**SUN SPOT**

**Γενικές πληροφορίες**

****

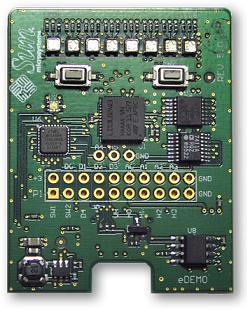
**Εικόνα 33: Sun SPOT**

To Sun SPOT (Sun Small Programmable Object Technology), είναι μια πειραματική πλατφόρμα που αναπτύχθηκε από την εταιρεία πληροφορικής Sun Microsystems για να ενθαρρύνει τους προγραμματιστές να δημιουργήσουν τα δικά τους gadgets, να αναπτύξουν καινοτόμες εφαρμογές που θα επιτρέπουν στους χρήστες να επικοινωνούν και μεταξύ τους και με τον περιβάλλοντα χώρο.

To Sun SPOT αποτελεί μία προγραμματιζόμενη συσκευή που έχει κατασκευαστεί με βάση τις αρχές ενός ειδικά σχεδιασμένου μηχανήματος εικονικής πραγματικότητας, του Squawk Java Virtual Machine, και για αυτό το λόγο έχει ενσωματωμένη την Java. Χρησιμοποιεί το πρωτόκολλο ΙΕΕΕ 802.15.4 για την ασύρματη επικοινωνία. Ένα σύνολο από Sun SPOTs μπορεί επομένως να αποτελέσει ένα ασύρματο δίκτυο αισθητήρων (Wireless Sensor Network – WSN) με το κάθε Sun SPOT να αποτελεί έναν κόμβο του δικτύου.

Η βασική μονάδα του Sun SPOT περιλαμβάνει αισθητήρες θερμοκρασίας, φωτισμού, κίνησης, ραδιοσυσκευή αποστολής σημάτων, ένα σύνολο από LEDs για την επικοινωνία με τον εξωτερικό κόσμο, δύο μπουτόν για την παροχή βασικής ανατροφοδότησης καθώς και θύρα USB για σύνδεση με τον υπολογιστή εκτός της ασύρματης σύνδεσης. Λόγω της εφαρμογής της Java στα Sun SPOTs ο προγραμματισμός τους είναι αρκετά εύκολος.

Κάθε αναπτυξιακό πακέτο για τα Sun SPOTs, (Sun SPOT Development Kit), περιλαμβάνει δύο Sun SPOTs (με επεξεργαστή, ασύρματη επικοινωνία, πίνακα αισθητήρων και μπαταρία), και ένα basestation (με επεξεργαστή κι ασύρματη επικοινωνία). Τα πακέτα περιλαμβάνουν επίσης και καλώδιο USB καθώς και CD για την εγκατάσταση του απαραίτητου λογισμικού σε έναν ηλεκτρονικό υπολογιστή για σύνδεση και επικοινωνία με τα Sun SPOTs.

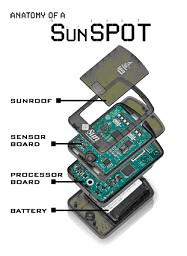


**Εικόνα 34: Βασική πλατφόρμα Sun**

**SPOT**

Το basestation συνδέεται με τον ηλεκτρονικό υπολογιστή και επιτρέπει τη δημιουργία προγραμμάτων τα οποία μπορούν να γραφτούν στον ηλεκτρονικό υπολογιστή και στη συνέχεια να “φορτωθούν” μέσω του basestation σε απομακρυσμένα Sun SPOTs. Ομοίως τα Sun SPOTs μέσω του basestation μπορούν να στείλουν τα δεδομένα που συλλέγουν στον ηλεκτρονικό υπολογιστή. Κάθε Sun SPOT μπορεί επίσης να χρησιμοποιηθεί σαν basestation αλλά σε αυτήν την περίπτωση ο πίνακας αισθητήρων του δεν μπορεί να χρησιμοποιηθεί.

Το αρχικό λογισμικό ανάπτυξης έχει εξεταστεί σε Windows XP, Macintosh OS X 10.4, Linux (Fedora Core 5, SuSE 10.1 και Ubuntu 6.06) και Solaris x86.



**Εικόνα 35: Μέρη από τα οποία αποτελείται ένα Sun**

**SPOT**

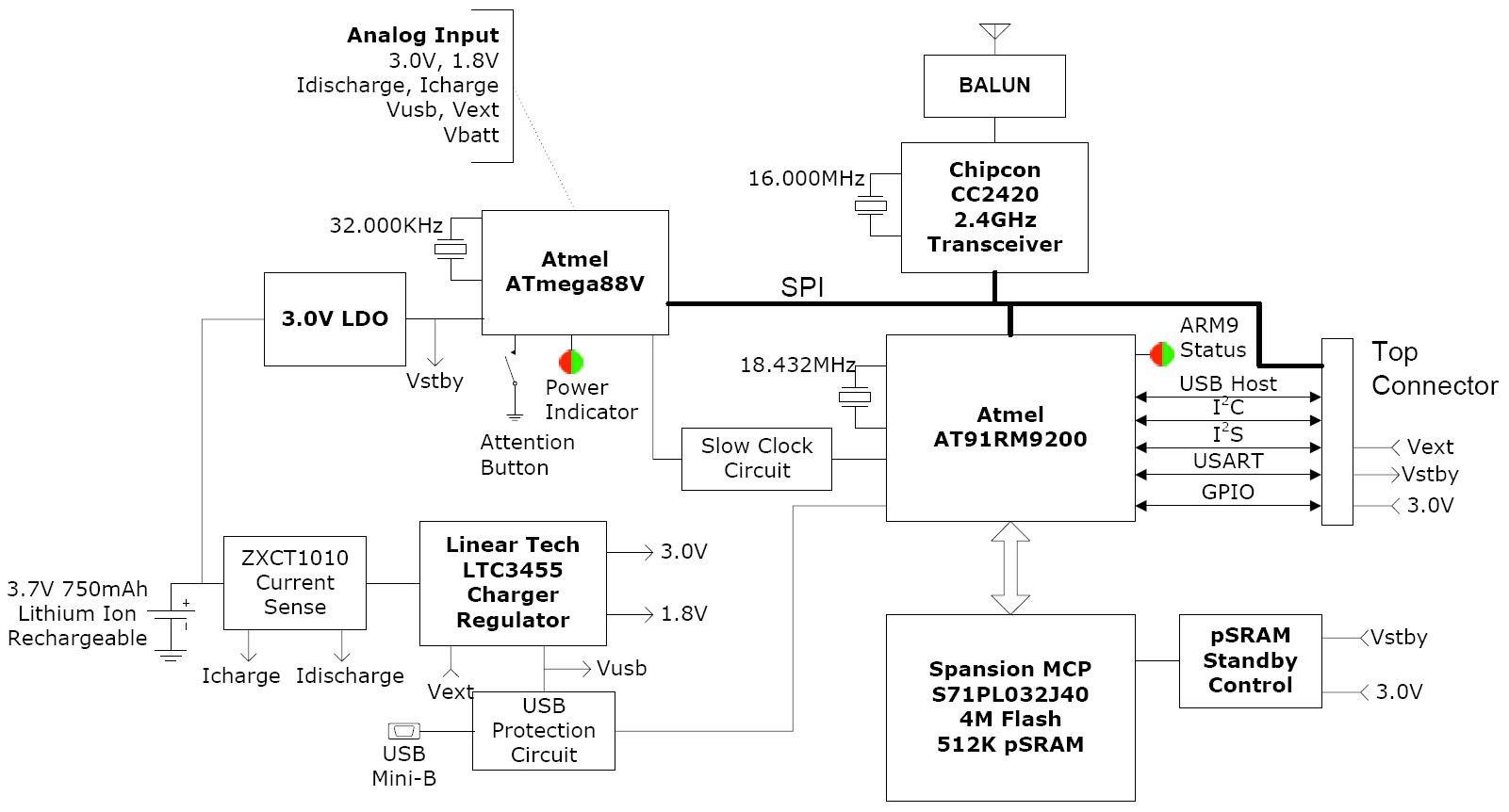
H έκδοση 3.0 των Sun SPOTs περιλαμβάνει έναν προσομοιωτή. Ο προσομοιωτής αυτός επιτρέπει την εκτέλεση μίας εφαρμογής Sun SPOT στον ηλεκτρονικό υπολογιστή. Κατ'επέκταση ο προσομοιωτής επιτρέπει τη δοκιμή ενός προγράμματος πριν αυτό “φορτωθεί” και εκτελεστεί σε ένα πραγματικό Sun SPOT. Αντί ενός φυσικού πίνακα αισθητήρων ο προσομοιωτής επιδεικνύει ένα εικονικό Sun SPOT με εικονικό πίνακα αισθητήρων, όπου ο χρήστης μπορεί να θέσει τιμές στις διάφορες εικονικές παραμέτρους, (φωτισμού, θερμοκρασίας, κίνησης κτλ), καθώς και να ελέγξει τα LEDS όπως ακριβώς και σε ένα πραγματικό Sun SPOT. Υποστηρίζεται επίσης η αποστολή και λήψη δεδομένων μέσω της ασύρματης σύνδεσης. Για την ασύρματη σύνδεση ορίζεται μία διεύθυνση σε κάθε εικονικό Sun SPOT ώστε κάθε εικονικό Sun SPOT να μεταδίδει και να λαμβάνει δεδομένα από άλλα εικονικά Sun SPOTs. Παρόμοια υποστηρίζεται και εικονικός basestation, ο οποίος μπορεί να “φορτώσει” προγράμματα σε εικονικά Sun SPOTs αλλά και να λάβει τα δεδομένα που του στέλνουν τα εικονικά Sun SPOTs. [27]

**Περιγραφή υλικού μέρους**

Ένα Sun SPOT έχεις τις εξής διαστάσεις: μήκος 71 mm, πλάτος 42.40 mm και ύψος 18 mm. Ένα Sun SPOT μπορεί χαρακτηριστικά να χωρέσει στην παλάμη ενός ενήλικα.

Ένα Sun SPOT αποτελείται από τα ακόλουθα στοιχεία:

* Έναν κύριο επεξεργαστή, ο οποίος είναι ένα ολοκληρωμένο κύκλωμα 180 MHz 32-bit ARM920T.
* Μνήμη 4MB Flash και 512K RAM.
* Έναν μικροεπεξεργαστή 8-bit Atmel Atmega 88, ο οποίος χρησιμοποιείται σαν ελεγκτής ισχύος.
* Μία εσωτερική επαναφορτιζόμενη μπαταρία, η οποία φορτίζεται μέσω της θύρας USB, (3.7V rechargeable, 750 mAh lithium – ion battery). Η μπαταρία αυτή, με ενεργό τον επεξεργαστή και την ασύρματη επικοινωνία, μπορεί να διαρκέσει μέχρι και 7 ώρες. Ο χρόνος αυτός μπορεί να εκταθεί αν καμία λειτουργία δε λαμβάνει χώρα, οπότε και το Sun SPOT τίθεται σε κατάσταση αδρανοποίησης. Με συνεχή χρήση των LEDS του πίνακα αισθητήρων, η μπαταρία μπορεί να διαρκέσει μέχρι και 3 ώρες.
* Αισθητήρες θερμοκρασίας και έντασης φωτισμού, ένα τριαξωνικό επιταχύμετρο, 8 τρίχρωμους LEDS, 6 αναλογικές εισόδους που διαβάζονται από έναν αναλογικό-σε-ψηφιακό μετατροπέα, 2 διακόπτες, 5 γενικής χρήσης I/O pins και 4 υψηλής τάσης εξόδους pins.
* Έναν πομποδέκτη TI CC2420, (παλαιότερα ChipCon), με ενσωματωμένη κεραία, ο οποίος χρησιμοποιεί το πρωτόκολλο IEEE 802.15.4 για την ασύρματη επικοινωνία και λειτουργεί στη συχνότητα των 2.4GHz. [27]



**Εικόνα 36: Δομή πλακέτας επεξεργαστή Sun SPOT**

**Περιγραφή λογισμικού**

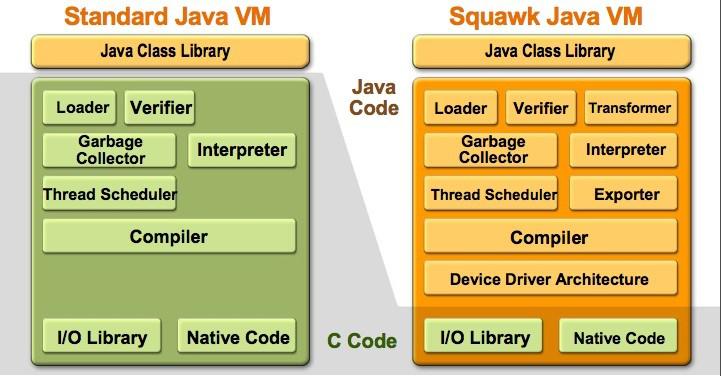
**Squawk Java Virtual Machine**

Το Squawk Java Virtual Machine είναι ένα μηχάνημα εικονικής πραγματικότητας κατασκευασμένο από την εταιρεία πληροφορικής Sun Microsystems για Java εφαρμογές σε ενσωματωμένα συστήματα και μικροσυσκευές, και αποτελεί το βασικό λογισμικό των Sun SPOTs. Προέκυψε από την ανάγκη να δημιουργηθεί ένα μηχάνημα εικονικής πραγματικότητας σε Java συμβατό με τη λειτουργία CLDC, (Connected Limited Device Configuration), που δε θα χρειάζεται την ύπαρξη λειτουργικού συστήματος για να εκτελεστεί. Έτσι, το Squawk JVM σε αντίθεση με τα περισσότερα μηχανήματα εικονικής πραγματικότητας που έχουν γραφτεί σε χαμηλού επιπέδου γλώσσες, όπως C/C++ ή assembly, έχει γραφτεί σε Java και δε χρειάζεται την ύπαρξη λειτουργικού συστήματος για να εκτελεστεί. Συνεπώς το Squawk JVM προσφέρει πολλά πλεονεκτήματα όπως ασφάλεια, συλλογή απορριμμάτων και χειρισμό εξαιρέσεων. Ταυτόχρονα απαιτεί μικρή μνήμη, διευκολύνει τη μεταγλώττιση, (compiling), και εκτέλεση του κώδικα και παρέχει τη δυνατότητα ταυτόχρονης εκτέλεσης πολλαπλών εφαρμογών, με την κάθε εφαρμογή να είναι πλήρως ανεξάρτητη και απομονωμένη από όλες τις υπόλοιπες. [28, 30]

Το Squawk JVM αποτελεί λογισμικό ανοιχτού κώδικα.

Ο μεταγλωττιστής στο Squawk JVM έχει γραφτεί σε C ενώ ο συλλέκτης απορριμμάτων, (garbage collector), μεταφράζεται από την Java στην C.

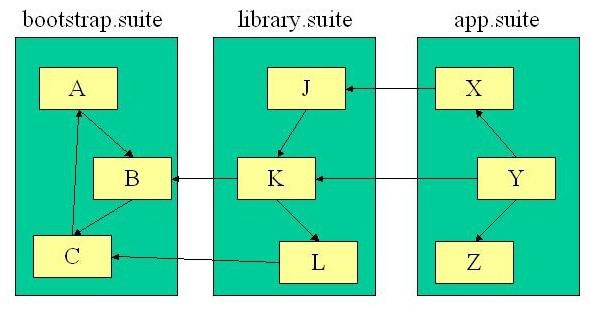
Η αρχιτεκτονική του Squawk JVM παρουσιάζεται στην εικόνα 37.



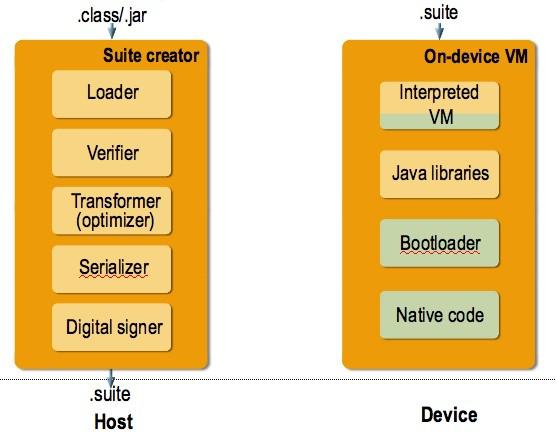
**Εικόνα 37: Αρχιτεκτονική Squawk JVM**

Παρουσιάζει τα εξής χαρακτηριστικά:

* Περιλαμβάνει ένα αμετάβλητο και συμπαγές σύνολο εντολών με κατά 35 – 45% μικρότερο μέγεθος από αυτό των αντίστοιχων J2ME αρχείων, (Java 2 Platform Micro – Edition).
* Παρέχει εσωτερική μετάφραση των κλάσεων σε ένα προσυνδεδεμένο συμπαγές σχήμα, επιτρέποντας έτσι την αποδοτική εκτέλεση των bytecodes που βρίσκονται στη μνήμη μόνο για ανάγνωση.
* Διευκολύνει τη συλλογή απορριμμάτων, (garbage collection), αφού οι μεταβλητές διακρίνονται σε δείκτες και σε καθορισμένες από την Java μεταβλητές, συνεπώς απαιτείται μόνο ένας χάρτης δεικτών για κάθε εφαρμογή, ενώ αφού δεν υπάρχει τίποτα στη στοίβα κατά τη διάρκεια της εκτέλεσης του κώδικα, δεν υπάρχει λόγος για στατική μετάφραση μεθόδων.
* Περιλαμβάνει suites. Μία suite είναι ένα σύνολο κλάσεων. Κάθε κλάση μέσα σε μία suite μπορεί να αναφέρεται σε κλάσεις της ίδιας suite ή σε κλάσεις της γονικής suite. Δημιουργούνται έτσι αλυσίδες από suites, μοντέλο πολύ πρακτικό γιατί οι suites απαιτούν το 1/3 της μνήμης που απαιτούν οι κλάσεις. Στην εικόνα 38 παρουσιάζεται μία αλυσίδα από suites.
* Για εξοικονόμηση μνήμης οι κλάσεις φορτώνονται σε ένα desktop μηχάνημα, όπου μετατρέπονται σε suites και στη συνέχεια φορτώνονται σε μικροσυσκευές και μεταγλωττίζονται από το Squawk JVM των μικροσυσκευών. Αυτό επιτρέπει μικρότερο μέγεθος για το εικονικό μηχάνημα στη μικροσυσκευή και γρηγορότερη εκτέλεση των εφαρμογών. Στην εικόνα 39 παρουσιάζεται η αρχιτεκτονική μεταγλώττισης του Squawk JVM. [30]



**Εικόνα 38: Αλυσίδα από suites**

****

**Εικόνα 39: Αρχιτεκτονική μεταγλώττισης εικονικής μηχανής**

.

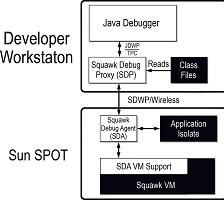
* Παρέχει application isolation. Στο Squawk JVM κάθε εφαρμογή αναπαρίσταται από ένα Java αντικείμενο. Αυτό το Java αντικείμενο είναι ένα στιγμιότυπο της κλάσης Isolate και μπορεί να χρησιμοποιηθεί για να ενημερωθεί για την κατάσταση της σχετιζόμενης εφαρμογής ή να επηρεάσει απευθείας την εφαρμογή μέσω συναρτήσεων όπως start(), pause(), resume() και exit(). Το Squawk JVM παρέχει τη δυνατότητα εκτέλεσης πολλαπλών εφαρμογών στο ίδιο μηχάνημα. Δεδομένης της χρήσης των suites, ο isolate μηχανισμός του Squawk JVM επιτρέπει την κοινή χρήση των suites ανάμεσα σε διαφορετικές εφαρμογές αλλά με την κάθε εφαρμογή να μην επηρεάζεται από την εκτέλεση των υπολοίπων. Έτσι μειώνεται η απαιτούμενη μνήμη κάθε εφαρμογής. Ο isolate μηχανισμός του Squawk JVM υποστηρίζει τα περισσότερα χαρακτηριστικά του JSR 121 Isolate Application Programming Interface αλλά διαφέρει στο ότι χρησιμοποιεί ένα απλούστερο και λιγότερο αυστηρό μοντέλο.
* Παρέχει isolation migration. Κάθε isolate εφαρμογή που εκτελείται σε ένα Squawk JVM μηχάνημα μπορεί να διακοπεί, να μετατραπεί σε ένα stream και να αποθηκευτεί σε ένα αρχείο και στη συνέχεια να επανεκκινήσει σε ένα άλλο μηχάνημα Squawk JVM. Μπορούν επομένως οι εκτελούμενες εφαρμογές να μετακινηθούν από ένα Sun SPOT σε ένα άλλο ή από ένα desktop μηχάνημα σε ένα άλλο ή και από έναν εξυπηρετητή, (server) σε έναν άλλο κ.ο.κ
* Υποστηρίζει πράσινα νήματα, (green threads). Τα πράσινα νήματα προσομοιώνουν πολυνηματικά περιβάλλοντα χωρίς να χρειάζονται κάποιο λειτουργικό σύστημα. Ένα πράσινο νήμα παραδίδει τον έλεγχο σε άλλα νήματα όταν το ίδιο σταματάει τη λειτουργία του, π.χ. με την Thread.yield(), ή όταν πραγματοποιεί μια εντολή που αποκλείει άλλα νήματα, π.χ. read().
* Υποστηρίζει χειρισμό εξαιρέσεων. Όταν πραγματοποιηθεί μία εξαίρεση, προστίθεται ένα bit στην Interrupt Status Word, (ISW), και απενεργοποιείται η εξαίρεση ώστε να μην επαναληφθεί. Κάθε φορά ο χρονοπρογραμματιστής ελέγχει το ISW, στέλνει σήμα στο Squawk JVM, το νήμα συνεχίζει την εκτέλεσή του από το σημείο που είχε διακοπεί και επανενεργοποιείται η εξαίρεση. [29, 30]

**3.3.1.1. Εκσφαλμάτωση**

Το Squawk JVM επιτρέπει οι εφαρμογές στα Sun SPOTs να εκσφαλματώνονται με τη χρήση Java περιβαλλόντων που υποστηρίζουν το Java Debug Wire Protocol, (JDWP), όπως το NetBeans και το JDB, (Java Debugger). Εξαιτίας των περιορισμών στη μνήμη που υπάρχουν στα SunSPOTs, το Squawk JVM δεν εφαρμόζει το JDWP απευθείας πάνω στο Sun SPOT αλλά μοιράζει τη διαδικασία της εκσφαλμάτωσης σε τρία στοιχεία, σε έναν debug proxy, ο οποίος τρέχει στο περιβάλλον του χρήστη, σε έναν debug agent, ο οποίος επικοινωνεί με τον debug proxy και ελέγχει την εφαρμογή που εκσφαλματώνεται και σε μία μικρή debug agent υποστήριξη που παρέχεται από το ίδιο το Squawk JVM. Ο debug proxy και ο debug agent επικοινωνούν μέσω ενός υποεπιπέδου του JDWP γνωστού ως Squawk Debug Wire Protocol, (SDWP).

Η αρχιτεκτονική αυτή επιτρέπει σε στοιχεία της Java Platform Debugger Architecture, (JPDA), που απαιτεί αρκετή μνήμη, να αποθηκεύονται στο περιβάλλον ανάπτυξης του χρήστη και όχι στο Squawk JVM, μειώνοντας έτσι το overhead της μνήμης. Πιο συγκεκριμένα ο debug proxy στο περιβάλλον ανάπτυξης του χρήστη έχει πρόσβαση στις αρχικές κλάσεις που εισήχθησαν σε suites στη συσκευή, επομένως έχει πρόσβαση και στους πίνακες, στις μεθόδους, στα πεδία και στα τοπικά ονόματα μεταβλητών που άλλαξαν σε μία suite.

55



**Εικόνα 40: Αρχιτεκτονική εκσφαλμάτωσης Squawk JVM**

Εκ κατασκευής το Squawk JVM περιλαμβάνει και ένα δεύτερο χαμηλότερου επιπέδου debugger για την εκσφαλμάτωση του υλικού των συσκευών. Καθένας από αυτούς τους debuggers προσθέτει ένα 10% overhead στο βρόχο μεταγλώττισης ελέγχοντας αν ένα breakpoint έχει οριστεί ή όχι. [28]

**3.3.1.2. Flash μνήμη**

Τα 4MB flash μνήμης στα Sun SPOTs έχουν χαμηλή ισχύ και μπορούν να εκτελέσουν το μέγιστο ένα εκατομμύριο κύκλους ανά τομέα. Το 1/3 αυτής της μνήμης διατηρείται για κώδικα συστήματος χωρίς να χρησιμοποιείται ολόκληρο ενώ τα 2/3 διατηρούνται για εφαρμογές και δεδομένα. Η εικόνα 41 δείχνει την κατανομή της μνήμης για τα διάφορα στοιχεία του Squawk JVM και τις σχετιζόμενες βιβλιοθήκες.

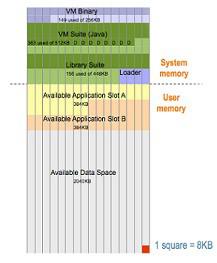
Η μνήμη του συστήματος κατανέμεται ως εξής: 256 KB διατηρούνται για το δυαδικό Squawk JVM και από αυτά τα 149 KB βρίσκονται σε χρήση, 512 KB διατηρούνται για τις suites του Squawk JVM και από αυτά τα 363 KB βρίσκονται σε χρήση, εκ των οποίων τα 64 KB διατηρούνται για τον debugger, 448 KB διατηρούνται για τις βιβλιοθήκες των suites και από αυτά τα 156 KB βρίσκονται στη διάθεση του χρήστη και 64 KB διατηρούνται και χρησιμοποιούνται από τον bootloader. Η μνήμη που διατηρείται για το χρήστη αποτελείται από δύο μέρη, καθένα εκ των οποίων αποτελείται από 384 KB, και από 2040 KB που διατίθενται για τις εφαρμογές. [31]

**3.3.1.3. RAM μνήμη**

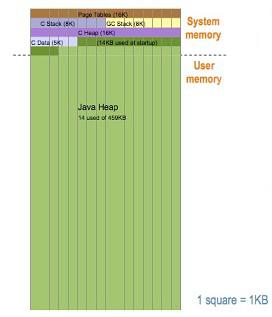
Κάθε Sun SPOT έχει μνήμη 512 KB SRAM. Λιγότερο από το 20% της SRAM διατηρείται για τη μνήμη συστήματος και η υπόλοιπη μνήμη διατίθεται για τα αντικείμενα των εφαρμογών. Η εικόνα 42 δείχνει την κατανομή της μνήμης RAM.

Η μνήμη του συστήματος κατανέμεται ως εξής: 16 KB διατηρούνται για τους πίνακες σελίδων, 8 KB διατηρούνται για τη C στοίβα, 8 KB διατηρούνται για τον συλλέκτη απορριμμάτων, 16 KB διατηρούνται για τον C σωρό, 5 KB διατηρούνται για τα C δεδομένα και 14 ΚΒ διατηρούνται για την εκκίνηση. Ο Java σωρός διατηρεί 459 KB. [31]

56



**Εικόνα 41: Κατανομή flash μνήμης**

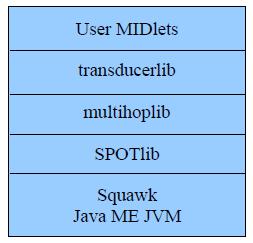
****

**Εικόνα 42: Κατανομή RAM μνήμης**

57

**3.3.2 Αρχιτεκτονική Sun SPOTs**

Η εικόνα 43 παρουσιάζει τη στοίβα λογισμικού σε ένα Sun SPOT.



**Εικόνα 43: Στοίβα λογισμικού σε ένα Sun**

**SPOT**

Στην κορυφή της στοίβας βρίσκεται η γραμμένη από το χρήστη Sun SPOT εφαρμογή, η οποία χρησιμοποιεί την Java MIDlet κλάση. Στον πάτο βρίσκεται το Squawk JVM. Δε μεσολαβεί κανένα λειτουργικό σύστημα, το Squawk JVM εκτελείται χωρίς να απαιτεί κάποιο λειτουργικό σύστημα. Στο ενδιάμεσο βρίσκονται διάφορες βιβλιοθήκες. Η SPOTlib παρέχει τις βασικές λειτουργίες εισόδου – εξόδου καθώς και πρόσβαση στις Sun SPOT συσκευές αλλά και στο χαμηλού επιπέδου MAC, (Medium Access Control), πρωτόκολλο. Η multihoplib παρέχει υψηλού επιπέδου radio πρωτόκολλα όπως radiogram και radiostream και είναι αρμόδια για τη δρομολόγηση των πακέτων των Sun SPOTs όταν τα τελευταία δε βρίσκονται σε άμεση επαφή μεταξύ τους. Η transducerlib πρόσβαση στο υλικό της eDemo πλακέτας του Sun SPOT, δηλαδή στο επιταχύμετρο, στα LEDs, στις αναλογικές εισόδους κ.α. [33]

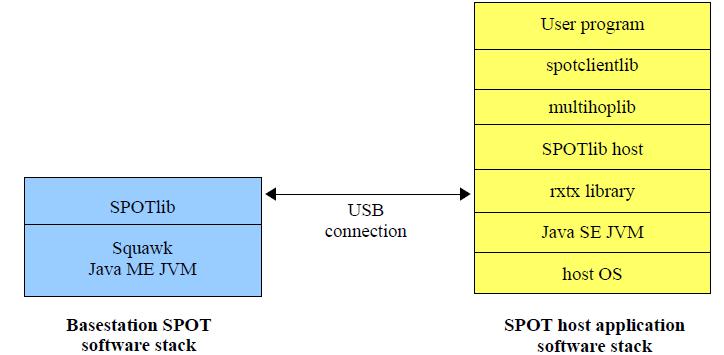
**3.3.3 Αρχιτεκτονική Sun SPOT εφαρμογής**

Η εικόνα 44 περιγράφει την αρχιτεκτονική μίας Sun SPOT εφαρμογής και της σύνδεσής της με τον basestation.

Στην κορυφή βρίσκεται η γραμμένη από το χρήστη Java εφαρμογή. Στον πάτο βρίσκεται το λειτουργικό σύστημα που φιλοξενεί την εφαρμογή, μπορεί να είναι Linux, Windows, Mac OS X ή Solaris. Ακριβώς πάνω από το λειτουργικό βρίσκεται το Java VM που περιλαμβάνει όλες τις Java βιβλιοθήκες. Ενδιάμεσα βρίσκονται διάφορες SPOT βιβλιοθήκες. Η SPOTlib παρέχει τις βασικές λειτουργίες εισόδου – εξόδου καθώς και πρόσβαση στις Sun SPOT συσκευές. Περιλαμβάνει επίσης πρόσβαση στο χαμηλού επιπέδου MAC radio πρωτόκολλο, που χρησιμοποιεί είτε την usb θύρα για να συνδεθεί με τον basestation είτε socket συνδέσεις για επικοινωνία με άλλες εφαρμογές. Η multihoplib παρέχει υψηλού επιπέδου radio πρωτόκολλα όπως radiogram και radiostream και είναι αρμόδια για τη δρομολόγηση των πακέτων. Η spotclientlib παρέχει έναν αριθμό από OTA ,(over-the-air), εντολές που μπορούν να σταλθούν από τον basestation σε άλλα Sun SPOTs. Η rxtx βιβλιοθήκη χρησιμοποιείται για λειτουργίες

58

εισόδου – εξόδου στον basestation μέσω της θύρας usb. [33]



**Εικόνα 44: Αρχιτεκτονική μίας Sun SPOT εφαρμογής και της σύνδεσής της με τον basestation**

**3.3.4 Αρχιτεκτονική basestation**

Ο basestation παρέχει έναν τρόπο ώστε εφαρμογές να επικοινωνούν με Sun SPOTs κάνοντας χρήση του basestation. Οι εφαρμογές εκτελούνται αποκλειστικά στον υπολογιστή και στον basestation δεν εκτελείται κανένας κώδικας χρήστη. Πακέτα που πρέπει να σταλθούν σε Sun SPOTs στέλνονται μέσω usb στον basestation, ο οποίος στη συνέχεια τα στέλνει στα Sun SPOTs με ασύρματη σύνδεση. Ομοίως όταν σταλθούν πακέτα στον basestation από Sun SPOTs, ο basestation τα προωθεί στην εφαρμογή που εκτελείται στον υπολογιστή. [33]

**3.3.5 Βιβλιοθήκες**

Οι βιβλιοθήκες για τα Sun SPOTs αποτελούν λογισμικό ανοιχτού κώδικα και αναπτύσσονται και ανανεώνονται συνεχώς. Χωρίζονται στις εξής κατηγορίες:

* **spots -core-libraries** – συνεχής ανάπτυξη βιβλιοθηκών για τη λειτουργία τωνSun SPOTs
* **spots -network -library** – συνεχής ανάπτυξη βιβλιοθηκών για ζητήματαδικτύωσης των Sun SPOTs
* **spots -sensorboard -library** – συνεχής ανάπτυξη βιβλιοθηκών για τους πίνακεςαισθητήρων των Sun SPOTs
* **spots -security** – συνεχής ανάπτυξη βιβλιοθηκών για ζητήματα ασφάλειας τωνSun SPOTs [37]



59

**3.4 Προγραμματίζοντας τα Sun SPOTs**

**3.4.1 MIDlets**

Οι εφαρμογές στα Sun SPOTs ονομάζονται MIDlets. Τα MIDlets χρησιμοποιούν τις προδιαγραφές του MIDP, (Mobile Information Device Profile), όπως αυτές ορίζονται από το πρωτόκολλο CLDC, (Connected Limited Device Configuration), και προορίζονται για Java περιβάλλοντα σε μικροσυσκευές. Τυπικές εφαρμογές των MIDlets αποτελούν τα παιχνίδια σε κινητά τηλέφωνα, τα οποία υποστηρίζουν γραφικά, χρήση πληκτρολογίου για επικοινωνία με το χρήστη και περιορισμένη σύνδεση στο Διαδίκτυο μέσω HTTP.

Οι τρεις πιθανές καταστάσεις στον κύκλο ζωής ενός MIDlet είναι οι εξής:

1. **paused** – το MIDlet έχει δημιουργηθεί και είναι ανενεργό
2. **active** – το MIDlet είναι ενεργό
3. **destroyed** – το MIDlet έχει καταστραφεί και αναμένεται η αποκατάστασή του απότον συλλέκτη απορριμμάτων

Κάθε MIDlet είναι ένα σύνολο από κλάσεις σχεδιασμένο να ελέγχεται και να εκτελείται από το εκάστοτε σύστημα ελέγχου λογισμικού της κινητής συσκευής, δηλαδή στην περίπτωση των Sun SPOTs από το Squawk JVM. Κάθε midlet πρέπει να περιέχει μία κλάση, η οποία θα επεκτείνει την javax.microedition.midlet.MIDlet κλάση. Οι μέθοδοι σε αυτήν την κλάση θα επιτρέπουν στο σύστημα ελέγχου λογισμικού της κινητής συσκευής να δημιουργεί, να εκκινεί, να διακόπτει και να καταστρέφει την εφαρμογή. Ομοίως, όλα τα MIDlets που εκτελούνται στα Sun SPOTs περιέχουν τις συναρτήσεις startApp(), pauseApp() και destroyApp().

Το MIDlet εκ κατασκευής βρίσκεται στην paused κατάσταση. Με την startApp() δεσμεύονται οι απαραίτητοι πόροι για την εκτέλεση της εφαρμογής και ενεργοποιείται το MIDlet. Όποτε η εκτέλεση γίνει ανενεργή ή συμβεί μία εξαίρεση, καλείται η pauseApp() και απελευθερώνονται προσωρινά οι αρχικά δεσμευμένοι πόροι. Όταν ολοκληρωθεί η εκτέλεση της εφαρμογής καλείται η destroyApp(), η οποία αποδεσμεύει τους δεσμευμένους πόρους του MIDlet και καταστρέφει το MIDlet, δηλαδή αποκαθίσταται από τον συλλέκτη απορριμμάτων.

Τα MIDlets ενσωματώνονται σε suites. Μία MIDlet suite αποτελείται από ένα .jad αρχείο, (Java Application Descriptor), και ένα .jar αρχείο, (Java Archive).

Το .jad αρχείο περιγράφει τη MIDlet suite, περιλαμβάνει δηλαδή το όνομα της, απαιτήσεις και χαρακτηριστικά της λειτουργίας της, τη θέση και το μέγεθος του .jar αρχείου καθώς και προδιαγραφές που ορίζονται από το MIDP ή και από τον προγραμματιστή των MIDlets. Η σύνταξη του .jad αρχείου είναι παρόμοια με αυτή της java.util.Properties κλάσης. Το .jad αρχείο χρησιμοποιείται επίσης για να φορτώσει τις εφαρμογές στα Sun SPOTs μέσω OTA, (over-the-air). Σε αυτή τη μέθοδο τα .jad και .jar αρχεία μεταφορτώνονται από το Sun SPOT μέσω HTTP σε έναν εξυπηρετητή δικτύου. Ο χρήστης στη συνέχεια κατεβάζει το .jad αρχείο και εγκαθιστά στο Sun SPOT τα MIDlets που απαιτούνται.

Το .jar αρχείο περιλαμβάνει ένα MANIFEST.MF αρχείο, το οποίο περιλαμβάνει πληροφορίες απαραίτητες για την εκτέλεση της εφαρμογής από το Squawk JVM, καθορίζει δηλαδή στην ουσία ποιες κλάσεις χρησιμοποιούν ποια MIDlets. Κάθε .jar αρχείο έχει αποκλειστικά και μόνο ένα MANIFEST.MF αρχείο και αυτό βρίσκεται στον υποκατάλογο META – INF. [32, 38, 39]

60

**3.4.1.1 Μορφή του MIDlet στα Sun SPOTs**

Ένα MIDlet για να φορτωθεί και να εκτελεστεί σε ένα Sun SPOT πρέπει να έχει συγκεκριμένη μορφή. Πιο συγκεκριμένα, ο γονικός κατάλογος που περιέχει το MIDlet, πρέπει να περιέχει και τα αρχεία build.xml και build.properties, τα οποία ελέγχουν το ant script που θα μεταγλωττίσει, εκσφαλματώσει και εκτελέσει την εφαρμογή. Ο γονικός αυτός κατάλογος πρέπει να περιέχει και τρεις υποκαταλόγους, τον src, τον nbproject και τον resources. Ο src θα περιέχει όλο τον πηγαίο κώδικα του MIDlet. Ο nbproject θα περιέχει αρχεία που έχουν δημιουργηθεί στην περίπτωση που ο κώδικας γραφτεί με χρήση του IDE Netbeans, (Integrated Development Environment Netbeans), και ο resources θα περιέχει το αρχείο MANIFEST.MF, το οποίο καθορίζει χαρακτηριστικά του MIDlet. Για να φορτωθεί σε Sun SPOTs και να εκτελεστεί ένα MIDlet πρέπει να έχει την παραπάνω μορφή. [33]

**3.4.2 Apache Ant**

To Apache Ant, (Another Neat Tool), είναι μία scripting γλώσσα που έχει σχεδιαστεί για τη γρηγορότερη και ευκολότερη εκσφαλμάτωση, μεταγλώττιση και εκτέλεση του κώδικα, αυτοματοποιώντας όλη αυτή τη διαδικασία, (build processing). Αποτελεί λογισμικό ανοιχτού κώδικα.

Το Ant χρησιμοποιεί την Java και απαιτεί Java περιβάλλοντα για την εκτέλεση του. Είναι παρόμοιο με το Make με τη διαφορά ότι για την περιγραφή της build διαδικασίας παράγει xml αρχεία. Εκ κατασκευής κάθε αρχείο ονομάζεται build.xml.

Αναπτύχθηκε αρχικά ως εργαλείο που ανεξαρτήτως πλατφόρμας θα αυτοματοποιούσε την build διαδικασία στον Apache Tomcat με χρήση xml αρχείων. Στην πορεία όμως διαμορφώθηκε ως build εργαλείο για όλα σχεδόν τα Java projects.

Τα Sun SPOTs χρησιμοποιούν το Ant κατά την build διαδικασία. Στην ουσία κάθε φορά που φορτώνεται ένα MIDlet δημιουργείται ένα build.xml, το οποίο εκτελείται και τελικά εκτελείται και η εφαρμογή του MIDlet. [40, 41]

**3.5 Επικοινωνία με basestation**

Το basestation αποτελεί τον κεντρικό κόμβο σε ένα δίκτυο από Sun SPOTs. Συνδέεται με τον υπολογιστή και επικοινωνεί ασύρματα με απομακρυσμένα Sun SPOTs. Φορτώνει εφαρμογές στα Sun SPOTs και συλλέγει τα δεδομένα που του στέλνουν. Ο κώδικας που εκτελείται κάθε φορά σε ένα basestation είναι μία τυπική Java εφαρμογή. Το basestation δε διαθέτει πίνακα αισθητήρων και δεν μπορεί να εφαρμόσει καμία αναπτυξιακή πλακέτα πάνω του. Το ρόλο αυτό μπορεί να αναλάβει και κάθε Sun SPOT αλλά σε αυτή την περίπτωση δε θα μπορεί να χρησιμοποιηθεί ο πίνακας αισθητήρων του.

Για την επικοινωνία μεταξύ basestation και Sun SPOTs χρησιμοποείται το πρωτόκολλο IEEE 802.15.4. Το πρωτόκολλο αυτό χρησιμοποιείται κυρίως για χαμηλής ταχύτητας ασύρματα δίκτυα προσωπικών επικοινωνιών, (Low – Rate Wireless Personal Area Networks, LR – WPAN). Η εμβέλεια του δικτύου φτάνει τα 10 μέτρα ενώ ο ρυθμός μετάδοσης αγγίζει τα 250 kbit/s.

Κάθε Sun SPOT διαθέτει μία IEEE extended MAC διεύθυνση, η οποία χρησιμοποείται για την αναγνώριση του Sun SPOT από τον basestation. Η διεύθυνση αυτή είναι μία 64-bit διεύθυνση με την ακόλουθη μορφή: 0014.4F01.xxxx.xxxx. Κάθε ψηφίο είναι ένα 16-δικό ψηφίο και τα 8 τελευταία ψηφία διαφέρουν από Sun SPOT σε Sun SPOT. Η διεύθυνση αυτή βρίσκεται τυπωμένη στην πλακέτα κάθε Sun SPOT.

61

Κατά την επικοινωνία του basestation με ένα Sun SPOT, το basestation πρέπει να λάβει επιβεβαίωση από το Sun SPOT ότι το πακέτο που έστειλε έχει παραληφθεί σωστά. Εάν δεν υπάρξει καμία επιβεβαίωση τότε συμβαίνει NoAckException εξαίρεση και διακόπτεται η εκτέλεση της εφαρμογής. Κατά την επικοινωνία του basestation με τον υπολογιστή μέσω της σύνδεσης usb εάν δεν υπάρξει NoAckException εξαίρεση τότε θεωρείται ότι η παράδοση του μηνύματος από τον υπολογιστή στο basestation έχει πραγματοποιηθεί σωστά. Όταν πραγματοποιηθεί NoAckException, δηλαδή το πακέτο δεν παραδοθεί από το basestation στο Sun SPOT, τότε η εφαρμογή παράγει warning.

Το basestation μπορεί να εκτελείται σε dedicated ή σε shared μορφή. Στην dedicated μορφή το basestation εκτελείται στο ίδιο JVM, (Java Virtual Machine), που εκτελείται και η εφαρμογή που τρέχει στον υπολογιστή με τον οποίο είναι συνδεδεμένο το basestation. JVM και basestation μοιράζονται την ίδια διεύθυνση σε αυτή την περίπτωση. Στην shared μορφή εκτελούνται δύο JVMs στον υπολογιστή που συνδέεται με το basestation. Το ένα ελέγχει το basestation και το άλλο εκτελεί την εφαρμογή. Σε αυτή τη μορφή η εφαρμογή που εκτελείται στον υπολογιστή έχει τη δική της διεύθυνση που έχει παραχθεί από το σύστημα και είναι διαφορετική από τη διεύθυνση που έχει το basestation. Η επικοινωνία επομένως στην shared μορφή απαιτεί δύο συνδέσεις σε αντίθεση με την dedicated μορφή όπου απαιτείται μία σύνδεση.

Το πλεονέκτημα της shared μορφής είναι ότι περισσότερες από μία εφαρμογές μπορούν να χρησιμοποιήσουν το ίδιο basestation ταυτόχρονα και να επικοινωνήσουν έτσι και μεταξύ τους. Το μειονέκτημα είναι ότι ο χειρισμός του καναλιού στο οποίο πραγματοποιείται η μετάδοση των πακέτων και ο χειρισμός του PAN id, (Personal Area Network identification), κατά την υλοποίηση της σύνδεσης δεν είναι εφικτός.

Εκ κατασκευής οι εφαρμογές που χρησιμοποιούν το basestation βρίσκονται σε dedicated μορφή. Για να αλλάξουν σε shared μορφή θα πρέπει στο φάκελο .sunspot.properties του γονικού φακέλου του χρήστη να εισαχθεί η εντολή basestation.shared = true.

Η επικοινωνία μεταξύ basestation και Sun SPOTs μπορεί να πραγματοποιηθεί με δύο τρόπους, είτε point-to-point είτε broadcast. Point-to-point επικοινωνία ονομάζεται η επικοινωνία αποκλειστικά μεταξύ δύο συσκευών ενώ broadcast επικοινωνία ονομάζεται η επικοινωνία μίας με περισσότερες από μία συσκευές. Και οι δύο αυτές μορφές επικοινωνίας επιτυγχάνονται μέσω δύο πρωτοκόλλων, του radiostream πρωτοκόλλου και του radiogram πρωτοκόλλου. [31, 32, 33]

**3.5.1 radiostream – radiogram πρωτόκολλα**

Το radiostream πρωτόκολλο παράγει ένα αξιόπιστο συνεχές ρεύμα εισόδου – εξόδου μεταξύ δύο συσκευών με χρήση sockets.

Με την εντολή

RadiostreamConnection conn = (RadiostreamConnection)Connector.open(“radiostream:<destAddr>:<portNo>”);

ανοίγει μία σύνδεση. destAddr είναι η 64-bit διεύθυνση της συσκευής προορισμού και portNo είναι ο αριθμός της θύρας που χρησιμοποιείται ως αναγνωριστικό της συγκεκριμένης σύνδεσης και κυμαίνεται μεταξύ 0 και 255. Το 0 δε θεωρείται έγγυρη IEEE διεύθυνση σε αυτή την εφαρμογή. Η σύνδεση που ανοίγει χρησιμοποιεί εκ κατασκευής το κανάλι 26 και το PAN identifier 3. Για να εδραιωθεί η σύνδεση θα πρέπει και οι δύο συσκευές της σύνδεσης να χρησιμοποιήσουν το ίδιο portNo και τις αντίστοιχες IEEE διευθύνσεις. Όταν εδραιωθεί η σύνδεση μπορεί καθεμία από τις δύο συσκευές να λάβει και να αποστείλει δεδομένα με τις παρακάτω εντολές.

62

DataInputStream dis = conn.openDataInputStream(); DataOutputStream dos = conn.openDataOutputStream();

Το radiogram πρωτόκολλο χρησιμοποιεί το μοντέλο πελάτη – εξυπηρετητή, (client – server), και παράγει datagrams μεταξύ των δύο συσκευών. Το πρωτόκολλο αυτό δεν παρέχει καμία εγγύηση για την παράδοση των datagrams. Τα datagrams ενδέχεται να μην παραδοθούν καθόλου ή να παραδοθούν με λάθος σειρά ή να παραδοθούν περισσότερες από μία φορές.

Με την εντολή

RadiogramConnection conn = (RadiogramConnection)Connector.open(“radiogram://:<portNo>”);

ανοίγει ο εξυπηρετητή μία σύνδεση. portNo είναι ο αριθμός της θύρας που χρησιμοποιείται ως αναγνωριστικό της συγκεκριμένης σύνδεσης και κυμαίνεται μεταξύ 0 και 255. Εκ κατασκευής χρησιμοποιούνται από τη σύνδεση το κανάλι 26 και το PAN identifier 3.

Με την εντολή

RadiogramConnection conn = (RadiogramConnection)Connector.open(“radiogram://<serveraddr>:<portNo>”);

ανοίγει ο πελάτης μία σύνδεση. serveraddr είναι η 64-bit ΙΕΕΕ διεύθυνση του εξυπηρετητή και portNo είναι ο αριθμός της θύρας που χρησιμοποιείται ως αναγνωριστικό της συγκεκριμένης σύνδεσης και κυμαίνεται μεταξύ 0 και 255. Το 0 δε θεωρείται έγγυρη IEEE διεύθυνση σε αυτή την εφαρμογή. Ομοίως εξυπηρετητής και πελάτης πρέπει να χρησιμοποιούν το ίδιο portNo.

Με την εντολή

Datagram dg = conn.newDatagram(conn.getMaximumLength()); δημιουργείται ένα άδειο datagram από τη σύνδεση.

Με την εντολή conn.send(dg);

αποστέλλεται ένα datagram και με την παρακάτω εντολή λαμβάνεται ένα datagram. conn.receive(dg);

Ακολουθούν κάποια αξιοσημείωτα χαρακτηριστικά των datagrams:

* + Datagrams που λαμβάνονται από μία σύνδεση δεν μπορούν να χρησιμοποιηθούν σε άλλη σύνδεση.
  + Η διεύθυνση στην οποία πραγματοποιείται μία σύνδεση είναι χαρακτηριστική για τη σύνδεση. Προσπάθειες να αποσταλούν datagrams σε διαφορετική διεύθυνση από αυτή της σύνδεσης θα καταλήγουν σε εξαίρεση.
  + Επιτρέπεται να ανοιχθούν συνδέσεις πελάτη και διακομιστή στο ίδιο μηχάνημα με χρήση του ίδιου αριθμού θύρας. Όλα τα εισερχόμενα σε αυτή τη θύρα datagrams θα δρομολογούνται στη σύνδεση του διακομιστή.
  + Εάν κλείσει η σύνδεση του διακομιστή, κλείνουν και όλες οι συνδέσεις του πελάτη σε αυτό το διακομιστή.
* broadcast επικοινωνία πραγματοποιείται με το radiogram πρωτόκολλο. Δεν είναι αξιόπιστη καθώς ενδέχεται τα datagrams να μην παραδοθούν καθόλου στους προορισμούς τους, αφού τα Sun SPOTs αγνοούν τις επιβεβαιώσεις των broadcast

63

μηνυμάτων που τους στάλθηκαν ή που έστειλαν τα ίδια. Με την εντολή

DatagramConnection conn = (DatagramConnection)Connector.open(“radiogram://broadcast:<portnum>”);

ανοίγει μία radiogram σύνδεση για broadcast επικοινωνία. portnum είναι ο αριθμός της θύρας που θα χρησιμοποιηθεί. Οι broadcast συνδέσεις δεν είναι αμφίδρομες, χρησιμοποιούνται δηλαδή μόνο για να σταλθούν πακέτα και όχι να ληφθούν. Για να ληφθούν πακέτα από μία broadcast σύνδεση θα πρέπει να ανοιχθεί μία σύνδεση διακομιστή στον ίδιο αριθμό θύρας. [32, 33]

**3.5.2 broadcasting και basestations**

Τα basestations όπως έχει ήδη αναφερθεί λειτουργούν σε dedicated ή σε shared μορφή. Έστω ένα Sun SPOT που βρίσκεται εκτός εμβέλειας ενός basestation και κάνει broadcast ένα radiogram με τις εκ κατασκευής δύο συνδέσεις ορισμένες. Ένα άλλο Sun SPOT που βρίσκεται εντός της εμβέλειας του basestation λαμβάνει το radiogram και το επανασυνδέει με το basestation. Επειδή το basestation λαμβάνει το radiogram μετά από δύο συνδέσεις, δεν το επανασυνδέει.

Στην dedicated μορφή επικοινωνίας η εφαρμογή που τρέχει στον υπολογιστή με τον οποίο είναι συνδεδεμένο το basestation, χρησιμοποιεί τη διεύθυνση του basestation και επομένως λαμβάνει το radiogram. Στην shared μορφή επικοινωνίας η εφαρμογή που τρέχει στον υπολογιστή με τον οποίο είναι συνδεδεμένο το basestation, βρίσκεται μία σύνδεση μακρυά από τη διαδικασία που ελέγχει το basestation και επομένως δε λαμβάνει το radiogram. Για αυτό το λόγο οι προγραμματιστές όταν χρησιμοποιούν την broadcast μορφή επικοινωνίας για απομακρυσμένα Sun SPOTs, πρέπει να λάβουν υπ'όψιν τους την επίδραση του basestation στον αριθμό των broadcast συνδέσεων που ορίζουν. [33]

**3.6 Εκτέλεση προγραμμάτων στα Sun SPOTs**

Η φόρτωση, εκτέλεση και εκσφαλμάτωση κώδικα στα Sun SPOTs μπορεί να πραγματοποιηθεί είτε μέσω της θύρας usb του υπολογιστή, είτε χρησιμοποιώντας κάποιο IDE, (Integrated Development Environment), όπως Netbeans ή Eclipse, είτε με χρήση του basestation και της over-the-air μεθόδου.

**3.6.1 USB**

Από τη γραμμή εντολών του λειτουργικού και στον κατάλογο όπου βρίσκεται η εφαρμογή που θα φορτωθεί, με την εντολή ant jar-app γίνεται compiled και δημιουργείται ένα .jar αρχείο με το όνομα <MIDlet-Name>\_<MIDlet-Version>.jar. Στη συνέχεια και έχοντας συνδέσει το Sun SPOT στη θύρα usb του υπολογιστή, με την εντολή ant jar-deploy φορτώνεται ο κώδικας στο Sun SPOT και τέλος με την εντολή ant run εκτελείται. [34]

**3.6.2 Over-the-air**

Στη μέθοδο αυτή χρησιμοποιείται το basestation. Για να επικοινωνήσει το basestation με άλλα Sun SPOTs πρέπει να είναι γνωστή η 64-bit ΙΕΕΕ διεύθυνση των Sun SPOTs.

64

Από τη γραμμή εντολών στον κατάλογο όπου βρίσκεται η εφαρμογή που θα φορτωθεί και έχοντας συνδέσει το basestation στη θύρα usb του υπολογιστή, με την εντολή ant info εμφανίζονται πληροφορίες για το συνδεδεμένο Sun SPOT, όπως εάν το Sun SPOT λειτουργεί ως basestation και ποιες libraries χρησιμοποιούνται από το Sun SPOT. Για να λειτουργεί το Sun SPOT ως basestation, θα πρέπει να εμφανιστούν οι εντολές Configured as a Basestation και OTA Command Server is disabled. Εάν οι εντολές αυτές δεν εμφανιστούν τότε με χρήση των εντολών ant startbasestation και ant disableota το basestation θα λειτουργεί ως basestation. Τα Sun SPOTs που θα συνδεθούν με τον basestation θα πρέπει να έχουν ενεργό τον OTA Command Server ειδάλλως δε θα υλοποιηθεί η σύνδεση. Ο OTA Command Server είναι στην ουσία ένα νήμα που κατά την εκτέλεσή του ακούει σε άλλες εντολές, δηλαδή στην ουσία δέχεται εντολές από άλλες συσκευές, από το basestation στην προκειμένη περίπτωση.

Στη συνέχεια με την εντολή ant -DremoteId = 0014.4F01.xxxx.xxxx deploy, όπου 0014.4F01.xxxx.xxxx είναι η IEEE διεύθυνση του απομακρυσμένου Sun SPOT, φορτώνεται η εφαρμογή μέσω του basestation στο Sun SPOT. Τέλος με την εντολή ant -DremoteId = 0014.4F01.xxxx.xxxx run εκτελείται η εφαρμογή στο απομακρυσμένο Sun SPOT.

Το output της εκτέλεσης θα εμφανιστεί στο παράθυρο της γραμμής εντολών. Το output της εκτέλεσης θα σταματήσει με χρήση του ctrl-C, η εφαρμογή όμως θα συνεχίσει να εκτελείται στο Sun SPOT. Για να διακοπεί η εκτέλεση και στο Sun SPOT θα πρέπει να απενεργοποιηθεί το Sun SPOT, να σταματήσει η λειτουργία του. [34]

**3.6.3 IDEs**

Τα IDEs που χρησιμοποιούνται για τη φόρτωση προγραμμάτων στα Sun SPOTs είναι το NetBeans και το Eclipse, ελεύθερα προσβάσιμα. Η διαδικασία φόρτωσης εφαρμογών σε Sun SPOTs μέσω IDEs περιλαμβάνει άνοιγμα της εφαρμογής με το IDE, building project και στη συνέχεια φόρτωση των προγραμμάτων στα Sun SPOTs, (deploying to Sun SPOTs). Τα παραπάνω βήματα υλοποιούνται με χρήση των εντολών των εκάστοτε εκδόσεων των IDEs. [34, 42, 43]

**3.7 HTTP πρωτόκολλο**

Το HTTP πρωτόκολλο, (Hypertext Transfer Protocol), υποστηρίζεται από τα Sun SPOTs και επιτρέπει σε απομακρυσμένες SPOT εφαρμογές να ανοίξουν http συνδέσεις προς οποιαδήποτε web υπηρεσία προσβάσιμη από έναν ηλεκτρονικό υπολογιστή. Με την εντολή

HttpConnection connection = (HttpConnection)Connector.open(“http://host: [port]/filepath”);

ανοίγει μία http σύνδεση προς τον host, το όνομα δηλαδή της web υπηρεσίας στο Διαδίκτυο, με τον αριθμό θύρας port, που μπορεί να παραληφθεί και με μονοπάτι προς το συγκεκριμένο πόρο που ζητείται από τον εξυπηρετητή δικτύου, (web server), που καθορίζεται από το filepath.

Για να έχει το http πρωτόκολλο πρόσβαση στο url που ζητείται, θα πρέπει η συσκευή να βρίσκεται εντός εμβέλειας ενός basestation συνδεδεμένου σε υπολογιστή που θα εκτελεί το Socket Proxy πρόγραμμα. Το πρόγραμμα αυτό είναι υπεύθυνο για την επικοινωνία με τον εξυπηρετητή δικτύου που καθορίζεται από το url.

Το http πρωτόκολλο είναι στην ουσία ένας πελάτης http στο Sun SPOT που

65

χρησιμοποιεί το socket πρωτόκολλο. Όταν ανοιχθεί η σύνδεση http, μία αρχική σύνδεση εδραιώνεται με τον Socket Proxy με χρήση του radiogram πρωτοκόλλου στον αριθμό θύρας 10 εκ κατασκευής ή στον αριθμό θύρας που έχει οριστεί στο MANIFEST.MF αρχείο της εφαρμογής που εκτελείται στο Sun SPOT. Ο Socket Proxy απαντάει τότε με ένα διαθέσιμο radio αριθμό θύρας στον οποίο μπορεί η συσκευή να συνδεθεί ώστε να αρχίσει η μεταφορά δεδομένων.

Εκ κατασκευής χρησιμοποιείται μία broadcast σύνδεση για να βρεθεί ο basestation που θα χρησιμοποιηθεί για να ανοίξει μία σύνδεση με τον Socket Proxy. Για να χρησιμοποιηθεί ένας συγκεκριμένος basestation πρέπει η παρακάτω εντολή να προστεθεί στο αρχείο MANIFEST.MF της εφαρμογής που εκτελείται.

com.sun.spot.io.j2me.socket.SocketConnection-BaseStationAddress: <IEEE address>

Για να αλλάξει ο αριθμός θύρας στον οποίο θα πραγματοποιηθεί η αρχική σύνδεση προς τον Socket Proxy πρέπει η παρακάτω εντολή να προστεθεί στο αρχείο MANIFEST.MF της εφαρμογής που εκτελείται.

com.sun.spot.io.j2me.socket.SocketConnection-BaseStationAddress: <radiogram port>

Στην περίπτωση που ο υπολογιστής που εκτελεί τον Socket Proxy βρίσκεται πίσω από έναν proxy διακομιστή, η συσκευή μπορεί να έχει πρόσβαση στο Διαδίκτυο αρκεί η παρακάτω εντολή να έχει οριστεί στο MANIFEST.MF αρχείο.

com.sun.squawk.io.j2me.http.Protocol-HttpProxy: <proxyaddress>:<port>

proxyaddress είναι το όνομα ή η IP διεύθυνση του http proxy διακομιστή και port είναι ο TCP αριθμός θύρας του proxy διακομιστή. [33]

**3.7.1 Socket Proxy GUI τρόπος**

Ο Socket Proxy έχει δύο τρόπους με τους οποίους μπορεί να εκτελεστεί. Εκ κατασκευής εκτελείται με τον τρόπο που περιγράφτηκε παραπάνω. Υπάρχει όμως και μία GUI εκδοχή, (Graphical User Interface), που επιτρέπει το output του κώδικα που εκτελείται να παρουσιαστεί γραφικά και να παρέχει στατιστική ανάλυση των δεδομένων. Για να λειτουργήσει ο Socket Proxy με αυτή τη μορφή πρέπει να χρησιμοποιηθεί η παρακάτω εντολή

ant socket-proxy-gui

Θα εμφανιστεί τότε ένα παράθυρο όπου θα παρουσιάζονται τα εξής:

* **basestation 's serial port** – ο αριθμός της θύρας με τον οποίο είναισυνδεδεμένος ο basestation
* **Radiogram port** – ο αριθμός της θύρας του radiogram που χρησιμοποιήθηκεγια την αρχική σύνδεση. Σε αυτό το επίπεδο μπορεί ο χρήστης να επιλέξει ποια στατιστικά δεδομένα επιθυμεί να δει στο γράφημα.
  1. System : όλα τα στατιστικά δεδομένα που σχετίζονται με τον proxy
  2. Error : στατιστικά δεδομένα που αποτελούν λάθη, εξαιρέσεις δηλαδή
  3. Warning : στατιστικά δεδομένα που αποτελούν warnings
  4. Info : πληροφορίες για εισερχόμενες και εξερχόμενες συνδέσεις
  5. I/O : στατιστικά δεδομένα που αφορούν δραστηριότητα εισόδου – εξόδου



66

* **Radio port usage** – μετρητής του Socket Proxy για κάθε αριθμό θύρας πουσυνδέεται με Sun SPOT. Στην περίπτωση που όλοι οι αριθμοί θύρας του Socket Proxy χρησιμοποιούνται από Sun SPOTs, ο Socket Proxy σταματάει να δέχεται καινούριες συνδέσεις μέχρι να απελευθερωθούν αριθμοί θύρας.



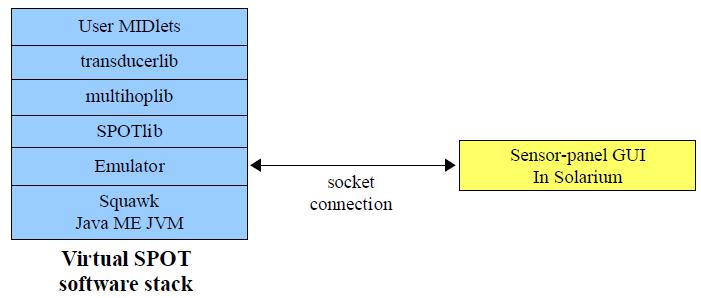
Το τι επιθυμεί να δει ο χρήστης στο γράφημα μπορεί να οριστεί στο socketproxy.loglevel στο αρχείο default.properties του γονικού καταλόγου των Sun SPOTs. Ο αριθμός θύρας που θα χρησιμοποιηθεί από το radiogram για την αρχική σύνδεση μπορεί να οριστεί στο socketproxy.initport στο default.properties αρχείο. [34]

**3.8 Solarium**

Το Solarium είναι μία Java εφαρμογή που εκτελείται στον υπολογιστή και χρησιμοποιείται για να εκτελέσει μια σειρά από ενέργειες στα Sun SPOTs. Έχει τις παρακάτω δυνατότητες:

* Ανακαλύπτει και εμφανίζει Sun SPOTs που είναι συνδεδεμένα στον υπολογιστή μέσω usb ή ασύρματης επικοινωνίας
* Μπορεί να φορτώσει και να ξεφορτώσει λογισμικό από τα Sun SPOTs, να εκκινήσει, διακόψει, επανεκκινήσει και τερματίσει εφαρμογές και να ανιχνεύσει την κατάσταση μίας συσκευής. Το πλαίσιο Radio View παρέχει έναν γραφικό τρόπο αναπαράστασης της ασύρματης συνδεσιμότητας των Sun SPOTs.
* Μπορεί να ελέγξει ένα δίκτυο από Sun SPOTs με χρήση του πλαισίου Deployment View. Το πλαίσιο αυτό καθορίζει ποια εφαρμογή πρέπει να φορτωθεί σε ποια συσκευή και διευκολύνει τη διαδικασία φόρτωσης τους.
* Περιλαμβάνει έναν προσομοιωτή, ο οποίος μπορεί να φορτώσει και να εκτελέσει εικονικές εφαρμογές σε εικονικά Sun SPOTs. [35]

**3.8.1 Λειτουργία προσομοιωτή**

****

**Εικόνα 45: Αρχιτεκτονική προσομοιωτή Solarium**

Με τη δημιουργία ενός εικονικού Sun SPOT στο Solarium, δημιουργείται μία καινούρια διαδικασία η οποία εκτελεί τον κώδικα του προσομοιωτή στο Squawk JVM. Ο κώδικας

67

του προσομοιωτή επικοινωνεί μέσω μίας socket σύνδεσης με τον εικονικό SPOT GUI κώδικα στο Solarium. Στην εικόνα 45 παρουσιάζεται η αρχιτεκτονική του προσομοιωτή.

Κάθε εικονικό Sun SPOT έχει το δικό του Squawk JVM να τρέχει μία ξεχωριστή διαδικασία στον υπολογιστή. Κάθε Squawk JVM περιλαμβάνει ολοκληρωμένη την radio στοίβα ως μέρος της Sun SPOT βιβλιοθήκης, η οποία επιτρέπει σε Sun SPOT εφαρμογές να επικοινωνήσουν με άλλες Sun SPOT εφαρμογές που εκτελούνται στον ίδιο υπολογιστή. [35]

**3.9 Αισθητήρες**

Οι αισθητήρες που περιλαμβάνονται στα Sun SPOTs είναι αισθητήρες φωτισμού, θερμοκρασίας και κίνησης. Για να χρησιμοποιηθεί οποιοσδήποτε από τους παραπάνω αισθητήρες αλλά και οτιδήποτε υπάρχει πάνω στην κεντρική πλακέτα του Sun SPOT, όπως LEDs, αναλογικές είσοδοι κ.α. πρέπει να χρησιμοποιηθεί το πακέτο com.sun.spot.sensorboard.EdemoBoard. Πρέπει επίσης να χρησιμοποιηθεί το πακέτο com.sun.spot.sensorboard.peripheral, το οποίο περιέχει διεπαφές για συγκεκριμένες εφαρμογές διάφορων περιφερειακών μονάδων που συνδέονται με την eDemo πλακέτα. [34, 36]

**3.9.1 Αισθητήρας φωτισμού**

Ο αισθητήρας φωτισμού ανιχνεύει την ένταση του φωτισμού και επιστρέφει έναν ακέραιο που κυμαίνεται από το 0 μέχρι το 750. Το 0 αντικατοπτρίζει το απόλυτο σκοτάδι. Η μέγιστη ευαισθησία του αισθητήρα αγγίζει μήκη κυμάτων 600 nm. Για να δημιουργήσουμε ένα αντικείμενο αισθητήρα φωτισμού χρησιμοποιείται ο παρακάτω κώδικας.

import com.sun.spot.sensorboard.EDemoBoard;

import com.sun.spot.sensorboard.peripheral.ILightSensor;

ILightSensor ourLightSensor = EdemoBoard.getInstance().getLightSensor();

Για να πάρουμε τιμές από τον αισθητήρα φωτισμού γίνεται χρήση του παρακάτω κώδικα. [34, 36]

int lightSensorReading = ourLightSensor.getValue();

**3.9.2 Αισθητήρας θερμοκρασίας**

Ο αισθητήρας θερμοκρασίας ανιχνεύει την ένταση της θερμοκρασίας περιβάλλοντος και επιστρέφει έναν ακέραιο που κυμαίνεται μεταξύ 0 και 1023. Ο αριθμός αυτός δεν αντιπροσωπεύει τη θερμοκρασία ούτε σε Celsius ούτε σε Fahrenheit βαθμούς, είναι ένας ακατέργαστος αριθμός. Το μοντέλο του αισθητήρα φωτισμού είναι ADT7411 με έναν 10-bit αναλογικό-σε-ψηφιακό μετατροπέα ο οποίος μπορεί να ανιχνεύσει θερμοκρασία περιβάλλοντος που κυμαίνεται από -40 σε +125 βαθμούς Celsius. Για να δημιουργηθεί ένα αντικείμενο αισθητήρα θερμοκρασίας χρησιμοποιείται ο παρακάτω κώδικας.

import com.sun.spot.sensorboard.EDemoBoard; import com.sun.spot.sensorboard.io.ITemperatureInput;

ITemperatureInput ourTempSensor = EdemoBoard.getADCTemperature();

68

Για να πάρουμε τιμές από τον αισθητήρα θερμοκρασίας χρησιμοποιείται ο παρακάτω κώδικας.

* ακατέργαστη τιμή θερμοκρασίας int temp = tempSensor.getValue();
* τιμή θερμοκρασίας σε Celsius

double celsiusTemp = ourTempSensor.getCelsius(); // τιμή θερμοκρασίας σε Fahrenheit

double fahrenheitTemp= ourTempSensor.getFahrenheit();

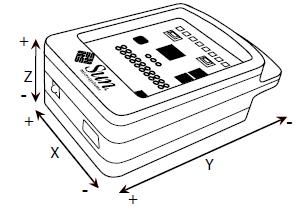
Για πηγές φωτισμού, όπως οι λάμπες φθορισμού, οι οποίες φαίνονται στο ανθρώπινο μάτι σταθερές αλλά στην πραγματικοτήτα η έντασή τους αλλάζει συνέχεια, χρησιμοποιείται η παρακάτω εντολή,

int getAverageValue(int n);

όπου λαμβάνονται n μετρήσεις, καθεμία κάθε 1 msec και υπολογίζεται η μέση ένταση φωτισμού τους.

Ο αισθητήρας θερμοκρασίας των Sun SPOTs βρίσκεται αναπόφευκτα κοντά σε πηγές θερμότητας πάνω στην πλακέτα των Sun SPOTs και ιδεατά πιο έγκυρες μετρήσεις θα προέκυπταν από τη χρήση ενός εξωτερικού αισθητήρα θερμοκρασίας που θα συνδεόταν στις θύρες εισόδου – εξόδου της πλακέτας των Sun SPOTs. [34, 36]

**3.9.3 Επιταχύμετρο – αισθητήρας κίνησης**

****

**Εικόνα 46: Διαστάσεις x,y,z στο Sun SPOT**

Το επιταχύμετρο μετράει την επιτάχυνση της βαρύτητας, ανιχνεύει δηλαδή την κίνηση σε τρεις διαστάσεις. Το μοντέλο του επιταχύμετρου είναι LIS3L02AQ και αποτελείται από έναν μικροαισθητήρα MEMS, (Micro-Electro-Mechanical System), ο οποίος μετατοπίζεται από την ονομαστική του θέση όταν εφαρμόζεται μία γραμμική επιτάχυνση, προκαλώντας έτσι μία ηλεκτρική δυσαναλογία, η οποία διαβάζεται τελικά από τον αναλογικό-σε-ψηφιακό μετατροπέα της κεντρικής πλακέτας των Sun SPOTs. Διαθέτει δύο τρόπους λειτουργίας που καθορίζουν το εύρος των τιμών που θα μετρήσουν, από -2g έως 2g και από -6g έως 6g. Η ακατέργαστη τιμή που λαμβάνει το επιταχύμετρο είναι ένας αριθμός που κυμαίνεται από το 0 μέχρι το 1023 και μετατρέπεται στη συνέχεια σε g δυνάμεις. Η μηδενική βαρύτητα βρίσκεται περίπου στο 466,5. [34, 36]

69

Το επιταχύμετρο μετράει την επιτάχυνση σε τρεις διαστάσεις. Η ανωτέρω εικόνα 46 παρουσιάζει πως αντικατοπτρίζονται οι διαστάσεις x, y, z στη συσκευή του Sun SPOT.

Για να δημιουργηθεί ένα αντικείμενο του επιταχύμετρου χρησιμοποιείται ο παρακάτω κώδικας.

import com.sun.spot.sensorboard.EDemoBoard; import com.sun.spot.sensorboard.IAccelerometer3d;

IAccelerometer3D ourAccel = EdemoBoard.getInstance().getAccelerometer();

Για να λάβει δεδομένα το επιταχύμετρο, θα πρέπει να οριστεί το εύρος τιμών στο οποίο θα μετρήσει. Το 0 θέτει το εύρος τιμών μεταξύ -2g και 2g, και το 1 θέτει το εύρος τιμών μεταξύ -6g και 6g. Στη συνέχεια το επιταχύμετρο θα πρέπει να μείνει ανενεργό για 100 milliseconds ώστε οι μετρήσεις του να είναι έγκυρες. Ακολουθεί ο σχετικός κώδικας.

// 0 είναι 2g, 1 είναι 6g

ourAccel.setRange(0); //χρησιμοποίησε 2g εύρος τιμών

* παύση για 100 milliseconds ώστε οι μετρήσεις να είναι έγκυρες try{

Thread.sleep(100);

} catch (InterruptedException ex) {

* αγνόησε εξαιρέσεις

}

Για να πάρουμε τιμές από το επιταχύμετρο χρησιμοποιείται ο παρακάτω κώδικας. int x-accel = ourAccel.getXaxis().getValue();

int y-accel = ourAccel.getYaxis().getValue(); int z-accel = ourAccel.getZaxis().getValue();

**3.10 Αναπτυξιακές πλακέτες**

Για τα Sun SPOTs έχουν κατασκευαστεί διάφορες αναπτυξιακές πλακέτες προκειμένου τα Sun SPOTs να μπορέσουν να επεκταθούν και σε άλλους τομείς της τεχνολογίας. Ακολουθεί μια λίστα και σύντομη περιγραφή αυτών των αναπτυξιακών πλακετών. [44]

* **spot-espot**
  + πλακέτα αυτή είναι ο βασικός επεξεργαστής των Sun SPOTs. Έχει περιγραφεί αναλυτικά στο κεφάλαιο 1.1. Σε αυτόν τον επεξεργαστή προστίθενται όλες οι πλακέτες που περιγράφονται παρακάτω. Έχει διαστάσεις 60mm x 38mm
* **spot-edemo**
  + πλακέτα αυτή είναι η πλακέτα που βρίσκεται ενσωματωμένη επί του παρόντος στα Sun SPOTs. Έχει περιγραφεί αναλυτικά στο κεφάλαιο 1.1.
* **s pot- eusbhost**
  + πρόσθετη αυτή πλακέτα παρέχει θύρα USB στα Sun SPOTs. Έχει διαστάσεις 1,9''x1,5'' και εφαρμόζει μέσα στο πλαστικό περίβλημα των Sun SPOTs. Περιέχει flash μνήμη, αποκωδικοποιητή για την διεύθυνση SPI, (Serial Peripheral Interface), προαιρετικά μετασχηματιστή από 3V σε 5V DC/DC και θύρα USB



70

τύπου A.

* **s pot-breadboard**

H breadboard επιτρέπει στους προγραμματιστές να δημιουργήσουν I/O κυκλώματα και να τα συνδέσουν στα Sun SPOTs. Συνδέεται με τον πίνακα αισθητήρων των Sun SPOTs μέσω ενός ribbon καλωδίου 20 pins. Το κέντρο της breadboard είναι ένας πίνακας 34 x 5 επαφών. Τα σήματα των Sun SPOTs καθοδηγούνται μέσω της σύνδεσης του ribbon καλωδίου με τις θηλυκές επαφές που βρίσκονται περιμετρικά του κεντρικού πίνακα της breadboard.

* **spot-breakout**
  + πρόσθετη αυτή πλακέτα περιλαμβάνει τερματικά καλωδίων και διεπαφές διαφόρων τύπων για να συνδεθούν μέσω αυτής στα Sun SPOTs servos, αισθητήρες και διαφόρων τύπων καλώδια. Αποφεύγεται έτσι η άμεση σύνδεση των παραπάνω με τα Sun SPOTs και επομένως και ο κίνδυνος καταστροφής των Sun SPOTs. Η πλακέτα έχει διαστάσεις 2,5''x2,5'' και συνδέεται με τον πίνακα αισθητήρων των Sun SPOTs μέσω ενός ribbon καλωδίου 20 pins.
* **spot-edaq**
  + πρόσθετη αυτή πλακέτα επιτρέπει στους χρήστες να αναπτύξουν υψηλής απόδοσης αναλογικές εφαρμογές στα Sun SPOTs. Η edaq πλακέτα θα περιέχει μία υψηλού επιπέδου Java Διεπαφή Προγραμματισμού Εφαρμογών, (API – Application Programming Interface), προεγκατεστημένο λογισμικό και αρκετά demo προγράμματα. Έχει διαστάσεις 1,9''x1,5'' και εφαρμόζει μέσα στο πλαστικό περίβλημα των Sun SPOTs. Η πλακέτα βρίσκεται ακόμα υπό ανάπτυξη, (beta).
* **spot- eflash**
  + πρόσθετη αυτή πλακέτα χρησιμοποιείται για διάβασμα, εγγραφή και αποθήκευση δεδομένων. Χρησιμοποιεί δύο mini SD flash cards, έχει διαστάσεις 1,9''x1,5'' και εφαρμόζει μέσα στο πλαστικό περίβλημα των Sun SPOTs. Ο ένας από τους δύο mini SD υποδοχείς χρησιμοποιεί την υψηλής ταχύτητας διεπαφή SD/MMC 4-bit της ARM9 ενώ ο άλλος υποδοχέας χρησιμοποιεί το πρωτόκολλο SPI.
* **spot-eproto**
  + πρόσθετη αυτή πλακέτα επιτρέπει στους χρήστες να συνδέσουν διεπαφές υλικού στα Sun SPOTs. Έχει διαστάσεις 1,9''x1,5'' και εφαρμόζει μέσα στο πλαστικό περίβλημα των Sun SPOTs. Διαθέτει flash μνήμη, αποκωδικοποιητή για την διεύθυνση SPI μετασχηματιστή από 3V σε 5V DC/DC και αρκετές θύρες διαστάσεων 0,1'' για τις συνδέσεις με τις διεπαφές υλικού. Οι θύρες που βρίσκονται πιο χαμηλά στην πλακέτα συνδέονται μέσω καλωδίου με τον εσωτερικό πίνακα και με το SPI chip.
* **spot-eprotomega**
  + πρόσθετη αυτή πλακέτα έχει ακριβώς τα ίδια χαρακτηριστικά με την πλακέτα eproto με την προσθήκη ενός επεξεργαστή Atmega 88. Έχει διαστάσεις 1,9''x1,5'' και εφαρμόζει μέσα στο πλαστικό περίβλημα των Sun SPOTs.
* **spot-eserial**
  + πρόσθετη αυτή πλακέτα χρησιμοποιείται για την RS232 σύνδεση των Sun SPOTs. Απαιτεί ένα επίπεδο ribbon καλώδιο για σύνδεση με υποδοχές DB9 ή DB25. Η RS232 διεπαφή συνδέεται με το Sun SPOT μέσω της θύρας UART0 του ARM9 επεξεργαστή. Έχει διαστάσεις 1,9''x1,5'' και εφαρμόζει μέσα στο πλαστικό



71

περίβλημα των Sun SPOTs.

* + **spot-ethumpre**
    - πρόσθετη αυτή πλακέτα περιέχει αισθητήρες θερμοκρασίας, υγρασίας και πίεσης. Έχει διαστάσεις 1,9''x1,5'' και εφαρμόζει μέσα στο πλαστικό περίβλημα των Sun SPOTs. Η πλακέτα περιέχει επεξεργαστή Atmel Atmega88, flash μνήμη και αποκωδικοποιητή.
  + **spot-egpsybt**
    - πρόσθετη αυτή πλακέτα περιέχει gps και bluetooth και βρίσκεται σε στάδιο ανάπτυξης.
  + **spot-etherspot**
    - πρόσθετη αυτή πλακέτα συνδέεται σε ethernet καλώδιο και προσθέτει δίκτυο 10/100 ethernet στο Sun SPOT. Χρησιμοποιεί πρωτόκολλο Power over Ethernet. Βρίσκεται σε στάδιο ανάπτυξης.

1. **Ασφάλεια**

Η ασφάλεια στις ασύρματες επικοινωνίες είναι ιδιαίτερα σημαντικό ζήτημα. Η περιορισμένη μνήμη, οι περιορισμένες υπολογιστικές δυνατότητες, η περιορισμένη ενέργεια, η δυνατότητα λειτουργίας σε διαφορετικά και πιθανώς “εχθρικά” υπολογιστικά περιβάλλοντα, η ανάγκη δικτύωσης των μικροσυσκεύων καθιστούν αναγκαία την ύπαρξη ισχυρής και ασφαλούς δικτύωσης με το λιγότερο δυνατό κόστος.

Στα Sun SPOTs εφαρμόζεται η τεχνολογία κρυπτογράφησης δημοσίου κλειδιού.

Στην κρυπτογράφηση δημοσίου κλειδιού χρησιμοποιούνται αλγόριθμοι ασύμμετρου κλειδιού, όπου σε αντίθεση με τους αλγόριθμους συμμετρικού κλειδιού δεν απαιτείται η αρχική ασφαλής ανταλλαγή των κλειδιών μεταξύ αποστολέα και παραλήπτη του μηνύματος αλλά χρησιμοποιείται διαφορετικό κλειδί για την κρυπτογράφηση του μηνύματος και διαφορετικό κλειδί για την αποκρυπτογράφηση του μηνύματος. Πιο συγκεκριμένα, χρησιμοποιούνται δύο κλειδιά, ένα μυστικό ιδιωτικό κλειδί και ένα δημόσιο κλειδί. Η χρήση αυτών των κλειδιών επιτρέπει την προστασία της αυθεντικότητας του μηνύματος καθώς δημιουργείται μια ψηφιακή υπογραφή για το μήνυμα που αποστέλλεται με τη χρήση του ιδιωτικού κλειδιού και στη συνέχεια το γνήσιο αυτής της υπογραφής αναγνωρίζεται από το δημόσιο κλειδί. Το δημόσιο κλειδί χρησιμποιείται δηλαδή για την κρυπτογράφηση του μηνύματος και το ιδιωτικό κλειδί για την αποκρυπτογράφηση.

Υποστηρίζεται γενικά η άποψη ότι οι συσκευές αισθητήρων δεν έχουν τη δυνατότητα να ικανοποιήσουν τις απαιτήσεις ενός συστήματος κρυπτογράφησης δημοσίου κλειδιού. Ωστόσο η χρήση υψηλά βελτιωμένων εφαρμογών του αλγόριθμου ασύμμετρου κλειδιού RSA έχει επιφέρει αρκετά θετικά αποτελέσματα. Έχει αναπτυχθεί μια παραλλαγή του παραπάνω αλγόριθμου, το “Elliptic Curve Cryptography” σύστημα, το οποίο χρησιμοποιείται στα Sun SPOTs. Το “Elliptic Curve Cryptography” σύστημα βελτιώνει σημαντικά την απόδοση στις 8- bit Κεντρικές Μονάδες Επεξεργασίας. Απαιτεί μικρή μνήμη και παρέχει μία ασφαλή διαδικτυακή server stack που συμπεριλαμβάνει τα πρωτόκολλα HTTP και SSL, τη λεγόμενη Sizzle, η οποία ενσωματώνεται σε μικροσυσκευές και επιτρέπει τον έλεγχο των μικροσυσκευών μέσω ενός φυλλομετρητή. [48]

72

**3.11.1 Περιγραφή συστήματος ασφαλείας**

Τα δεδομένα που στέλνει ένα Sun SPOT σε ένα άλλο ή σε έναν basestation είναι εκ κατασκευής κρυπτογραφημένα. Αυτό συμβαίνει ώστε ο κώδικας που εκτελείται στο Sun SPOT να περιέχει έγκυρα bytecodes και για να αποφευχθεί η φόρτωση επικίνδυνου για τα Sun SPOTs λογισμικού από εξωτερικούς χρήστες.

Εκ κατασκευής κάθε Sun SPOT δε σχετίζεται με κανένα χρήστη και δεν περιέχει κανένα δημόσιο κλειδί. Όταν ένας χρήστης χρησιμοποιήσει για πρώτη φορά ένα Sun SPOT, δηλαδή φορτώσει μία εφαρμογή σε ένα Sun SPOT, είτε μέσω usb είτε over-the-air, δημιουργείται αυτόματα ένα ζευγάρι δημοσίου – ιδιωτικού κλειδιού και σχετίζεται με το συγκεκριμένο χρήστη. To δημόσιο κλειδί μεταφέρεται στη συσκευή και μόνο ο ιδιοκτήτης μπορεί να φορτώσει εφαρμογές στη συγκεκριμένη συσκευή. Αυτό πραγματοποιείται με την ψηφιακή υπογραφή που αποκτά το Sun SPOT μέσω του ιδιωτικού κλειδιού του χρήστη και στη συνέχεια με την αναγνώριση αυτής της υπογραφής μέσω του δημοσίου κλειδιού που υπάρχει αποθηκευμένο στο Sun SPOT.

Άπαξ και οριστεί το δημόσιο κλειδί, μόνο ο ιδιοκτήτης του μπορεί να το αλλάξει, αν και στην πραγματικοτήτα οποιοσδήποτε έχει φυσική πρόσβαση στο Sun SPOT μπορεί να αλλάξει το δημόσιο κλειδί του. Εάν ο χρήστης Β επιθυμεί να χρησιμοποιήσει το Sun SPOT που προηγουμένως ανήκε στο χρήστη Α, μπορεί να το κάνει με δύο τρόπους.

* Εάν ο χρήστης Β δεν έχει φυσική πρόσβαση στη συσκευή, ο χρήστης Α μπορεί να χρησιμοποιήσει την εντολή

ant deletepublickey

για να διαγράψει το υπάρχον δημόσιο κλειδί από τη συσκευή. Ο χρήστης Α μπορεί επίσης να πραγματοποιήσει την ίδια διαδικασία απομακρυσμένα με την εντολή

ant deletepublickey -DremoteId=0014.4F01.xxxx.xxxx,

(όπου 0014.4F01.xxxx.xxxx η διεύθυνση MAC της συσκευής)

Ο χρήστης Β μπορεί να φορτώσει τότε μία εφαρμογή στη συσκευή με την εντολή ant deploy -DremoteId=0014.4F01. xxxx.xxxx

ή στην περίπτωση που ο χρήστης Β χρησιμοποιεί διαφορετικές βιβλιοθήκες από το χρήστη Α, μπορεί να χρησιμοποιήσει την εντολή

ant flashlibrary -DremoteId=0014.4F01.xxxx.xxxx

Ο χρήστης Β θα είναι αυτόματα ο καινούριος ιδιοκτήτης της συσκευής. Κατά την περίοδο που η συσκευή δεν έχει ιδιοκτήτη, (αφού ο χρήστης Α έχει εκτελέσει deletepublickey και προτού ο χρήστης Β εκτελέσει deploy ή flashlibrary), το Sun SPOT είναι εκτεθειμένο σε εχθρικές επιθέσεις, (ένας τρίτος χρήστης θα μπορούσε να γίνει ο ιδιοκτήτης του Sun SPOT πριν προλάβει να το πραγματοποιήσει αυτό ο χρήστης Β). Για το λόγο αυτό συνιστάται η αλλαγή των δημοσίων κλειδιών μέσω usb.

* Εάν ο χρήστης Β έχει φυσική πρόσβαση στη συσκευή, μπορεί να συνδέσει τη συσκευή μέσω usb και να εκτελέσει

ant deploy ή ant flashlibrary

Εάν μια εφαρμογή πρέπει να εφαρμοστεί από περισσότερα από ένα Sun SPOTs, θα πρέπει όλοι οι χρήστες και όλα τα Sun SPOTs να μοιράζονται το ίδιο ζευγάρι δημοσίου – ιδιωτικού κλειδιού. Θα πρέπει να αντιγραφεί επομένως το ζευγάρι αυτό από έναν χρήστη που θα έχει οριστεί ως κύριος χρήστης σε όλους τους άλλους χρήστες. Αυτό γίνεται αντιγράφοντας το αρχείο sdk.key από τον υποκατάλογο sunspotkeystore του

73

home directory του κύριου χρήστη σε κάθε υποκατάλογο sunspotkeystore κάθε home directory όλων των άλλων χρηστών. Στη συνέχεια πρέπει να φορτωθεί το δημόσιο κλειδί σε κάθε Sun SPOT. Αυτό μπορεί να γίνει φορτώνοντας την εφαρμογή σε κάθε Sun SPOT μέσω usb με την εντολή ant deploy.

Με τη διαγραφή του sdk.key αρχείου διαγράφεται και το δημόσιο κλειδί ενός Sun SPOT. Στη συνέχεια όταν θα φορτωθεί κάποια εφαρμογή στο Sun SPOT θα δημιουργηθεί αυτόματα ένα καινούριο δημόσιο κλειδί.

Εφαρμογές και μορφοποιημένες βιβλιοθήκες ελέγχονται πάντα για τη γνησιότητά τους και εάν δεν μπορεί να αναγνωριστεί η ψηφιακή υπογραφή με τη χρήση του δημοσίου κλειδιού, η εφαρμογή δεν μπορεί να εκτελεστεί. Για να ελεγχθεί η γνησιότητα μιας μορφοποιημένης βιβλιοθήκης θα πρέπει να εκτελεστεί η εντολή ant flashlibrary ώστε να ανανεωθεί η ψηφιακή υπογραφή της βιβλιοθήκης. Στην περίπτωση της φόρτωσης εφαρμογών μέσω over-the-air όλες οι ανανεώσεις στην configuration page πραγματοποιούνται πριν ανανεωθεί η σελίδα, οπότε αποφεύγονται έτσι πιθανές επιθέσεις.

Το σύστημα ασφαλείας στα Sun SPOTs έχει τα εξής μειονεκτήματα:

* + Δεν παρέχεται καμία ασφάλεια στα Sun SPOTs όταν ένας κακόβουλος χρήστης έχει άμεση πρόσβαση στα Sun SPOTs.
  + Το ζευγάρι δημοσίου – ιδιωτικού κλειδιού είναι αποθηκευμένο σε ένα .txt αρχείο στο σύστημα αρχείων του ιδιοκτήτη του Sun SPOT και επομένως οποιοσδήποτε έχει πρόσβαση στο σύστημα αρχείων του ιδιοκτήτη έχει πρόσβαση και στο ζευγάρι δημοσίου – ιδιωτικού κλειδιού του Sun SPOT. [33]

1. **Αναπτυξιακά προγράμματα**

Τα παρακάτω προγράμματα βρίσκονται υπό ανάπτυξη και αποτελούν λογισμικό ανοιχτού κώδικα.

* **Remote Control Camera** – Δημιουργία διαδικτυακής κάμερας της οποίας ηγωνία θέασης θα ελέγχεται από Sun SPOTs. Ο basestation θα λειτουργεί ως εξυπηρετητής για τον διαδικτυακό έλεγχο. Για τον έλεγχο της κάμερας χρησιμοποιούνται δύο σερβοκινητήρες τύπου HS-311 και HS-81MG αντίστοιχα. [47]
* **Balancing Robot** – Δημιουργία τηλεκατευθυνόμενου robot με χρήση Sun SPOTsκαι μικρο-σερβοκινητήρων. Χρησιμοποιήθηκε το επιταχύμετρο των Sun SPOTs. Όταν μετακινείται το Sun SPOT κατά μία γωνία στέλνεται μήνυμα στους σερβοκινητήρες να περιστρέψουν τους έλικές τους κατά αυτή τη γωνία. [46]
* **Airstore** – Δημιουργία ενός κοινού χώρου αποθήκευσης, μίας κοινής βάσηςδεδομένων στην οποία θα μπορούν να αποθηκεύονται τα δεδομένα πολλών Sun SPOTs για διαφορετικές εφαρμογές. [45]
* **Autospots** – Παροχή απομακρυσμένης και αυτόνομης καθοδήγησης σε RC,(Remote Control), αυτοκίνητα και φορτηγά. Η απομακρυσμένη καθοδήγηση εφαρμόζεται με τη χρήση ενός Sun SPOT που ελέγχει το αυτοκίνητο ή φορτηγό και που ακολουθεί τις οδηγίες που του παρέχονται ασύρματα από ένα άλλο Sun SPOT. Η αυτόνομη καθοδήγηση παρέχεται με τη χρήση ενός Sun SPOT που ανιχνεύει προηγουμένως καταγεγραμμένες διαδρομές από αισθητήρες. [45]
* **SDK – Demos** – Demo εφαρμογές που παρέχονται με το sdk, (Sun SPOTDevelopment Kit), για χρήση και περαιτέρω ανάπτυξη από τους χρήστες. [45]