, σχηματίζοντας ένα multihop δίκτυο (δίκτυο στο οποίο υπάρχουν ενδιάμεσοι κόμβοι για την μετάδοση της πληροφορίας). Ένας PAN συντονιστής επιλέγεται να διαχειρίζεται την λειτουργία του δικτύου. Ο PAN συντονιστής μπορεί να λειτουργεί το PAN του με ένα υπερπλαίσιο (superframe) ή χωρίς. Στην πρώτη περίπτωση το υπερπλαίσιο ξεκινάει με ένα σήμα beacon που εξυπηρετεί για λόγους συγχρονισμού, όπως επίσης και για να προσδιορίσει τη δομή του υπερπλαισίου και να δώσει πληροφορίες ελέγχου στο PAN. Το υπερπλαίσιο διαιρείται σε μία ενεργή και σε

μία ανενεργή ποσότητα (όπου ο PAN συντονιστής μπορεί να περάσει σε κατάσταση αναμονής και να κρατήσει ενέργεια). Η ενεργή ποσότητα διαιρείται σε σταθερού μεγέθους θυρίδες και περιέχει μία Περίοδο Ανταγωνισμού Πρόσβασης (Contention Access Period – CAP), όπου οι κόμβοι συναγωνίζονται για πρόσβαση στο κανάλι χρησιμοποιώντας ένα πρωτόκολλο με χρονοθυρίδες CSMA-CA, και μια Περίοδο Ελεύθερη Ανταγωνισμού (Contention Free Period – CFP), όπου οι κόμβοι μεταδίδουν δίχως να ανταγωνίζονται για το κανάλι στις Εγγυημένες Χρονοθυρίδες (Guaranteed Time Slots – GTS). Η ανάθεση και η επίβλεψη επιτυγχάνεται με τον PAN συντονιστή. Όταν μια τερματική συσκευή χρειάζεται να δώσει δεδομένα σε έναν συντονιστή (μη GTS) πρέπει να αναμένει το σήμα beacon να συγχρονιστεί και αργότερα να ανταγωνιστεί για πρόσβαση στο κανάλι. Από την άλλη, η επικοινωνία από έναν συντονιστή σε μια τερματική συσκευή είναι έμμεση. Ο συντονιστής αποθηκεύει το μήνυμα και ανακοινώνει την εκκρεμή παράδοση σε ένα beacon. Οι τερματικές συσκευές κυρίως βρίσκονται σε κατάσταση αναμονής τον περισσότερο χρόνο και ενεργοποιούνται περιοδικά για να δουν εάν έχουν να λάβουν μηνύματα από τον συντονιστή περιμένοντας το σήμα beacon. Όταν παρατηρήσουν ότι ένα μήνυμα είναι διαθέσιμο, το ζητούν συγκεκριμένα κατά την διάρκεια του CAP. Όταν ένας συντονιστής θέλει να μιλήσει σε άλλο συντονιστή πρέπει να συγχρονιστεί με το beacon του και να ενεργήσει σαν τερματική μονάδα. Η άλλη επιλογή για επικοινωνία στο PAN είναι χωρίς υπερπλάισιο. Ο PAN συντονιστής δεν στέλνει ποτέ beacons και η επικοινωνία πραγματοποιείται σε βάση ενός CSMA-CA χωρίς θυρίδες. Ο συντονιστής είναι συνέχεια ανοιχτός και έτοιμος να λάβει δεδομένα από μια τερματική συσκευή, ενώ η μεταφορά δεδομένων σε άλλη περίπτωση είναι βασισμένη σε λειτουργία συνεχόμενου ελέγχου (poll – based). Η τερματική συσκευή ξυπνάει περιοδικά και σφυγμομετρεί τον συντονιστή για εκκρεμή μηνύματα. Ο συντονιστής μετά στέλνει αυτά τα μηνύματα ή σήματα ότι δεν υπάρχει διαθέσιμο μήνυμα. Η επικοινωνία από συντονιστή σε συντονιστή δεν αποτελεί πρόβλημα εφόσον και οι δύο κόμβοι είναι ανοιχτοί όλη την ώρα. Εκτός από την μεταφορά δεδομένων, το MAC επίπεδο προσφέρει σάρωση καναλιών και λειτουργικότητες συσχέτισης - αποσυσχέτισης. Η διαδικασία σάρωσης περιλαμβάνει σάρωση αρκετών λογικών καναλιών στέλνοντας ένα beacon μήνυμα αίτησης και ακούγοντας (ενεργή σάρωση, για FFDs) ή απλά ακούγοντας (παθητική σάρωση, για RFDs) για beacons έτσι ώστε να εντοπίσουν υπάρχοντα PANs και συντονιστές. Υψηλότερα επίπεδα αποφασίζουν σε ποιο PAN θα συνδεθούν και αργότερα ρωτούν το MAC επίπεδο να ξεκινήσει μία διαδικασία συσχέτισης για το επιλεγμένο PAN. Αυτό εμπεριέχει το να στείλει μία αίτηση σε ένα συντονιστή και να περιμένει το αντίστοιχο μήνυμα αποδοχής. Εάν γίνει δεκτός στο PAN, ο κόμβος λαμβάνει μία 16-bit «σύντομη» διεύθυνση που μπορεί να χρησιμοποιήσει αργότερα στην θέση της 64-bit «εκτεταμένης» IEEE διεύθυνσης.

2.2.2 Πρωτόκολλο ZigBee.

Το ZigBee αποτελεί ένα πρωτόκολλο για μια μορφή υψηλού επιπέδου επικοινωνίας. Χρησιμοποιεί μικρούς και χαμηλής ισχύος πομπούς που βασίζονται στο πρότυπο IEEE 802 και χρησιμοποιούνται σε δίκτυα προσωπικού χώρου (Private Area Network, PAN). Η ονομασία του παραπέμπει στο χορό των μελισσών κατά την επιστροφή τους στην κυψέλη. Οι συσκευές που χρησιμοποιούν αυτόν τον τρόπο επικοινωνίας μπορούν να είναι ασύρματοι διακόπτες φωτός, ηλεκτρονικές μετρητικές συσκευές που χρησιμοποιούνται στο εσωτερικό οικιών και άλλοι βιομηχανικοί εξοπλισμοί που απαιτούν ασύρματη μεταφορά δεδομένων μικρής εμβέλειας. Η τεχνολογία που ορίζεται από τις προδιαγραφές του ZigBee σχεδιάστηκε με σκοπό να

είναι απλούστερη και λιγότερο δαπανηρή από ότι οι τεχνολογίες άλλων ασύρματων δικτύων προσωπικού χώρου (Wide Private Area Network, WPAN), όπως η τεχνολογία Bluetooth για παράδειγμα. Το ZigBee απευθύνεται σε εφαρμογές που χρησιμοποιούν ραδιοσυχνότητες και απαιτούν ένα χαμηλό ποσοστό ανταλλαγής δεδομένων, μεγάλη διάρκεια ζωής της μπαταρίας και ασφαλή δικτύωση. Έχει μια καθορισμένη ταχύτητα της τάξης των 250 kbit/s που είναι κατάλληλη για περιοδική επικοινωνία ή αποστολή ενός μοναδικού σήματος από έναν αισθητήρα ή μια συσκευή εισόδου. Το Πρωτόκολλο ZigBee χρησιμοποιεί τα υψηλότερα επίπεδα από το πρωτόκολλο δικτύωσης υπολογιστών. Το επίπεδο δικτύου (Network Layer – NWK) είναι υπεύθυνο για να οργανώνει και να παρέχει δρομολόγηση σε ένα multihop δίκτυο, χτισμένο πάνω σε IEEE 802.15.4 λειτουργία, ενώ το Επίπεδο Εφαρμογών (Application Layer – APL) έχει σαν στόχο να παρέχει ένα πλαίσιο για διαχείριση και επικοινωνία των συσκευών, μέσω εφαρμογών – τρόπων λειτουργίας που παρέχει ο κατασκευαστής. Το APL περιλαμβάνει το Πλαίσιο Εφαρμογών (Application Framework), τις συσκευές Zigbee (Zigbee Device Objects – ZDO) και το Υπό- επίπεδο εφαρμογών (Application Sub Layer – APS). Το Πλαίσιο Εφαρμογών μπορεί να έχει μέχρι και 240 εφαρμογές, δηλαδή εφαρμογές για επικοινωνία διαφόρων module που προσδιορίζονται από τον χρήστη και είναι μέρος ενός συστήματος Zigbee. Το ZDO παρέχει υπηρεσίες που επιτρέπουν στα APOs να ανακαλύψουν το ένα το άλλο και να οργανωθούν σε μια διαμοιρασμένη εφαρμογή. Το APS προσφέρει μια διεπαφή στα δεδομένα και τις υπηρεσίες ασφαλείας στα APOs και το ZDO.

Το Zigbee αναγνωρίζει τρεις τύπους συσκευών. Μία τερματική συσκευή Zigbee που αντιστοιχεί σε έναν IEEE RFD ή FFD και ενεργεί σαν μία απλή συσκευή. Ένα δρομολογητή Zigbee (router) που είναι ένας FFD με δυνατότητες δρομολόγησης και το συντονιστή Zigbee (ένας μοναδικός στο δίκτυο) που είναι ένας FFD που διαχειρίζεται ολόκληρο το δίκτυο.

Εκτός από την τοπολογία αστέρα (που εντάσσεται φυσικά στην αντίστοιχη τοπολογία στο IEEE 802.15.4), το επίπεδο δικτύου του Zigbee επίσης υποστηρίζει και πιο σύνθετες τοπολογίες όπως του δέντρου και του πλέγματος (peer to peer).

Ανάμεσα στις λειτουργικότητες που παρέχονται από το επίπεδο δικτύου είναι η multihop δρομολόγηση, η ανακάλυψη της διαδρομής, η συντήρηση, η ασφάλεια και η σύνδεση/αποσύνδεση ενός δικτύου με συνεχόμενες σύντομες 16-bit διευθύνσεις που ανατίθενται στις νέες συνδεδεμένες συσκευές.

Ένα multihop δίκτυο εδραιώνεται μέσω της διαδικασίας σύνδεσης. Όταν μια συσκευή επιθυμεί να συνδεθεί να σε ένα υπάρχον δίκτυο, το επίπεδο δικτύου ζητείται να ξεκινήσει μια διαδικασία αναζήτησης δικτύου. Με υποστήριξη από τη διαδικασία σάρωσης του MAC επιπέδου (επίπεδο δικτύου), μαθαίνει για γειτονικούς δρομολογητές που ανακοινώνουν τα δίκτυα τους. Αφού το ανώτερο επίπεδο έχει αποφασίσει σε ποιο δίκτυο θα συνδεθεί (αρκετά δίκτυα Zigbee μπορεί να επικαλύπτονται χωρικά, χρησιμοποιώντας διαφορετικά κανάλια συχνοτήτων επικοινωνίας), το επίπεδο δικτύου επιλέγει έναν «πατέρα» κόμβο (στο επιθυμητό δίκτυο) από την γειτονιά του και ζητά από το επίπεδο MAC να ξεκινήσει μία διαδικασία συσχέτισης. Με το που λαμβάνει μία ένδειξη της αίτησης συσχέτισης από το MAC επίπεδο, το επίπεδο δικτύου του αναθέτει μια 16-bit μικρού μήκους διεύθυνση και επιτρέπει στο MAC επίπεδο να απαντήσει επιτυχώς στην αίτηση συσχέτισης. Ο κόμβος θα χρησιμοποιεί την μικρού μήκους διεύθυνση για όλη την περαιτέρω επικοινωνία στο δίκτυο.

Οι σχέσεις πατέρα – παιδιού που δημιουργούνται σαν αποτέλεσμα ενώσεων, διαμορφώνουν ολόκληρο το δίκτυο σε μορφή ενός δέντρου με τον συντονιστή Zigbee ως ρίζα του, τους δρομολογητές Zigbee σαν εσωτερικούς κόμβους και οι τερματικές

συσκευές Zigbee σαν φύλλα του. Αυτή η δομή δέντρου βασίζεται επίσης στον διαμοιρασμένο αλγόριθμο ανάθεσης διευθύνσεων δικτύου. Ο αλγόριθμος δρομολόγησης εξαρτάται από την τοπολογία που χρησιμοποιείται στο δίκτυο αισθητήρων. Σε μια τοπολογία δέντρου η δρομολόγηση μπορεί να γίνει μόνο ανάμεσα σε δεσμούς πατέρα – παιδιού που εγκαθίστανται σαν αποτέλεσμα ενεργειών ένωσης (αυτό λέγεται δρομολόγηση «βασισμένη σε δέντρο»). Οι δρομολογητές διατηρούν μόνο τις διευθύνσεις τους και τις πληροφορίες που σχετίζονται με τις διευθύνσεις των παιδιών τους και των πατεράδων τους. Δεδομένου του τρόπου που ανατίθενται οι διευθύνσεις, ένας δρομολογητής που χρειάζεται να προωθήσει ένα μήνυμα μπορεί εύκολα να καθορίσει εάν ο προορισμός ανήκει σε ένα δέντρο με ρίζες σε έναν από τους δρομολογητές παιδιά του ή είναι μια από τις τερματικές συσκευές παιδιά. Εάν αυτό ισχύει, δρομολογεί το πακέτο στο κατάλληλο παιδί, διαφορετικά δρομολογεί το πακέτο στον πατέρα του. Αυτού του είδους ο αλγόριθμος δρομολόγησης δεν είναι απαραίτητα ο πιο ενεργειακά οικονομικός αλλά είναι πολύ απλός στην εφαρμογή του και επιτρέπει στους δρομολογητές να λειτουργούν σε ένα δίκτυο με ενεργοποιημένη τη δυνατότητα beacon. Με άλλα λόγια, όλοι οι δρομολογητές Zigbee (και ο συντονιστής Zigbee) στέλνουν beacons, επικοινωνούν μέσω ενός πρωτοκόλλου CSMA-CA με χρονοθυρίδες και κοιμούνται στον ανενεργό χρόνο. Η κεντρική ιδέα είναι να υπάρχει μικρός ενεργός χρόνος σε σύγκριση με την περίοδο του beacon και οι γειτονικοί δρομολογητές να ξεκινούν την επικοινωνία τους κατάλληλα έτσι ώστε να είναι κατάλληλα μετατοπισμένη χρονικά του ενός σε σχέση με του άλλου ώστε να αποφεύγεται η αλληλοεπικάλυψη. Η επικοινωνία από ένα παιδί στον πατέρα συμβαίνει στο CAP (Contention Access Period) του πατέρα ενώ η επικοινωνία από ένα πατέρα στο παιδί είναι έμμεση. Σε κάθε περίπτωση ένας κόμβος πρέπει να συγχρονίζεται με το beacon του πατέρα για να ανταλλάξει δεδομένα μαζί του, ενώ οδηγεί την επικοινωνία με τα παιδιά του σύμφωνα με τη χρονοθυρίδα του.

Η τοπολογία δικτύου σε πλέγμα είναι πιο πολύπλοκη στη διαχείριση και η αποστολή beacons δεν επιτρέπεται αλλά είναι πιο εύρωστη και ανθεκτική σε σφάλματα. Οι δρομολογητές διατηρούν έναν πίνακα δρομολόγησης (Routing Table – RT) και καλούν έναν αλγόριθμο ανακάλυψης διαδρομής να κατασκευάσει / αναβαθμίσει αυτές τις κατασκευές δεδομένων στους κόμβους διαδρομής. Η αναζήτηση διαδρομής είναι μια διαδικασία που απαιτείται για την καθιέρωση καταχωρήσεων στον πίνακα δρομολόγησης των κόμβων κατά μήκος της διαδρομής ανάμεσα σε δύο κόμβους που επιθυμούν να επικοινωνήσουν. Ένας Πίνακας Αναζήτησης Διαδρομής (Route Discovery Table – RDT) διατηρείται από δρομολογητές και τον συντονιστή για να εφαρμοστεί η αναζήτηση διαδρομής. Η αναζήτηση διαδρομής στο Zigbee βασίζεται στο γνωστό αλγόριθμο δρομολόγησης Ad Hoc On Demand Distance Vector – AODV. Όταν ένας κόμβος χρειάζεται να δρομολογήσει προς ένα συγκεκριμένο προορισμό, μεταδίδει ένα μήνυμα αίτησης διαδρομής (Route Request Message – RREQ) που διαδίδεται σε όλο το δίκτυο μέχρι να φτάσει στον προορισμό. Όσο ταξιδεύει στο δίκτυο, ένα μήνυμα RREQ συσσωρεύει (σε ένα από τα πεδία του) μία τιμή κόστους που είναι το άθροισμα όλων των κοστίων όλων των συνδέσεων που διέσχισε. Το κόστος μιας σύνδεσης μπορεί να ρυθμιστεί σε μία σταθερή τιμή ή να υπολογίζεται δυναμικά βάσει μίας εκτίμησης ποιότητας της σύνδεσης παρεχόμενη από την διεπαφή του IEEE 802.15.4. Το κάθε μήνυμα RREQ κουβαλάει μία τιμή RREQ ID την οποία αυξάνει ο εκκινητής κάθε φορά που στέλνει ένα νέο μήνυμα RREQ. Με αυτόν τον τρόπο η RREQ ID και η διεύθυνση της πηγής μπορούν να χρησιμοποιηθούν σαν μοναδική αναφορά για μία διαδικασία αναζήτησης διαδρομής. Η λήψη ενός RREQ ενεργοποιεί μία έρευνα μέσα

στον RDT για μία καταχώρηση που να ταιριάζει με την αναζήτηση διαδρομής. Εάν δεν βρεθεί αντιπαραβολή, δημιουργείται μία νέα καταχώρηση RDT για την διαδικασία αναζήτησης και ξεκινάει ένας χρονομετρητής αναζήτησης αίτησης (με την λήξη του χρονομετρητή η καταχώρηση του RDT θα αφαιρεθεί). Αντιστρόφως εάν βρεθεί μια καταγραφή στον RDT, ο κόμβος συγκρίνει το κόστος για το μήνυμα RREQ και την αντίστοιχη τιμή στην καταχώρηση του RDT. Εάν η πρώτη είναι μεγαλύτερη πετάει το μήνυμα RREQ, αντίθετα ανανεώνει την καταχώρηση του RDT. Τελικά, εάν ο κόμβος δεν είναι ο προορισμός της αναζήτησης διαδρομής, κατανέμει μία RT καταχώρηση για τον προορισμό, με κατάσταση Αναζήτηση, και αναμεταδίδει το RREQ αφού ανανεώσει το πεδίο του κόστους διαδρομής. Εάν ο κόμβος είναι ο τελικός προορισμός, απαντάει στον δημιουργό με ένα μήνυμα απάντησης διαδρομής (Route Reply – RREP) που ταξιδεύει ανάποδα την διαδρομή.

Το RREP μήνυμα απευθύνεται στον δημιουργό της διαδικασίας αναζήτησης διαδρομής και κουβαλάει μαζί του μία τιμή πεδίου για το υπολειπόμενο κόστος την οποία ο κάθε κόμβος αυξάνει καθώς προωθούν το μήνυμα. Κατά την λήψη ενός μηνύματος απάντησης διαδρομής (RREP), ο κόμβος ανακτά τις καταχωρήσεις RDT και RT για την συσχετισμένη αναζήτηση διαδρομής. Εάν ο κόμβος είναι ο δημιουργός του RREQ και αυτή είναι η πρώτη RREP που λαμβάνει, θέτει την καταχώρηση στον RT σε ενεργός και καταγράφει το υπολειπόμενο κόστος και το επόμενο άλμα στην καταχώρηση του RDT. Σε όλες τις άλλες περιπτώσεις συγκρίνει το υπολειπόμενο κόστος από το RREP με αυτό της καταχώρησης στον RDT. Εάν η πρώτη είναι μεγαλύτερη ο κόμβος απορρίπτει το μήνυμα RREP, διαφορετικά ανανεώνει την καταχώρηση του RDT (υπολειπόμενο κόστος) και την καταχώρηση στον RT (επόμενο άλμα). Ένας κόμβος που δεν είναι ο δημιουργός πρέπει επιπλέον να προωθήσει το RREP προς τον δημιουργό. Να επισημανθεί ότι οι ενδιαμέσοι κόμβοι ποτέ δεν αλλάζουν την καταχώρηση της κατάστασης του RT σε Ενεργός σαν αποτέλεσμα της λήψης ενός μηνύματος RREP. Θα αλλάξουν την κατάσταση της καταχώρησης κατά την λήψη (και δρομολόγηση) ενός μηνύματος δεδομένων για τον δεδομένο προορισμό.

Μία εφαρμογή Zigbee αποτελείται από μία σειρά από Αντικείμενα Εφαρμογών (Application Objects – APOs) διασκορπισμένα σε αρκετούς κόμβους στο δίκτυο. Ένα APO είναι ένα κομμάτι λογισμικού, υλοποιημένο από την ίδια την εταιρία, που ελέγχει μία μονάδα υλικού (αισθητήριο, διακόπτης, λάμπα) διαθέσιμη στην συσκευή. Σε κάθε APO ανατίθεται ένας μοναδικός τοπικά καταληκτικός αριθμός τον οποίο τα άλλα APO μπορούν να χρησιμοποιούν σαν μία επέκταση στην διεύθυνση συσκευής του δικτύου για να αλληλεπιδρούν μαζί του. Το Αντικείμενο Συσκευής Zigbee (Zigbee Device Object – ZDO) είναι ένα ειδικό αντικείμενο που προσφέρει υπηρεσίες στα APO: τους επιτρέπει να ανακαλύπτουν συσκευές στο δίκτυο και την υπηρεσία που εφαρμόζουν. Επίσης παρέχει επικοινωνία, και υπηρεσίες δικτύου και ασφάλειας. Το Υποεπίπεδο Εφαρμογών (Application Sublayer – APS) παρέχει υπηρεσίες μεταφοράς δεδομένων για τα APO και το ZDO. Μία εφαρμογή Zigbee πρέπει να συμμορφώνεται σε ένα υπάρχον (Zigbee Alliance) προφίλ εφαρμογών. Ένα προφίλ εφαρμογών καθορίζει την διάταξη των μηνυμάτων και τα πρωτόκολλα για την αλληλεπίδραση ανάμεσα στα APO που συλλογικά συγκροτούν μία διαμοιρασμένη εφαρμογή. Το πλαίσιο του προφίλ εφαρμογών επιτρέπει στους διάφορους μηχανικούς – εταιρίες να σχεδιάζουν και να πουλάνε συσκευές Zigbee που μπορούν να λειτουργούν ταυτόχρονα το ένα με το άλλο σε ένα δεδομένο προφίλ εφαρμογών. Κάθε APO ενσωματώνει μια σειρά ιδιοτήτων (οντότητες δεδομένων που αντιπροσωπεύουν την εσωτερική κατάσταση, κ.λ.π.) και παρέχει λειτουργίες - υπηρεσίες για τον καθορισμό - ανάκτηση των τιμών αυτών των

ιδιοτήτων ή ειδοποίησης όταν η τιμή κάποιας ιδιότητας αλλάζει. Στο πλαίσιο ενός προφίλ, μία ομάδα σχετικών ιδιοτήτων ονομάζεται «Σύμπλεγμα» («Cluster») και αναγνωρίζεται με έναν αριθμό id. Τυπικά ένα cluster αντιπροσωπεύει ένα είδος διεπαφής (ή μέρος της) του APO στα άλλα APO. Το προφίλ εφαρμογών πρέπει να καθορίζει έναν από τους δύο δυνατούς τύπους υπηρεσίας επικοινωνίας. Για τον τύπο υπηρεσίας Key Value Pair (KVP) το πρότυπο Zigbee έχει προκαθορίσει διατάξεις μηνυμάτων που πρέπει να συμπληρώνονται κατάλληλα από τα APO για να ζητούν μία δεδομένη λειτουργία σε ιδιότητες που βρίσκονται σε ένα απομακρυσμένο APO. Οι αλληλεπιδράσεις μεταξύ των APO περιορίζονται από τις λειτουργίες που ορίζονται από ιδιότητες. Ο τύπος υπηρεσίας Γενικού Μηνύματος (Generic Message) είναι κατάλληλος για εφαρμογές που δεν χωράνε στον τύπο υπηρεσίας του KVP και αφήνει την ευθύνη στο προφίλ εφαρμογών για να καθορίσει τους τύπους μηνύματος και τα περιεχόμενα τους Ένα ειδικό προφίλ εφαρμογών, που ονομάζεται Προφίλ Συσκευής (Device Profile), πρέπει να εφαρμοστεί από όλους τους κόμβους σε ένα δίκτυο Zigbee. Το υπεύθυνο αντικείμενο για αυτό το προφίλ είναι το ZDO. Το Προφίλ Συσκευής απαιτεί τα εκτελεστικά του αντικείμενα (ZDOs) να υποστηρίζουν διαδικασίες αναζήτησης συσκευής/υπηρεσίας όπου ένας κόμβος προσπαθεί να ανακαλύψει υπάρχοντες κόμβους στο δίκτυο, ενεργά τερματικά σημεία σε κάποιο κόμβο και/ή τις υπηρεσίες που εφαρμόζουν (διαθέσιμα cluster ids). Οι διαδικασίες αναζήτησης είναι κρίσιμες στην χορήγηση διευθύνσεων στα APO. Στον άμεσο τρόπο χορήγησης διευθύνσεων ένα μήνυμα απευθύνεται σε μία συγκεκριμένη διεύθυνση προορισμού (διεύθυνση προορισμού 16-bit) και σε έναν αριθμό καταληκτικού σημείου. Ο αποστολέας κόμβος είναι υπεύθυνος να ανακαλύψει και τα δύο δια μέσου των υπηρεσιών αναζήτησης του ZDO. Ο τρόπος έμμεσης χορήγησης διευθύνσεων απαιτεί μόνο τον αποστολέα να παρέχει ένα cluster id αλλά χρειάζεται υποστήριξη από ένα γειτονικό (ή τοπικό) δρομολογητή Zigbee (ή συντονιστή) να εντοπίσει τον κόμβο προορισμού για το μήνυμα. Αυτό είναι εφικτό χάρις στο APS του δρομολογητή Zigbee που διατηρεί έναν δεσμευτικό πίνακα που συσχετίζει διάφορες πληροφορίες όπως πηγαία διεύθυνση, πηγαίο καταληκτικό σημείο, cluster id, με μία λίστα από αντίστοιχες πληροφορίες όπως διεύθυνση προορισμού, καταληκτικό σημείο προορισμού και να υπάρχει ένα APS για κάθε συσκευή που πρέπει να φθάσει το μήνυμα. Ένα μήνυμα που στέλνεται από μία τερματική συσκευή με έμμεση χορήγηση διεύθυνσης φθάνει στον κόμβο πατέρα. Σε αυτή την περίπτωση το APS συμβουλευεται τον δεσμευτικό του πίνακα για να καθορίσει τους πραγματικούς προορισμούς και να τους στείλει κατάλληλα μηνύματα με άμεση χορήγηση διεύθυνσης. Η προσθήκη και η αφαίρεση μιας καταχώρησης στον δεσμευτικό πίνακα διατάσσεται από το ZDO σε απάντηση στις τοπικές - απομακρυσμένες δεσμευτικές αιτήσεις, όπως καθορίζεται από το Προφίλ Συσκευής.

2.3 RF modules και τα χαρακτηριστικά τους

Η οικογένεια Xbee των προϊόντων με ενσωματωμένο RF κύκλωμα παρέχει OEMs (original equipment manufacturer) τα οποία έχουν κοινό τρόπο σύνδεσης (footprint) τον οποίο μοιράζονται πολλές διαφορετικές πλατφόρμες, συμπεριλαμβανομένων των τοπολογιών ZigBee - Mesh αλλά και των δικτύων πολλαπλών σημείων (multipoint), για υλοποιήσεις τόσο στα 2.4GHz όσο και στα 900MHz. Τα OEMs τα οποία εκμεταλλεύονται τις λύσεις Xbee μπορούν να αντικαταστήσουν ένα Xbee με κάποιο άλλο, αναλόγως των απαιτήσεων που θα έχει εκάστοτε εφαρμογή, με ελάχιστο χρηματικό ή χρονικό κόστος ανάπτυξης και μειωμένο ρίσκο.

Τα Xbee RF modules χρησιμοποιούνται για υλοποίηση συνδέσεων πολλαπλών σημείων (multipoint) και είναι ιδανικά για εφαρμογές οι οποίες έχουν απαιτήσεις πολύ μικρής καθυστέρησης και προβλέψιμου χρόνου επικοινωνίας. Παρέχουν γρήγορη και αξιόπιστη επικοινωνία σε τοπολογίες σημείου προς σημείο (point to point), μεταξύ δύο κόμβων (peer to peer) καθώς και σε τοπολογίες πολλαπλών σημείων ή αστέρα (multipoint/star), επιτρέποντας την αξιόπιστη σύνδεση μεταξύ των σημείων του συστήματος με χαρακτηριστική ευκολία. Είτε χρησιμοποιούνται αποκλειστικά για την αντικατάσταση των καλωδίων σε απλές σειριακές επικοινωνίες, είτε ως τμήμα ενός περισσότερο σύνθετου δικτύου αισθητήρων, τα συγκεκριμένα προϊόντα μεγιστοποιούν την ασύρματη απόδοση και την ευκολία της ανάπτυξης και εγκατάστασης του δικτύου.

Χαρακτηριστικά/Πλεονεκτήματα

- 802.15.4 / Multipoint τοπολογίες (πολλαπλών σημείων),
- Συχνότητες 2.4GHz για χρήση σε όλο τον κόσμο (ISM Band),
- Συχνότητες 900MHz για συστήματα μεγάλων αποστάσεων,
- Πλήρως συμβατά με άλλα προϊόντα δικτύωσης Drop-in της εταιρίας (Digi), συσκευών επέκτασης δικτύου κτλ.
- Κοινός τρόπος διασύνδεσης Xbee για πλήθος συσκευών RF.
- Κατάσταση χαμηλής κατανάλωσης ενέργειας (sleep mode)
- Πολλαπλές λύσεις κεραιών
- Βιομηχανικό εύρος θεοκρασιών
- Διαθέσιμες εναλλακτικές επιλογές χαμηλής κατανάλωσης και μικρών αποστάσεων

Τα προϊόντα Xbee OEM RF ανήκουν στην οικογένεια προϊόντων δικτύωσης Drop-in της Digi η οποία περιλαμβάνει λύσεις δικτύωσης από άκρη σε άκρη (end-to-end). Παρέχοντας άψογη διεπαφή ή καλύτερα επικοινωνία με παρόχους δικτύου (gateways), συσκευές δικτύων και συσκευές επέκτασης δικτύων (extenders), δίνουν τη δυνατότητα πολλαπλών λύσεων συνδεσιμότητας με διαφορετικά προϊόντα. Παρακάτω παρουσιάζονται τα χαρακτηριστικά τριών module από τα συγκεκριμένα προϊόντα:

- **Xbee 802.15.4 (Series 1)**

Platform	Xbee 802.15.4 (Series 1)
Performance	
RF Data Rate	250 kbps
Indor/Urban Range	100 ft (30 m)
Outdoor/RF Line-of-Sight Range	300 ft (100 m)
Transmit Power	1mW (+0 dBm)
Receiver Sensitivity (1% PER)	(-)92 dBm
Features	
Serial Data Interface	3.3V CMOS UART
Configuration Method	API or AT Commands, local or over-the-air
Frequency Band	2.4GHz
Einterference Immunity	DSSS (Direct Sequence Spread Spectrum(
Serial Data Rate	1200 bps - 250 kbps
ADC Inputs	(6) 10-bit ADC Inputs
Digital I/O	8
Antenna Options	Chip. Wire Whip, U.FL, &RPSMA
Networking & Security	
Encryption	128-bit AES
Reliable Packet Delivery	Retries/Acknowledgements

Ασύρματη επικοινωνία πομπών Xbee μέσω Arduino

IDs and Channels	PAN ID, 64-bit IEEE MAC
Power Requirements	
Supply Voltage	2.8 - 3.4 VDC
Transmit Current	45 mA @ 3.3VDC
Receive Current	50 mA @ 3.3VDC
Power-Down Current	< 10uA @ 25 oC
Regulatory Approvals	
FCC(USA)	OUR-XBEE
IC(Canada)	4214a-XBEE
ETSI(Europe)	Yes
C-TICK Australia	Yes
Telec(Japan)	Yes



Εικόνα 2.2: Το module Xbee 802.15.4 (Series 1)

Ασύρματη επικοινωνία πομπών Xbee μέσω Arduino

Platform	Xbee 802.15.4 (Series 2)
Performance	
RF Data Rate	250 kbps
Indor/Urban Range	300 ft (100 m)
Outdoor/RF Line-of-Sight Range	1 mi (1.6 km)
Transmit Power	60mW (+18 dBm)
Receiver Sensitivity (1%)	(-)100 dBm
Features	
Serial Data Interface	3.3V CMOS UART
Configuration Method	API or AT Commands, local or over-the-air
Frequency Band	2.4GHz
Einterference Immunity	DSSS (Direct Sequence Spread Spectrum(
Serial Data Rate	1200 bps - 250 kbps
ADC Inputs	(6) 10-bit ADC Inputs
Digital I/O	8
Antenna Options	Chip. Wire Whip, U.FL, &RPSMA
Networking & Security	
Encryption	128-bit AES
Reliable Packet Delivery	Retries/Acknowledgements
IDs and Channels	PAN ID, 64-bit IEEE MAC, 12 Channels
Power Requirements	
Supply Voltage	2.8 - 3.4 VDC
Transmit Current	215 mA @ 3.3VDC
Receive Current	55 mA @ 3.3VDC
Power-Down Current	< 10uA @ 25 oC
Regulatory Approvals	
FCC(USA)	OUR-XBEEPRO
IC(Canada)	4214a-XBEEPRO
ETSI(Europe)	Yes Max TX 10 mW
C-TICK Australia	Yes
Telec(Japan)	Yes

➤ Χαρακτηριστικά Xbee 802.15.4 (Series 2)



Εικόνα 2.3: Το module Xbee 802.15.4 (Series 2)

Platform	Xbee-PRO XSC
Performance	
RF Data Rate	19 kbps / 9.6 kbps
Indor/Urban Range	Up to 1200 ft (370 m)
Outdoor/RF Line-of-Sight Range	up to 6 mi (9.6 km)
Transmit Power	100 mW (+20 dB m)
Receiver Sensitivity (1% PER)	(-)106 dBm
Features	
Serial Data Interface	3.3V CMOS UART (5V Tolerant)
Configuration Method	AT Commands
Frequency Band	902 MHz to 928 MHz
Einterference Immunity	FHSS (Frequency Hoppind Spread Spectrum)
Serial Data Rate	1200 bps - 57.6 kbps
ADC Inputs	None

Digital I/O	None
Antenna Options	Wire Whip, U.FL, RPSMA
Networking & Security	
Encryption	No
Reliable Packet Delivery	Retries/Acknowledgements
IDs and Channels	PAN ID, 32-bit IEEE MAC, 7 Channels
Power Requirements	
Supply Voltage	3.0- 3.6 VDC
Transmit Current	265 mA typical
Receive Current	65 mA typical
Power-Down Current	45 uA pin Sleep
Regulatory Approvals	
FCC(USA)	MCQ-XBEEEXSC
IC(Canada)	1846A-XBEEEXSC
ETSI(Europe)	No
C-TICK Australia	No
Telec(Japan)	No

➤ **Χαρακτηριστικά Xbee-PRO XSC**



Εικόνα 2.4: Το module Xbee-PRO XSC

2.3.1 XBee-PRO ZB (Programmable)

Αξιοποιώντας το σύνολο των χαρακτηριστικών ZigBee PRO, αυτές οι μονάδες γίνονται διαλειτουργικές με άλλες συσκευές ZigBee, συμπεριλαμβανομένου συσκευών από άλλους κατασκευαστές, υπό κάποιες προϋποθέσεις.

Τα προϊόντα της οικογένειας XBee είναι πολύ εύκολα στη χρήση, αφού δεν απαιτούν επιπλέον ρυθμίσεις ή ανάπτυξη. Επιπλέον, οι προγραμματιζόμενες εκδόσεις XBee-PRO ZB, κάνουν απλή και εύκολη την παραμετροποίηση των εφαρμογών ZigBee, αφού προγραμματίζοντας απευθείας τη μονάδα (module) δεν χρειάζεται να χρησιμοποιηθεί ξεχωριστός επεξεργαστής. Ακόμη, λόγω του ότι το πρόγραμμα/κώδικας που εξασφαλίζει την ασύρματη επικοινωνία είναι απομονωμένο, δεν υφίσταται ρίσκο για την ποιότητα ή την ασφάλεια της ασύρματης επικοινωνίας.

2.3.2 XBee ZB SMT (embedded surface mount RF)

Τα XBee και XBee-PRO ZB modules είναι επιφανειακής στήριξης RF modules και παρέχουν αποτελεσματική και οικονομική ασύρματη συνδεσιμότητα σε ηλεκτρονικές συσκευές. Μπορούν να λειτουργήσουν/συνεργαστούν απρόσκοπτα με συσκευές ZigBee PRO με διαφορετικό σύνολο χαρακτηριστικών, συμπεριλαμβανομένων και συσκευών από διαφορετικούς κατασκευαστές, όπως και τα XBee-PRO ZB.

Χρησιμοποιώντας την τεχνολογία επιφανειακής στήριξης (Surface Mount Technology, SMT), τα XBee και ZBee-PRO ZB modules είναι ιδανικά για εφαρμογές που σχετίζονται με τον τομέα της ενέργειας και του ελέγχου όπου η αποτελεσματικότητα μιας συγκεκριμένης υλοποίησης αποτελεί κρίσιμο παράγοντα. Η σειριακή διεπαφή (Serial Peripheral Interface SPI) παρέχει μια διεπαφή υψηλών ταχυτήτων και βελτιστοποιεί την επικοινωνία με ενσωματωμένους μικροελεγκτές, μειώνοντας σημαντικά το κόστος ανάπτυξης όπως και το χρόνο διάθεσης στην αγορά (time to market).

Χαρακτηριστικά/Πλεονεκτήματα

- Λειτουργικότητα με άλλες συσκευές συμβατές με ZigBee
- Οι προγραμματιζόμενες εκδόσεις του XBee-PRO ZB επιτρέπουν την ανάπτυξη εξειδικευμένων εφαρμογών
- Επιλογές παραμετροποίησης μέσω απλών AT ή API εντολών
- 15 I/O θύρες γενικού σκοπού
- Ανανεώσεις του firmware μέσω UART, SPI ή ασύρματα
- Ρεύμα κατάστασης αναμονής (sleep mode) κάτω του 1 μ A.
- Link budget 110 dB για το XBee και 119 dB για το XBee-Pro ZB



Εικόνα 2.5: Το module Xbee-(PRO) ZB SMT

ΚΕΦΑΛΑΙΟ 3

Arduino

3.1 Γενικά

Το Arduino είναι μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα και σχεδιασμού, που βασίζεται σε ευέλικτο και εύκολο στη χρήση υλικό και λογισμικό. Προορίζεται για καλλιτέχνες, σχεδιαστές, υλοποίηση χόμπι και δραστηριοτήτων, και γενικότερα για οποιοδήποτε ενδιαφέρεται να δημιουργήσει αλληλεπιδραστικά αντικείμενα ή περιβάλλοντα. Για να μιλήσουμε λίγο πιο τεχνικά, υπάρχει ένα κύκλωμα που χρησιμοποιεί μικροελεγκτή, το οποίο μας δίνει ένα αριθμό πυλών οι οποίες μπορεί να λειτουργήσουν είτε ως είσοδοι είτε ως έξοδοι στα κυκλώματά μας.

Οι πλατφόρμες Arduino κατασκευάζονται κυρίως από την εταιρία Smart Project. Ωστόσο, το Arduino ξεκίνησε ως έργο προς ανάπτυξη το 2005 στην Ιταλία, στο Ινστιτούτο Αλληλεπίδρασης Σχεδίασης Invea ώστε οι φοιτητές του Ινστιτούτου να αναπτύσσουν ενσωματωμένα συστήματα οικονομικά και αποδοτικά αξιοποιώντας τις δυνατότητες και τις ευκαιρίες που μπορεί να προσφέρει το ελεύθερο λογισμικό.

Γενικότερα, το Arduino θα λέγαμε ότι είναι ένα εργαλείο που μπορούμε να κατασκευάσουμε ένα υπολογιστικό σύστημα με την έννοια ότι αυτό θα ελέγχει συσκευές του φυσικού κόσμου, σε αντίθεση με τον κοινό Ηλεκτρονικό Υπολογιστή. Βασίζεται σε ευέλικτο, εύκολο στη χρήση υλικό και λογισμικό, σε μια αναπτυξιακή πλακέτα που ενσωματώνει επάνω έναν μικροελεγκτή και συνδέεται με τον Η/Υ για να προγραμματιστεί μέσα από ένα απλό περιβάλλον ανάπτυξης. Με το Arduino δημιουργούνται συσκευές οι οποίες εξυπηρετούν διάφορους σκοπούς έχοντας την δυνατότητα να δέχονται ερεθίσματα από το περιβάλλον τους (μέσω των αισθητήρων) και να αντιδρούν ανάλογα με το πως έχουν προγραμματιστεί.

Τα παραπάνω δεν ακούγονται πρωτότυπα. Υπάρχουν και άλλες πλατφόρμες και υλοποιήσεις που μπορούν να κάνουν τα ίδια πράγματα. Ποια είναι η ειδοποιός διαφορά; Το Arduino βασίζεται σε τεχνολογίες ανοιχτού κώδικα. Μπορεί να κατασκευαστεί από τον καθένα, μπορεί να ενσωματωθεί σε συσκευές ακόμα και για εμπορικούς σκοπούς και το σημαντικότερο είναι ότι υπάρχει μια ολόκληρη κοινότητα που χρησιμοποιεί το Arduino σε κατασκευές άρα υπάρχει μεγάλος όγκος ελεύθερης πληροφορίας. Γενικά, τα Projects στον εν λόγω Μικροελεγκτή μπορούν να είναι αυτόνομα (σε επίπεδο hardware) ή να επικοινωνούν με κάποιο software στον Η/Υ του προγραμματιστή (προγράμματα όπως τα Flash, Processing, MaxMSP). Το Arduino χρησιμοποιεί τώρα ένα ειδικά προγραμματιζόμενο Atmega382 αντί του chip FTDI ώστε αυτό να επιτρέπει τόσο την πιο γρήγορη ταχύτητα μεταφοράς όσο και τη γρήγορη σειριακή επικοινωνία.

Ο μικροεπεξεργαστής ενός Arduino συνήθως προγραμματίζεται εκ των προτέρων ώστε να παρέχει κάποιο φορτωτή εκκίνησης (BootLoader). Ο φορτωτής εκκίνησης υπάρχει ώστε να απλοποιεί την διαδικασία της αποθήκευσης των προγραμμάτων στην Flash Memory του Arduino μέσω σειριακής USB θύρας.

Επιπλέον, η γλώσσα προγραμματισμού, οι διάφορες βιβλιοθήκες και το ολοκληρωμένο περιβάλλον ανάπτυξης που υπάρχουν για τον προγραμματισμό της πλατφόρμας Arduino αποτελούν ανοιχτό λογισμικό προσφέροντας έτσι ανεκτίμητη γνώση σε όλους.

Βασικά Πλεονεκτήματα πλατφόρμας Arduino:

- Οικονομική: Η πλατφόρμα Arduino αποτελεί οικονομική λύση διότι είναι φθηνότερη. Επιπλέον, είναι αρχιτεκτονικά ανοιχτή και μπορεί ο οποιοσδήποτε να την αναπτύξει από μόνος του.

- **Μεταφέρσιμη:** Σε σχέση με τις υπάρχουσες πλατφόρμες στο εμπόριο η πλατφόρμα Arduino παρέχει πλήρη μεταφερσιμότητα με αποτέλεσμα να μπορεί να προγραμματιστεί στα περισσότερα λειτουργικά συστήματα.
- **Επεκτάσιμη:** Το υλικό και το λογισμικό της πλατφόρμας Arduino είναι ανοιχτά και ελεύθερα για όλους. Καθημερινά, χιλιάδες υποστηρικτές του ελεύθερου λογισμικού αναπτύσσουν διάφορες βιβλιοθήκες για την υποστήριξη της πλατφόρμας. Παράλληλα, τόσο η αρχιτεκτονική όσο και το υλικό της πλατφόρμας εξελίσσονται συνεχώς.

Παρακάτω ακολουθούν μερικές από τις πλατφόρμες Arduino που έχουν αναπτυχθεί και όπου η κάθε μία είτε αποτελεί εξέλιξη κάποιας άλλης, είτε έχει αναπτυχθεί για κάποιο συγκεκριμένο σκοπό :

- Arduino Uno
- Arduino Diecimila
- Arduino Duemilanove
- Arduino Mega1280
- Arduino mega2560
- Arduino mini
- Arduino nano
- Arduino usb
- Arduino stamp
- Arduino fio
- Arduino ng
- Arduino ng+
- Arduino extreme
- Arduino bluetooth
- Lilypad arduino
- Serial arduino



Εικόνα 3.1: Το λογότυπο Arduino

Το λογισμικό του Arduino

Το περιβάλλον ανάπτυξης του Arduino (IDE) έχει συγγραφεί με την γλώσσα προγραμματισμού Java και αυτό το καθιστά μεταφέρσιμο στα περισσότερα λειτουργικά συστήματα. Το IDE του Arduino περιέχει έναν έξυπνο συντάκτη, μεταγλωττιστή της C, C++, τερματικό για σειριακή επικοινωνία με το Arduino, κ.α. Πιο

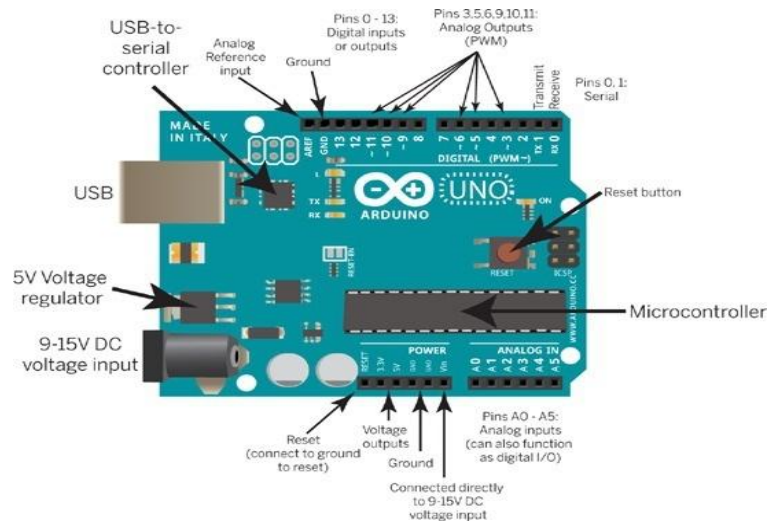
συγκεκριμένα, η γλώσσα προγραμματισμού που χρησιμοποιείται για την συγγραφή προγραμμάτων στο Arduino είναι η Wiring (C, C++). Το IDE του Arduino χρησιμοποιεί εργαλεία GNU toolchain και AVR Libc για να παρέχει την μεταγλώττιση προγραμμάτων από C, C++ σε κατάλληλες AVR εντολές γλώσσας μηχανής, καθώς και το εργαλείο avrduude για την αποστολή του εκτελέσιμου προγράμματος στην Flash memory του Arduino. Η ψηφιακή σχεδίαση του υλικού μέρους του Arduino είναι ανοιχτή και προσβάσιμη από όλους μια και είναι δημοσιευμένη υπό την άδεια Creative Commons Attribution Share-Alike 2.5. Επίσης, το περιβάλλον ανάπτυξης (IDE) του Arduino είναι ελεύθερο λογισμικό και είναι δημοσιευμένο υπό την άδεια GNU General Public License Version 2.

3.2 Μέρη ενός Arduino

Το Arduino έχει 14 ψηφιακούς ακροδέκτες Εισόδου/Εξόδου οι οποίοι μπορούν να τεθούν ως είσοδοι ή ως έξοδοι με τις εντολές-συναρτήσεις pinMode(), digitalWrite(), και digitalRead() που θα αναλυθούν παρακάτω. Λειτουργούν στα 5 Volts και έχουν την δυνατότητα να παρέχουν ή να καταβυθίζουν ένταση της τάξεως των 40mA. Σε κάθε pin υπάρχει εσωτερικά ένας Pull-up αντιστάτης στα 20-50KΩ. Επιπλέον, έχει 6 αναλογικούς ακροδέκτες Εισόδου. Αυτοί μπορούν να διαβάσουν αναλογικές τιμές όπως η τάση μιας μπαταρίας κτλ και να τις μετατρέψουν σε έναν αριθμό από 0-1023. Η μέτρηση της τάσης γίνεται από προκαθορισμένα από 0 έως 5 volts. Εκτός αυτού, 6 εκ των 14 ψηφιακών ακροδεκτών οι P3, P5, P6, P9, P10 και P11 έχουν την δυνατότητα να προγραμματιστούν ώστε να λειτουργούν ως αναλογικές Έξοδοι. Κάποιοι ακροδέκτες έχουν συγκεκριμένες λειτουργίες.

- Σειριακή Λειτουργία: 0 (RX) και 1 (TX). Χρησιμοποιούνται για λήψη (RX) και εκπομπή (TX) TTL σειριακών δεδομένων.
- Εξωτερικές Διακοπές: 2 και 3. Αυτοί οι ακροδέκτες μπορούν να ενεργοποιούν διακοπές αν ανιχνευθεί παλμός χαμηλής τάσης. Με την συνάρτηση attachInterrupt(). Ο σκανδαλισμός των διακοπών μπορεί να γίνεται στο λογικό 0,1.
- PWM: 3, 5, 6, 9, 10, και 11 pins. Παρέχουν Έξοδο 8-bit PWM με την συνάρτηση analogWrite().
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Αυτοί οι ακροδέκτες επιτρέπουν επικοινωνία SPI, η οποία αν και παρέχεται από το hardware δεν είναι ακόμα διαθέσιμη στην γλώσσα προγραμματισμού του Arduino.
- LED: 13. Στον ακροδέκτη 13 υπάρχει ένα ενσωματωμένο LED. Όταν ο ακροδέκτης έχει τιμή HIGH, το LED ανάβει ενώ όταν το pin είναι LOW δεν ανάβει.

Ασύρματη επικοινωνία πομπών Xbee μέσω Arduino



Εικόνα 3.2: Μέρη της πλατφόρμας Arduino Uno

3.3 Σειριακή Θύρα

Χρησιμοποιείται για επικοινωνία μεταξύ της πλατφόρμας Arduino και ενός υπολογιστή ή με άλλες συσκευές. Επομένως, όλες οι πλακέτες έχουν τουλάχιστον μια σειριακή θύρα. Επικοινωνεί με τις ψηφιακές ακίδες 0 (RX) και 1 (TX), καθώς και με τον υπολογιστή μέσω USB. Έτσι, εάν χρησιμοποιείται αυτή η λειτουργία(USB), δεν μπορούν ταυτόχρονα να χρησιμοποιηθούν οι ακίδες 0 και 1 για ψηφιακή είσοδο ή έξοδο. Αξίζει να αναφερθεί, η ενσωματωμένη σειριακή οθόνη στο περιβάλλον του Arduino μπορεί να χρησιμοποιηθεί για να επικοινωνεί με την πλακέτα Arduino. Κάνοντας κλικ στο κουμπί Serial Monitor στην γραμμή εργαλείων και επιλέγοντας την ίδια ταχύτητα που χρησιμοποιείται στην κλήση της Serial.begin().

Οι βασικές συναρτήσεις της σειριακής θύρας είναι:

- begin() (αρχικοποίηση της σειριακής)
- end() (κλείσιμο της σειριακής)
- available() (έλεγχος αν υπάρχουν δεδομένα να διαβαστούν)
- read() (ανάγνωση των εισερχόμενων σειριακών δεδομένων)
- peek() (επιστρέφει το επόμενο byte από την σειριακή)
- flush() (άδειασμα του buffer της σειριακής από δεδομένα που έχει)
- print() (γράψιμο δεδομένων στη σειριακή)
- println() (το ίδιο με την Print(), αλλά με αλλαγή γραμμής στο τέλος)
- write() (γράφει δυαδικά δεδομένα στη σειριακή)

3.4 Χαρακτηριστικά του Arduino

- Microcontroller: ATmega328
- Τάση λειτουργίας: 5V
- Τάση εισόδου: 7-12V
- Τάση εισόδου (όριο): 6-20V

- Digital I/O Pins: 14 (εκ των οποίων 6 περιέχουν PWM εξόδους)
- Analog Input Pins: 6
- DC ρεύματος I/O Pin: 40 mA
- DC τρέχουσα για 3.3V Pin: 50 mA
- Flash Memory: 32 KB εκ των οποίων 0,5 KB που χρησιμοποιούνται από τον bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz

3.5 Βασικές μνήμες

Οι πλατφόρμες Arduino διαθέτουν τρεις βασικές μνήμες:

- Flash memory (32 Kbytes) στην οποία τοποθετείται κάθε φορά το πρόγραμμα που πρόκειται να εκτελεστεί καθώς και ο φορτωτής εκκίνησης που διευκολύνει την διαδικασία του προγραμματισμού της πλατφόρμας.
- SRAM memory (στατική μνήμη τυχαίας προσπέλασης των 2 Kbytes) η οποία χρησιμοποιείται για την προσωρινή αποθήκευση των στατικών και των μεταβλητών δεδομένων του προγράμματος που εκτελείται.
- EEPROM memory (1 Kbytes) στην οποία αποθηκεύονται οι τιμές των μεταβλητών όταν η πλατφόρμα σβήσει(OFF). Χρησιμοποιείται για την αποθήκευση ρυθμίσεων και άλλων παραμέτρων ανάμεσα στα Reset του Arduino.

Πρέπει να προστεθεί, η μνήμη Flash και η μνήμη EEPROM είναι σταθερές (οι πληροφορίες παραμένουν μετά την απενεργοποίησης του ρεύματος). Η μνήμη SRAM είναι ασταθής και οι πληροφορίες χάνονται όταν εναλλάσσεται το ρεύμα. Επειδή δεν υπάρχει πολύ διαθέσιμη SRAM, αν τελειώσει, το πρόγραμμα μπορεί να αποτύχει με απροσδόκητους τρόπους. Μπορεί να φαίνεται ότι φορτώνει με επιτυχία, αλλά δεν τρέχει, ή τρέχει παράξενα. Για να ελεγχθεί εάν αυτό συμβαίνει, μπορούν να μειωθούν τα σχόλια ή οι σειρές ή άλλες δομές δεδομένων στο sketch (χωρίς να αλλάξει ο κώδικας). Εάν λειτουργεί με επιτυχία στη συνέχεια, κατά πάσα πιθανότητα έχει εξαντληθεί η SRAM. Ένας τρόπος για να αντιμετωπιστεί αυτό το πρόβλημα είναι αν υπάρχουν πίνακες αναζήτησης ή άλλοι μεγάλοι πίνακες, τότε μπορεί να χρησιμοποιηθεί ο μικρότερος τύπος δεδομένων που είναι αναγκαίος για να αποθηκευτούν οι τιμές που χρειάζονται.

3.6 Τροφοδοσία

Το Arduino Uno τροφοδοτείται είτε από εξωτερική τροφοδοσία που παρέχεται είτε μέσω μιας υποδοχής των 2.1mm (θετικός πόλος στο κέντρο) που βρίσκεται στην κάτω αριστερή γωνία του Arduino είτε απευθείας από την θύρα USB του υπολογιστή. Η επιλογή της πηγής γίνεται αυτόματα από το αναπτυσσόμενο. Ως εξωτερική τροφοδοσία ορίζεται είτε μια μπαταρία, είτε μετασχηματιστής των 9Volt από 220V. Η μπαταρία μπορεί να συνδεθεί στις υποδοχές του Arduino Vin και GND όπου τοποθετούνται ο θετικός πόλος και ο αρνητικός αντίστοιχα. Από την άλλη αν τροφοδοτηθεί με μετασχηματιστή απλά πρέπει να τοποθετηθεί το βύσμα στην υποδοχή που υπάρχει

θετικό πόλο στο κέντρο. Η πλακέτα μπορεί να λειτουργήσει με εξωτερική πηγή από 6 έως 20 Volts. Αν ωστόσο τροφοδοτηθεί με λιγότερα από 7 Volt τα pin εξόδου 5Volt δεν θα καταφέρουν να εξάγουν τάση 5 Volts. Αντίθετα, αν δώσουμε πάνω από 12 Volts θα υπερθερμανθεί ο σταθεροποιητής τάσης στην πλακέτα και ενδεχομένως να καταστραφεί. Συνεπώς, μια ιδανική τάση είναι τα 9 Volts.

Οι ακροδέκτες τροφοδοσίας είναι οι εξής:

VIN. Ακροδέκτης για μη σταθεροποιημένη τάση. Συνήθως εδώ συνδέεται μια εξωτερική πηγή τροφοδοσίας.

5V. Ακροδέκτης σταθεροποιημένης τάσης 5Volt. Η ρυθμιζόμενη παροχή ηλεκτρικού ρεύματος που χρησιμοποιείται για την τροφοδοσία του μικροελεγκτή ή άλλων ηλεκτρονικών στοιχείων της πλακέτας. Αυτό μπορεί να προέρχεται είτε από Vin με ενσωματωμένο ρυθμιστή, ή να παρέχεται από USB ή άλλη ρυθμιζόμενη παροχή 5V

3V3. Μέγιστη κατανάλωση ρεύματος είναι 50mA.

GND. Γειωμένες ακίδες

3.7 Επικοινωνία

Το Arduino Uno έχει την δυνατότητα να επικοινωνεί με τον Ηλεκτρονικό Υπολογιστή, έναν άλλον Arduino ή άλλους μικροελεγκτές. Το ολοκληρωμένο ATmega382 παρέχει σειριακή επικοινωνία TTL 5 Volt UART, η οποία είναι διαθέσιμη από τους ακροδέκτες (λήψη RX) 0 και (εκπομπή TX) 1 του ολοκληρωμένου. Επιπλέον, η αναπτυξιακή πλακέτα του Arduino παρέχει σειριακή επικοινωνία με τον Ηλεκτρονικό Υπολογιστή για προγραμματισμό με την βοήθεια ενός ειδικά προγραμματιζόμενου ενσωματωμένου ολοκληρωμένου ATmega382 αντί του chip FTDI. Ωστόσο, αυτό επιτρέπει την πιο γρήγορη ταχύτητα μεταφοράς και γρήγορης σειριακής επικοινωνίας. Με την σύνδεση του Arduino μέσω της θύρας USB αυτό εμφανίζεται ως εικονική σειριακή θύρα COM στο λογισμικό του υπολογιστή. Το firmware ATmega382 χρησιμοποιεί τα προγράμματα οδήγησης USB COM και δεν χρειάζεται να υπάρχει εξωτερικός παράγοντας. Επομένως, στα Windows απαιτείται μόνο ένα αρχείο .inf. Ένα Arduino περιλαμβάνει ένα τμηματικό όργανο ελέγχου το οποίο επιτρέπει την απλή μορφή κειμένου δεδομένων που αποστέλλονται προς και από τη πλακέτα Arduino. Οι RX και TX λυχνίες LED στην πλακέτα θα αναβοσβήνουν όταν γίνεται μετάδοση δεδομένων μέσω του USB-to-chip σειριακή και USB σύνδεση με τον υπολογιστή (αλλά όχι για σειριακή επικοινωνία στις ακίδες 0 και 1).

3.8 Γλώσσα Προγραμματισμού/εντολές

Μετά την εγκατάσταση του Arduino IDE μπορούμε να γράψουμε τα πρώτα μας τμήματα κώδικα.

Η λογική του Arduino είναι πολύ απλή - στην ουσία υπάρχουν δύο βασικές συναρτήσεις, η setup()

και η loop() οι οποίες δουλεύουν ως εξής:

- **setup()** - εδώ βάζουμε όλες τις εντολές που πρέπει να τρέξουν μία φορά, όταν ενεργοποιείται η μονάδα μας (όταν δηλαδή δίνουμε ρεύμα ή όταν πατηθεί το πλήκτρο reset που υπάρχει). Συνήθως μπαίνουν αρχικοποιήσεις τιμών μεταβλητών και οπωσδήποτε ο χαρακτηρισμός των εισόδων/εξόδων που θα χρησιμοποιήσουμε (αν δηλαδή ένα συγκεκριμένο Pin θα είναι είσοδος ή εξόδος).

- **loop()** - εδώ γράφουμε το πρόγραμμά μας. Οι εντολές που υπάρχουν θα τρέξουν κι όταν φτάσει στο τέλος θα ενεργοποιηθεί ξανά η loop(), συνεχίζοντας από την αρχή της, και ξανά. Αυτό θα συμβαίνει συνεχώς, όσο έχει ρεύμα το Arduino ή μέχρι να

πατηθεί το πλήκτρο reset. Έτσι, η βασική λειτουργία του Arduino είναι ότι τρέχει η συνάρτηση setup() μία φορά στην αρχή και ακολούθως η loop() ξανά και ξανά μέχρι να το κλείσουμε (να μην τροφοδοτείται με ρεύμα) ή να πατήσουμε το πλήκτρο reset. Στην περίπτωση του Reset ξανατρέχει η συνάρτηση setup() μία φορά και ακολούθως η loop() ξανά και ξανά, όπως δηλαδή ακριβώς και όταν αρχικά ενεργοποιείται με ρεύμα ο μικροελεγκτής. Στην περίπτωση που έχουμε κάνει αλλαγές στο πρόγραμμά μας και το φορτώσουμε στον μικροελεγκτή (θα δούμε παρακάτω τη διαδικασία αυτή) αρκεί να πατήσουμε το πλήκτρο Reset ώστε να φορτώσει το πρόγραμμά μας από την αρχή με τον τρόπο που περιγράφηκε. Ένα τυπικό πρόγραμμα έχει την παρακάτω δομή:

```
void setup() {
/* οι εντολές εδώ θα τρέξουν μόνο στην ενεργοποίηση ή μετά από Reset */
}
void loop() {
/* οι εντολές εδώ θα τρέχουν ξανά και ξανά,
μέχρι να απενεργοποιηθεί ή να πατηθεί το Reset */
}
```

➤ Δηλώσεις μεταβλητών

Όπως σε όλες τις γλώσσες προγραμματισμού, μπορώ να δηλώσω ονόματα μεταβλητών. Οι τύποι μεταβλητών που υποστηρίζονται στο Arduino είναι αρκετοί. Για έναν αρχάριο χρήστη οι παρακάτω τύποι θα είναι αρκετοί:

- **boolean**, με τιμές το 0 και 1 (ή True – False)
- **byte**, με τιμές από 0 έως και 255
- **int**, ακέραιος με δυνατές τιμές από -32768 έως και 32767
- **long**, ακέραιος με δυνατές τιμές από -2147483648 έως και 2147483647
- **float**, δεκαδικοί αριθμοί
- **char**, ένας χαρακτήρας (μέγεθος ένα Byte)
- **string**, πίνακας χαρακτήρων

Ένα παράδειγμα δήλωσης μεταβλητών δίνεται παρακάτω:

```
int ledPin = 13; // ορίζω ακέραια μεταβλητή ledPin και αρχικοποιώ την τιμή της σε 13
float SinVal; // ορίζω πραγματική μεταβλητή SinVal
```

➤ Συναρτήσεις διαχείρισης θυρών εισόδου – εξόδου (Pins)

Όπως αναφέρθηκε, η κύρια λειτουργία του μικροελεγκτή βασίζεται στο να ελέγχει τις θύρες που διαθέτει και είτε να δίνει ρεύμα είτε να παίρνει ρεύμα από αυτές. Στην αρχικοποίηση κάθε

προγράμματος (μέσα στη συνάρτηση setup) θα χρειαστεί να χαρακτηρίσουμε τα Pins που χρησιμοποιούμε ως είσοδο ή ως έξοδο. Η συνάρτηση pinMode(Pin, Mode) χρησιμοποιείται με το όνομά της και ορίσματα α) τον αριθμό Pin και β) την κατάσταση λειτουργίας που χαρακτηρίζεται με τη λέξη INPUT (είσοδος) ή OUTPUT (έξοδος).

Όπως έχουμε αναφέρει έχουμε 14 ψηφιακά Pins, 6 εκ των οποίων είναι PWM, με ονόματα 0..13 και έξι αναλογικά με ονόματα A0..A5. Για παράδειγμα:

```
pinMode(12, OUTPUT);
pinMode(ledPin, OUTPUT);
pinMode(A2, INPUT);
```

➤ Ψηφιακή έξοδος

Και τα 14 pins του Arduino μπορούν δουλεύουν ως ψηφιακές έξοδοι, δηλαδή δίνουν έξοδο 0 ή 5V.

Αυτό γίνεται με χρήση της συνάρτησης **digitalWrite(Pin, Value)**, όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα δώσουμε τάση εξόδου, ενώ η

τάση εξόδου μπορεί να είναι 0 V ή 5 V, οι οποίες αναπαρίστανται με προκαθορισμένες τιμές στην παράμετρο value

- LOW : θα δώσει 0 V στην έξοδο (pin)
- HIGH : θα δώσει 5 V στην έξοδο (pin)

Για παράδειγμα:

```
digitalWrite(ledPin, HIGH);
```

Προσοχή: Η αντίστοιχη θύρα θα πρέπει να έχει οριστεί ως εξόδου στη διαδικασία setup(), με χρήση της συνάρτησης pinMode.

Για παράδειγμα:

```
pinMode(10, OUTPUT);
```

➤ Ψηφιακή είσοδος

Και τα 14 ψηφιακά pins του Arduino μπορούν δουλεύουν ως ψηφιακές εισόδους, δηλαδή να “διαβάσουν” ως είσοδο τάση με τιμή είτε 0 είτε 5V. Αυτό γίνεται με χρήση της συνάρτησης **digitalRead(Pin)**, όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα πάρουμε είσοδο, ενώ η συνάρτηση επιστρέφει με το όνομά της την τιμή εισόδου. Η τάση εισόδου μπορεί να είναι 0V ή 5V, οι οποίες αναπαρίστανται με προκαθορισμένες τιμές στην τιμή που διαβάζουμε:

- LOW : όταν λάβει τάση 0 V στην είσοδο (pin)
- HIGH : όταν λάβει τάση 5 V στην είσοδο (pin)

Για παράδειγμα:

```
Val = digitalRead(ledPin);
```

Προσοχή: Η αντίστοιχη θύρα θα πρέπει να έχει οριστεί ως εισόδου στη διαδικασία setup(), με χρήση της συνάρτησης pinMode. Για παράδειγμα:

```
pinMode(10, INPUT);
```

➤ Αναλογική έξοδος (PWM pins)

Κάποια από τα 14 Pins του Arduino έχουν την ένδειξη PWM, δηλαδή μπορούν να προσομοιώσουν την αναλογική έξοδο μέσω παλμοκωδικής διαμόρφωσης Έτσι, **με τιμές από το 0 μέχρι το 255** προσομοιώνουμε (αναλογικά) το διάστημα από 0 έως 5V. Αυτό γίνεται με χρήση της συνάρτησης **analogWrite(Pin, Value)**, όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα δώσουμε ρεύμα εξόδου, ενώ η τάση εξόδου κυμαίνεται από 0 V μέχρι και 5 V, οι οποίες τιμές της τάσης αναλογικά αναπαρίστανται με τιμές στη μεταβλητή value. Τιμή 0 δίνει 0V στην έξοδο (pin), τιμή 255 δίνει τάση 5V στην έξοδο (pin), ενώ αναλογικά μπορούμε να δώσουμε ενδιάμεσες τάσεις (π.χ. 122 για τάση 2,5V). Για παράδειγμα:

```
analogWrite(ledPin, 122);
```

Υπενθύμιση: Τη λειτουργία αυτή μπορούν να υποστηρίξουν μόνο τα PWM pins κι όχι όλα τα ψηφιακά. Τα PWM pins είναι τα 3, 5, 6, 9, 10, 11.

Προσοχή: Η αντίστοιχη θύρα θα πρέπει να έχει οριστεί ως εξόδου στη διαδικασία setup(), με χρήση της συνάρτησης pinMode. Για παράδειγμα:

```
pinMode(10, OUTPUT);
```

➤ Αναλογική είσοδος

Το Arduino έχει 6 αναλογικές εισόδους, οι οποίες χαρακτηρίζονται με τα σύμβολα A0, A1, A2, A3, A4, A5. Μπορούμε να συνδέσουμε κάποιο αναλογικό εξάρτημα (π.χ. ένα ποτενσιόμετρο) και να το διαβάσουμε ως είσοδο. Αυτό γίνεται με χρήση της συνάρτησης **analogRead(Pin)**, όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα πάρουμε είσοδο, ενώ η συνάρτηση επιστρέφει με το όνομά της την τιμή εισόδου. Η τιμή εισόδου κυμαίνεται από 0 μέχρι και 1023. Συνήθως χρησιμοποιούμε μια μεταβλητή για να καταχωρήσουμε την τιμή. Για παράδειγμα:

```
int r = analogRead(A1);
```

Προσοχή: Η αντίστοιχη θύρα θα πρέπει να έχει οριστεί ως είσοδος στη διαδικασία setup(), με χρήση της συνάρτησης pinMode. Για παράδειγμα:

pinMode(A1, INPUT);

➤ Συνάρτηση καθυστέρησης – delay()

Στο πρόγραμμά μας μπορούμε να ορίσουμε μια καθυστέρηση ώστε να διαρκέσει για το χρόνο που εμείς ορίζουμε ένα γεγονός. Αυτό το επιτυγχάνουμε με χρήση της συνάρτησης delay(time) όπου στη θέση time δίνουμε το χρόνο σε ms (1/1000 sec). Η εντολή delay(time) σημαίνει ότι σταματά στο σημείο αυτό η εκτέλεση του προγράμματός μας για το χρόνο time. Για παράδειγμα:

delay(1000); //σταματά την εκτέλεση του προγράμματος για 1000 ms = 1 sec
delay(500); //σταματά την εκτέλεση στο σημείο αυτό για 500 ms = 0.5 sec

➤ Συνάρτηση καθυστέρησης - delayMicroseconds()

Ακριβώς αντίστοιχα με την παραπάνω συνάρτηση delay, μπορούμε να χρησιμοποιήσουμε την delayMicroseconds(), όπου ο χρόνος καθυστέρησης δίνεται πλέον σε microseconds (1/106 sec). Για παράδειγμα:

delayMicroseconds(10); // σταματά την εκτέλεση για 10μsec = 1/100000 sec

➤ Συνάρτηση καταγραφής χρόνου - millis()

Το Arduino έχει ενσωματωμένο ρολόι, το οποίο μετράει το χρόνο από τη στιγμή που ενεργοποιείται (ή του γίνεται reset). Η πληροφορία αυτή μας είναι διαθέσιμη σε κάθε σημείο με κλήση της συνάρτησης millis(), η οποία μας επιστρέφει το χρόνο σε milliseconds (1/1000 sec) που έχει περάσει από την ενεργοποίηση της μονάδας μας. Αυτό μας βοηθά να μετράμε το χρόνο στα προγράμματά μας, ειδικά στις περιπτώσεις που θέλουμε να “θυμόμαστε” πράγματα. Για παράδειγμα, αν θέλουμε να ελέγξουμε αν έχει περάσει συγκεκριμένο διάστημα από τότε που έγινε κάτι (π.χ. πατήθηκε τελευταία φορά ένα πλήκτρο), μπορούμε να καταγράψουμε το γεγονός σε μια μεταβλητή χρόνου και τον χρόνο αυτό να τον αφαιρούμε από τον επόμενο κτλ. Για παράδειγμα:

lastPress = millis(); if (lastPress – millis() > 1000) {...}

➤ Η σειριακή θύρα επικοινωνίας (Serial)

Το Arduino παρέχει μια σειριακή θύρα επικοινωνίας μεταξύ της πλακέτας και του υπολογιστή ή κάποιας συσκευής που θέλουμε. Για το σκοπό αυτό χρησιμοποιείται η σύνδεση με καλώδιο USB (όταν πρόκειται για τον υπολογιστή) ή τα pins 0 και 1 όταν θέλουμε κάποια πιο εξειδικευμένη σύνδεση (π.χ. με κάποια άλλη συσκευή). Για το λόγο αυτό προτείνεται, αν δεν είναι απαραίτητο στις εφαρμογές μας, να μην χρησιμοποιούνται τα pins αυτά. Για να ενεργοποιήσουμε τη σειριακή θύρα επικοινωνίας αρκεί να δώσουμε στη διαδικασία setup() την εντολή Serial.begin(BaudRate), όπου το BaudRate εκφράζει το ρυθμό με τον οποίο θα μεταδίδονται τα bits (μια τιμή στα 9600 είναι συνήθως αρκετή). Για παράδειγμα: Serial.begin(9600); Μπορούμε να χρησιμοποιήσουμε τη σειριακή θύρα στις εφαρμογές για αμφίδρομη επικοινωνία, δηλαδή να στείλουμε και να λάβουμε δεδομένα. Μία απλή περίπτωση χρήση της επικοινωνίας αυτής είναι για εκσφαλμάτωση (debugging) των προγραμμάτων μας, να μπορούμε δηλαδή να δούμε τι τιμές μας δίνουν μετρητές και τι τιμές έχουν οι μεταβλητές μας μέσω της οθόνης σειριακής επικοινωνίας. Μια εντολή που μας βοηθάει σε αυτό είναι η print(), που εκτυπώνει ένα μήνυμα ή τιμές ή η println() που λειτουργεί ακριβώς το ίδιο αλλά εκτυπώνοντας με αλλαγή γραμμής κάθε φορά. Για παράδειγμα:

Serial.print(“Η epikoinwnia ksekinhse”); /* Θα εμφανίσει το μήνυμα αυτό στην οθόνη χωρίς να αλλάξει γραμμή μετά */ Serial.println(distance); /* Θα εμφανίσει την τιμή της μεταβλητής

distance σε μια γραμμή */ Όταν έχετε συνδέσει το Arduino σας με τη θύρα USB στον υπολογιστή, η σειριακή οθόνη ενεργοποιείται από το εικονίδιο πάνω δεξιά “Σειριακή Οθόνη” (εικόνα 2), και στο παράθυρο που ανοίγει μπορείτε να βλέπετε όλα τα

μηνύματα που στέλνονται από τον κώδικα που έχει φορτωθεί ήδη και τρέχει στην πλακέτα.

➤ Δομή επιλογής

Στον προγραμματισμό πολλές φορές θα χρειαστεί να ελέγξουμε κάποια συνθήκη για να αποφασίσουμε αν θα εκτελεστεί ένα τμήμα κώδικα ή αν θα εκτελεστεί κάποιο άλλο αντί για αυτό στη θέση του. Αυτό το επιτυγχάνουμε με τη χρήση της δομής επιλογής, η οποία συντάσσεται:

```
if <συνθήκη> { <εντολές 1> } else { <εντολές 2> }
```

όπου, στη <συνθήκη> έχουμε τον έλεγχο που θέλουμε να γίνει, συνήθως χρησιμοποιώντας τους τελεστές σύγκρισης (>, <, =, <>, >=, <=), π.χ. potVal > 500. Η συνθήκη μπορεί να είναι και πιο σύνθετη, χρησιμοποιώντας τους λογικούς τελεστές (|| για το Η', && για το ΚΑΙ), π.χ. (potVal > 500) && (timePass >= 1000) .

Στα μπλοκ { <εντολές> } εκτελούνται αντίστοιχα οι εντολές που θέλουμε σε κάθε περίπτωση. Αν ισχύει η <συνθήκη> θα εκτελεστούν οι <εντολές 1>, αν δεν ισχύει οι <εντολές 2>. Σε κάθε περίπτωση, το τελευταίο κομμάτι else { <εντολές 2>} δεν είναι απαραίτητο να υπάρχει.

➤ Δομή επανάληψης (For)

Πολλές φορές θα χρειαστεί να επαναλάβουμε κάποια διαδικασία αρκετές φορές. Στην περίπτωση αυτή έχουμε εντολές οι οποίες επαναλαμβάνουν ένα σύνολο εντολών όσες φορές θέλουμε, είτε μετρώντας τις επαναλήψεις είτε ελέγχοντας κάθε φορά μία συνθήκη. Η συχνότερη μορφή που συναντάμε σε μια επανάληψη είναι αυτή με τον προκαθορισμένο αριθμό βημάτων. Η σύνταξη της εντολής αυτής είναι η εξής:

```
for (<αρχική τιμή>; <συνθήκη_τερματισμού>;<βήμα>) { <εντολές> }
```

όπου χρησιμοποιείται μια μεταβλητή ελέγχου ως εξής

<αρχική τιμή> δίνουμε την αρχική τιμή, π.χ. i = 0

<βήμα> δίνουμε την αλλαγή κάθε επανάληψης, π.χ. i+5 (το i++ που θα δείτε σημαίνει i+1)

<συνθήκη_τερματισμού> η συνθήκη για να τελειώσει η επανάληψη, π.χ. i < 10 (όσο ισχύει αυτή θα τρέχει) π.χ. for (i=1;i<10;i=i+1) { brightness = brightness + 5; analogWrite(ledPin, brightness); };

Υπάρχουν εντολές επανάληψης που δεν έχουν προκαθορισμένο αριθμό βημάτων, αλλά συνεχίζουν επ' άοριστο ελέγχοντας μια συνθήκη. Στα φύλλα εργασίας που ακολουθούν δεν θα χρειαστεί να χρησιμοποιήσετε μια τέτοια εντολή, αλλά για λόγους αναφοράς έχουμε τις παρακάτω εντολές:

- while <συνθήκη> { <εντολές> } // **όσο ισχύει** η <συνθήκη> τρέχουν οι εντολές
 - repeat {<εντολές>} until <συνθήκη> // οι <εντολές> τρέχουν **όσο δεν ισχύει η συνθήκη**

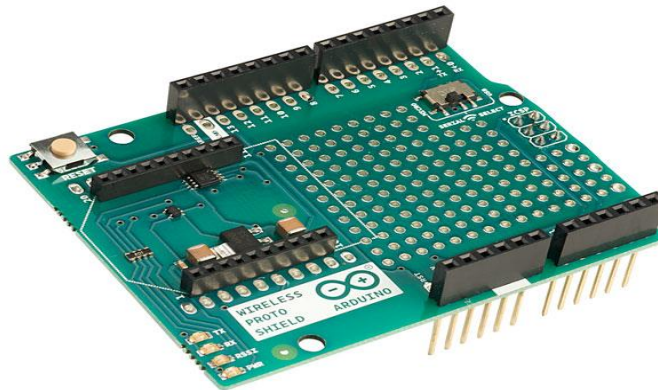
ΚΕΦΑΛΑΙΟ 4

ΕΡΓΑΣΙΑ

4.1 Το κομμάτι της κατασκευής

Αφού λοιπόν έχουμε αναλύσει το θεωρητικό μέρος της εργασίας ήρθε η ώρα να δούμε πρακτικά κάποια πράγματα. Στο κομμάτι της κατασκευής θα δούμε την χρήση του Xbee την δυνατότητα επικοινωνίας δυο Arduino μέσω αυτών καθώς και τον προγραμματισμό τους. Αρχικά να πούμε πως για τα υλικά χρειάστηκαν περίπου 110 ευρώ και δυο προγράμματα, το πρώτο είναι το λογισμικό Xctu το οποίο μπορούμε να το βρούμε δωρεάν στο "<https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu#productsupport-utilities>" το δεύτερο πρόγραμμα που χρειαζόμαστε είναι φυσικά το Arduino το οποίο και αυτό είναι δωρεάν και μπορεί κανείς να το κατεβάσει στο "<https://www.arduino.cc/en/Main/Software>". Στη συνέχεια ακολουθεί μια λίστα με τα υλικά που χρειαστήκαμε:

- 2x Arduino Wireless Proto Shield



Εικόνα 4.1: Arduino Wireless Proto Shield

- Keyboard(τύπου ps2)



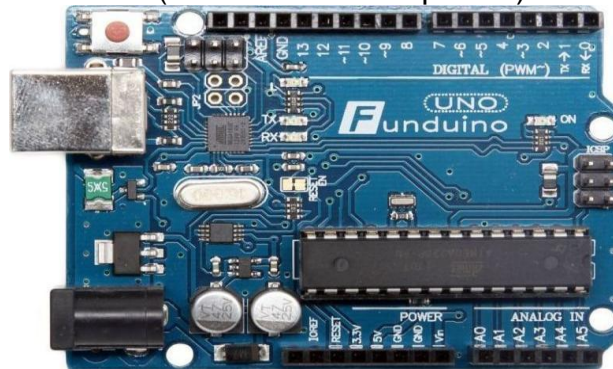
Εικόνα 4.2: Keyboard(τύπου ps2)

- I2C 20x4 Arduino LCD Display Module(blue)



Εικόνα 4.3: Arduino LCD Display Module(blue)

- 2x XBee Pro 63mW Wire Antenna - Series 2
- 2x Funduino UNO Rev3 (Arduino Uno Compatible)



Εικόνα 4.4: Funduino Uno

- 8x HR Jumper wires 40P 20cm M/F



Εικόνα 4.5: Jumper wires

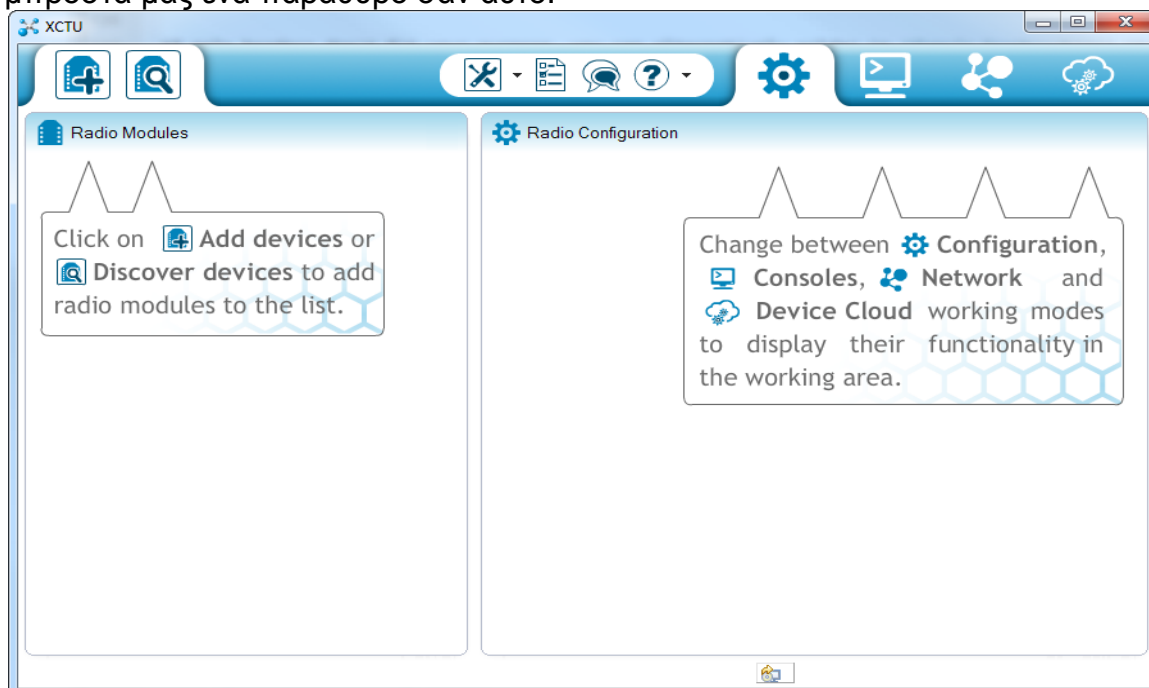
Σε αυτό το σημείο θα ήθελα να προσθέσω ότι κάθε εξάρτημα αν και απαραίτητο για την κατασκευή μας δεν είναι και το μοναδικό στο είδος του. Για παράδειγμα αγοράσαμε Xbee series 2, θα μπορούσαμε να χρησιμοποιήσουμε το series 1 ή ακόμα και το series 3. Όπως επίσης και η πλακέτα Funduino που χρησιμοποιήθηκε καθαρά για οικονομικούς λόγους καθώς είναι πολύ φτηνή, θα μπορούσε να ήταν κάποια άλλη

αρκεί να ήταν συμβατή με την Arduino Uno. Τέλος και η οθόνη θα μπορούσε να αντικατασταθεί με κάποια άλλη ακόμα μεγαλύτερη και με καλύτερη ανάλυση αλλά τότε θα έπρεπε να διαφοροποιηθεί ο κώδικας καθώς θα χρειαζόμασταν επιπλέον την προσθήκη ενός ακόμα shield όπως είναι το "Tellymate Shield"(σου επιτρέπει να στέλνεις σήματα από το Arduino στην τηλεόραση σου). Απλώς τα εξαρτήματα αυτά είναι πολύ εύκολο να βρεθούν στην αγορά και είναι σχετικά από τις οικονομικότερες λύσεις που υπάρχουν.

4.2 Το κομμάτι του προγραμματισμού

4.2.1 Προγραμματισμός με Xctuu

Έχοντας πει όλα τα παραπάνω ας ξεκινήσουμε με το κομμάτι του προγραμματισμού. Αρχικά θα πρέπει να ρυθμίσουμε/παραμετροποιήσουμε τους πομποδέκτες μας τα προαναφερόμενα Xbee. Για να το πετύχουμε αυτό λοιπόν αφού κατεβάσουμε και εγκαταστήσουμε το πρόγραμμα Xctuu θα πρέπει να ρυθμίσουμε τα δυο Xbee με τέτοιο τρόπο ώστε το ένα να λειτουργεί ως πομπός(θα στέλνει χαρακτήρες από το πληκτρολόγιο) και το άλλο ως δέκτης(θα λαμβάνει πληροφορίες από τον πομπό, στην περίπτωση μας θα είναι κάποιοι χαρακτήρες από το πληκτρολόγιο). Να προσθέσω πως ένας πομπός μπορεί να μεταδώσει πληροφορίες σε πολλαπλούς δέκτες αρκεί να γίνει η σωστή ρύθμιση. Ξεκινώντας λοιπόν όταν ανοίξουμε το Xctuu θα εμφανιστεί μπροστά μας ένα παράθυρο σαν αυτό:



Εικόνα 4.6: Αρχική οθόνη Xctuu

Αν και σχετικά εύκολο και ευχάριστο στην χρήση του το Xctuu μπορεί να φανεί αρκετά δύσκολο και περίπλοκο στην αρχή. Η λύση σε αυτό το πρόβλημα είναι πολύ απλό καθώς υπάρχουν αρκετά tutorials τα οποία μπορούμε να τα βρούμε με λίγο ψάξιμο στο Google, καθώς και videos στο YouTube που δείχνουν αναλυτικά βήμα βήμα τον τρόπο με τον οποίο πρέπει να γίνει η σωστή ρύθμιση αλλά και το σημαντικότερο, τον λόγο για τον οποίο χρησιμοποιούμε μια συγκεκριμένη παράμετρο. Το βίντεο που με βοήθησε βρίσκεται στην διεύθυνση "<https://www.youtube.com/watch?v=cf3RLBq4t5s>" και παρουσιάζει πολύ

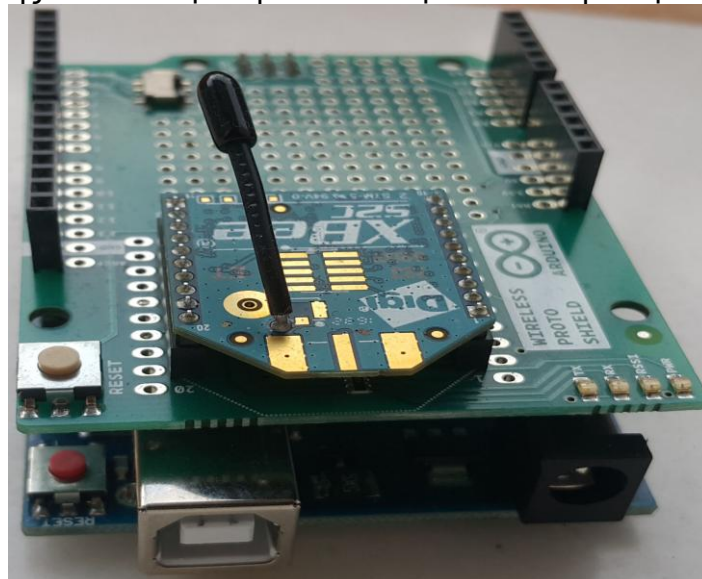
αναλυτικά τον τρόπο που μπορούμε να συνδέσουμε ένα Xbee με τον υπολογιστή μας να το προγραμματίσουμε και τέλος πως μπορούμε να ελέγξουμε τυχόν λάθη στην σύνδεση μεταξύ πομπού και δέκτη.



ZIGBEE XBEE-S2C -How to configure

Εικόνα 4.7: Βίντεο για την αρχικοποίηση του Xbee

Ο τρόπος σύνδεσης του Xbee με την πλακέτα φαίνεται στην παρακάτω εικόνα:



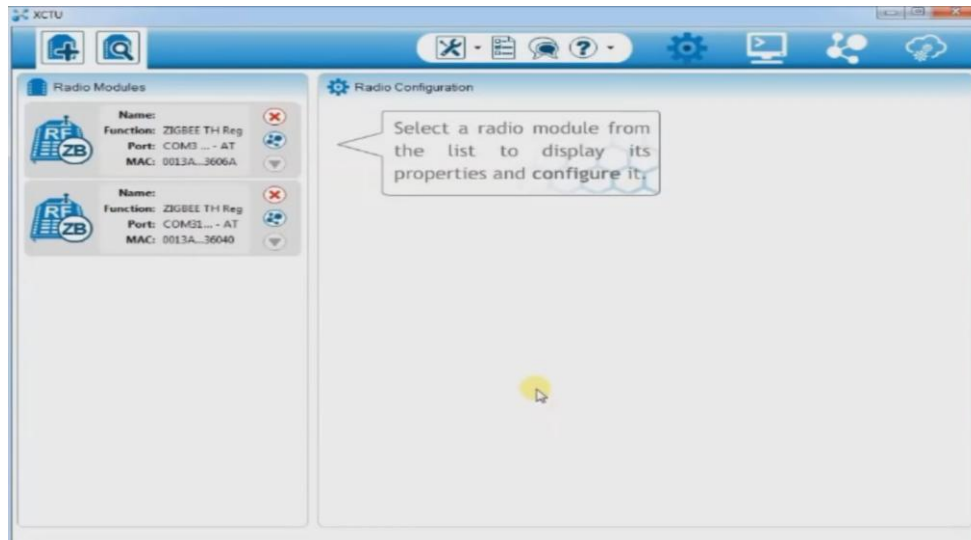
Εικόνα 4.8: Σύνδεση πλακέτας Arduino και Xbee

Το πρώτο που βλέπουμε είναι το Xbee μας αμέσως μετά ακολουθεί το Arduino Wireless Proto Shield το οποίο με την σειρά του είναι συνδεδεμένο στην κύρια πλακέτα μας την Funduino UNO.

Επόμενο βήμα λοιπόν είναι να "εντοπίσουμε" στο περιβάλλον Xctuu τα δυο Xbee που έχουμε συνδέσει μέσω καλωδίου usb. Στο παράθυρο που είναι ανοιχτό μπροστά μας βλέπουμε πάνω αριστερά έναν μεγεθυντικό φακό. Αφού τον πατήσουμε θα εμφανιστεί νέο παράθυρο και θα εμφανίσει όλες τις θύρες COM που διαθέτει ο υπολογιστής μας, εμείς πρέπει να επιλέξουμε τις 2 θύρες στις οποίες

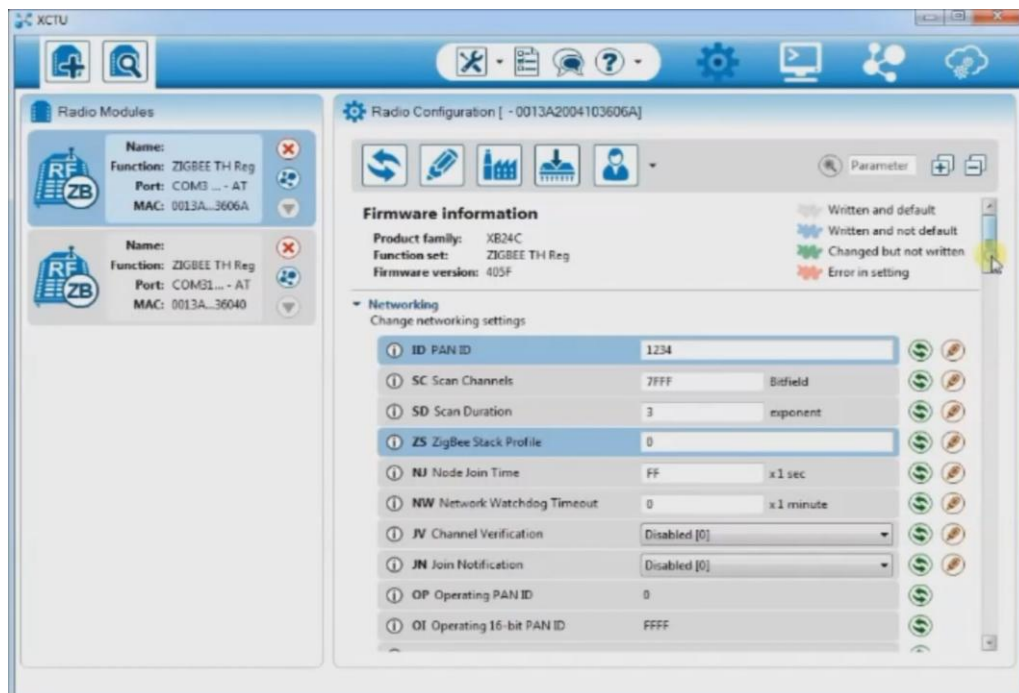
Ασύρματη επικοινωνία πομπών Xbee μέσω Arduino

μόλις συνδέσαμε τα rf modules μας. Πατάμε "next" και χωρίς να πειράξουμε κάποια ρύθμιση στο παράθυρο που εμφανίζεται πατάμε "finish". Μετά από μερικά δευτερόλεπτα αν έχουμε κάνει όλα σωστά θα πρέπει να εμφανιστεί νέο παράθυρο και να εμφανίζει το μήνυμα "Search finished 2 device(s) found" πατάμε λοιπόν "add selected devices". Φορτώσαμε λοιπόν στο πρόγραμμα τα 2 Xbee τα οποία θα πρέπει να εμφανίζονται στο αρχικό μας παράθυρο όπως στην εικόνα



Εικόνα 4.9: Εμφάνιση των 2 xbee στο λογισμικό Xctu

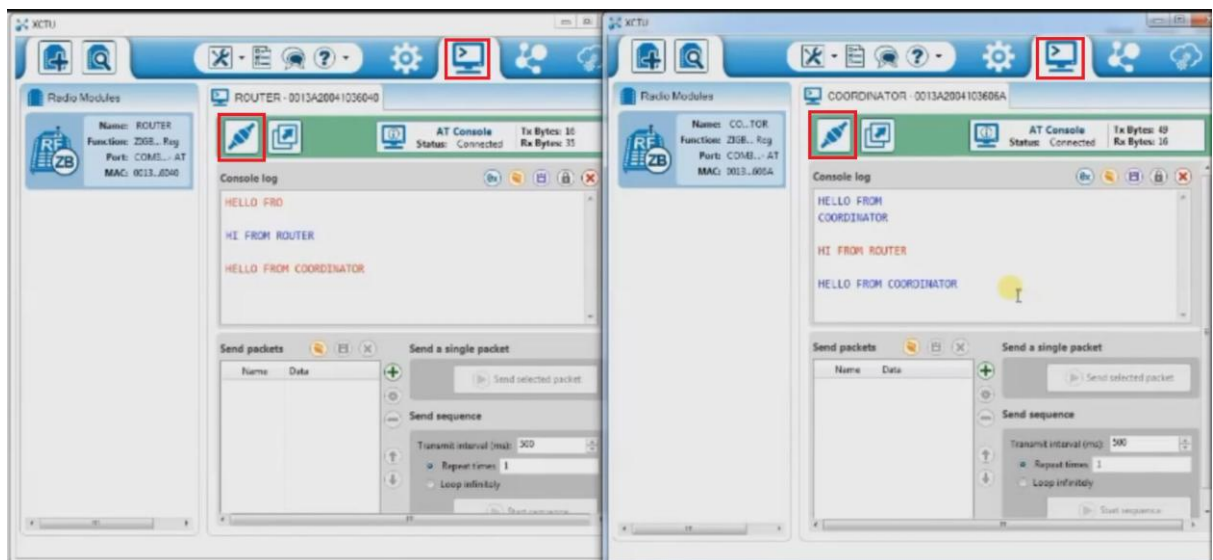
Κάνουμε κλικ στο πρώτο εικονίδιο και εμφανίζεται δεξιά μια σειρά παραμέτρων. Ευτυχώς δεν χρειάζεται να τις αλλάξουμε όλες παρά μόνο μερικές. Ξεκινάμε με το "PAN ID"



Εικόνα 4.10: Παραμετροποίηση του Xbee

στο πλαίσιο αυτό θα πρέπει να βάλουμε μια τιμή μεταξύ 0 και FFFF. Όλα τα Modules που θέλουμε να ανήκουν σε αυτό το δίκτυο θα πρέπει να έχουν ακριβώς το ίδιο ID. Οπότε ας βάλουμε το "1234". Κατεβαίνοντας προς τα κάτω στις ρυθμίσεις θα δούμε κάπου το "DL Destination Address low" θα ορίσουμε το πρώτο μας Xbee να είναι ο πομπός για αυτό το λόγο θα βάλουμε την FFFF καθώς έτσι το Module μας θα λειτουργεί σε "broadcast mode", κατ'αυτόν τον τρόπο θα μπορεί να επικοινωνεί με τα υπόλοιπα rf που ανήκουν στο ίδιο δίκτυο. Ακριβώς από κάτω υπάρχει το "NI Node Identifier" είναι το όνομα που δίνουμε στο Xbee ώστε να μπορούμε να το αναγνωρίσουμε αργότερα, ας βάλουμε το "Pombos". Τελευταία ρύθμιση που πρέπει να κάνουμε είναι στο "CE Coordinator Enable" να βάλουμε "Enabled[1]". Μόλις τα κάνουμε όλα αυτά πατάμε το μπλε μολυβάκι που βρίσκεται από πάνω και περιμένουμε μερικά δευτερόλεπτα να ολοκληρωθεί η διαδικασία. Θα ρυθμίσουμε τώρα το δεύτερο Xbee με παρόμοιο τρόπο. Πατάμε στο πρόγραμμα το 2 δεύτερο εικονίδιο για να εμφανιστούν οι ρυθμίσεις του άλλου πομποδέκτη μας. Βάζουμε λοιπόν στο "PAN ID" ακριβώς το ίδιο, δηλαδή "1234". Και στο "NI Node Identifier" ας βάλουμε τώρα "Dektis". Αυτό που αλλάζει τώρα είναι το "DL Destination Address low" αφού το Xbee μας θα λειτουργεί ως δέκτης πρέπει να βάλουμε το μηδέν. Τέλος αλλάζουμε μια νέα παράμετρο που βρίσκεται στο "JC Channel Verification" και θέτουμε την τιμή σε "Enable[1]". Αφού τα κάνουμε όλα αυτά πατάμε πάλι το μπλε μολυβάκι ώστε να αποθηκευτούν οι ρυθμίσεις μας. Στη συνέχεια το Xctu μας δίνει την δυνατότητα να δοκιμάσουμε την σύνδεση μεταξύ των Xbee.

Κλείνουμε τελείως το Xctu. Και το ανοίγουμε πάλι 2 φορές, αυτή τη φορά δεν θα φορτώσουμε και τα 2 Xbee ταυτόχρονα αλλά θα φορτώσουμε τον έναν(πομπός) στο ένα παράθυρο και τον άλλον(δέκτης) στο άλλο.



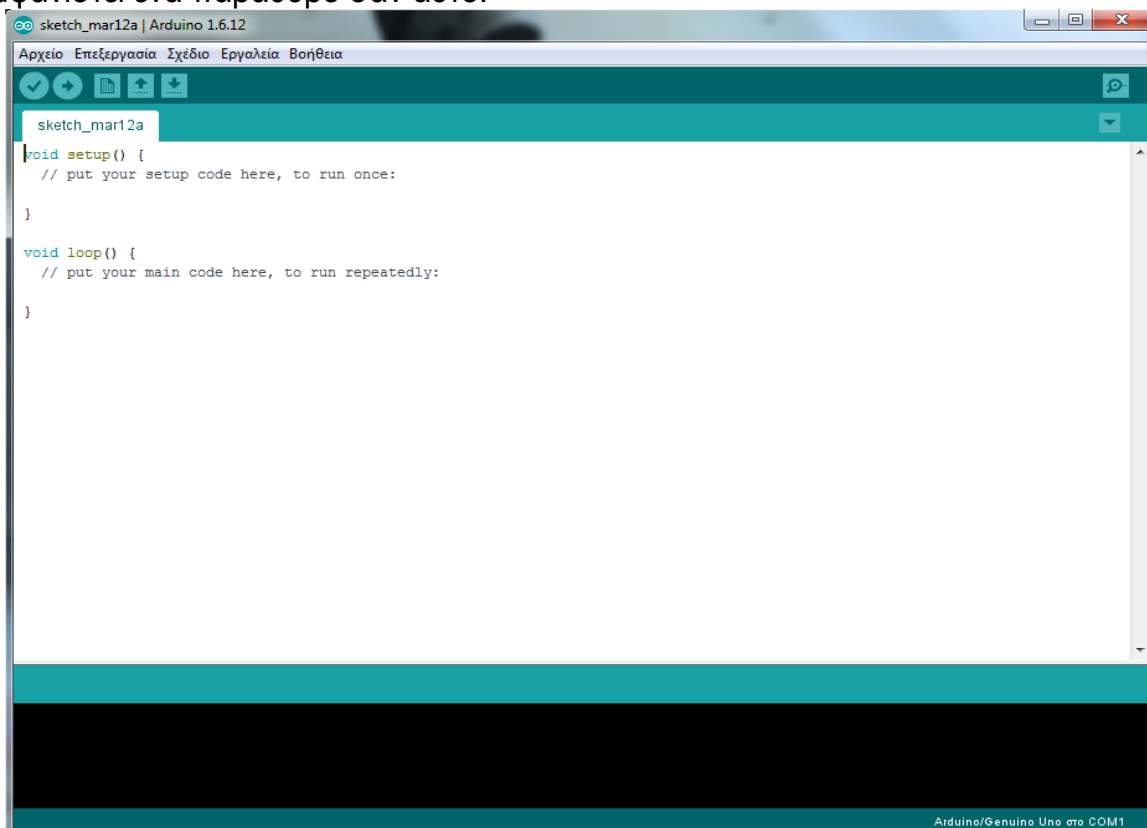
Εικόνα 4.11: Εμφάνιση των 2 Xbee σε ξεχωριστά παράθυρα

Στην εικόνα βλέπουμε τα 2 παράθυρα του προγράμματος και πατώντας τα 2 κουμπιά που είναι με κόκκινο είμαστε σε θέση να ανταλλάσουμε μηνύματα μεταξύ τους. Τα κόκκινα γράμματα είναι αυτά που λαμβάνει το Xbee και τα μπλε αυτά που μεταδίδει.

Ολοκληρώνοντας λοιπόν το κομμάτι του προγραμματισμού των πομποδεκτών θα πρέπει να προγραμματίσουμε τις πλακέτες μας με την βοήθεια του προγράμματος Arduino.

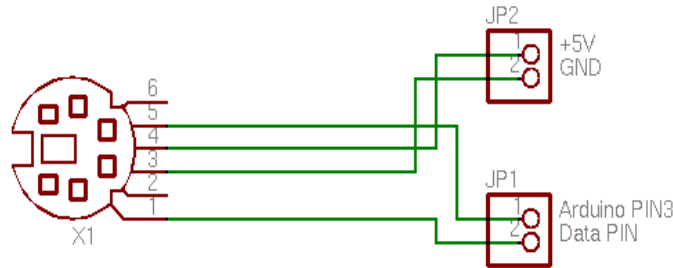
4.2.2 Προγραμματισμός με Arduino

Αφού τελειώσαμε με τα Xbee ήρθε η ώρα να προγραμματίσουμε και τις πλακέτες μας. Θα υπάρξουν δυο κομμάτια κώδικα ο ένας είναι αυτός της οθόνης και ο άλλος του πληκτρολογίου. Αναλυτικότερα στον πρώτο κώδικα αρχικά αρχικοποιούμε το πληκτρολόγιο ώστε να μπορεί να το αναγνωρίσει η πλακέτα και στη συνέχεια με την βοήθεια εντολών θέλουμε οτιδήποτε πατάει ο χρήστης στο πληκτρολόγιο να στέλνεται ασύρματα στην άλλη πλακέτα. Η λειτουργία λοιπόν της άλλης πλακέτας θα είναι αρχικά να αποθηκεύει σε μια μεταβλητή τον χαρακτήρα που μόλις έλαβε και στη συνέχεια να εμφανίζει τον χαρακτήρα αυτόν στην οθόνη lcd. Εγκαθιστούμε λοιπόν το λογισμικό Arduino. Μόλις τελειώσει η εγκατάσταση και το ανοίξουμε θα πρέπει να εμφανιστεί ένα παράθυρο σαν αυτό.



Εικόνα 4.12: Αρχική οθόνη του προγράμματος Arduino

Στην αρχή θα αναλύσουμε το κομμάτι που αφορά την σύνδεση πληκτρολογίου με το Arduino. Στο παρακάτω σχήμα φαίνεται ο τρόπος με τον οποίο συνδέεται κάθε καλώδιο πάνω στα pins της πλακέτας.



Εικόνα 4.13: Τρόπος σύνδεσης του πληκτρολογίου και Arduino

Ένα καλώδιο λοιπόν θα πάει στην τάση 5V ένα στην γείωση Gnd και τέλος τα άλλα 2 στο DataPin και στο IRQpin(interrupt request from a device) αντίστοιχα.

- Παρακάτω φαίνεται ο κώδικας που χρησιμοποιήθηκε για την διασύνδεση πληκτρολογίου και πλακέτας

```
CORDINATOR_KEY

#include <Wire.h>
#include <PS2Keyboard.h>
#include <SoftwareSerial.h>

const int DataPin = 3;
const int IRQpin = 2;

PS2Keyboard keyboard;

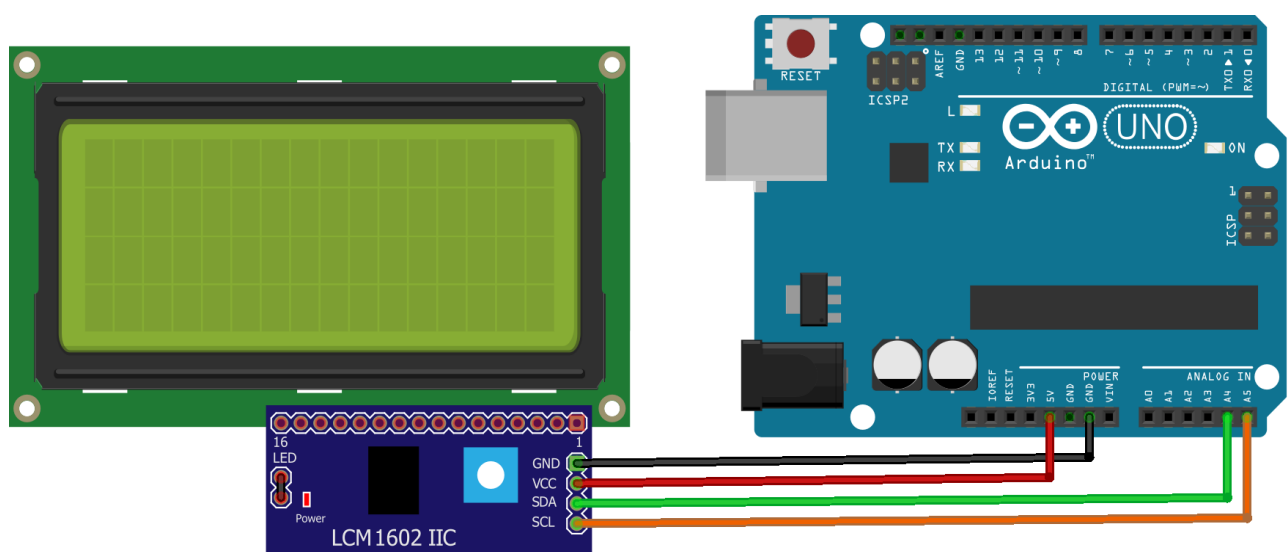
void setup() {
  delay(1000);
  keyboard.begin(DataPin, IRQpin);
  Serial.begin(9600);
}

void loop() {
  if (keyboard.available()) {
    char c = keyboard.read();
    Serial.println(c);
  }
}
```

Θα ξεκινήσουμε αναλύοντας τις 3 πρώτες γραμμές. Κάνουμε λοιπόν "include" τις απαραίτητες βιβλιοθήκες. Την "Wire.h" την "PS2Keyboard.h" και τέλος την "SoftwareSerial.h". Η πρώτη βιβλιοθήκη επιτρέπει την επικοινωνία μεταξύ I2C / TWI συσκευών. Η δεύτερη περιλαμβάνει εντολές και παραμέτρους που χρειάζονται για να

συνδέσουμε το πληκτρολόγιο πάνω στο Arduino. Και τέλος η τελευταία βιβλιοθήκη επιτρέπει την συριακή επικοινωνία μέσω θύρας Usb. Στην συνέχεια έχουμε δήλωση δυο μεταβλητών και ανάθεση τιμών σε αυτές 3 και 2 αντίστοιχα. Εδώ ορίζουμε σε ποια Pins έχουμε συνδέσει τα 2 καλώδια που μεταφέρουν πληροφορίες από το πληκτρολόγιο στην πλακέτα. Ακλουθεί μια εντολή "PS2Keyboard keyboard;" εδώ κατασκευάζουμε ένα αντικείμενο με το όνομα keyboard το οποίο έχει όλα τα γνωρίσματα της βιβλιοθήκης "PS2Keyboard.h" έτσι μπορούμε να χρησιμοποιούμε εντολές όπως "keyboard.read()" που μας επιτρέπουν να κάνουμε διάφορες λειτουργίες. Στη συνέχεια έχουμε μια συνάρτηση την void() {...} σε αυτό το κομμάτι κώδικα γίνονται πάντα κάποιες παραμετροποιήσεις. Αναλυτικότερα: έχουμε την εντολή "delay(1000)" το πρόγραμμα κάνει μια καθυστέρηση 1 sec για λόγους ομαλής λειτουργίας του κώδικα. Ακόμα έχουμε την εντολή "Serial.begin(9600);" που ορίζει το data rate για την συριακή μετάδοση δεδομένων το οποίο μετριέται σε bps(bits per second), οι τιμές που μπορεί να πάρει το όρισμα αυτό μπορεί να είναι 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, ή 115200 αλλά συνήθως χρησιμοποιούμε την 9600. Τέλος η εντολή "keyboard.begin(DataPin, IRQpin);" θέτει το αντικείμενο που δημιουργήσαμε σε λειτουργία δίνοντας του ως ορίσματα ουσιαστικά το 2 και το 3 αφού έτσι έχουμε δηλώσει τις μεταβλητές αυτές στην αρχή και όπως προείπαμε συμβολίζουν τα pins στα οποία έχουμε συνδέσει τα 2 καλώδια. Η τελευταία συνάρτηση που συναντάμε είναι η void loop(){...}, η οποία θεωρητικά θα πρέπει να εκτελείται δίχως να σταματά. Ο λόγος είναι η επόμενη εντολή "if (keyboard.available())" έχουμε έναν ατέρμονα βρόχο ο οποίος ελέγχει με απλά λόγια αν όλα λειτουργούν σωστά, δίχως σφάλματα αν υπάρχει δηλαδή διαθέσιμη σύνδεση. Από την στιγμή λοιπόν που θα λειτουργεί το πληκτρολόγιο δίχως πρόβλημα η συνθήκη if θα είναι πάντα αληθής. Στην συνέχεια με την εντολή "char c = keyboard.read();" αποθηκεύουμε σε μια μεταβλητή τύπου char(χαρακτήρας) τον χαρακτήρα που θα πατήσει ο χρήστης. Και τέλος με την εντολή "Serial.println(c);" στέλνουμε τον χαρακτήρα αυτόν στην σειριακή θύρα. Ο οποίος χαρακτήρας με την εντολή "char c=Serial.read();" θα διαβαστεί από την άλλη πλακέτα.

Τελειώνοντας με το κομμάτι του πληκτρολογίου ας αναλύσουμε την συνδεσμολογία της άλλης πλακέτας με την οθόνη lcd καθώς και τον κώδικα που θα χρειαστούμε.



Εικόνα 4.14: Τρόπος σύνδεσης της οθόνης lcd και Arduino

fritzing

Και εδώ θα χρειαστούμε 4 καλώδια ένα για την τροφοδοσία 5V ένα για την γείωση και τα άλλα δυο για το Sda(data line) και Scl(clock line) που αντιστοιχούν στο A4 και A5 pin της πλακέτας Arduino αντίστοιχα. Να προσθέσουμε ακόμα ότι το μπλε τετραγωνάκι που φαίνεται στην εικόνα της οθόνης είναι ένα ποτενσιόμετρο που ρυθμίζει την αντίθεση της οθόνης.

- Παρακάτω φαίνεται ο κώδικας που χρησιμοποιήθηκε για την διασύνδεση οθόνης lcd και πλακέτας

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <PS2Keyboard.h>
#include <SoftwareSerial.h>
|
LiquidCrystal_I2C lcd(0x3F, 20, 4);
int i;
int a=0;
int b=0;

void setup() {
  lcd.init();
  lcd.backlight();
  Serial.begin(9600);
  lcd.setCursor(5,0);
  lcd.print("WELCOME!!");
  delay(3000);
  lcd.clear() ;
  lcd.blink_on() ;
}

void loop() {
  while(Serial.available()>0 ) {
    if(b==20 &&a<4){
      lcd.setCursor(0,a+1);
      b=0;
      a++;
    }

    char c=Serial.read();
    i++;
    if(i==1){
      b++;
      if (c == PS2_ESC) {
        lcd.clear();
        lcd.setCursor(0,0);
        b=0;
        a=0;
      }
      if (c == PS2_ENTER&&a<4) {
        a++;
        lcd.setCursor(0,a);
      }
    }
  }
}
```



```

b=0;
}
if (c == PS2_BACKSPACE&&b>1) {
  lcd.setCursor(b-2,a);
  lcd.print(" ");
  lcd.setCursor(b-2,a);
  b=b-2;
}
if (c == PS2_RIGHTARROW) {
  lcd.setCursor(0,3);
  for (int i = 0; i < 20; ++i)
  {
    lcd.write(' ');
  }
  lcd.setCursor(0,3);
  a=3;
  b=0;
}
if (c == PS2_LEFTARROW) {
  lcd.setCursor(0,1);
  for (int i = 0; i < 20; ++i)
  {
    lcd.write(' ');
  }
  lcd.setCursor(0,1);
  a=1;
  b=0;
}
if (c == PS2_DOWNARROW) {
  lcd.setCursor(0,2);
  for (int i = 0; i < 20; ++i)
  {
    lcd.write(' ');
  }
  lcd.setCursor(0,2);
  a=2;
  b=0;
}
if (c == PS2_UPARROW) {
  lcd.setCursor(0,0);
  for (int i = 0; i < 20; ++i)
  {
    lcd.write(' ');
  }
  lcd.setCursor(0,0);
  a=0;
  b=0;
}
if (c != PS2_ESC&&c!=PS2_ENTER&&c!=PS2_RIGHTARROW &&c!=PS2_LEFTARROW &&
c!=PS2_UPARROW &&c!=PS2_DOWNARROW &&c!=PS2_BACKSPACE &&a<4 ) {
  lcd.print(c);}
else {
  char c=Serial.read();
  i=0;
}
delay(10);
}
}

```

Οι τέσσερις πρώτες γραμμές του κώδικα αναφέρονται στις βιβλιοθήκες. Τρεις από αυτές είναι ίδιες με εκείνες που αναλύσαμε και πριν, έχουμε και μια καινούρια όμως την "LiquidCrystal_I2C.h". Η βιβλιοθήκη αυτή είναι απαραίτητη για την λειτουργία της οθόνης lcd και περιέχει όλες τις παραμέτρους και εντολές που χρειάζονται. Η επόμενη εντολή "LiquidCrystal_I2C lcd(0x3F,20,4)" είναι η παραμετροποίηση της οθόνης στην παρένθεση έχει τρία ορίσματα το πρώτο είναι το destination address το δεύτερο αναφέρεται στο πλήθος χαρακτήρων που μπορούν να υπάρξουν σε μια γραμμή και τέλος ο αριθμός 4 αναφέρεται στο πλήθος γραμμών που έχει η οθόνη. Στη συνέχεια έχουμε δήλωση τριών μεταβλητών το i το a και το b. Η πρώτη θα αναλυθεί αργότερα, οι άλλες δυο θα χρησιμοποιηθούν ως μετρητές η πρώτη ως μετρητής γραμμών και η άλλη ως χαρακτήρων. Ακλουθεί η συνάρτηση void setup(){...} οι δυο εντολές που βλέπουμε είναι η "lcd.init()" και η "lcd.backlight()", αναφέρονται στην οθόνη και η πρώτη ουσιαστικά την θέτει σε λειτουργία "on"(init από το initialize) η δεύτερη ενεργοποιεί τον φωτισμό της οθόνης. Η εντολή "Serial.begin(9600)" εξηγήθηκε και πριν. Οι επόμενες τρεις εντολές εμφανίζουν στην οθόνη ένα μήνυμα "WELCOME!!" για 3 δευτερόλεπτα. Λειτουργούν ως εξής: η "lcd.setCursor(5,0)" θέτει τον κέρσορα στην πρώτη γραμμή και πέμπτη στήλη εκτυπώνει το μήνυμα με την εντολή "lcd.print" και κάνει μια καθυστέρηση 3 sec ώστε να διαβαστεί το μήνυμα. Στην συνέχεια με την εντολή "lcd.clear()" καθαρίζουμε κάθε περιεχόμενο της οθόνης και τέλος με την εντολή "lcd.blink_on()" εμφανίζουμε έναν κέρσορα που αναβοσβήνει, είναι προαιρετική εντολή και λειτουργεί καθαρά για αισθητικούς λόγους.

Η επόμενη συνάρτηση είναι η void setup(){...}. Ξεκινάει με μια εντολή την while η οποία έχει ως συνθήκη να εκτελείται μόνο αν το "Serial.available()>0" η τιμή αυτή είναι πάντα θετική όταν δεν προκύπτει κάποιο error. Αν λοιπόν ισχύει η συνθήκη ο κώδικας προχωράει στην εκτέλεση 4 εντολών. Η πρώτη είναι μια if που σκοπό έχει την αλλαγή γραμμής της οθόνης αν οι χαρακτήρες που έχει η προηγούμενη έφτασαν τους 20. Λειτουργεί ως εξής: με την εντολή "if(b==20 && a<4)" ελέγχω αν έχω 20 χαρακτήρες και αν οι γραμμές δεν έχουν ξεπεράσει το 3(εφόσον η οθόνη έχει μόνο 4 γραμμές άρα και το a μπορεί να πάρει μόνο 4 τιμές (από 0 μέχρι 3), αν λοιπόν ισχύουν αυτές οι συνθήκες αλλάζουμε γραμμή στην οθόνη με την εντολή "lcd.setCursor(0,a+1)" το 0 αναφέρεται στην αρχή της γραμμής και το a+1 στην αμέσως επόμενη γραμμή. Στην συνέχεια μηδενίζουμε τον μετρητή των χαρακτήρων και αυξάνουμε το a, αφού αλλάξαμε γραμμή.

Επόμενη εντολή που ακλουθεί είναι η σημαντικότερη καθώς χάρη σ'αυτή μπορέσαμε να μεταφέρουμε δεδομένα από την μια πλακέτα στην άλλη. Η εντολή λοιπόν "char c=Serial.read()" αποθηκεύει σε μια μεταβλητή c τύπου char έναν χαρακτήρα τον οποίο διάβασε από την σειριακή θύρα. Αφού διαβάσει λοιπόν έναν χαρακτήρα αυξάνουμε τον μετρητή i, αν λοιπόν το i είναι ίσο με το 1 "if(i==1)" τότε και το b αυξάνει κατά ένα. Στη συνέχεια του κώδικα ακολουθούν δηλώσεις ειδικών πλήκτρων, όπως για παράδειγμα είναι το enter, το backspace, το esc και άλλα. Ο λόγος είναι πως το πρόγραμμα μας δεν καταλαβαίνει τα πλήκτρα αυτά ως ειδικά αλλά ως ακόμα έναν χαρακτήρα έτσι έπρεπε με διάφορες εντολές να κάνουμε τα πλήκτρα αυτά να κάνουν κάποιες λειτουργίες όπως για παράδειγμα η αλλαγή γραμμής, το σβήσιμο ενός χαρακτήρα, σβήσιμο όλων των περιεχομένων της οθόνης κα. Πρώτο μας πλήκτρο λοιπόν είναι το "esc". Με το πάτημα αυτού του πλήκτρου έχουμε ορίσει να καθαρίζει η οθόνη από όλα τα περιεχόμενα της ο τρόπος με τον οποίο λειτουργεί είναι ο εξής: αρχικά με μια εντολή if ελέγχουμε αν το πλήκτρο που πάτησε ο χρήστης είναι το πλήκτρο esc η εντολή αυτή είναι η "if (c == PS2_ESC)". Στην συνέχεια αν η συνθήκη είναι αληθής εκτελούμε την εντολή "lcd.clear()" και θέτουμε τον κέρσορα στην αρχή της οθόνης (θέση 0,0) στην συνέχεια μηδενίζουμε τους μετρητές a και b.

Ακλουθεί το ειδικό πλήκτρο "enter" το οποίο έχουμε ορίσει με το πάτημα του να αλλάζει η γραμμή. Και σε αυτή την περίπτωση με μια εντολή if ελέγχω αν πατήθηκε αυτό το πλήκτρο, αν η συνθήκη if είναι αληθής τότε αυξάνω τον μετρητή a καθώς θα αλλάξω γραμμή οπότε και ο μετρητής πρέπει να αυξηθεί κατά 1 θέτω τον κέρσορα στην από κάτω γραμμή με την εντολή "lcd.setCursor(0,a)" και μηδενίζω φυσικά το b αφού είμαστε πλέον σε νέα γραμμή οπότε και οι χαρακτήρες είναι 0.

Ύστερα ορίσαμε το πλήκτρο "backspace" να είναι υπεύθυνο για το σβήσιμο ενός χαρακτήρα σε μια γραμμή. Αυτή τη φορά στην συνθήκη If συναντάμε έναν ακόμα περιορισμό το $b > 1$ ο λόγος είναι πως για να έχει νόημα το πλήκτρο backspace θα πρέπει να έχει πατηθεί τουλάχιστον ένας χαρακτήρας. Ο τρόπος λοιπόν που λειτουργεί το πλήκτρο αυτό είναι ο εξής: αφού πατηθεί το πλήκτρο θέτουμε τον κέρσορα στην παραπροηγούμενη θέση του, πλην 2 δηλαδή. Ο λόγος που μπαίνει το 2 και όχι το ένα είναι διότι το backspace που θεωρείται ως χαρακτήρας θα πιάσει μια θέση στην οθόνη και ακόμη μια θέση πιάνει ο χαρακτήρας που θέλουμε να σβήσουμε. Στην συνέχεια εκτυπώνουμε ένα κενό χαρακτήρα με την εντολή "lcd.print(" ")" το οποίο δίνει την εντύπωση πως ο χαρακτήρας που θέλουμε να διορθώσουμε σβήνει, ενώ στην ουσία απλά πάνω του εκτυπώνουμε έναν κενό χαρακτήρα και ξαναθέτουμε τον κέρσορα στην προηγούμενη θέση. Τέλος μειώνουμε τον μετρητή b κατά 2.

Στις παρακάτω γραμμές του κώδικα ορίζουμε 4 ακόμα ειδικά πλήκτρα το "uparrow" το "leftarrow" το "downarrow" και τέλος το "rightarrow", τα πλήκτρα αυτά σβήνουν όλο το περιεχόμενο μιας γραμμής. Το πάνω βέλος της πρώτης το αριστερό της δεύτερης το κάτω της τρίτης και το δεξί της τέταρτης. Τα πλήκτρα αυτά ακολουθούν την ίδια λογική οπότε θα αναλύσω τον τρόπο με τον οποίο λειτουργεί ένα εξ'αυτών. Ας αναλύσουμε λοιπόν το ειδικό πλήκτρο "UPARROW" το οποίο και συναντάμε προς το τέλος του κώδικά μας, (όπως αναφέρθηκε και πριν σκοπός του πλήκτρου αυτού είναι να καθαρίσει όλα τα περιεχόμενα της πρώτης γραμμής). Λειτουργεί ως εξής: πρώτη εντολή είναι φυσικά μια if η οποία ελέγχει αν ο χρήστης πάτησε το πλήκτρο για το οποίο μιλάμε. Αν ισχύει η συνθήκη θέτουμε τον κέρσορα στη θέση 0,0 δηλαδή στην αρχή της οθόνης και ξεκινάμε μια επανάληψη for η οποία θα γίνει 20 φορές μέσα στην επανάληψη αυτή θα γεμίσω την οθόνη με κενούς χαρακτήρες ' ' τέλος θα θέσω ξανά τον κέρσορα από την αρχή και θα μηδενίσω τις τιμές των μετρητών a και b.

Τελειώνοντας ακλουθεί μια ακόμα εντολή if και else. Η πρώτη λοιπόν κάνει μια σύγκριση αν το πλήκτρο που πάτησε ο χρήστης δεν είναι κάποιος ειδικός χαρακτήρας όπως esc,enter,backspace κλπ τότε το πρόγραμμα θα εμφανίζει τον χαρακτήρα που πάτησε ο χρήστης(lcd.print(c)). Αλλιώς περιμένει από τον χρήστη να πατήσει κάποιο πλήκτρο, εδώ βλέπουμε και την χρήση του i που αναφέρθηκε από την αρχή ουσιαστικά είναι ένα σημείο αναφοράς (true or false) λειτουργεί σαν break point του κώδικα καθώς όσο το i παραμένει 0 ο κώδικας δεν κάνει απολύτως τίποτα, ενώ από την στιγμή που πατηθεί πλήκτρο και επιτυχώς αναγνωριστεί από την συσκευή μας είναι ελεύθερο να προχωρήσει και να κάνει όλες τις λειτουργίες που αναφέρθηκαν πριν. Τελευταία γραμμή του κώδικα είναι μια delay αμελητέου χρόνου η οποία όμως χρησιμεύει στην αποφυγή τυχόν σφαλμάτων κατά την εκτέλεση του κώδικα.

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ

5.1 Σύνοψη πτυχιακής εργασίας

Συνοψίζοντας λοιπόν έχουμε καταφέρει να δείξουμε πως με απλές ενέργειες μπορούμε να επιτύχουμε την ασύρματη επικοινωνία μέσω πομπών Xbee. Ακλουθώντας 3 απλά βήματα. Το πρώτο είναι να προμηθευτούμε υλικά για τον σκοπό μας, υλικά τα οποία είναι πολύ εύκολο να βρεθούν στην αγορά και ανάλογα τις απαιτήσεις μας έχουν το αντίστοιχο κόστος. Το δεύτερο βήμα είναι να κατανοήσουμε τον τρόπο λειτουργίας των δυο απαραίτητων προγραμμάτων του Xctuu και Arduino. Απαιτείται ελάχιστος χρόνος και ακόμα πιο ελάχιστες γνώσεις καθώς στο διαδίκτυο υπάρχουν πλήθος απλών και εύκολων παραδειγμάτων, συνεπώς ακόμα και κάποιος με μηδενικές γνώσεις πάνω στο αντικείμενο θα μπορούσε να ασχοληθεί με κάποιο project. Εκτός από αυτά όμως υπάρχουν και βίντεο που εξηγούν με λεπτομέρεια πως δουλεύει το κάθε τι. Τέλος το τρίτο βήμα είναι να σκεφτούμε τον κώδικα μας πως θέλουμε να λειτουργεί με ποιες εντολές μπορούμε να το επιτύχουμε αυτό, συμβουλευόμενοι πάντα τα ήδη υπάρχοντα στο διαδίκτυο έτοιμα προγράμματα τα οποία ίσως μοιάζουν με αυτό που θέλουμε να φτιάξουμε επομένως μπορούμε κάλλιστα να πάρουμε κάποιες ιδέες από εκεί. Τελειώνοντας λοιπόν να πω, πως στην παρούσα πτυχιακή χάρη στην τεχνολογία που προσφέρει το Xbee σε συνδυασμό με το Arduino έγινε εφικτή η ασύρματη επικοινωνία δίνοντας μας την δυνατότητα να στέλνουμε μηνύματα/προτάσεις από το πληκτρολόγιο και αυτά με την σειρά τους να εμφανίζονται στην led οθόνη μας.

5.1 Προοπτικές

Η δυνατότητα της ασύρματης επικοινωνίας δεν περιορίζεται φυσικά μόνο στο να μπορούμε να στέλνουμε κάποιους χαρακτήρες ασύρματα. Μας δίνεται για παράδειγμα η δυνατότητα λήψης μετρήσεων στάθμης νερού ή στάθμης πετρελαίου της δεξαμενής που έχουμε στο σπίτι μας. Ακόμα χρησιμοποιώντας αισθητήρες στο σπίτι μας θα μπορούσαμε να στέλνουμε ασύρματα δεδομένα όπως αν κάποιο παράθυρο η πόρτα είναι ανοιχτή. Αντίστοιχη τεχνολογία σιγά σιγά γίνεται μέρος του λεγομένου "Εξυπνου Σπιτιού" το οποίο έχει αισθητήρες παντού, που εντοπίζουν την θερμοκρασία, το ηλιακό φως, την κίνηση μέσα στο σπίτι μας κα. Όλοι αυτοί οι αισθητήρες μπορούν να προγραμματιστούν να εκτελούν πολλαπλά σενάρια και να παραμετροποιηθούν με την χρήση μιας ασύρματης οθόνης. Φυσικά στο παράδειγμα που υλοποιήσαμε δεν έχουμε ασύρματη οθόνη ώστε να εκχωρούμε δυναμικά τιμές/παραμέτρους. Μπορούμε όμως για παράδειγμα αντί για την οθόνη lcd μας να βάλουμε ένα μηχανισμό στην πόρτα η οποία διαθέτει έναν κινητήρα ο οποίος με την σειρά του μπορεί να ενεργοποιηθεί αν πατηθεί στο πληκτρολόγιο το πλήκτρο "A" και να σταματήσει με το πλήκτρο "B". Υπάρχει ακόμα και η δυνατότητα να σταματήσει μετά από κάποια δευτερόλεπτα που επιθυμούμε εμείς. Ακόμα μια χρήση του Arduino θα ήταν αν βάζαμε στους διακόπτες της αυλής του σπιτιού μας ρελέ ρευματώθησης (καστάνιας) έτσι θα μπορούσαμε με το πάτημα ενός πλήκτρου να ανοίξουμε όλα τα φώτα στην αυλή. Το Arduino διαθέτει πληθώρα shield ένα εξ αυτών είναι αυτό που αναγνωρίζει την φωνή μας σε ένα πιο περίπλοκο σενάριο λοιπόν αν θέλαμε να γίνουμε ακόμα πιο εφορευτικοί θα μπορούσαμε με την φωνή μας να ανοίξουμε τα φώτα αυτά. Υπάρχουν λοιπόν άπειρες δυνατότητες που μας δίνει το Arduino με την βοήθεια της ασύρματης επικοινωνίας ανάλογα πάντα με τις ανάγκες και τις απαιτήσεις που έχει ο καθένας. Χάρη στην ευκολία χρήσης που προσφέρει το καθιστά προσιτό για οποιονδήποτε επιθυμεί μελλοντικά να ασχοληθεί με την τεχνολογία αυτή.

ΚΕΦΑΛΑΙΟ 6

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Ανάπτυξη εφαρμογών με το Arduino,"Παναγιώτης Μ. Παπάζογλου",2017
- [2] Ενσωματωμένα συστήματα, οι μικροελεγκτές AVR και ARDUINO,"Δημήτρης Πογαρίδης",2015
- [3] Ο προγραμματισμός του μικροελεγκτή AVR ATmega328 με τη χρήση της πλατφόρμας ARDUINO," Δημήτρης Καλοφωλιάς",2017
- [4] Arduino for Dummies,"John Nussey",2016
- [5] Building Wireless Sensor Networks: with ZigBee, XBee, Arduino, and Processing,"Robert Faludi",2010
- [6] The Hands-on XBEE Lab Manual: Experiments that Teach you XBEE Wireless Communications,"Jonathan Titus",2012
- [7] XBee ZigBee Development Workshop Kindle Edition,"by Agus Kurniawan",2017
- [8] Αξιολόγηση απόδοσης ασύρματων δικτυακών τοπολογιών αρχιτεκτονικής Mesh, Πτυχιακή εργασία των σπουδαστών Βενέρη Γεώργιου & Μπούτζουκα Αγησίλαου,"ΤΕΙ Κρητης",2010
- [9] Ασύρματα πρωτόκολλα χαμηλής ισχύος, Πτυχιακή εργασία του Τζεμανάκη Ευτύχιου,"ΤΕΙ Κρητης",2014
- [10] Σεμινάριο Ηλεκτρονικού Τομέα/Εφαρμογές Arduino,"1ο ΕΠΑΛ Περάματος -7ο ΕΚ Πειραιά",2014
- [11] Προγραμματίζοντας με τον μικροελεγκτή Arduino,"Εμμανουήλ Πουλάκης",2015
- [12] <https://grobotronics.com/>
- [13] <https://www.arduino.cc/>
- [14] <https://el.wikipedia.org/wiki/Arduino>
- [15] <https://en.wikipedia.org/wiki/XBee>
- [16] <https://www.digi.com/pdf/xbee-802-15-4-protocol-comparison>
- [17] <https://www.hackster.io/xbee/projects>