

Σύστημα ασφαλείας χώρου με χρήση ασύρματων αισθητήρων Sun SPOT

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μαρασίδης Γρηγόρης
Α.Μ. 507

ΤΕΙ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΤΜΗΜΑ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ
ΔΙΚΤΥΩΝ

Ιούνιος 2013

Επιβλέπων καθηγητής –

ΕΥΧΑΡΙΣΤΙΕΣ

Σύστημα ασφαλείας χώρου με χρήση ασύρματων αισθητήρων Sun SPOT

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Περιεχόμενα

1. ΕΙΣΑΓΩΓΗ	6
1.1 ΜΙΑ ΣΥΝΤΟΜΗ ΑΝΑΔΡΟΜΗ ΤΩΝ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ.....	8
1.2 Ο ΡΟΛΟΣ ΤΩΝ ΔΙΚΤΥΩΝ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ.....	13
1.3 ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ	17
2. ΑΣΥΡΜΑΤΟΙ ΑΙΣΘΗΤΗΡΕΣ SUN SPOT	18
2.1 ΕΙΣΑΓΩΓΙΚΑ.....	18
2.2 ΓΕΝΙΚΑ	22
2.3 ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	23
ΠΕΡΙΒΑΛΛΟΝ ΧΕΙΡΙΣΜΟΥ	25
ΧΡΗΣΙΜΟ ΛΟΓΙΣΜΙΚΟ ΓΙΑ SUN SPOT ΣΥΣΚΕΥΕΣ	26
ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ	27
ΔΥΝΑΤΟΤΗΤΑ ΚΛΙΜΑΚΩΣΗΣ(SCALABILITY)	27
3. ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	29
3.1 ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΑΣ ΣΤΑΘΜΟΥ ΕΛΕΥΘΕΡΗΣ ΠΕΡΙΟΧΗΣ.....	30
3.2 ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΑΣ ΣΤΑΘΜΟΥ ΒΑΣΗΣ.....	32
4. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	35
4.1 ΠΑΡΑΤΗΡΗΣΕΙΣ.....	35

4.1.1	ΕΥΑΙΣΘΗΣΙΑ ΤΟΥ ΑΙΣΘΗΤΗΡΑ.....	36
4.1.2	ΑΝΤΟΙΧΗ ΜΠΑΤΑΡΙΑΣ.....	36
4.1.3	ΕΝΔΕΧΟΜΕΝΕΣ ΕΠΙΘΕΣΕΙΣ.....	37
4.2	ΓΕΝΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΣΦΑΛΜΑ! ΔΕΝ ΕΧΕΙ ΟΡΙΣΤΕΙ ΣΕΛΙΔΟΔΕΙΚΤΗΣ.	
	ΠΑΡΑΡΤΗΜΑ	86
	ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΣΤΑΘΜΟΥ ΒΑΣΗΣ.....	42
	ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΣΤΑΘΜΟΥ ΕΛΕΥΘΕΡΗΣ ΠΕΡΙΟΧΗΣ.....	70
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	86

1. Εισαγωγή

With the evolution of the internet and the expansion of computer networks, *network security* is becoming one of the most important factors for companies to consider. By increasing network security, you decrease the chance of privacy spoofing, identity or information theft and so on. Hacker tools have become more and more sophisticated, intelligence is no longer a requirement to hack someone's computer or server. Of course, there are individuals that have developed sophisticated skills and know how to breach into a user's privacy in several ways, but these types of individuals are less common than in the past.

Today, most malicious users do not possess a high level of programming skills and instead make use of tools available on the Internet. There are several stages that an attacker has to pass through to successfully carry out an attack. Originally, the term hacker defined an individual who possessed strong programming skills and was involved in developing new ways to protect networks against attacks.



These days, a hacker is more commonly known as someone that uses computing skills to break into someone's account or computer and compromise their private information. You've probably heard other terms that define such individuals like cracker, black hat, phreaker, spammer or phisher.

All these terms define a person that uses his or her computing skills to steal important data. These individuals use different techniques that define them as a malicious user. For example, a spammer is someone who uses email services to send malicious emails that often carry viruses. A phisher is an individual who's specialized in duplicating real content like emails, websites or services, in order to trick a user into providing confidential information.

Με τη εξέλιξη του Internet και την εξαπλωση των δικτυων υπολογιστων, **η διαδικτυακή ασφάλεια** γίνεται όλο και περισσότερο ένας από τους πιο σημαντικούς παράγοντες για τις επιχειρήσεις. Αυξάνοντας τη δικτυακή ασφάλεια, μειώνουμε τις πιθανότητες του privacy spoofing, identity theft κτλ. Τα εργαλεία των χάκερ γίνονται όλο και πιο φιλοσοφημένα μέρα με τη μέρα, δν είναι πλέον απαραίτητες βαθιές τεχνικές γνώσεις για να πραγματοποιηθεί μια επίθεση σε έναν υπολογιστή ή ένα server. Φυσικά θα υπάρχουν άτομα τα οποία θα έχουν αναπτύξει φιλοσοφημένες τεχνικές και θα γνωρίζουν πώς να εισχωρήσουν στα προσωπικά δεδομένα ενός χρήστη με διάφορους τρόπους, αλλά αυτά τα άτομα είναι λιγότερο συνηθισμένα στις μέρες μας από ότι στο παρελθόν.

Σήμερα οι περισσότεροι κακόβουλοι χρήστες δεν κατέχουν υψηλό επίπεδο προγραμματιστικών ικανοτήτων και έναντι αυτού κανουν χρήση των έτοιμων εφαρμογών που υπάρχουν διαθέσιμα στο Διαδίκτυο. Υπάρχουνε πολλά επίπεδα μέσα από τα οποία θα πρέπει να περάσει ο εισβολέας για να μπορέσει επιτυχώς να πραγματοποιήσει μια επίθεση. Αρχικά ο όρος χάκερ όριζε ένα άτομο το οποίο κατείχε δυνατές προγραμματιστικές γνώσεις και συχνά προσπαθούσε να αναπτύξει καινούριους τρόπους προστατίας από δικτυακές επιθέσεις. Αυτές τις μέρες ο όρος χάκερ είναι πιο γνωστός σαν κάποιος που χρησιμοποιεί γνώσεις πληροφορικής ώστε να εισχωρήσει στο λογαριασμό ή στον υπολογιστή ενός χρήστη και να κλέψει να προσωπικά τους δεδομένα. Πιθανώς να έχετε ακούσει και άλλους όρους οι οποίοι ορίζουν τέτοιου είδους άτομα όπως cracker, black hat, phreaker, spammer ή phisher.

Όλοι αυτοί οι όροι έχουν να κάνουν με ανθρώπους που χρησιμοποιούν τις υπολογιστικές και προγραμματιστικές τους γνώσεις για να κλέψουν σημαντικά δεδομένα. Αυτά τα άτομα χρησιμοποιούν διαφορετικές τεχνικές οι οποίες τους κατηγοριοποιούν ως κακόβουλους χρήστες. Για παράδειγμα ο spammer είναι κάποιος που χρησιμοποιεί το email κάποιου άλλου για να στείλουν καλόβουλα email τα οποία συχνά περιέχουν ιούς. Ο phisher είναι ένα άτομο που ειδικεύεται στο να αντιγράφει πραγματικό υλικό όπως email, websites, services, για να κοροϊδέψει ένα χρήστη στό να του παραχωρήσει ευαίσθητα δεδομένα.

1.1 Μια σύντομη αναδρομή των επικοινωνιακών συστημάτων

Ο άνθρωπος από τα αρχαία χρόνια της ιστορίας του ήρθε αντιμέτωπος με ένα από τα πιο βασικά προβλήματα: τη μετάδοση της πληροφορίας. Η δημιουργία ενός τρόπου επικοινωνίας ήταν αναγκαία. Η γνώση μίας πληροφορίας και η δυνατότητα μετάδοσης της από ένα άτομο σε ένα άλλο και στη συνέχεια από ένα σημείο σε ένα άλλο απομακρυσμένο ήταν ζωτικής σημασίας. Ιστορικά λοιπόν από το 3500 π.Χ. αναπτύχθηκε η γλώσσα και σήματα καπνού και τύμπανα χρησιμοποιήθηκαν για την επικοινωνία εξ' αποστάσεως. Πери το 1500π.Χ. οι πολιτισμοί της μέσης ανατολής ανέπτυξαν τις αρχικές μορφές της αλφαβήτου. Περίπου 1000 χρόνια μετά βλέπουμε τη χρήση ταχυδομικών περιστεριών στους Ολυμπιακούς Αγώνες. 105μ.Χ. ανακαλύφθηκε το χαρτί στην Κίνα από τον Ts'ai Lun ενώ πολύ αργότερα το 1440μ.Χ. ο Γερμανός Johannes Gutenberg ανακάλυψε το πρώτο σύστημα εκτύπωσης με κινητά τυπογραφικά στοιχεία, που είναι γνωστό στο δυτικό κόσμο. Ο Γάλλος Blaise Pascal ανέπτυξε το 1600μ.Χ. πρότυπα υπολογιστικών μηχανών, που οδήγησαν στους σημερινούς Η/Υ.

Το 1821 ο Charles Wheatstone αναπαράγει ήχο σε ένα πρωτόγονο μουσικό κουτί το 1821 το πρώτο μικρόφωνο εφευρέθηκε. 1830: Ο Αμερικάνος επιστήμονας και καθηγητής Joseph Henry μετάδωσε το πρώτο ηλεκτρικό σήμα δείχνοντας πως ο ηλεκτρομαγνητισμός μπορεί να κάνει πολύ περισσότερα από την δημιουργία ηλεκτρικού ρεύματος και την ανύψωση φορτίων. 7 χρόνια μετά ο Morse κάνει δημόσια επίδειξη του τηλέγραφού του. 1843: Επινοήθηκε το FAX από τον Σκωτσέζο φυσικό Alexander Bain. 1864: Ο Άγγλος James Clerk Maxwell διετύπωσε μια θεωρία περί ηλεκτρομαγνητισμού. Η θεωρία αυτή έθεσε τις βάσεις για την ανακάλυψη του ραδιοφώνου. 1876: Ο Αμερικανός Alexander Graham Bell κατοχύρωσε με δίπλωμα ευρεσιτεχνίας το τηλέφωνο. 1877: Ο Thomas A. Edison κατοχύρωσε με δίπλωμα ευρεσιτεχνίας το φωνογράφο. 1930:Γερμανοί μηχανικοί εφηύραν το μαγνητόφωνο (καταγραφή ήχων σε ταινίες). 1944:Η εταιρία IBM κατασκεύασε τον υπολογιστή Harvard Mark I. 1947:Οι Αμερικανοί John Bardeen, Walter Brattain και William Shockley εφηύραν το τρανζίστορ για λογαριασμό της εταιρείας Bell Telephone Laboratories.1947:Ο Βρετανός μηχανικός Dennis Gabor εφηύρε την ολογραφία. 1950:τα τηλεοπτικά δίκτυα επιχειρούν τις πρώτες μαγνητοσκοπήσεις (videotape). 1952:

Η Ιαπωνική εταιρεία SONY προωθεί στην αγορά τα πρώτα ραδιόφωνα τρανζίστορ τσέπης. 1978:Ο Μουσικός Οργανισμός της Αμερικής (MCA) προωθεί στην αγορά τις πρώτες 50 συσκευές που παίζουν οπτικούς δίσκους (videodisk), που στην Αμερική κοστίζουν 20.000 δολάρια η κάθε μία. 1982:Η Ιαπωνική εταιρεία Hattari-Seiko κατασκευάζει την επίπεδη οθόνη χρησιμοποιώντας τεχνολογία υγρών διόδων (LCD).

1983:Εισάγονται στο εμπόριο οι πρώτοι σύμπηκνοι δίσκοι (CD).1988:Ερευνητές στο M.I.T. (Τεχνολογικό Ινστιτούτο της Μασαχουσέτης) αναπτύσσουν έναν τρόπο να συμπυκνώνουν εικόνες που διαρκούν μέχρι μία ώρα σε έναν πτυκτό δίσκο που τον αναφέρουν ως “κινηματογράφο της τσέπης”. 1988:Εγκαθίσταται το πρώτο τηλεφωνικό καλώδιο με οπτικές ίνες που διασχίζει τον Ατλαντικό. Κοστίζει 362 εκατ.δολάρια και με αυτό μπορούν να γίνουν 40.000 ηλεφωνικές συνδιαλέξεις ταυτόχρονα. 1989: Ο Αμερικάνος Nube, κατασκευαστής υπερυπολογιστών ανακοίνωσε ότι έχει αναπτύξει μία μηχανή που μπορεί να εκτελέσει 27 δισ. υπολογισμούς το δευτερόλεπτο

1.2 Ο ρόλος τών δικτύων επικοινωνιακών συστημάτων

- **Οικονομική παραγωγικότητα**

Τα δίκτυα επιτρέπουν τη διεθνή επικοινωνία με τους προμηθευτές, τους πελάτες και άλλα ενδιαφερόμενα μέρη. Τα δίκτυα μπορούν να συμβάλουν στο άνοιγμα μαζικών, παγκόσμιων αγορών στις αναπτυσσόμενες χώρες. Αυτό βέβαια εγείρει το φάσμα της μείωσης των μισθών και την καταστροφή του περιβάλλοντος στα αναπτυγμένα έθνη. Στις αναπτυγμένες χώρες θα πρέπει να επισημάνουμε τις θετικές επιπτώσεις - περισσότερες ευκαιρίες επενδύσεων και εμπορίου, την αύξηση της αποτελεσματικότητας, το χαμηλότερο κόστος των αγαθών και των υπηρεσιών . Ταυτόχρονα, θα πρέπει να υπάρξει μέριμνα για τις αρνητικές συνέπειες επιδιώκοντας διεθνή περιβαλλοντικά και εργασιακά πρότυπα.

- **Εκπαίδευση**

Τα πλεονεκτήματα των δικτύων στην ακαδημαϊκή κοινότητα είναι προφανή – οι βάσεις δεδομένων είναι κοινές, οι εργασίες αναρτώνται στο διαδίκτυο και σχολιάζονται, η έρευνα αναπτύσσεται με συνεργασίες και αυξάνεται η διεθνής επικοινωνία μεταξύ πανεπιστημίων και σπουδαστών. Στις αρχές του αιώνα, η έρευνα συγκεντρώνεται σε λίγα κέντρα. Με την ανάπτυξη των δικτύων των Η/Υ αυξήθηκε η διεθνής επικοινωνία - περιοδικά και συνέδρια – και οδήγησε στην παγκόσμια διασπορά της έρευνας.

- **Φροντίδα Υγείας**

Βλέπουμε, επίσης, την εφαρμογή των δικτύων στην υγειονομική περίθαλψη στις αναπτυσσόμενες χώρες: η παροχή του ηλεκτρονικού ταχυδρομείου, των ηλεκτρονικών εκδόσεων, η πρόσβαση σε βάσεις δεδομένων, η εξ αποστάσεως εκπαίδευση, οι συμβουλευτικές υπηρεσίες μέσω Διαδικτύου και υποστήριξη, και η συνεργασία μεταξύ βιβλιοθηκών. Οι ηλεκτρονικοί φάκελοι αντικαθιστούν τους συμβατικούς φακέλους. Επιτρέπουν την ηλεκτρονική αποθήκευση, ανάκτηση και τροποποίηση των πληροφοριών των ασθενών και οι ιατρικές υπηρεσίες / τμήματα μπορούν να συνεργάζονται με σκοπό την καλύτερη παροχή ιατρικής φροντίδας.

Τα τμήματα Ακτινολογίας αποθηκεύουν ηλεκτρονικά, εμφανίζουν και μεταφέρουν μεγάλες ψηφιακές εικόνες σε οποιοδήποτε οργανισμό η υπηρεσία απαιτείται. Οι ιατροί μπορούν χρησιμοποιώντας υπολογιστή στο σπίτι τους να αποκτήσουν πρόσβαση στα CPOE (Computerized Physician Order Entry) και να επικοινωνήσουν μέσω του δικτύου με ιατρικά τμήματα δίνοντας οδηγίες αγωγής για συγκεκριμένο ασθενή. Μπορεί κανείς να φανταστεί πολλές εφαρμογές δικτύωσης στον τομέα της υγείας σε μια χώρα σαν την Κούβα ή την Κίνα, όπου «ξυπόλυτοι γιατροί» και άνθρωποι παραϊατρικών υπηρεσιών υπηρετούν φτωχές κοινότητες και αγροτικές περιοχές.

-

- **Δημοκρατία και τα Ανθρώπινα Δικαιώματα**

Τα δίκτυα συνεισφέρουν στη δημοκρατία παρέχοντας στους ανθρώπους που ζουν κάτω από δικτατορικά καθεστώτα πληροφορίες και ιδέες, και τη δυνατότητα να συντονίσουν την πολιτική δραστηριότητα εντός των εθνών τους. Ο Kedzie παρουσιάζει μια πολυμεταβλητή στατιστική ανάλυση, η οποία δείχνει ότι η διασύνδεση είναι καλύτερος προγνωστικός δείκτης της δημοκρατίας από το σχολείο, το ΑΕΠ, το προσδόκιμο ζωής, την εθνική ομοιογένεια ιδίως σε περιοχές αναδυόμενης δημοκρατίας. Καταλήγει στο συμπέρασμα ότι «η πιο πιθανή σχέση μεταξύ της δημοκρατίας και της δικτυωμένης επικοινωνίας (και ίσως και της οικονομικής ανάπτυξης) μπορεί να είναι ένας ενάρετος κύκλος με θετικές ανατροφοδοτήσεις και στις δύο κατευθύνσεις".

Πολλές οργανώσεις που υποστηρίζουν τα ανθρώπινα δικαιώματα και τη δημοκρατία στις αναπτυσσόμενες χώρες χρησιμοποιούν το Διαδίκτυο. Ο Σύλλογος για την Προοδευτική Επικοινωνίας (APC), έχει ηγετική θέση σε αυτή την προσπάθεια από το 1989, συντονίζει τη λειτουργία και την ανάπτυξη των δικτύων αφιερωμένων στην ειρήνη, την οικολογία και τα ανθρώπινα δικαιώματα. Από τον Σεπτέμβριο του 1995, η APC μπορεί να έχει έναν μόνιμο αντιπρόσωπο στον ΟΗΕ, και το δικαίωμα να υποβάλει γραπτές δηλώσεις προς το συμβούλιο, και να προτείνει θέματα στην ημερήσια διάταξη για την εξέταση από το Συμβούλιο .

- **Ποιότητα Ζωής**

Έχουμε τη ρύπανση του περιβάλλοντος και περιορισμένες πηγές ενέργειας. Η εξάπλωση των δικτύων μας επιτρέπει να υποκαταστήσει η επικοινωνία τις μεταφορές. Το πιο σημαντικό είναι ότι οι άνθρωποι της υπαίθρου δεν χρειάζεται να μετακινηθούν. Οι άνθρωποι συρρέουν στις πόλεις σε αναζήτηση καλύτερης εκπαίδευσης, υγειονομικής περίθαλψης, και απασχόλησης. Με την ανέγερση όμως υπερβολικού αριθμού ψηλών κτιρίων το περιβάλλον υποβαθμίζεται και εμφανίζονται οι παρενέργειες της αστικοποίησης όπως η κυκλοφοριακή συμφόρηση και η αύξηση της εγκληματικότητας. Αν οι συνθήκες της αγροτικής ζωής βελτιωθούν, λιγότεροι θα αισθάνονται υποχρεωμένοι να μετακινηθούν στα μεγάλα αστικά κέντρα. Θα μπορούσε η χρήση των δικτύων να περιορίσει την αστικοποίηση, παρέχοντας πλεονεκτήματα στις μικρές πόλεις και στις αγροτικές περιοχές; Η αγροτική ζωή μπορεί να είναι επιθυμητή εάν επιλέγεται ελεύθερα, αντί να επιβάλλεται από την ανάγκη. Η επιλογή προϋποθέτει την ανάπτυξη της τεχνολογίας των επικοινωνιών που διευρύνει τους ορίζοντες και μας βοηθά στην καλύτερη ποιότητα ζωής.

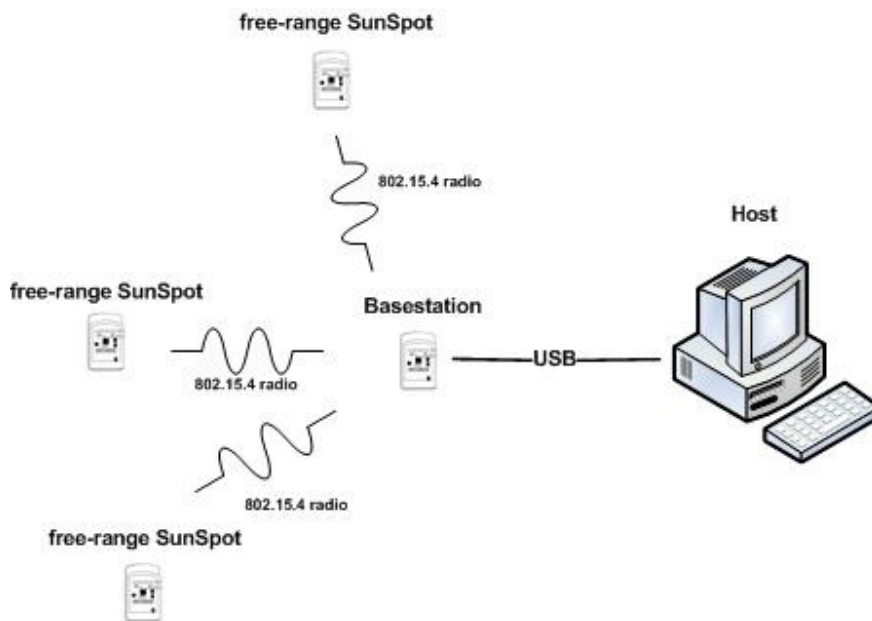
1.3 Σκοπός της εργασίας

Ο βασικός στόχος της εργασίας είναι η υλοποίηση ενός συστήματος ασφαλείας χώρου με τη χρήση ασύρματων αισθητήρων Sun Spot. Μετά την ολοκλήρωση του συστήματος, θα γίνει ανάλυση των αποτελεσμάτων ως προς την αποδοτικότητα του.

2. Ασύρματοι αισθητήρες Sun SPOT

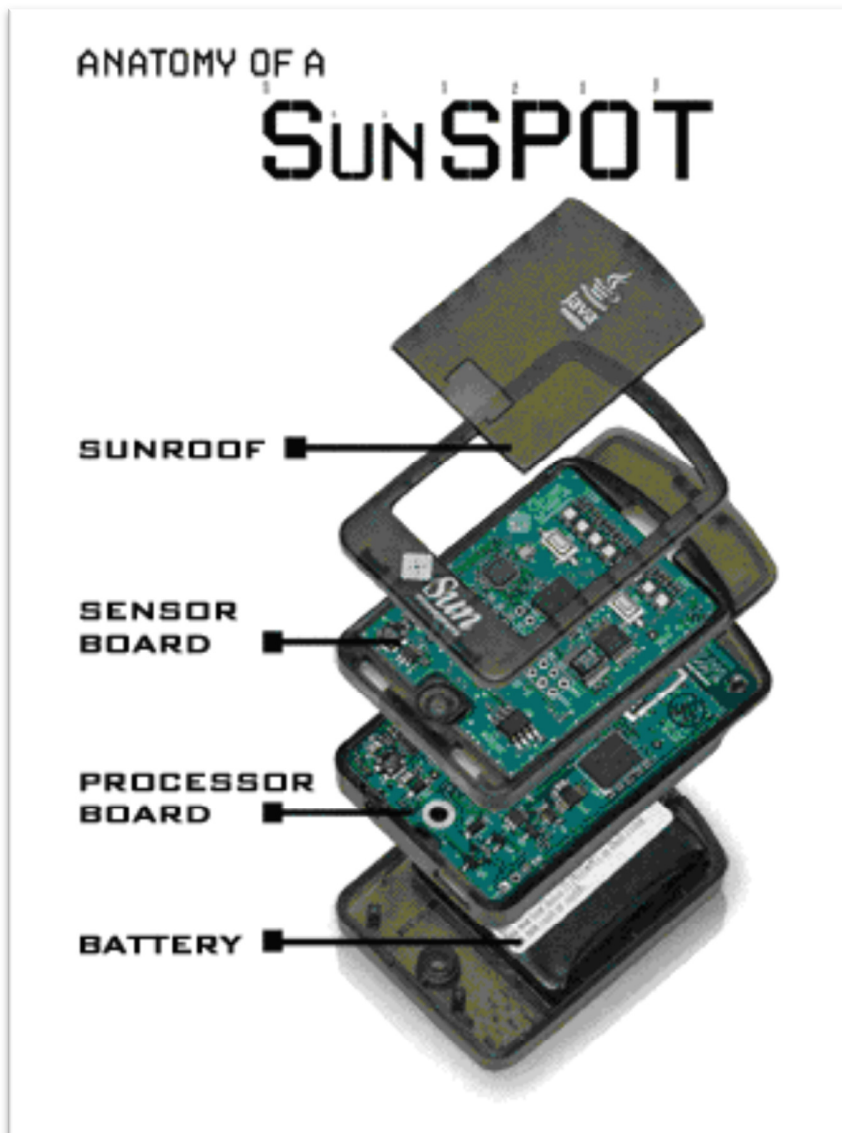
2.1 Εισαγωγικά

Σήμερα, το διαδίκτυο αποτελείται από εκατομμύρια υπολογιστές, όμως η φύση του αλλάζει καθημερινά. Νέοι τύποι συσκευών αρχίζουν να συνδέονται μέρα με τη μέρα στο διαδίκτυο. Αυτοκίνητα, ιατρικές συσκευές, παιχνίδια θα μπορούν να διαχειρίζονται και να ελέγχουν δεδομένα μεταξύ τους αλλά και ανά τον κόσμο. Αυτές οι συσκευές θα ξεπεράσουν κατά πολύ τον αριθμό των μέχρι τώρα υπολογιστών. Ήδη δισεκατομμύρια κινητά με λογισμικό java επικοινωνούν μεταξύ τους σε καθημερινή βάση. Προβλέπεται, κάποια μέρα στο κοντινό μέλλον να επικοινωνούν μεταξύ τους τρισεκατομμύρια τέτοιων συσκευών.



Για χρόνια, η εταιρία Sun πίστευε ότι μια μέρα τα πάντα θα είναι κομμάτι του vinternet. Αυτό το όραμα, του «internet όλων των πραγμάτων»

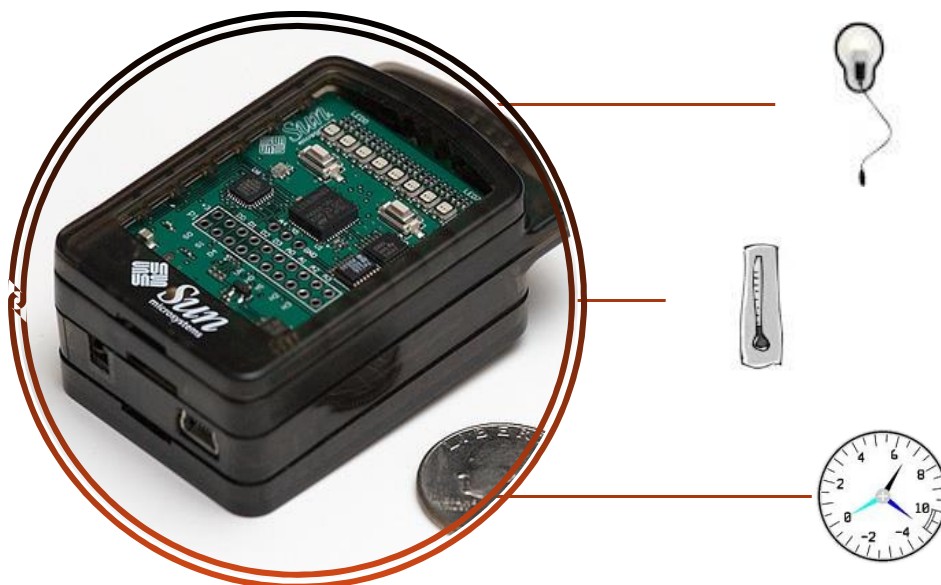
περιλαμβάνει όχι μόνο υπολογιστές που θα επικοινωνούν μεταξύ τους και θα ανταλλάσσουν δεδομένα, αλλά θα είναι όλα συνδεδεμένα μεταξύ τους όλα θα επικοινωνούν συνεχώς. Αυτοκίνητα, δίκυκλα, κινητά, ενσωματωμένα συστήματα ακόμη και δέντρα θα μπορούν να συλλέγουν και να παραδίδουν αμφίδρομα πληροφορίες μέσω αυτών των συσκευών.



Τα τελευταία χρόνια η SUN έχει βάλει στόχο να δημιουργήσει ένα δίκτυο υποδομής ώστε να χειρίζεται μεγάλες ποσότητες δεδομένων δουλειές δηλαδή που μπορεί να επιτελέσει ένας server, ένα σύστημα αποθήκευσης δεδομένων, λογισμικό διαχείρισης δεδομένων και πάει λέγοντας. Και τώρα με τα SUN SPOT (Small Programmable Object Technology) ένα τεχνολογικό εγχείρημα της SUN κατάφερε να αυτή την έμπνευση να την ενσαρκώσει σε μια συσκευή.

2.2 Γενικά

- Τα Sun SPOT (Sun Small Programmable Object Technology) είναι προγραμματιζόμενοι (*NetBeans, Eclipse* κ.α.) ασύρματοι αισθητήρες οι οποίοι ανεπτύχθησαν από τη *Sun Microsystems*.
- Βασίζονται στο πρωτόκολλο *IEEE 802.15.4*
- Είναι χτισμένα με το Squawk Java Virtual Machine
- Οι σταθμοί βάσης (base station) μπορούν να χρησιμοποιούν οποιαδήποτε βιβλιοθήκη Java
- Οι σταθμοί ελεύθερης περιοχής διαθέτουν αισθητήρες φωτινότητας, θερμοκρασίας και επιτάχυνσης



2.3 Τεχνικά χαρακτηριστικά

Processing

- 180 MHz 32 bit ARM920T core - 512K RAM - 4M Flash
- 2.4 GHz IEEE 802.15.4 radio with integrated antenna
- AT91 timer chip
- USB interface

Sensor Board

- 2G/6G three-axis accelerometer
- Temperature sensor
- Light sensor
- 8 tri-color LEDs
- 6 analog inputs
- 2 momentary switches
- 5 general purpose I/O pins and 4 high current output pins

Battery

- 3.7V rechargeable 750 mAh lithium-ion battery
- 30 uA deep sleep mode
- Automatic battery management provided by the software

Network

- IEEE 802.15.4 Wireless Protocol

Security

- Optimized implementations of RSA and Elliptic Curve Cryptography (ECC)

Περιβάλλον Χειρισμού

Βάζοντας τη πλατφόρμα των SunSPOT πάνω σε τεχνολογία java γίνεται πιο εύκολο να γραφεί κώδικας για μικρές ασύρματες συσκευές αισθητήρων και άλλων ηλεκτρονικών συσκευών κατανάλωσης. Έτσι, οι developers μπορούν να γράψουν ένα πρόγραμμα σε java, να το φορτώσουν σε μια συσκευή να το τρέξουν και να βρουν σφάλματα στον κώδικα. Χρησιμοποιώντας java είναι επίσης ευκολότερο να μεταφέρουμε εφαρμογές μεταξύ των διάφορων πλατφόρμων και οι συσκευές SunSPOT προσφέρουν μια μικρή, ευέλικτη ασύρματη πλατφόρμα ώστε να τοποθετηθούν αυτές οι εφαρμογές. Για τα εκατομμύρια των χρηστών που χειρίζονται και γράφουν κώδικα σε JAVA μέσω αυτής της συσκευής μπορούν να επιτύχουν αρκετά πράγματα.

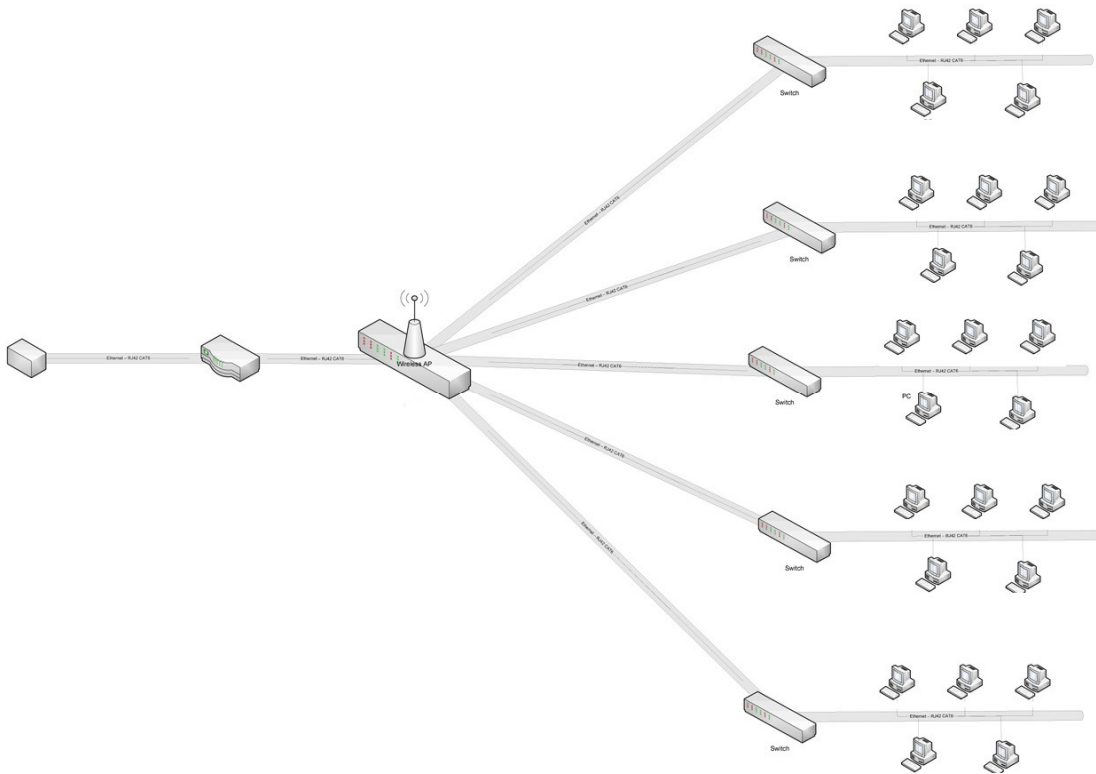
Χρήσιμο Λογισμικό για Sun SPOT Συσκευές

Οι συσκευές που χρησιμοποιούν JAVA είναι γνωστές για την ικανότητα τους να είναι ανεξάρτητες από το hardware. Τα SunSPOT χρησιμοποιούν ένα μικρό J2ME το οποίο τρέχει κατευθείαν στον επεξεργαστή χωρίς ένα λειτουργικό σύστημα. Και το J2ME και τα SunSPOT είναι **ανοικτού κώδικα**. Με τον όρο **Λογισμικό ανοικτού κώδικα** εννοείται λογισμικό του οποίου ο πηγαίος κώδικας διατίθεται ελεύθερα σε αυτούς που θέλουν να τον εξετάσουν, και/ή τροποποιήσουν η χρησιμοποιήσουν σε άλλες εφαρμογές. Υπάρχουν πολλές διαφορετικές άδειες χρήσης που μπορεί να συνοδεύουν το λογισμικό ανοικτού κώδικα. Σε γενικές γραμμές το λογισμικό ανοικτού κώδικα δεν σημαίνει απαραίτητα δωρεάν λογισμικό, ούτε ελεύθερο λογισμικό (σύμφωνα με τον ορισμό που δίνει στο ελεύθερο λογισμικό το Ίδρυμα Ελεύθερου Λογισμικού), αλλά αναφέρεται κυρίως στην ελευθερία του κάθε χρήστη να εξετάσει και να χρησιμοποιήσει την γνώση και τις δυνατότητες που του προσφέρει ο κώδικας προγραμματισμού.

Εργαλεία Ανάπτυξης

Αντιθέτως με άλλα συστήματα, οι SUNSPOT χρήστες μπορούν να χρησιμοποιούν εργαλεία ανάπτυξης εφαρμογών όπως τα NETBEANS ή ECLIPSE για να προγραμματίζουν τις εφαρμογές που επιθυμεί ο χρήστης.

Δυνατότητα Κλιμάκωσης(Scalability)



Οποιαδήποτε συσκευή αυτής της τεχνολογίας μπορεί να είναι ικανή να επηρεάζεται από μια μεγάλη κλιμάκωση(παραδείγματος χάρη ολοκληρωμένα συστήματα από έναν έως χιλιάδες επεξεργαστές είτε η αύξηση σε petabytes σε χωρητικότητα).

3. Γενική περιγραφή του συστήματος

Το δοκιμαστικό σύστημα περιλαμβάνει

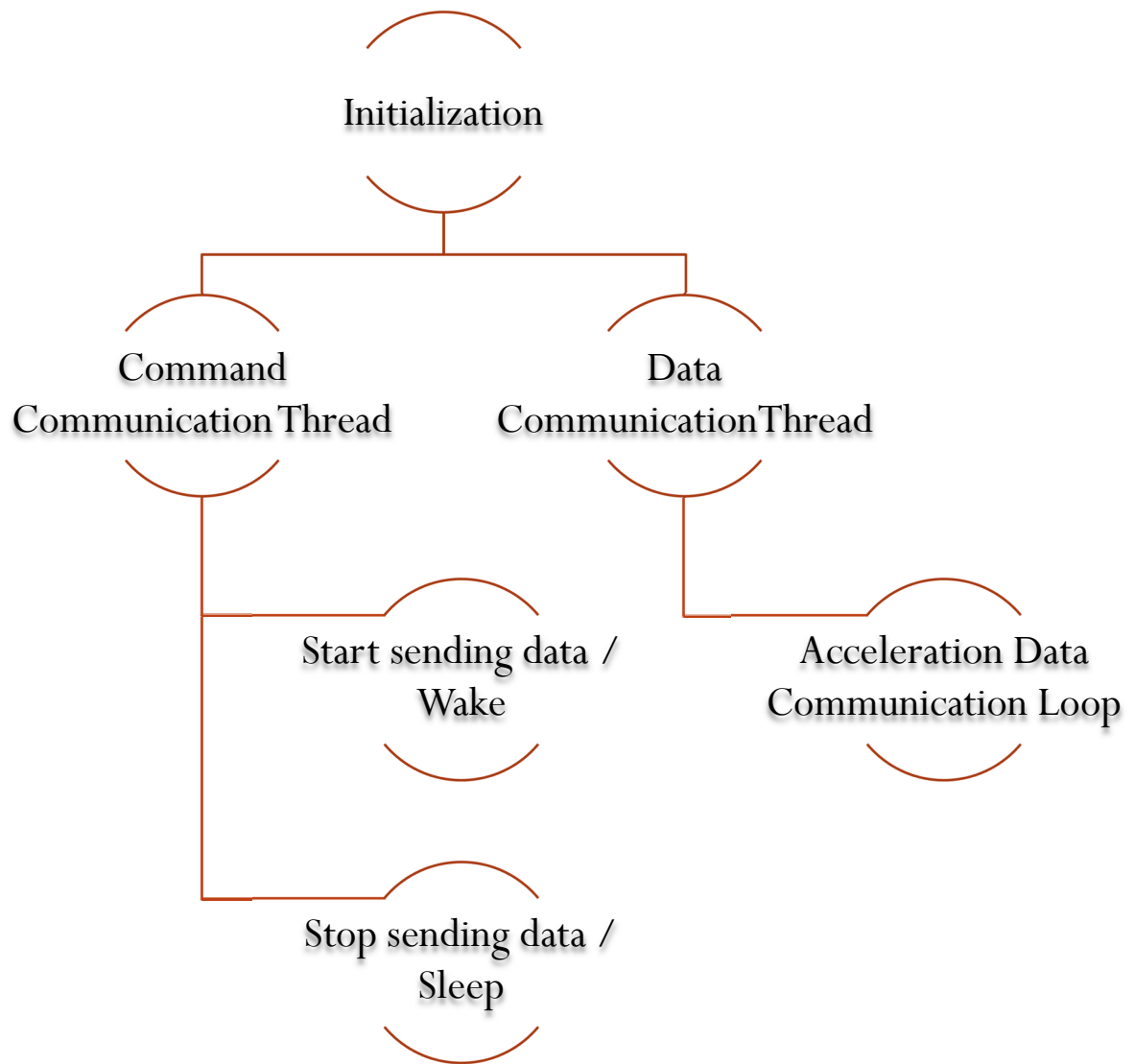
- Ένα σταθμό βάσης
- Δύο σταθμούς ελεύθερης περιοχής
- Έναν υπολογιστή συνδεδεμένο με το σταθμό βάσης

Οι δύο σταθμοί ελεύθερης περιοχής τοποθετούνται σε μια πόρτα. Στη συνέχεια ο κάθε ένας από αυτούς ανοίγει ένα κανάλι επικοινωνίας με το σταθμό βάσης. Αφού εδραιωθεί η επικοινωνία μεταξύ των σταθμών ελεύθερης περιοχής με το σταθμό βάσης ξεκινάει η αποστολή δεδομένων επιτάχυνσης προς το σταθμό βάσης, βάση των οποίων η εφαρμογή θα προσπαθήσει να αντιληφθεί αν σημειώθηκε παραβίαση του χώρου. Επιπλέον θα υπάρχει επιλογή λειτουργίας failover κατά τα την εκτέλεση της οποίας, ο ένας σταθμός ελεύθερης περιοχής θα μπαίνει σε κατάσταση εξοικονόμησης ενέργειας.

3.1 Περιγραφή λειτουργίας σταθμού ελεύθερης περιοχής

Ο σταθμός ελεύθερης περιοχής αφού ενεργοποιηθεί ή γίνει reset ξεκινάει τη ακόλουθη διαδικασία:

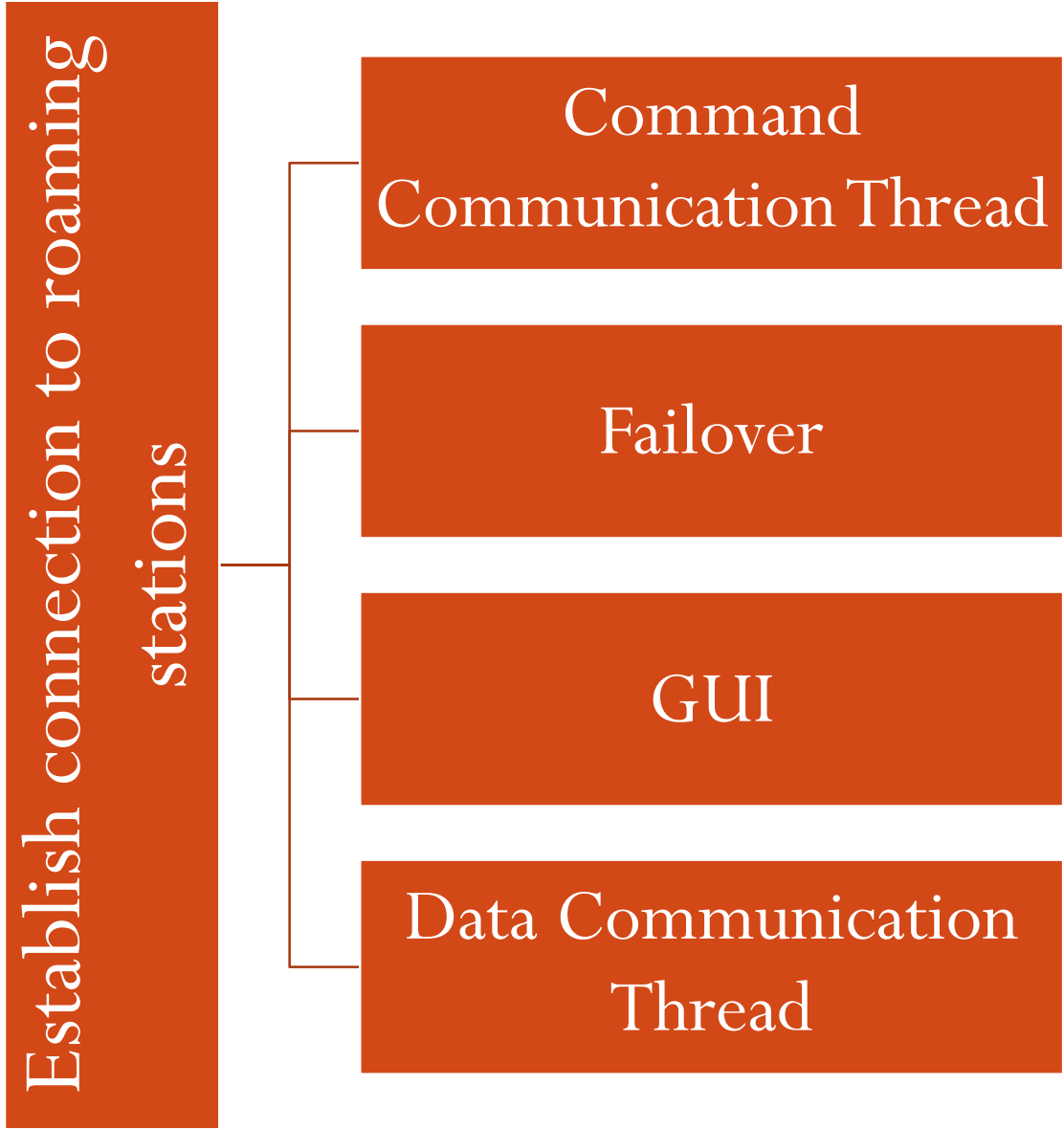
- Ανοίγει ένα broadcast connection
- Ακούει τα δεδομένα από το broadcast με σκοπό να βρεί τη διεύθυνση του σταθμού βάσης
- Μόλις γίνει αυτό, προσπαθεί να ανοίξει ένα κανάλι δεδομένων με το σταθμό βάσης
- Μέσα από αυτό το κανάλι μεταφέρονται οι εντολές του σταθμού βάσης προς το σταθμό ελεύθερης περιοχής
- Ο σταθμός ελεύθερης περιοχής θα ξεκινήσει την αποστολή δεδομένων προς το σταθμό βάσης ότα λάβει από τον τελευταίο την εντολή start
- Εάν κατά τη διάρκεια της αποστολής δεδομένων ληφθεί η εντολή stop ο σταθμός ελεύθερης περιοχής θα μπει σε λειτουργία εξοικονόμησης ενέργειας(Deep Sleep) για ένα προκαθορισμένο χρονικό διάστημα. Μετά το τέλος αυτού του διαστήματος η συσκευή κάνει reset και τα επαναλαμβάνει τα παραπάνω βήματα



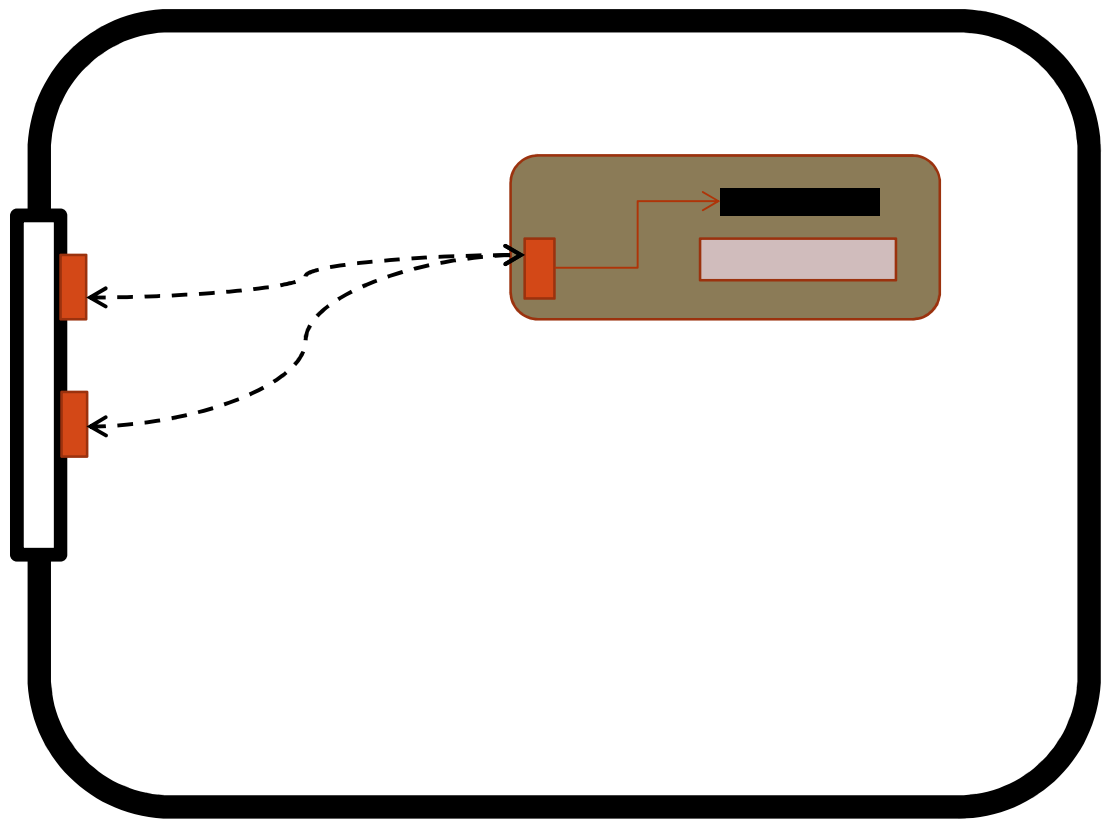
3.2 Περιγραφή λειτουργίας σταθμού βάσης

Αντίστοιχα, ο σταθμός βάσης ακολουθεί την παρακάτω διαδικασία:

- Ανοίγει ένα broadcast connection
- Στέλνει στο broadcast κανάλι τη διεύθυνση του
- Ανοίγει ένα κανάλι δεδομένων για κάθε σταθμό ελεύθερης περιοχής
- Ανάλογα με το user input στέλνει εντολές έναρξης/διακοπής αποστολής δεδομένων προς τους σταθμούς ελεύθερης περιοχής
- Λαμβάνει και καταγράφει τα δεδομένα που στέλνονται από τους σταθμούς ελεύθερης περιοχής και αν οι μετρήσεις είναι πάνω από το όριο(threshold) μας ενημερώνει για παραβίαση του χώρου



ΣΧΗΜΑ ΔΟΚΙΜΑΣΤΙΚΗΣ ΕΚΤΕΛΕΣΗΣ



4. Συμπεράσματα

4.1 Παρατηρήσεις

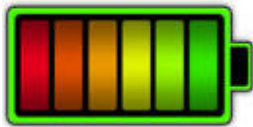
Κατά τη διάρκεια του δοκιμαστικού, ενδιαφέρον παρουσίασαν οι παρακάτω παράμετροι:

- Η ευαισθησία των αισθητήρων επιτάχυνσης των ελεύθερης περιοχής
- Η αντοχή της μπαταρίας των σταθμών ελεύθερης περιοχής
- Τα χαρακτηριστικά της κάθε πόρτας
- Η ιδιαιτερότητα του χώρου
- Χρήση επιπλέον μέτρων ασφαλείας
- Ενδεχόμενα επιθέσεων
- Διαχείριση του συστήματος
- Εγκατάσταση του συστήματος
- Φυσική προστασία

4.1.1 Ευαισθησία του αισθητήρα

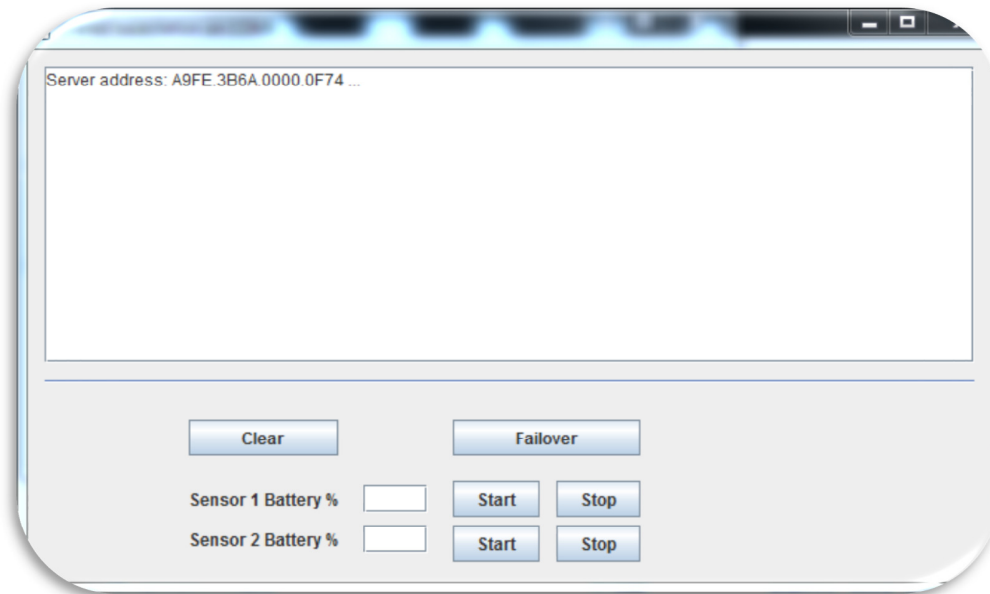
Σε γενικές γραμμές, η ευαισθησία του αισθητήρα είναι μέτρια για αυτο το σκοπό. Ανάλογα με την εκάστοτε περίσταση, αφήνοντας την πόρτα ξεκλειδωτη υπάρχει πιθανότητα το άνοιγμα να γίνει τόσο απαλά που να μην καταφέρει να το ανιχνεύσει ο αισθητήρας. Επίσης ανάλογα με την κάθε περίπτωση η ευαισθησία των αισθητήρων θα πρέπει να ρυθμίζεται κατά την εγκατάσταση του συστήματος για την καλύτερη απόκριση και μείωση των εκάστοτε false positives.

4.1.2 Αντοιχή μπαταρίας



Με τη σωστή διαχείριση του χρόνου παραμονής των συσκευών σε deep sleep οι αισθητήρες μπορούν να αντέξουν μέρες χωρίς φόρτιση.

4.1.3 Διαχείριση Συστήματος



Με τη χρήση του διαθέσιμου GUI της εφαρμογής η διαχείριση του συστήματος γίνεται πολύ απλή.

4.1.4 Εγκατάσταση Συστήματος



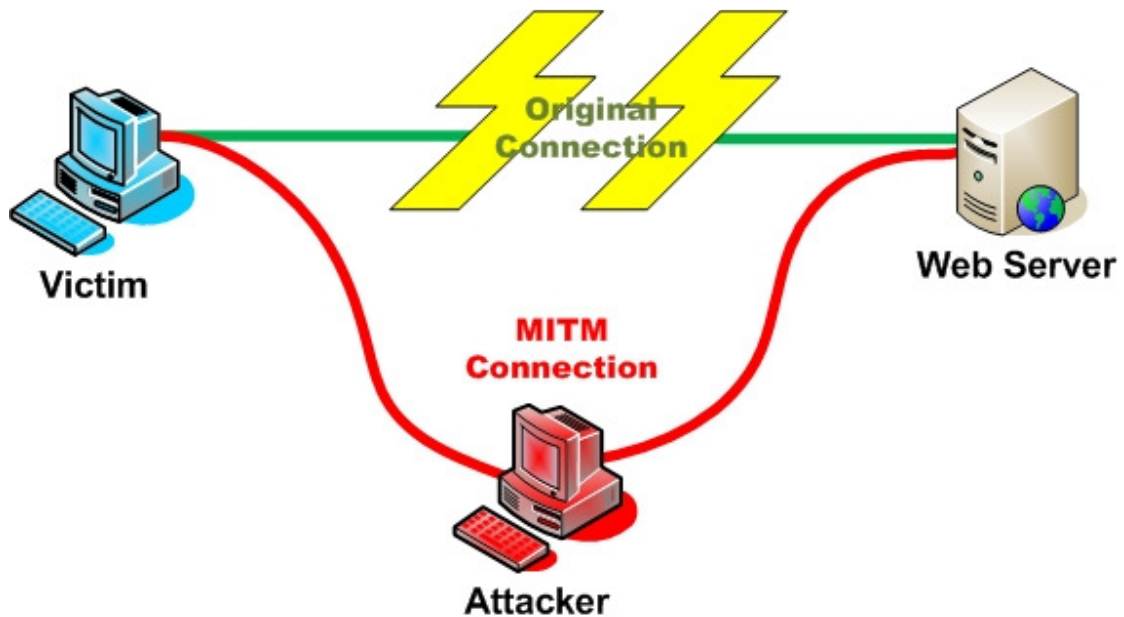
Για την εγκατάσταση του συστήματος είναι απαραίτητο ένα PC με εγκατεστημένο και πλήρως ρυθμισμένο το περιβαλλον SunSPOT SDK. Η εγκατάσταση γίνεται με τη φόρτωση του κώδικα στα SunSPOT τρέχοντας τα εκτελέσιμα .jar αρχεία για τον σταθμό βάσης και τους σταθμούς ελεύθερης περιοχής αντίστοιχα.

Σε περίπτωση που χρειάζεται παραμετροποίηση του κώδικα απαιτείται ένας πύο έμπειρος developer με γνώσεις προγραμματισμού java και SunSPOT SDK για να κάνει τις απαραίτητες αλλαγές στο όλο σύστημα.

4.1.5 Φυσικές επιθέσεις

Προφανώς εάν ο διαχειριστής του συστήματος είναι κακόβουλος χρήστης τότε βρίσκεται σε θέση να διακόψει τη λειτουργία του συστήματος . Επίσης εάν οποιοσδήποτε κακόβουλος χρήστης έχει φυσική πρόσβαση οποιαδήποτε στιγμή στο σύστημα μπορεί να διαταράξει ή να παρακάμψει το σύστημα. Σωστή τοποθέτηση του συστήματος και περιορισμός της φυσικής πρόσβασης σε τρίτους μπορούν να προσφέρουν μια επιπλέον προστασία

4.1.6 Επιθέσεις επιπέδου δικτύου με χρήση υλικού



Εάν ο εν δυνάμει εισβολέας έχει στη διάθεσή του τον πηγαίο κώδικα του συστήματος και ένα ξεχωριστό σέτ Sun SPOT, τότε υπάρχει πιθανότητα να μπορέσει να αλλιώσει τη λειτουργία του συστήματος, πχ απενεργοποιώντας τους αισθητήρες ή προκαλώντας επίθεση DoS (Denial of Service) στο σύστημα. Για να αποτραπεί αυτό το ενδεχόμενο, τα Sun SPOT προσφέρουν επιλογή κρυπτογράφησης από το SDK με public/private key encryption ώστε τα δεδομένα να μπορούν να διαβαστούν μόνο από τις επιλεγμένες συσκευές.

Γενικά συμπεράσματα και προτάσεις

Όσον αφορά την ασφάλεια υπολογιστικών συστημάτων, ποτέ δεν είναι εφικτό να ληφθούν μέτρα ασφαλείας που να καθιστούν ένα σύστημα απόρθητο. Αυτό συμβαίνει για πολλούς λόγους. Για αρχή η ασφάλεια συστημάτων είναι ένα πολυδιάστατο αντικείμενο και κάθε μεμονωμένο στοιχείο που αποτελεί ένα σύστημα μπορεί να κρύβει πολλά κενά ασφαλείας. Αυτά μπορούν να βρίσκονται σε οποιοδήποτε τομέα (buffer overflows, routing poisoning, arp poisoning, keyloggers, sniffing, compatibility issues, protocol stack attacks, rootkits, reverse engineering, etc) και να λειτουργήσουν συνδιαστικά ανάλογα με το σκοπό και τις γνώσεις του εκάστοτε επιτιθέμενου.

Οι περισσότερες από αυτές τις περιπτώσεις βέβαια μπορούν να αντιμετωπιστούν με απλό τρόπο για ένα απλό σύστημα ασφαλείας αποσυνδέοντας τον υπολογιστή στον οποίο συνδέεται μέσω usb ο σταθμός βάσης από οποιοδήποτε δίκτυο.

Αυτό όμως δεν θα είναι εφικτό σε περίπτωση που το συγκεκριμένο σύστημα επεκταθεί χρησιμοποιώντας τις διάφορες java βιβλιοθήκες ώστε να ανοίκει σε ένα μεγαλύτερο κεντροποιημένο ή κατακεντρωμένο σύστημα διαχείρισης όπου κάθε χώρος θα μπορεί να ελέγχεται μέσω δικτύου ανεξάρτητα από τη φυσική τοποθεσία του. Επίσης διάφορες τεχνικές και συνδιασμοί συστημάτων μπορούν να ενισχύσουν το κάθε σύστημα πχ. IP cameras με object tracking software σε python μπορούν να συνδιαστούν και να αλληλεπιδράσουν μέσω των sockets του localhost ή δικτυακά δίνοντας ένα κάλο παράδειγμα για ένα πιο ολοκληρωμένο διαδικτυακό σύστημα ασφαλείας χώρου.

Πηγαίος κώδικας Σταθμού Βάσης

SendDataDemoHostApplication.java

```
package org.sunspotworld.demo;

import com.sun.spot.io.j2me.radiogram.*;

import com.sun.spot.peripheral.ota.OTACommandServer;

import java.io.IOException;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.microedition.io.*;
```

```

public class SendDataDemoHostApplication {

    Date date;

    GUI gui;

    DatagramConnection connection1,connection2,broadcastConn;

    Datagram rDatagram1,rDatagram2,broadcastDg;

    String ourAddress;

    DoorSensor sensor1, sensor2;

    TCPServer ser;

    int batteryLevel1, batteryLevel2;

    public void run() throws Exception {

        date = new Date();

        gui = new GUI();

        gui.setVisible(true);

        sensor1 = new DoorSensor("0014.4F01.0000.619A");

        sensor2 = new DoorSensor("0014.4F01.0000.5F80");

        ourAddress = System.getProperty("IEEE_ADDRESS");//ADDRESS!!

        gui.add("Server address: " + ourAddress + " ...");
    }
}

```

```

try {

    broadcastConn = (DatagramConnection)
Connector.open("radiogram://broadcast:"+100);

    broadcastDg = broadcastConn.newDatagram(50);// only sending 12 bytes of data

    connection1 = (DatagramConnection)
Connector.open("radiogram://" + sensor1.address + ":67");

    rDatagram1 = connection1.newDatagram(50);// only sending 12 bytes of data

    connection2 = (DatagramConnection)
Connector.open("radiogram://" + sensor2.address + ":68");

    rDatagram2 = connection2.newDatagram(50);// only sending 12 bytes of data

} catch (Exception e) {

    System.err.println("Caught " + e + " in connection initialization.");

}

(new Thread(new BeaconRecThread())).start();

(new Thread(new ReceiveDataThread1())).start();

(new Thread(new ReceiveDataThread2())).start();

//(new Thread(new TCPServer())).start();

```

```
}
```

```
public class BeaconRecThread implements Runnable{
```

```
    public void run(){
```

```
        try {
```

```
            while(true){
```

```
                if(gui.getStartFlag1()){
```

```
                    broadcastDg.reset();
```

```
                    broadcastDg.writeUTF(sensor1.address+"start");
```

```
                    broadcastConn.send(broadcastDg);
```

```
                    gui.setStartFlag1(false);
```

```
                    gui.add("command sent.. ");
```

```
                }
```

```
                else if(gui.getStartFlag2()){
```

```
                    broadcastDg.reset();
```

```
                    broadcastDg.writeUTF(sensor2.address+"start");
```

```
                    broadcastConn.send(broadcastDg);
```

```
                    gui.setStartFlag2(false);
```

```

        gui.add("command sent.. ");

    }

    else if(gui.getStopFlag1()){

        broadcastDg.reset();

        broadcastDg.writeUTF(sensor1.address+"stop");

        broadcastConn.send(broadcastDg);

        gui.setStopFlag1(false);

        gui.add("command sent.. ");

    }

    else if(gui.getStopFlag2()){

        broadcastDg.reset();

        broadcastDg.writeUTF(sensor2.address+"stop");

        broadcastConn.send(broadcastDg);

        gui.setStopFlag2(false);

        gui.add("command sent.. ");

    }

    broadcastDg.reset();

    broadcastDg.writeUTF(ourAddress);

```

```
broadcastConn.send(broadcastDg);

//gui.add("Beacon sent..");

Thread.sleep(2000);

if(gui.getFailoverFlag()){

    if(batteryLevel1>=batteryLevel2){

        broadcastDg.reset();

        broadcastDg.writeUTF(sensor1.address+"sleep");

        broadcastConn.send(broadcastDg);

        // Thread.sleep(1000);

    }

    else if(batteryLevel1<batteryLevel2){

        broadcastDg.reset();

        broadcastDg.writeUTF(sensor2.address+"sleep");

        broadcastConn.send(broadcastDg);

        //Thread.sleep(1000);

    }

}

}
```



```

    } catch (Exception ex) { gui.add("Exception: "+ex);}

}

}

public class ReceiveDataThread1 implements Runnable{

    public void run(){

        try {

            while(true){

                connection1.receive(rDatagram1);

                batteryLevel1 = rDatagram1.readInt();    // read the sensor value

                gui.readBatteryLevel1(batteryLevel1);

                // if(batteryLevel==100){command="stop";gui.add("command sent..");}

                double accelerationValue = rDatagram1.readDouble();

                String s = "ALARM! " + accelerationValue + " " + date.toString();

                if(accelerationValue >= 0.001){ gui.add(s);//ser.getData(s);

                }

                // gui.add("  BatteryLevel = " + batteryLevel + " Acceleration = " +
accelerationValue);

            }

}

```

```

    } catch (Exception ex) { gui.add("Exception: "+ex);}

}

}

public class ReceiveDataThread2 implements Runnable{

    public void run(){

        try {

            while(true){

                connection2.receive(rDatagram2);

                batteryLevel2 = rDatagram2.readInt();    // read the sensor value

                gui.readBatteryLevel2(batteryLevel2);

                // if(batteryLevel==100){command="stop";gui.add("command sent..");}

                double accelerationValue = rDatagram2.readDouble();

                String s = "ALARM! " + accelerationValue + " " + date.toString();

                if(accelerationValue >= 0.001){ gui.add(s);//ser.getData(s);

                }

                // gui.add("  BatteryLevel = " + batteryLevel + " Acceleration = " +
accelerationValue);

            }

        } catch (Exception ex) { gui.add("Exception: "+ex);}

```

```

    }

}

public class DoorSensor{

    String address;

    public DoorSensor(String addr){

        address=addr;

    }

}

public static void main(String[] args) throws Exception {

    // register the application's name with the OTA Command server & start OTA
running

    OTACommandServer.start("SendDataDemo");

    SendDataDemoHostApplication app = new SendDataDemoHostApplication();

    app.run();

}

}

```


GUI.java

```
/*
```

```
* To change this template, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
package org.sunspotworld.demo;
```

```
import javax.microedition.io.Datagram;
```

```
/**
```

```
*
```

```
* @author greg
```

```
*/
```

```
public class GUI extends javax.swing.JFrame {
```

```
    Boolean startflag1,startflag2,stopflag1,stopflag2,failoverflag;
```

```
/**
```

```
* Creates new form GUI
```

```
*/

public GUI() {

    startflag1=false;

    startflag2=false;

    stopflag1=false;

    stopflag2=false;

    failoverflag=false;

    initComponents();

}

public boolean getFailoverFlag(){return failoverflag;}

public boolean getStartFlag1(){return startflag1;}

public void setStartFlag1(boolean flag){startflag1=flag;}

public boolean getStopFlag1(){return stopflag1;}

public void setStopFlag1(boolean flag){stopflag1=flag;}

public boolean getStartFlag2(){return startflag2;}

public void setStartFlag2(boolean flag){startflag2=flag;}

public boolean getStopFlag2(){return stopflag2;}
```

```

public void setStopFlag2(boolean flag){stopflag2=flag;}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

public void readBatteryLevel1(int lvl){s1battText.setText(Integer.toString(lvl));
}

public void readBatteryLevel2(int lvl){s2battText.setText(Integer.toString(lvl));
}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();

    jTextArea1 = new javax.swing.JTextArea();

    jButton1 = new javax.swing.JButton();

    s1battText = new javax.swing.JTextField();

```

```
jSeparator1 = new javax.swing.JSeparator();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

s2battText = new javax.swing.JTextField();

s1startButton = new javax.swing.JButton();

s1stopButton = new javax.swing.JButton();

s2stopButton = new javax.swing.JButton();

s2startButton = new javax.swing.JButton();

failoverButton = new javax.swing.JToggleButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jTextArea1.setEditable(false);

jTextArea1.setColumns(20);

jTextArea1.setRows(5);

jScrollPane1.setViewportView(jTextArea1);

jButton1.setText("Clear");
```



```
jButton1.addActionListener(new java.awt.event.ActionListener() {  
  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
  
        jButton1ActionPerformed(evt);  
  
    }  
  
});
```

```
s1battText.addActionListener(new java.awt.event.ActionListener() {  
  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
  
        s1battTextActionPerformed(evt);  
  
    }  
  
});
```

```
s1battText.addPropertyChangeListener(new java.beans.PropertyChangeListener() {  
  
    public void propertyChange(java.beans.PropertyChangeEvent evt) {  
  
        s1battTextPropertyChange(evt);  
  
    }  
  
});
```

```
jLabel1.setText("Sensor 1 Battery %");
```

```
jLabel2.setText("Sensor 2 Battery %");

s2battText.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        s2battTextActionPerformed(evt);

    }

});

s2battText.addPropertyChangeListener(new java.beans.PropertyChangeListener() {

    public void propertyChange(java.beans.PropertyChangeEvent evt) {

        s2battTextPropertyChange(evt);

    }

});

s1startButton.setText("Start");

s1startButton.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        s1startButtonActionPerformed(evt);

    }

});
```

```
    }  
});  
  
s1stopButton.setText("Stop");  
  
s1stopButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        s1stopButtonActionPerformed(evt);  
    }  
});  
  
s2stopButton.setText("Stop");  
  
s2stopButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        s2stopButtonActionPerformed(evt);  
    }  
});  
  
s2startButton.setText("Start");
```

```

s2startButton.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        s2startButtonActionPerformed(evt);

    }

});

```

```

failoverButton.setText("Failover");

```

```

failoverButton.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        failoverButtonActionPerformed(evt);

    }

});

```

```

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(layout.createSequentialGroup())
```

```
        .addContainerGap()
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addComponent(jScrollPane1)
```

```
    .addComponent(jSeparator1)))
```

```
.addGroup(layout.createSequentialGroup())
```

```
    .addGap(115, 115, 115)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,  
false)
```

```
    .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
    .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
    .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup()

        .addComponent(s2battText,
javax.swing.GroupLayout.PREFERRED_SIZE, 46,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(18, 18, 18)

        .addComponent(s2startButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(s2stopButton))

    .addGroup(layout.createSequentialGroup()

        .addComponent(s1battText,
javax.swing.GroupLayout.PREFERRED_SIZE, 46,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(18, 18, 18)

```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,  
false)
```

```
    .addGroup(layout.createSequentialGroup()
```

```
        .addComponent(s1.startButton)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
    .addComponent(s1.stopButton))
```

```
    .addComponent(failoverButton,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
```

```
    .addGap(0, 239, Short.MAX_VALUE)))
```

```
    .addContainerGap()
```

```
);
```

```
layout.setVerticalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(layout.createSequentialGroup()
```

```
            .addComponent(s1.startButton)
```

```
            .addContainerGap()
```

```
.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,  
213, Short.MAX_VALUE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,  
10, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE  
)
```

```
.addComponent(jButton1)
```

```
.addComponent(failoverButton))
```

```
.addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE  
)
```

```
.addComponent(s1battText, javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addComponent(jLabel1)
```



```

        .addComponent(s1startButton)

        .addComponent(s1stopButton))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
    )

        .addComponent(s2startButton)

        .addComponent(s2stopButton))

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
    )

        .addComponent(jLabel2)

        .addComponent(s2battText,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

    .addGap(14, 14, 14)

);

```

```
    pack();

} // </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    jTextArea1.setText(null);

    s1battText.setText(null);

    s2battText.setText(null);

}

private void s1battTextActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void s1battTextPropertyChange(java.beans.PropertyChangeEvent evt) {

    // TODO add your handling code here:

}
```

```
private void s2battTextActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
}
```

```
private void s2battTextPropertyChange(java.beans.PropertyChangeEvent evt) {  
  
    // TODO add your handling code here:  
  
}
```

```
private void s2startButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    setStartFlag2(true);  
  
}
```

```
private void s1startButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    setStartFlag1(true);  
  
}
```

```
private void s1stopButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    setStopFlag1(true);// TODO add your handling code here:  
  
}
```

```
private void s2stopButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    setStopFlag2(true);  
  
}
```

```
private void failoverButtonActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    if(failoverButton.isSelected()){ failoverflag=true;}  
  
    else{ failoverflag=false;}  
  
}
```

```
/**  
  
 * @param args the command line arguments  
  
 */
```

```
public void add(String s){jTextArea1.append(s+"\n");}

// Variables declaration - do not modify

private javax.swing.JToggleButton failoverButton;

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JTextArea jTextArea1;

private javax.swing.JTextField s1battText;

private javax.swing.JButton s1startButton;

private javax.swing.JButton s1stopButton;

private javax.swing.JTextField s2battText;

private javax.swing.JButton s2startButton;

private javax.swing.JButton s2stopButton;

// End of variables declaration

}
```


Πηγαίος κώδικας Σταθμού ελεύθερης περιοχής

SensorSampler.java

```
package org.sunspotworld.demo;

import com.sun.spot.io.j2me.radiogram.*;

import com.sun.spot.peripheral.Battery;

import com.sun.spot.peripheral.IBattery;

import com.sun.spot.peripheral.Spot;

import com.sun.spot.peripheral.radio.BroadcastConnectionState;

import com.sun.spot.peripheral.radio.LowPan;

import com.sun.spot.resources.Resources;

import com.sun.spot.resources.transducers.IAccelerometer3D;

import com.sun.spot.resources.transducers.ITriColorLED;

import com.sun.spot.resources.transducers.ILightSensor;

import com.sun.spot.resources.IResource;
```

```

import com.sun.spot.util.Utils;

import java.io.IOException;

import javax.microedition.io.*;

import javax.microedition.midlet.MIDlet;

import javax.microedition.midlet.MIDletStateChangeException;

public class SensorSampler extends MIDlet {

    DatagramConnection connection, broadcastConn;

    Datagram broadcastDg, connectionDg=null;

    IAccelerometer3D accelerationSensor;

    IBattery battery;

    String ourAddress;

    String command;

    ITriColorLED led1 = (ITriColorLED)Resources.lookup(ITriColorLED.class,
"LED1");

```



```
ITriColorLED led2 = (ITriColorLED)Resources.lookup(ITriColorLED.class,  
"LED2");
```

```
ITriColorLED led3 = (ITriColorLED)Resources.lookup(ITriColorLED.class,  
"LED3");
```

```
ITriColorLED led4 = (ITriColorLED)Resources.lookup(ITriColorLED.class,  
"LED4");
```

```
ITriColorLED led5 = (ITriColorLED)Resources.lookup(ITriColorLED.class,  
"LED5");
```

```
ITriColorLED led6 = (ITriColorLED)Resources.lookup(ITriColorLED.class,  
"LED6");
```

```
ITriColorLED led7 = (ITriColorLED)Resources.lookup(ITriColorLED.class,  
"LED7");
```

```
ITriColorLED led8 = (ITriColorLED)Resources.lookup(ITriColorLED.class,  
"LED8");
```

```
protected void startApp() throws MIDletStateChangeException {
```

```
    connection = null;
```

```
    connectionDg = null;
```

```
broadcastConn = null;
```

```
broadcastDg = null;
```

```
accelarationSensor =
```

```
(IAccelerometer3D)Resources.lookup(IAccelerometer3D.class);
```

```
battery =(IBattery) Spot.getInstance().getPowerController().getBattery();
```

```
ourAddress = System.getProperty("IEEE_ADDRESS");//ADDRESS!!
```

```
System.out.println("Starting sensor sampler application on " + ourAddress + " ...");
```

```
// Listen for downloads/commands over USB connection
```

```
new com.sun.spot.service.BootloaderListenerService().getInstance().start();
```

```
try {
```

```
    init();
```

```
} catch (IOException ex) {ex.printStackTrace();}
```

```
}
```

```

public void init() throws IOException{

    System.out.println("Initializing...");

    String remoteAddr = null;

    String packet = null;

    try {

        System.out.println("Opening broadcast conection...");

        broadcastConn = (DatagramConnection) Connector.open("radiogram://:100");

        broadcastDg
        broadcastConn.newDatagram(broadcastConn.getNominalLength());

    }catch (Exception e) {

        System.err.println("Caught " + e + " in broadcast connection initialization.");

    }

    System.out.println("Broadcast conection ok...");

    System.out.println("Listening for remote address...");

    setLightsBlue();

```

```

setAllLightsOn();

while(remoteAddr==null){

    broadcastConn.receive(broadcastDg);

    packet=broadcastDg.readUTF();

if(packet.endsWith("start")||(packet.endsWith("stop"))||(packet.endsWith("sleep"))){System.out.println("wrong packet retrying");}

    else {remoteAddr=packet;}

}

System.out.println("received address: "+remoteAddr);

try {

    System.out.println("Establishing connection...");

    if(ourAddress.endsWith("5F80")){connection = (DatagramConnection)
Connector.open("radiogram://" +remoteAddr+":68"); }

    else{connection = (DatagramConnection)
Connector.open("radiogram://" +remoteAddr+":67"); }

    connectionDg = connection.newDatagram(connection.getNominalLength());

}catch (Exception e) {

```

```

        System.err.println("Caught " + e + " in connection initialization.");

        notifyDestroyed();

    }

    System.out.println("Ready");

    setLightsGreen();

    setAllLightsOn();

    Utils.sleep(3000);

    setAllLightsOff();

    (new Thread(new ReceiveBroadcastThread())).start();

}

```

```

public class ReceiveBroadcastThread implements Runnable{

    final SendDataThread datasend = new SendDataThread();

    public void run(){

        System.out.println("Entering command loop...");

        while(true){

```

```

command = null;

try{

    broadcastConn.receive(broadcastDg);

    command = broadcastDg.readUTF();}catch(Exception e){ }

System.out.println(command);

synchronized(datasend){

    if(command.equalsIgnoreCase(ourAddress+"start"))

    {

        System.out.println("Start command found! Starting data loop...");

        datasend.resume();

        System.out.println("Data loop started!");

    }

    else if(command.equalsIgnoreCase(ourAddress+"stop"))

    {

        System.out.println("Stop command found! Stopping data loop if already
started...");

        datasend.suspend();

        System.out.println("Data loop stopped.");

    }
}

```

```

else if(command.equalsIgnoreCase(ourAddress+"sleep"))

{

    System.out.println("Sleep command found!");

    try {

        connection.close();

        broadcastConn.close();

    } catch (IOException ex) {ex.printStackTrace();}

    Utils.sleep(20000);

    notifyDestroyed();

}

}

}

}

}

```

```

public class SendDataThread implements Runnable{

    private boolean suspendFlag,sleepFlag;

```

```

Thread dataThread;

public SendDataThread(){

    dataThread = new Thread(this,"data");

    dataThread.start();

}

public void run(){

    suspendFlag=true;

    sleepFlag=false;

    try {

        while(true){

            synchronized(this){

                while(suspendFlag){

                    setLightsRed();

                    setAllLightsOn();

                    wait();}

            }

            synchronized(this){

                while(sleepFlag){

```



```

        sleepFlag=false;

        Utils.sleep(20000);

    }

}

Utils.sleep(100);

int batteryLevel=0;try{batteryLevel =
battery.getBatteryLevel();}catch(Exception e){System.err.println("Caught " + e + " in
battery level reading"); batteryLevel=100;}

    setLightsBattery();

    setBatteryLights(batteryLevel);

    double accelerationReading = accelarationSensor.getAccelZ();

    connectionDg.reset();

    connectionDg.writeInt(batteryLevel);

    connectionDg.writeDouble(accelerationReading);

    connection.send(connectionDg);

    System.out.println("Sending data: battery "+batteryLevel+" - accel:
"+accelerationReading);

}

} catch (Exception ex) {System.out.println("Exception: "+ex);}

```

```

    }

    public synchronized void resume(){suspendFlag=false;notify();}

    public synchronized void suspend(){suspendFlag=true;notify();}

    public synchronized void sleepNow(){sleepFlag=true;notify();}

}

public void setBatteryLights(int level){

    if

(level<16){led1.setOn();led2.setOff();led3.setOff();led4.setOff();led5.setOff();led6.setOf
f();led7.setOff();led8.setOff();}

    else

if(level>15&&level<31){led1.setOn();led2.setOn();led3.setOff();led4.setOff();led5.setOf
f();led6.setOff();led7.setOff();led8.setOff();}

    else

if(level>31&&level<46){led1.setOn();led2.setOn();led3.setOn();led4.setOff();led5.setOff
();led6.setOff();led7.setOff();led8.setOff();}

    else

if(level>45&&level<57){led1.setOn();led2.setOn();led3.setOn();led4.setOn();led5.setOff
();led6.setOff();led7.setOff();led8.setOff();}

    else

if(level>56&&level<68){led1.setOn();led2.setOn();led3.setOn();led4.setOn();led5.setOn(
);led6.setOff();led7.setOff();led8.setOff();}

```

```

else

if(level>67&&level<79){led1.setOn();led2.setOn();led3.setOn();led4.setOn();led5.setOn(
);led6.setOn();led7.setOff();led8.setOff();}

else

if(level>78&&level<90){led1.setOn();led2.setOn();led3.setOn();led4.setOn();led5.setOn(
);led6.setOn();led7.setOn();led8.setOff();}

else

if(level>89&&level<101){led1.setOn();led2.setOn();led3.setOn();led4.setOn();led5.setO
n();led6.setOn();led7.setOn();led8 .setOn();}

}

public void
setAllLightsOn(){led1.setOn();led2.setOn();led3.setOn();led4.setOn();led5.setOn();led6.s
etOn();led7.setOn();led8 .setOn();}

public void
setAllLightsOff(){led1.setOff();led2.setOff();led3.setOff();led4.setOff();led5.setOff();led
6.setOff();led7.setOff();led8 .setOff();}

public void setLightsBattery(){

led1.setRGB(255, 0, 0);

led2.setRGB(255, 0, 0);

led3.setRGB(0, 255, 0);

```

```
    led4.setRGB(0, 255, 0);

    led5.setRGB(0, 255, 0);

    led6.setRGB(0, 255, 0);

    led7.setRGB(0, 255, 0);

    led8.setRGB(0, 255, 0);

}

public void setLightsGreen(){

    led1.setRGB(0, 255, 0);

    led2.setRGB(0, 255, 0);

    led3.setRGB(0, 255, 0);

    led4.setRGB(0, 255, 0);

    led5.setRGB(0, 255, 0);

    led6.setRGB(0, 255, 0);

    led7.setRGB(0, 255, 0);

    led8.setRGB(0, 255, 0);

}

public void setLightsBlue(){

    led1.setRGB(0, 0, 255);
```

```
    led2.setRGB(0, 0, 255);

    led3.setRGB(0, 0, 255);

    led4.setRGB(0, 0, 255);

    led5.setRGB(0, 0, 255);

    led6.setRGB(0, 0, 255);

    led7.setRGB(0, 0, 255);

    led8.setRGB(0, 0, 255);

}

public void setLightsRed(){

    led1.setRGB(255, 0, 0);

    led2.setRGB(255, 0, 0);

    led3.setRGB(255, 0, 0);

    led4.setRGB(255, 0, 0);

    led5.setRGB(255, 0, 0);

    led6.setRGB(255, 0, 0);

    led7.setRGB(255, 0, 0);

    led8.setRGB(255, 0, 0);

}
```

```
protected void pauseApp() {  
  
    // This will never be called by the Squawk VM  
  
}  
  
protected void destroyApp(boolean arg0) throws MIDletStateChangeException {  
  
    // Only called if startApp throws any exception other than  
MIDletStateChangeException  
  
}  
  
}
```

Παράρτημα

Βιβλιογραφία

[1]<http://www.sunspotworld.com/>

[2]http://en.wikipedia.org/wiki/Sun_SPOT

[3] http://en.wikipedia.org/wiki/Network_topology

[4] http://en.wikipedia.org/wiki/Wireless_sensor_network

[5] Building Wireless Sensor Networks (2011), Robert Faludi

[6]“Wireless Sensor Networks”, John A.Stankovic, University of Virginia(2006)

[7] Romer, Kay; Friedemann Mattern (December 2004). "The Design Space of Wireless Sensor Networks". IEEE Wireless Communications

[8] C. Chong, S.P.Kumar “Sensor Networks: Evolution, Opportunities, and Challenges” Proceedings of the IEEE, vol. 91, no. 8, August 2003.

[9]“Providing trust in wireless sensor networks using a bio-inspired technique”, Felix Gomez Marmol & Gregorio Martinez Perez, University of Murcia (2010)

[10]“Freemote: A WIRELESS SENSOR NETWORKS EMULATION SYSTEM”, Raphael Kummer, Timothee Maret, Peter Kropf and Jean-Frederic Wagen, TIC institute, University of applied Science of Fribourg(2009)

[11]“An overview of MANETs Simulation”, Luc Hogue University of Luxembourg, Pascal Bouvry University of Luxembourg, Frederic Guinand University of Havre(2008)

[12]“Using BIP for Modeling and Verification of Networked Systems-A Case Study on TinyOS-based Networks”, Ananda Basu , Laurent Mounier, Marc Poulhies.(2003)