

Ατομική Διπλωματική Εργασία

**ΠΛΑΤΦΟΡΜΑ ΕΝΔΙΑΜΕΣΟΥ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΣΥΛΛΟΓΗ
ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΧΑΜΗΛΟΥ ΚΟΣΤΟΥΣ ΣΥΣΚΕΥΕΣ**

Χρυσοβαλάντης Περίανδρου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2014

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Πλατφόρμα ενδιάμεσου λογισμικού για συλλογή δεδομένων από χαμηλού κόστους
συσκευές**

Χρυσοβαλάντης Περιάνδρου

Επιβλέπων Καθηγητής
Γιώργος Πάλλης

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

Μάιος 2014

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή μου Δρ. Γιώργο Πάλλη, για την βοήθειά του, την υποστήριξη του και για την πολύ καλή συνεργασία που είχαμε καθ' όλο το χρονικό διάστημα εκπόνησης της διπλωματικής αυτής εργασίας. Οι συμβουλές του, οι οδηγίες του αλλά και κατευθύνσεις που με συμβούλευε να ακολουθήσω, συνέβαλαν στο μέγιστο, για την επιτυχή κατά την άποψη μου ολοκλήρωση της εργασίας αυτής.

Θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ στον Χαρίτων Ευσταθιάδη για την πολύτιμη υποστήριξη που μου πρόσφερε, καθώς επίσης και για τις πολύτιμες συμβουλές που μου έδινε όλο αυτό το διάστημα.

Τέλος, θα ήθελα να εκφράσω ιδιαίτερες ευχαριστίες στην οικογένεια μου που με στήριξαν ηθικά, ψυχολογικά και οικονομικά σε ολόκληρη την πορεία μου κατά τη διάρκεια φοίτησης μου στο Πανεπιστήμιο Κύπρου, αλλά και κατά τη δύσκολη αυτή περίοδο ολοκλήρωσης της διπλωματικής μου εργασίας.

Περίληψη

Η ραγδαία ανάπτυξη της τεχνολογίας και η «εισβολή» των κινητών συσκευών smart phone στη ζωή μας, καθώς επίσης και η ανάπτυξη και εξέλιξη της ιδέας του «Internet of Things», έχουν δημιουργήσει νέες ανάγκες, νέες απαιτήσεις αλλά και νέες προοπτικές στην Επιστήμη της Πληροφορικής. Απαιτήσεις και προοπτικές που έχουν αλλάξει τον τρόπο σκέψης και υλοποίησης των πληροφορικών συστημάτων και των εφαρμογών. Η καθημερινότητα μας έχει κατακλυστεί από μικρές συσκευές, με πολύ μεγάλη υπολογιστική ισχύ, μεγάλο αποθηκευτικό χώρο και πολλούς ενσωματωμένους αισθητήρες. Οι μικροσυσκευές βρίσκονται παντού, σε αντικείμενα καθημερινής χρήσης, σε αυτοκίνητα, ακόμα και σε ρούχα. Στα πλαίσια αυτά, έχουν δημιουργηθεί και πολύ μικροί αισθητήρες για την κάλυψη των αναγκών.

Στόχος της παρούσας διπλωματικής εργασίας είναι σε πρώτη φάση η βελτίωση της πλατφόρμας ενδιάμεσου λογισμικού FeelTheWorld[33], η οποία είναι υλοποιημένη σε πλατφόρμα Android SDK, που διαχειρίζεται αισθητήρες κινητών τηλεφώνων, όπως επίσης και εξωτερικούς αισθητήρες. Σε δεύτερη φάση, είναι η τροποποίηση της πλατφόρμας σε λειτουργικό σύστημα Linux, η οποία θα επιτρέπει στους προγραμματιστές να αναπτύξουν εύκολα, γρήγορα και αποδοτικά εφαρμογές που μπορούν να διαχειριστούν και αισθητήρες. Απώτερος σκοπός είναι οι εφαρμογές να μπορούν να εγκατασταθούν σε χαμηλού κόστους συσκευές που τρέχουν Linux όπως και το Raspberry PI.

Μετά την υλοποίηση της πλατφόρμας, έγινε η ανάπτυξη δύο εφαρμογών με σκοπό την αξιολόγηση της. Η πρώτη εφαρμογή παρακολουθεί τα δεδομένα του περιβάλλοντος και τα παρουσιάζει με διαγράμματα σε ιστοσελίδα, και η δεύτερη αποτελεί την ανάπτυξη μοντέλου περιβάλλοντος έξυπνου σπιτιού.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Ορισμός Προβλήματος	2
	1.2 Υποκίνηση Εργασίας	3
	1.3 Στόχοι Εργασίας	3
	1.4 Περίγραμμα Αναφοράς	5
Κεφάλαιο 2	Υπόβαθρο.....	6
	2.1 Αστική Αντίληψη	7
	2.1.1 Συστήματα Αστικής Αντίληψης	7
	2.1.2 Συμμετέχουσα Αντίληψη	7
	2.1.3 Καιροσκοπική Αντίληψη	8
	2.2 Raspberry PI	8
	2.3 Διαδίκτυο των Πραγμάτων	10
	2.3.1 Διαδίκτυο των Πραγμάτων	10
	2.3.2 Έξυπνο Σπίτι	11
	2.4 Πλατφόρμα ενδιαμέσου λογισμικού	12
	2.5 Έξυπνα τηλέφωνα	12
	2.6 Πλατφόρμα ενδιαμέσου λογισμικού Feel The World	13
Κεφάλαιο 3	Περιγραφή – Ανάλυση Συστήματος.....	14
	3.1 Γενική Περιγραφή	15
	3.2 Προδιαγραφές Συστήματος	16
	3.3 Εργαλεία και βιβλιοθήκες	16
	3.3.1 Eclipse	16

3.3.2 Sensaris Senspods	17
3.3.3 Linux	18
3.4 Ανάλυση Αρχιτεκτονικής	18
3.4.1 Αρχιτεκτονική Πλατφόρμας Ενδιάμεσου Λογισμικού	18
3.4.2 Ανάλυση Αρχιτεκτονικής	22
Κεφάλαιο 4 Υλοποίηση Συστήματος.....	26
4.1 Διάγραμμα κλάσεων	27
4.2 Λεπτομέρειες Υλοποίησης	33
Κεφάλαιο 5 Δοκιμή – Αξιολόγηση.....	39
5.1 Εφαρμογές που αναπτύχθηκαν	40
5.1.1 Sense Me	40
5.1.2 Smart Home	46
5.2 Μεθοδολογία Αξιολόγησης	52
5.3 Μετρικές Αξιολόγησης	52
Κεφάλαιο 6 Συμπεράσματα – Μελλοντική Εργασία.....	57
6.1 Συμπεράσματα	58
6.2 Μελλοντική εργασία	59
Βιβλιογραφία.....	61
Παράρτημα Α.....	A-1

Κεφάλαιο 1

Εισαγωγή

1.1	Ορισμός Προβλήματος	2
1.2	Υποκίνηση Εργασίας	3
1.3	Στόχοι Εργασίας	3
1.4	Περίγραμμα Εργασίας	5

Στο κεφάλαιο αυτό γίνεται μια εισαγωγή στο πεδίο με το οποίο ασχοληθήκαμε στην παρούσα διπλωματική εργασία. Αρχικά γίνεται μια περιγραφή της παρούσας κατάστασης όπως αυτή έχει διαμορφωθεί στις μέρες μας, λόγο της ραγδαίας ανάπτυξης της τεχνολογίας. Ακολούθως παρουσιάζονται τα κίνητρα της διπλωματικής εργασίας, οι στόχοι που καθορίστηκαν και η συνεισφορά της. Τέλος, γίνεται αναφορά στο περίγραμμα της ατομικής διπλωματικής εργασίας.

1.1 Ορισμός Προβλήματος

Η ραγδαία ανάπτυξη της τεχνολογίας, η μείωση του μεγέθους των επεξεργαστών και της μνήμης, η εξέλιξη της τεχνολογίας των αισθητήρων, καθώς επίσης και η ανάγκη για αυτοματοποίηση κάποιων καθημερινών λειτουργιών ενός ηλεκτρονικού υπολογιστή, είχαν σαν αποτέλεσμα την αλλαγή στον τρόπο χρήσης του Διαδικτύου. Πλέον, το Διαδίκτυο δεν χρησιμοποιείται μόνο για πρόσβαση σε ιστοσελίδες ή σε απομακρυσμένα δεδομένα. Με όλες αυτές τις μικροσυσκευές να διαβάζουν δεδομένα και να τα ανεβάζουν ή να ζητούν δεδομένα, το Διαδίκτυο αναγκάστηκε να αλλάξει τρόπο λειτουργίας. Έτσι ενσωματώθηκε ο όρος «Διαδίκτυο των πραγμάτων»[2]. Η ανάπτυξη αυτή έχει επιφέρει τρομερές αλλαγές στα μεγέθη των ηλεκτρονικών συσκευών, όπως επίσης και στην λειτουργία τους. Πανίσχυρες συσκευές, για κάθε λειτουργία, μικρότερες από την παλάμη ενός χεριού άρχισαν να κατακλύζουν την αγορά στα μέσα της προηγούμενης δεκαετίας. Τρανό παράδειγμα είναι τα έξυπνα τηλέφωνα[1], τα οποία σταδιακά έχουν κερδίσει το μεγαλύτερο μερίδιο της αγοράς των κινητών τηλεφώνων.

Με την εξέλιξη της τεχνολογίας και την ευρεία χρήση των έξυπνων τηλεφώνων, έχει δημιουργηθεί η ανάγκη για εφαρμογές που να ακολουθούν τις νέες τάσεις της τεχνολογίας. Πάρα πολλές έρευνες στρέφονται στην δημιουργία κάθε είδους εφαρμογών. Εφαρμογές που να συμβαδίζουν με τις ανάγκες των ανθρώπων για άμεση και συνεχή ενημέρωση, για ανάγνωση και επεξεργασία δεδομένων από αισθητήρες αλλά και για αυτοματοποίηση λειτουργιών. Μεγαλύτερη όμως ανάγκη υπάρχει για πλατφόρμες ενδιάμεσου λογισμικού, δηλαδή βιβλιοθήκες με έτοιμο αλλά και αποδοτικό κώδικα, που να βοηθούν προγραμματιστές που αναπτύσσουν τέτοιες εφαρμογές, ούτως ώστε η ανάπτυξη να γίνεται πιο εύκολη, πιο γρήγορη και πιο αποδοτική. Πολύ σημαντική είναι η απουσία τέτοιων εργαλείων από το λειτουργικό σύστημα Linux, το οποίο χρησιμοποιείται και από συσκευές με χαμηλό κόστος αγοράς.

1.2 Υποκίνηση Εργασίας

Τα τελευταία χρόνια, έχει αυξηθεί η ανάπτυξη εφαρμογών, που ακολουθούν τα πρότυπα του Διαδικτύου των πραγμάτων, καθώς και την ανάγνωση δεδομένων από αισθητήρες. Η μεγάλη πολυπλοκότητα, αλλά και ο μεγάλος χρόνος που χρειάζεται για την ανάπτυξη τέτοιων εφαρμογών, λειτουργούσε σαν αρωγός σε αυτή την προσπάθεια. Έτσι η δημιουργία των πλατφορμών ενδιάμεσου λογισμικού γινόταν πιο επιτακτική. Με αφορμή πλατφόρμες που έχουν δημιουργήσει μεγάλες εταιρίες του εξωτερικού, αλλά και την πλατφόρμα Feel The World, που έχει δημιουργηθεί τα προηγούμενα χρόνια, στα πλαίσια διπλωματικών εργασιών στο Τμήμα Πληροφορικής, αποφασίσαμε όπως δημιουργήσουμε μια πλατφόρμα ενδιάμεσου λογισμικού σε συσκευές χαμηλού κόστους με λειτουργικό σύστημα Linux[5]. Η πλατφόρμα έχει σαν σκοπό την απλοποίηση εφαρμογών για ηλεκτρονικούς υπολογιστές και ειδικότερα σε χαμηλού κόστους συσκευές, όπως το Raspberry PI[6] κάνοντας πιο απλή και αποδοτική την προσπάθεια του προγραμματιστή. Η δημιουργία εφαρμογών που χειρίζονται αισθητήρες είναι σε πολύ αρχικά στάδια, όσο αφορά το λειτουργικό σύστημα Linux. Αυτό απεικονίζεται και από την μη ύπαρξη κάποιας πλατφόρμας ενδιάμεσου λογισμικού, καθιστώντας την δική μας πρωτοπόρο στο είδος της. Επίσης, σημαντικός παράγοντας υποκίνησης ήταν και η οικονομική ύφεση, αλλά και η αυξημένες τιμές των έξυπνων τηλεφώνων.

1.3 Στόχοι Εργασίας

Η διπλωματική εργασία έχει ως στόχους:

1. Την επέκταση της πλατφόρμας ενδιάμεσου λογισμικού FeelTheWorld και ενίσχυση της με την εισαγωγή πακέτων που διευκολύνουν την χρήση βάσης δεδομένων, καθώς και την εισαγωγή πρωτεριοτήτων.
2. Την δημιουργία μιας πλατφόρμας λογισμικού, η οποία θα διευκολύνει την ανάπτυξη εφαρμογών που ασχολούνται με την πρόσβαση, επεξεργασία, χρήση και παρουσίαση δεδομένων που αφορούν αισθητήρες.
3. Οι εφαρμογές που θα αναπτύσσονται θα μπορούν να εγκατασταθούν σε συσκευές χαμηλού κόστους με λειτουργικό σύστημα Linux, ενώ η πλατφόρμα θα διασφαλίζει ότι

δεν θα υπάρχουν προβλήματα λόγω της μειωμένης επεξεργαστικής ισχύς, αλλά και τους περιορισμένους πόρους τέτοιων συσκευών.

Η πλατφόρμα ενδιάμεσου λογισμικού απευθύνεται στους προγραμματιστές που ασχολούνται με την ανάπτυξη εφαρμογών που αφορούν αισθητήρες. Επίσης, πολύ σημαντικό είναι οι λειτουργίες της πλατφόρμας να παρέχουν και την επιλογή του πραγματικού χρόνου(real time), δηλαδή την άμεση ανάκτηση των δεδομένων από τους αισθητήρες, την επεξεργασία και την παρουσίαση ή την χρήση τους.

Τα δεδομένα που εισάγονται στο σύστημα από τους αισθητήρες, θα πρέπει να μπορούν να αποστέλλονται και να αποθηκεύονται σε ένα εξυπηρετητή, ο οποίος θα μπορεί επιπλέον να τα επεξεργάζεται. Αυτά τα δεδομένα θα πρέπει να είναι ορατά σε όλους τους χρήστες, οι οποίοι θα μπορούν να τα ανακτούν με διάφορους τρόπους.

Σε συνδυασμό με την πλατφόρμα FeelTheWorld, θα μπορούμε να χειριζόμαστε δεδομένα και από έξυπνα τηλέφωνα, συνδυάζοντας τα με τα δεδομένα από τις συσκευές χαμηλού κόστους.

Η πλατφόρμα πρέπει να παρέχει μια αφαιρετική αναπαράσταση των δεδομένων για να υπάρχει η δυνατότητα να χρησιμοποιηθεί από μεγάλο εύρος εφαρμογών. Επίσης, αφαιρετική πρέπει να είναι και η αναπαράσταση των αισθητήρων για να μπορεί να γίνει εύκολα η επέκταση της πλατφόρμας, με την εισαγωγή και άλλων αισθητήρων. Τέλος, η πλατφόρμα ενδιάμεσου λογισμικού πρέπει να είναι σχεδιασμένη με τέτοιο τρόπο ώστε να μπορεί εύκολα και ένα άτομο χωρίς ιδιαίτερη προγραμματιστική εμπειρία, να αναπτύξει κάποια εφαρμογή.

1.4 Περίγραμμα Εργασίας

Τελειώνοντας το πρώτο κεφάλαιο, στο οποίο παρουσιάστηκε ο σκοπός της διπλωματικής εργασίας, οι στόχοι της και αναφέρθηκαν κάποια γενικά χαρακτηριστικά και λειτουργίες που θα εμπεριέχονται στην πλατφόρμα, ακολουθεί μια σύντομη περιγραφή για το τι θα περιέχουν τα επόμενα κεφάλαια.

Στο δεύτερο κεφάλαιο, παρουσιάζεται το γνωστικό υπόβαθρο της διπλωματικής εργασίας. Ακολουθούν ορισμοί για την αστική αντίληψη, το Raspberry PI αλλά και για το Διαδίκτυο των πραγμάτων.

Στο τρίτο κεφάλαιο, γίνεται η περιγραφή και η ανάλυση του συστήματος. Αναλύονται οι απαιτήσεις, οι προδιαγραφές και η αρχιτεκτονική.

Στο τέταρτο κεφάλαιο παρουσιάζεται η υλοποίηση και τα διαγράμματα των κλάσεων, όπως επίσης και σημαντικές λεπτομέρειες στο κώδικα υλοποίησης.

Το πέμπτο κεφάλαιο αφορά την δοκιμή της πλατφόρμας με την δημιουργία δύο εφαρμογών. Ακολουθεί η αξιολόγηση της πλατφόρμας με μετρικές και διαγράμματα.

Τέλος, στο έκτο κεφάλαιο παραθέτονται τα συμπεράσματα αλλά και οι μελλοντικές εργασίες που μπορούν να γίνουν σε σχέση με αυτή την εργασία.

Κεφάλαιο 2

Υπόβαθρο

2.1 Αστική Αντίληψη	7
2.1.1 Συστήματα Αστικής Αντίληψης	7
2.1.2 Συμμετέχουσα Αντίληψη	7
2.1.3 Καιροσκοπική Αντίληψη	8
2.2 Raspberry PI	8
2.3 Διαδίκτυο των Πραγμάτων	10
2.3.1 Διαδίκτυο των Πραγμάτων	10
2.3.2 Έξυπνο Σπίτι	11
2.4 Πλατφόρμα ενδιάμεσου λογισμικού	12
2.5 Έξυπνα τηλέφωνα	12
2.6 Πλατφόρμα ενδιάμεσου λογισμικού Feel The World	13

Σε αυτό το κεφάλαιο γίνεται αναφορά στα συστήματα αστικής αντίληψης, συμμετέχουσας και καιροσκοπικής αντίληψης. Ακολουθώς, γίνεται περιγραφή του Raspberry PI. Στη συνέχεια, γίνεται αναφορά στο διαδίκτυο των πραγμάτων και στα έξυπνα σπίτια, ενώ τέλος γίνεται εκτενής παρουσίαση της πλατφόρμας ενδιάμεσου λογισμικού Feel The World.

2.1 Αστική Αντίληψη

2.1.1 Συστήματα Αστικής Αντίληψης

Τα συστήματα αστικής αντίληψης είναι ένας τομέας της πληροφορικής, ο οποίος έχει αναπτυχθεί ιδιαίτερα τα τελευταία χρόνια, και θα συνεχίσει να αναπτύσσεται, λόγω της συνεχούς ανάπτυξης των αισθητήρων, της συνεχιζόμενης ανάγκης για γνώση του περιβάλλοντος, όπως επίσης και της ανάγκης για έλεγχο, του τι υπάρχει γύρω μας. Πολλά ερευνητικά κέντρα και επιστήμονες ανά τον κόσμο έχουν ασχοληθεί με αυτό το θέμα, και πάρα πολλές μελέτες και εφαρμογές έχουν δημοσιευθεί. Η εξέλιξη των αισθητήρων και η μείωση του μεγέθους τους είχαν σαν αποτέλεσμα την αύξηση της χρήσης τους, την πρακτική χρήση τους σε εφαρμογές, όπως επίσης και την ενσωμάτωσή τους σε μικροσυσκευές, όπως τα έξυπνα τηλέφωνα κ.τ.λ. Επίσης, σημαντικός παράγοντας είναι και η βελτίωση της απόδοσης των ασύρματων συνδέσεων, όπως το Bluetooth[7] και το wifi[8], που βοήθησαν στην χρήση αισθητήρων που το μέγεθος τους δεν μπορούσε να μειωθεί.

Τα ζητήματα που απασχόλησαν μέχρι τώρα τις έρευνες, ήταν ζητήματα όπως το πως θα συλλέγονται οι τιμές, που θα αποθηκεύονται και πως θα χρησιμοποιούνται. Επίσης, σημαντικός τομέας της ανάπτυξης των συστημάτων αστικής αντίληψης, είναι η ανάπτυξη αλγορίθμων, που θα μπορούν όχι μόνο να εξάγουν έγκυρα αποτελέσματα, αλλά να μπορούν να κάνουν και προβλέψεις, και όλα αυτά σε πραγματικούς χρόνους.

Τον πιο σημαντικό ρόλο σε όλα τα συστήματα έχει ο άνθρωπος. Σε αυτό το σύστημα, η οπτική γωνία με την οποία η κάθε ερευνητική ομάδα βλέπει το σύστημα, δίνει και στον άνθρωπο τον ανάλογο ρόλο. Οι πιο σημαντικές προσεγγίσεις είναι δύο, η συμμετέχουσα και η καιροσκοπική αντίληψη.

2.1.2 Συμμετέχουσα Αντίληψη

Ο ρόλος του ανθρώπου είναι πολύ σημαντικός στην ανάπτυξη κάθε συστήματος. Ο τρόπος με τον οποίο ο άνθρωπος αντιμετωπίζεται από το σύστημα, δίνει και τον τελικό χαρακτήρα

του συστήματος. Στην περίπτωση των συστημάτων αστικής αντίληψης, μία προσέγγιση είναι η συμμετέχουσα αντίληψη. Η συμμετέχουσα αντίληψη[9] είναι όταν ο χρήστης, χωρίς προσωπικό ή οικονομικό συμφέρον, επιλέγει να ικανοποιήσει ένα αίτημα του συστήματος. Πολύ σημαντική σε τέτοια συστήματα είναι η διαφύλαξη της ιδιωτικότητας των χρηστών. Αυτό επιτυγχάνεται με την προστασία των δεδομένων καθώς και με την δυνατότητα στο χρήστη να επιλέγει τα δεδομένα τα οποία επιθυμεί να μοιράζεται. Η προσέγγιση η οποία ακολουθείτε σε τέτοια συστήματα είναι η παροχή εργαλείων στον χρήστη που να τον βοηθούν στην δημοσίευση, αναζήτηση, ερμηνευση και επαλήθευση των πληροφοριών που έχουν συλλεγεί.

2.1.3 Καιροσκοπική Αντίληψη

Υπάρχει όμως και μία άλλη αντίληψη, στην οποία ο χρήστης δεν ενημερώνεται. Σύμφωνα με την καιροσκοπική αντίληψη, ο χρήστης δεν γνωρίζει ποτέ ένα τέτοιο σύστημα είναι σε λειτουργία καθώς ούτε και ποια ιδιότητα του συστήματος τέθηκε σε ισχύ. Το σύστημα αξιολογεί και κρίνει την χρησιμότητα της κάθε λειτουργίας ξεχωριστά, και την ενεργοποιεί αυτόματα χωρίς ο χρήστης να ενημερωθεί. Συνήθως, η αξιολόγηση γίνεται με βάση στοιχεία από αισθητήρες, δηλαδή αν η τιμή κάποιου αισθητήρα, έχει συγκεκριμένη τιμή τότε το σύστημα ενεργοποιεί αυτόματα. Τέτοια συστήματα περιορίζονται με τους περιορισμούς ιδιωτικότητας και διαφάνειας που ορίζει ο χρήστης. Αν δεν υπάρχουν περιορισμοί τότε τίθενται θέματα παραβίασης της ιδιωτικότητας του χρήστη. Τέτοια παραβίαση παρουσιάζεται όταν το σύστημα γνωστοποιεί πληροφορίες του χρήστη, ενώ οι περιορισμοί του χρήστη το απαγορεύουν.

2.2 Raspberry PI

Το Raspberry PI είναι ένας υπολογιστής μεγέθους πιστωτικής κάρτας. Έχει κατασκευαστεί στο Ηνωμένο Βασίλειο από την Pi Foundation Raspberry, με σκοπό την προώθηση του ηλεκτρονικού υπολογιστή στην διδασκαλία των σχολείων. Αντίθετα με το σκοπό παραγωγής του, λόγω του μικρού μεγέθους του, αλλά και για το χαμηλό κόστος του, το Raspberry Pi έχει εκτενέστερη χρήση.

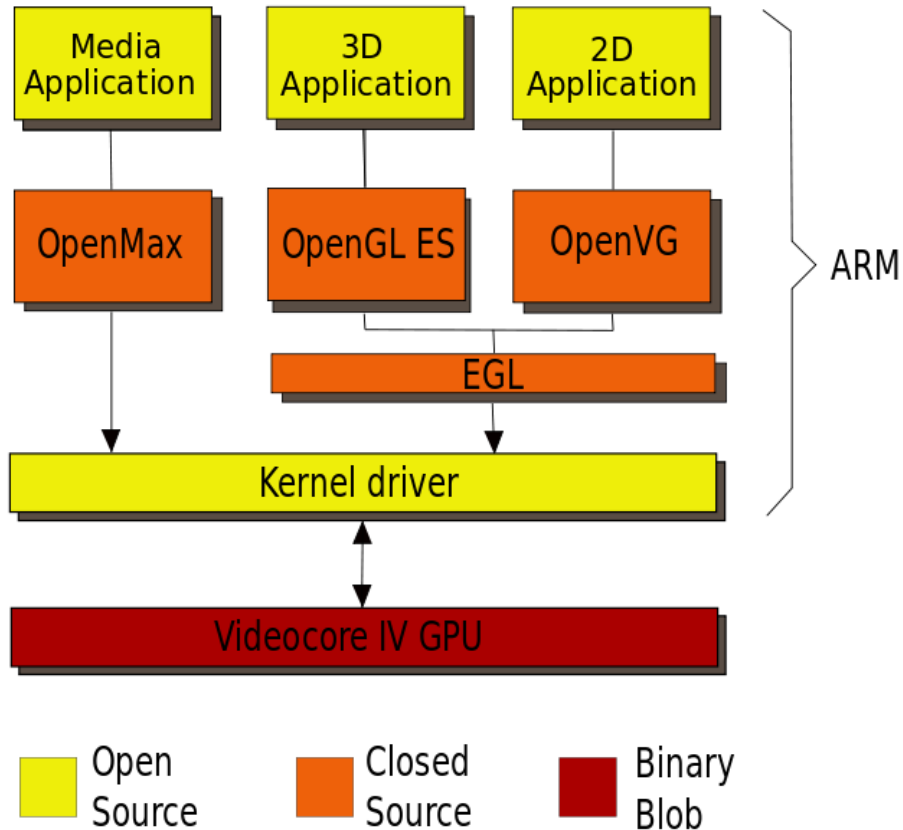
Σύστημα: BCM2835 Broadcom σε ένα chip.

Επεξεργαστή: ARM, 700 MHz , VideoCore IV GPU

RAM: 256 MB - 512 MB.

Αποθήκευση: Κάρτα SD (δεν περιλαμβάνει ενσωματωμένο σκληρό δίσκο).

Λειτουργικό σύστημα: Linux. Έχει μία θύρα USB και μία HDMI.



Εικόνα 2.1 Αρχιτεκτονική Raspberry Pi.

Το hardware του GPU φορτώνεται από την κάρτα SD στο GPU κατά την εκκίνηση του Raspberry Pi, μέσω μιας εικόνας firmware. Η εικόνα firmware είναι μια δυαδική κηλίδα (binary blob). Οι οδηγοί (ARM drivers), που είναι αρχικά κλειστού κώδικα (closed source), απελευθερώνονται στην συνέχεια, αλλά η περισσότερη δουλειά γίνεται από τον κλειστό κώδικα. Η εφαρμογή λογισμικού (application software) χρησιμοποιεί βιβλιοθήκες κλειστού κώδικα όπως είναι η OpenMax, η OpenGL ES και η OpenVG. Επίσης, η εφαρμογή

λογισμικού καλεί το Kernel driver που με τη σειρά του επικοινωνεί με την Videocore IV GPU. Οι εφαρμογές Video χρησιμοποιούν την OpenMax, οι 3D χρησιμοποιούν την OpenGL ES και οι 2D την OpenVG. Οι Open Max και EGL χρησιμοποιούν το Kernel driver .

2.3 Διαδίκτυο των Πραγμάτων

2.3.1 Διαδίκτυο των Πραγμάτων

Το «διαδίκτυο των πραγμάτων» (ΔτΠ) προτάθηκε για πρώτη φορά το 1999 από τον Kevin Ashton αν και η συζήτηση περί αυτού είχε ξεκινήσει στα μέσα του 1991. Έγινε δημοφιλής μέσω του προγράμματος Auto-ID Center από το MIT και έχει σαν κεντρική ιδέα, ότι αν όλα τα αντικείμενα και οι άνθρωποι στην καθημερινή τους ζωή, ήταν εξοπλισμένα με μικροσυσκευές για αναγνώριση δεδομένων, τότε θα μπορούσαμε να διαχειριζόμαστε και να καταγράφουμε τα δεδομένα σε ηλεκτρονικούς υπολογιστές. Κύριος ορισμός είναι: «Είναι μια μελλοντική κατάσταση, κατά την οποία καθημερινά αντικείμενα - όπως κινητά τηλέφωνα, αυτοκίνητα, οικιακές συσκευές, ρούχα, ακόμη και τρόφιμα - θα συνδέονται ασύρματα στο διαδίκτυο μέσω έξυπνων μικροκυκλωμάτων και θα μπορούν να συλλέγουν και να ανταλλάσσουν δεδομένα»[10].

Αυτή η αλλαγή, από το παραδοσιακό Διαδίκτυο στο Διαδίκτυο των Πραγμάτων, έγινε εφικτή με την σμίκρυνση των επεξεργαστών και της μνήμης, την εξέλιξη της τεχνολογίας των αισθητήρων, καθώς και με την τεράστια πρόοδο στο τομέα της αναγνώρισης ραδιοσυχνοτήτων, που θεωρήθηκε σαν προϋπόθεση για το ΔτΠ κατά το αρχικό διάστημα. Ακολούθησαν κι άλλοι τρόποι επικοινωνίας όπως η επικοινωνία εγγύς πεδίου (barcodes, QR κώδικες και ψηφιακό watermarking).

Η μέχρι τώρα χρήση του Διαδικτύου των Πραγμάτων βοηθά ανθρώπους να ελέγχουν εργαλεία ή να βλέπουν τα δεδομένα τους από απόσταση. Παρόλα αυτά το ΔτΠ έχει πολύ πιο σημαντικές ικανότητες. Η πρόβλεψη του τέλους των αποθεμάτων μιας αποθήκης και η έγκαιρη παραγγελία καινούργιων προϊόντων, η πρόβλεψη των καιρικών φαινομένων, η

αυτοματοποίηση του αυτοκινήτου ή ακόμα και του σπιτιού μας είναι μόνο κάποια απλά παραδείγματα του τι δυνατότητες έχει το ΔτΠ. Με τον άνθρωπο να είναι ο κινητήριο μοχλός, δηλαδή να είναι αυτός που εισάγει τα δεδομένα, το διαδίκτυο τείνει να αλλάξει τον κόσμο[11].

2.3.2 Έξυπνο Σπίτι

Το έξυπνο σπίτι[12] είναι το σπίτι το οποίο έχει συνδεδεμένες συσκευές με ηλεκτρονικό υπολογιστή, με τρόπο ώστε να αυτοματοποιείται η χρήση τους. Ο έλεγχος των οικιακών συσκευών γίνεται με την βοήθεια αισθητήρων, οι οποίοι εγείρουν κάποια λειτουργία όταν γίνει κάποιο συμβάν. Τα συστήματα μπορεί να κυμαίνονται από απλό τηλεχειριστήριο για το φωτισμό, έως πολύπλοκα δίκτυα ελέγχου που κάνουν χρήση υπολογιστή με ποικίλους βαθμούς ευφυίας και αυτοματισμούς. Η απλότητα στο χειρισμό μέσω smartphone και tablet, καθώς επίσης και η ευκολία στη συνδεσιμότητα, έχουν ανεβάσει την δημοτικότητα των έξυπνων σπιτιών.

Ο οικιακός αυτοματισμός μπορεί να περιλαμβάνει κεντρικό έλεγχο φωτισμού, θέρμανσης, εξαερισμού και κλιματισμού. Επίσης, αυτόματος έλεγχος των συσκευών, των κλειδαριών ασφαλείας και άλλων συστημάτων. Παρέχει βελτιωμένη άνεση και ενεργειακή απόδοση, ενώ το σπίτι περιβάλλεται με ενισχυμένη ασφάλεια. Πολύ σημαντικό κομμάτι είναι ο οικιακός αυτοματισμός για τους ηλικιωμένους και τα άτομα με ειδικές ανάγκες. Αυτόματες κυλιόμενες σκάλες και συστήματα ενημέρωσης σε περίπτωση έκτακτης ανάγκης, μπορούν να προσφέρουν αυξημένη ποιότητα ζωής για αυτά τα άτομα.

Ένα άλλο μεγάλο θέμα με το οποίο ασχολούνται αρκετές έρευνες είναι ο απομακρισμένος έλεγχος. Ο οικιακός αυτοματισμός δίνει την δυνατότητα στον ιδιοκτήτη να ελέγχει τις συσκευές του σπιτιού του από απόσταση. Αυτό έχει σαν αποτέλεσμα την αύξηση της ασφάλειας, αφού μπορείς πλέον να δεις την κατάσταση του σπιτιού σου μέσα από το Διαδίκτυο από οποιοδήποτε χώρο. Επιπλέον, αυξάνεται η λειτουργικότητα του σπιτιού, αφού δεν χρειάζεται να αναμένεις μέχρι να ζεσταθεί το νερό για το μπάνιο αφού μπορείς να το κάνεις αυτόματα με μια εντολή, ενώ φτάνεις στο σπίτι.

2.4 Πλατφόρμα ενδιάμεσου λογισμικού

Πλατφόρμα ενδιάμεσου λογισμικού είναι ένα ενδιάμεσο επίπεδο λογισμικού, το οποίο βρίσκεται μεταξύ του λειτουργικού συστήματος και της εφαρμογής. Έχει σαν στόχο να αποφορτίσει τον προγραμματιστή από γενικά προβλήματα και να τον βοηθήσει στην υλοποίηση της εφαρμογής. Ο προγραμματιστής ασχολείται μόνο με τα ειδικά προβλήματα της εφαρμογής, που αφορούν την υλοποίηση και όχι με προβλήματα που αφορούν την καταναεμημένη εκτέλεση της εφαρμογής από το σύστημα. Η χρήση τέτοιων πλατφορμών εντοπίζεται κυρίως στην υποστήριξη και απλοποίηση σύνθετων και καταναεμημένων εφαρμογών. Τα πιο σημαντικά πλεονεκτήματα μιας πλατφόρμας ενδιάμεσου λογισμικού είναι η αποδοτικότητα και η αφαιρετικότητα. Υπάρχουν αρκετές πλατφόρμες που κάνουν χρήση και έλεγχο αισθητήρων. Μερικές από αυτές είναι: iStuff Mobile (Aachen Germany)[26], Campaignr (UCLA U.S.A)[29], Nevicell (Microsoft)[28], Jigsaw (Nokia + Dartmouth College U.S.A)[27]. Όλες αποτελούν πλατφόρμες ενδιάμεσου λογισμικού για λειτουργικά συστήματα όπως Sympian OS[30], Google Android και Mac OS[31], ενώ δεν υπάρχει κάποια πλατφόρμα που να καλύπτει τις ανάγκες των συσκευών με λειτουργικό σύστημα Linux. Αυτό, κατατάσσει την πλατφόρμα την οποία αναπτύξαμε, στις πρωτοποριακές έρευνες.

2.5 Έξυπνα τηλέφωνα

Τα έξυπνα τηλέφωνα αποτελούν την εξέλιξη των απλών κινητών τηλεφώνων. Βασίζονται σε κάποιο λειτουργικό σύστημα όπως Android, IOS και Windows και έχουν πολύ μεγάλη υπολογιστική ισχύ. Παράλληλα διαθέτουν πολλούς τρόπους συνδεσιμότητας όπως Bluetooth και wi-fi. Η μεταπήδηση από το κανονικό κινητό τηλέφωνο στο έξυπνο, έγινε σταδιακά προσθέτοντας λειτουργίες. Λειτουργίες όπως αυτή ενός προσωπικού ψηφιακού βοηθού (PDA[32]), φορητών media players, ψηφιακές φωτογραφικές μηχανές και βιντεοκάμερες, όπως επίσης και μονάδες πλοήγησης GPS. Η εκμετάλλευση των δυνατοτήτων των έξυπνων τηλεφώνων, καθιερώθηκε σαν ένα πεδίο έρευνας που συνεχώς αναπτύσσεται και εξελίσσεται. Πλέον τα έξυπνα τηλέφωνα χρησιμοποιούνται για την αναφορά φυσικών καταστροφών, για τις έρευνες ρύπανσης του περιβάλλοντος όπως και

για θέματα υγείας. Χρησιμοποιούνται για την μελέτη της οδικής κυκλοφορίας και την εκτίμηση της κυκλοφοριακής συμφόρησης. Τέλος, αντιλαμβάνεται την κατάσταση κίνησης, αν δηλαδή ο κάτοχος του τρέχει, οδηγά ή χρησιμοποιεί ποδήλατο.

2.6 Πλατφόρμα ενδιάμεσου λογισμικού Feel The World

Το Feel The World είναι μία πλατφόρμα ενδιάμεσου λογισμικού, το οποίο έχει αναπτυχθεί σε λογισμικό Google Android[13]. Είναι μία πλατφόρμα που εμπεριέχει κλάσεις και μεθόδους συναφείς με το θέμα της αστικής αντίληψης. Αναφέρεται στον ερευνητικό τομέα της συνεχούς αίσθησης, δηλαδή έχει την ικανότητα να είναι ενεργοποιημένη συνέχεια και να εργάζεται στο παρασκήνιο, λαμβάνοντας υπόψη και εξοικονομώντας τους πόρους της κινητής συσκευής. Το Feel The World παρέχει ένα αφαιρετικό τρόπο διαχείρισης των ενσωματωμένων, αλλά και των εξωτερικών αισθητήρων της συσκευής. Γίνεται συλλογή, επεξεργασία και αποθήκευση των δεδομένων από τους αισθητήρες. Με απλότητα το σύστημα δίνει την δυνατότητα στον σχεδιαστή να επιλέξει διάφορες παραμέτρους, όπως το ρυθμό ανάγνωσης των αισθητήρων, αλλά και το χρονικό διάστημα ανάγνωσης. Επίσης, με την συλλογή και των πόρων, δίνεται η δυνατότητα για δυναμική εκτέλεση της εφαρμογής εξοικονομώντας πόρους. Η πλατφόρμα αναλαμβάνει την συμπίεση-αποσυμπίεση των πακέτων των δεδομένων και την αποστολή τους σε εξυπηρετητή για περισσότερη επεξεργασία. Επίσης, σημαντική δυνατότητα που έχει η πλατφόρμα είναι να αναγνωρίζει την ύπαρξη wi-fi και να ρυθμίζει το χρονοδιάγραμμα των αποστολών των πακέτων. Τέλος, η πλατφόρμα έχει σχεδιαστεί και υλοποιηθεί με σκοπό να μπορεί να χρησιμοποιηθεί και από άτομα με μικρή προγραμματιστική εμπειρία.

Κεφάλαιο 3

Περιγραφή – Ανάλυση Συστήματος

3.1 Γενική Περιγραφή	15
3.2 Προδιαγραφές Συστήματος	16
3.3 Εργαλεία και βιβλιοθήκες	16
3.3.1 Eclipse	16
3.3.2 Sensaris Senspods	17
3.3.3 Linux	18
3.4 Ανάλυση Αρχιτεκτονικής	18
3.4.1 Αρχιτεκτονική Πλατφόρμας Ενδιάμεσου Λογισμικού	18
3.4.2 Ανάλυση Αρχιτεκτονικής	22

Σε αυτό το κεφάλαιο γίνεται γενική περιγραφή του συστήματος. Γίνεται ανάλυση των απαιτήσεων και καθορισμός των προδιαγραφών που πρέπει να πληρεί το σύστημα. Επίσης, αναφέρονται οι βιβλιοθήκες και τα εργαλεία που έχουν χρησιμοποιηθεί για την υλοποίηση του συστήματος. Τέλος, γίνεται η ανάλυση της αρχιτεκτονικής της πλατφόρμας που έχουμε δημιουργήσει.

3.1 Γενική Περιγραφή

Η πλατφόρμα που αναπτύχθηκε σε αυτή την διπλωματική έχει ως κύριο στόχο να βοηθήσει τον προγραμματιστή να αναπτύξει γρήγορα και αποτελεσματικά εφαρμογές που κάνουν χρήση αισθητήρων, βάσεων δεδομένων και αποστολής δεδομένων σε εξυπηρετητές. Παρέχει ένα αφαιρετικό τρόπο χρήσης και χειρισμού αισθητήρων, δίνοντας έτσι πολλές δυνατότητες σε προγραμματιστές με μικρή εμπειρία σε αυτό το τομέα. Το πιο σημαντικό είναι η ικανότητα για εγκατάσταση και ενσωμάτωση της πλατφόρμας σε χαμηλού κόστους συσκευή Linux, έτσι μια πιθανή επιλογή, είναι το Raspberry PI.

Το πρώτο κομμάτι της διπλωματικής ήταν η βελτίωση και ενίσχυση της πλατφόρμας ενδιαμέσου λογισμικού Feel The World. Έτσι δημιουργήσαμε και προσθέσαμε πακέτα για δημιουργία και χρήση βάσης δεδομένων SQLite[14]. Προσφέροντας έτοιμες και αφαιρετικές μεθόδους για δημιουργία ή διαγραφή μιας βάσης δεδομένων, καθώς επίσης και μεθόδους για αποθήκευση, διαγραφή και ανάκτηση δεδομένων από μία βάση η πλατφόρμα εξελίχθηκε και μπορούσε να καλύψει μεγαλύτερο φάσμα εφαρμογών. Επιπλέον, αναπτύχθηκε πακέτο για την προτεραιότητα της αποστολής και της παραλαβής δεδομένων. Λαμβάνοντας υπόψη τον αισθητήρα ή τους αισθητήρες που χρησιμοποιούνται, καθώς επίσης, και την επιλογή του χρήστη για την συχνότητα ανάγνωσης δεδομένων και αποστολής τους, αναπτύξαμε ένα πακέτο με μεθόδους που ρυθμίζουν το πότε θα γίνει η αποστολή, πότε θα γίνει η παραλαβή και πότε θα γίνει ανάγνωση των δεδομένων. Αυτή η προτεραιότητα καθορίζεται και από το είδος της εφαρμογής, για παράδειγμα αν η εφαρμογή κάνει μόνο ανάγνωση και παρουσίαση των δεδομένων, τότε δεν χρειάζεται να υπάρχει προτεραιότητα ως προς το πότε θα στείλει η θα παραλάβει πακέτα από τον εξυπηρετητή. Όσο αφορά την προτεραιότητα των αισθητήρων, καθορίζεται αυτόματα αλλά μπορεί να αλλάξει κατόπιν εντολής του σχεδιαστή.

Στην δεύτερη φάση της διπλωματικής αυτής, η πιο πάνω πλατφόρμα, μεταφέρθηκε από Android SDK σε λογισμικό για χαμηλού κόστους συσκευές. Αυτό είχε σαν αποτέλεσμα να αλλάξουν όλα τα πακέτα και να εναρμονιστούν σε ένα άλλο τρόπο λειτουργίας. Πολύ σημαντική ήταν η αλλαγή του χειρισμού των αισθητήρων, αφού

στο Android, που αφορά κινητές συσκευές, υπάρχουν ενσωματωμένοι αισθητήρες, αλλά και έτοιμες βιβλιοθήκες για σύνδεση μέσω Bluetooth, σε αντίθεση με άλλα λειτουργικά συστήματα για συσκευές χαμηλού κόστους, που η διασύνδεση γίνεται μέσω εντολών από το σύστημα.

3.2 Προδιαγραφές Συστήματος

Οι προδιαγραφές του συστήματος αποτελούν τη λίστα με τις λειτουργίες του, δηλαδή το τι θα κάνει το σύστημα. Στη περίπτωση μιας πλατφόρμας ενδιάμεσου λογισμικού, οι προδιαγραφές είναι μια λίστα από το τι θα έχει στην διάθεση του ο χρήστης για να μπορεί να αναπτύξει μια εφαρμογή.

Πέραν των πακέτων για βάση δεδομένων και προτεραιοτήτων το Feel The World παρέχει πακέτα για αποστολή και παραλαβή δεδομένων από εξυπηρετητή, για συμπίεση και αποσυμπίεση αρχείων, για παρουσίαση των δεδομένων, για δημιουργία αρχείων ήχου και εικόνας, για παρατήρηση της μπαταρίας και του επεξεργαστή, για χρήση των αισθητήρων του κινητού ή και εξωτερικών αισθητήρων της εταιρίας Sensaris[15] με ασύρματη σύνδεση Bluetooth. Η πλατφόρμα που έχουμε δημιουργήσει διατηρεί όλες τις ιδιότητες του Feel The World, με διαφορετικό τρόπο υλοποίησης.

3.3 Εργαλεία και βιβλιοθήκες

3.3.1 Eclipse

Το Eclipse[16] είναι ένα από τα γνωστά και δημοφιλή περιβάλλοντα ανάπτυξης εφαρμογών. Η κύρια γλώσσα προγραμματισμού είναι η Java. Η δυνατότητα να προσθέτει διάφορα plug - ins, εξυπηρετούν στην ανάπτυξη εφαρμογών με άλλες γλώσσες προγραμματισμού, όπως C , C + + και Python. Απευθύνεται κυρίως σε έμπειρους προγραμματιστές. Παρέχει εργαλεία για να βοηθήσει τη δουλειά του προγραμματιστή, όπως είναι ο αυτόματος διορθωτής κώδικα. Ο κύριος λόγος χρήσης του Eclipse σε αυτή τη διπλωματική εργασία είναι ότι όλα τα προγράμματα είναι γραμμένα σε Java.

To Feel The World είναι γραμμένο σε Java και αναπτυγμένο στο Eclipse, έτσι και η επέκταση του Feel The World θα έπρεπε να ενσωματωθεί στο Eclipse. Στη συνέχεια, η μετατροπή της πλατφόρμας από Android SDK[22] σε Linux έγινε και αυτή στο Eclipse, πάλι γιατί η ανάπτυξη της καινούργιας πλατφόρμας έγινε σε γλώσσα προγραμματισμού Java[21].

3.3.2 Sensaris Senspods

Σε αυτή τη διπλωματική εργασία, κάνουμε χρήση αισθητήρων της εταιρίας Sensaris[15], η οποία ασχολείται με την σχεδίαση και τη δημιουργία αισθητήρων που χρησιμοποιούνται σε δίκτυα αισθητήρων. Η εταιρεία αποτελεί μία αξιόπιστη επιλογή για την αγορά αισθητήρων και αυτό φέρεται και από την συνεργασία που έχει με πανεπιστήμια όπως το MIT και με οργανισμούς όπως η IBM και η Microsoft. Για την χρήση των αισθητήρων της παρέχει API[17], για να βοηθήσει τους προγραμματιστές, να έχουν την μέγιστη εκμετάλλευση των αισθητήρων.

Για την παρούσα διπλωματική εργασία χρησιμοποιήθηκαν 7 αισθητήρες, με τις παρακάτω δυνατότητες:

1. Μέτρηση έντασης ήχου
2. Μέτρηση θερμοκρασίας
3. Μέτρηση υγρασίας
4. Μέτρηση αερίου COx
5. Μέτρηση αερίου NOx
6. Μέτρηση αερίου Όζον
7. Μέτρηση κίνησης
8. Μέτρηση μπαταρίας
9. Ημερομηνία και ώρα
10. Μέτρηση ακτινοβολίας UV
11. Μέτρηση O3
12. GPS
13. Μέτρηση ποσότητας διοξειδίου του άνθρακα

Οι αισθητήρες έχουν τα εξής χαρακτηριστικά:

1. Processor Texas Instruments MSP430
2. Operating system inspired from FreeRTOS
3. Bluetooth Bluegiga Class 2 radio
4. Power: Rechargeable long lasting 3.7 V Lithium ion battery 1100 mAh for extended lifetime
5. Storage: 2G Bit micro SD Card

3.3.3 Linux

Η επιλογή του λειτουργικού συστήματος είναι αρκετά σημαντική. Επιλέξαμε το Linux για τους πιο κάτω λόγους:

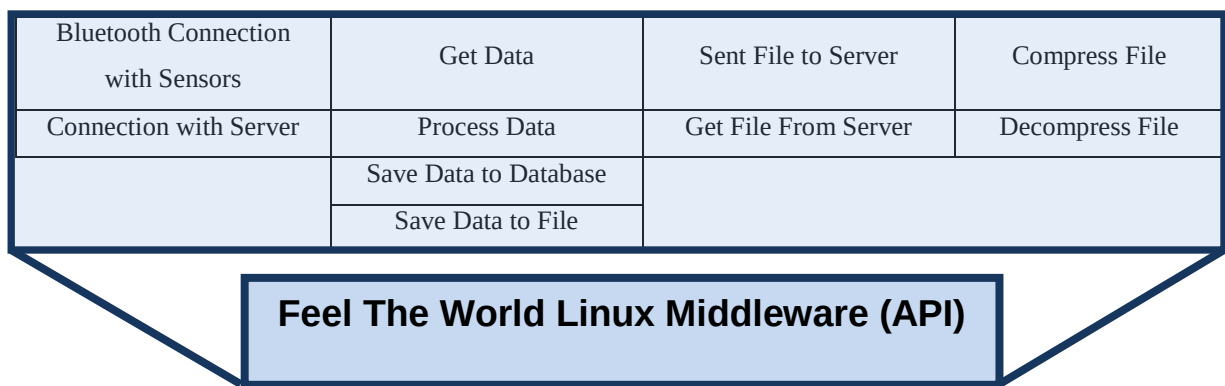
1. Πολλές προγραμματιστικές ελευθερίες
2. Χαμηλό κόστος
3. Αποτελεί το λειτουργικό σύστημα της χαμηλού κόστους συσκευής Raspberry PI
4. Εύκολο στο χειρισμό ή στην ανάπτυξη εφαρμογών.
5. Τρέχει προγράμματα γραμμένα σχεδόν σε όλες τις γλώσσες
6. Ελευθερίες στην διαχείριση των πόρων
7. Συμβατότητα με την γλώσσα προγραμματισμού JAVA, στην οποία ήταν γραμμένη η πλατφόρμα ενδιάμεσου λογισμικού Feel The World και στην οποία θα εγγραφεί και η καινούργια πλατφόρμα

3.4 Ανάλυση Αρχιτεκτονικής

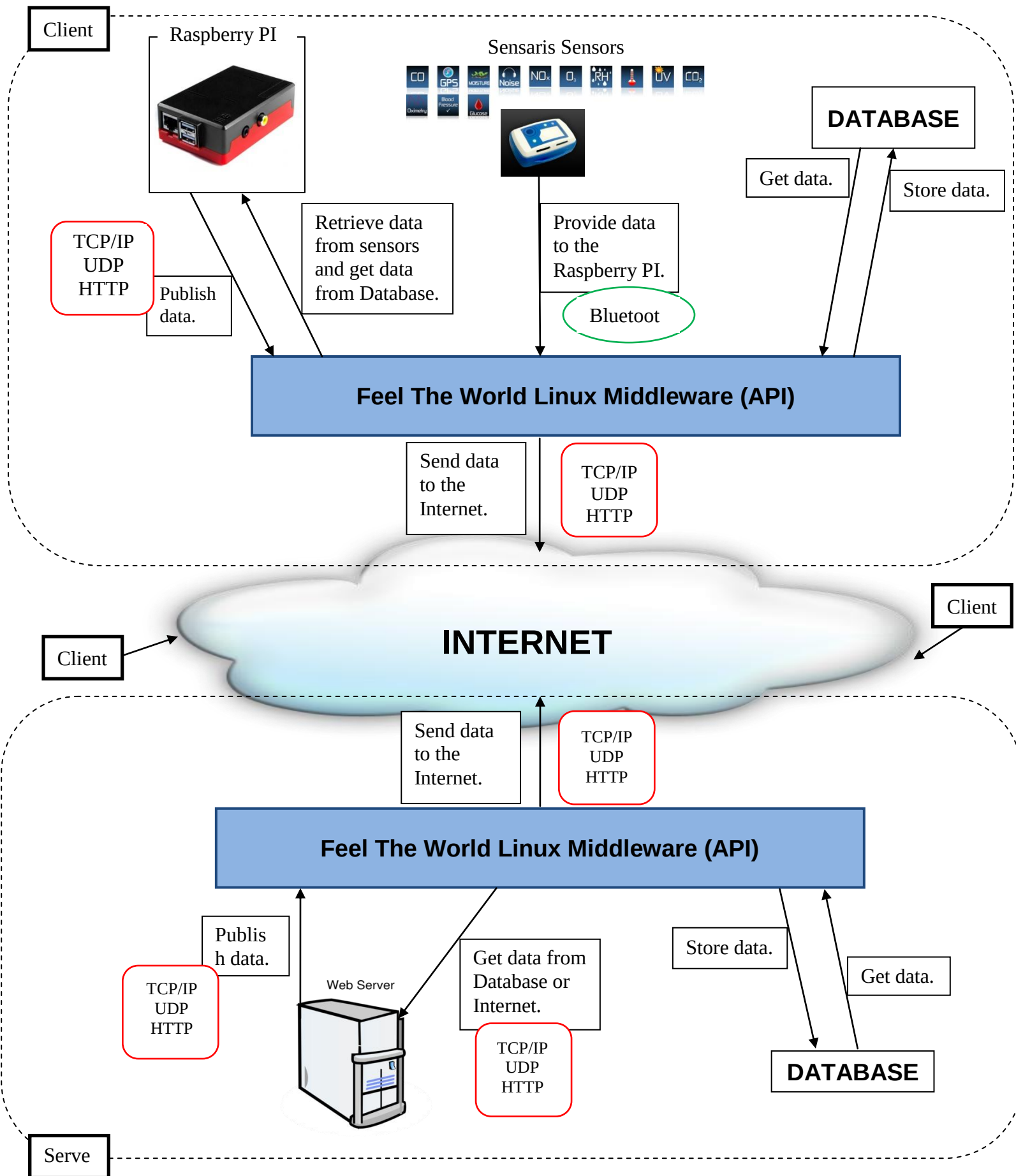
3.4.1 Αρχιτεκτονική Πλατφόρμας Ενδιάμεσου Λογισμικού

Η πλατφόρμα έχει αναπτυχθεί με σκοπό να μπορεί να εγκατασταθεί σε συσκευές χαμημού κόστους, αλλά και σε κάθε ηλεκτρονικό υπολογιστή που τρέχει Linux. Με αυτό το τρόπο μπορούν και εξυπηρετητές να έχουν την πλατφόρμα. Μέσον της πλατφόρμας το κάθε

Raspberry PI μπορεί να διαβάζει τα δεδομένα, να τα επεξεργάζεται και να τα παρουσιάζει στο Διαδίκτυο σε ξεχωριστό χώρο, αλλά και ένας εξυπηρετητής να περιέχει όλα τα δεδομένα, από όλα τα Raspberry PI σε ένα ενιαίο χώρο. Στο σχήμα 3.1 απεικονίζεται η γενική αρχιτεκτονική αυτής της πλατφόρμας, όπως επίσης και τα πρωτόκολλα που χρησιμοποιούνται. Σε αυτή τη φάση λήφθηκαν υπόψιν όλοι οι περιορισμοί, όπως τα χαρακτηριστικά των αισθητήρων, η συνδεσιμότητα μέσω wi-fi και Bluetooth, η αποστολή και η παραλαβή των πακέτων, το κόστος της μπαταρίας που έχουν οι αισθητήρες, κ.τ.λ.



Σχήμα 3.1: Η Αρχιτεκτονική Feel The World Linux.



Σχήμα 3.2: Η Αρχιτεκτονική Feel The World Linux.

Μεγάλο πλεονέκτημα για την πλατφόρμα είναι ότι μπορεί να παρέχει υπηρεσίες όχι μόνο σε εφαρμογές πελάτη, αλλά και εξυπηρετητή. Με αυτό το τρόπο, μπορούν να αναπτυχθούν οι εξής εφαρμογές με τις πιο κάτω λειτουργίες:

- Πελάτης:

1. Διασύνδεση μέσω Bluetooth με αισθητήρες.
2. Ανάκτηση δεδομένων από τους αισθητήρες.
3. Αποθήκευση των δεδομένων σε Βάση Δεδομένων.
4. Ανάκτηση δεδομένων από την Βάση Δεδομένων.
5. Επεξεργασία των δεδομένων από τους αισθητήρες για την παραγωγή σημαντικών και χρήσιμων συμπερασμάτων.
6. Συμπίεση και αποστολή πακέτων σε εξυπηρετητή. Η συμπίεση γίνεται για να μειώσει το χρόνο και την καθυστέρηση μεταφοράς των δεδομένων, όπως επίσης και για εξοικονόμηση ενέργειας.
7. Παρουσίαση των δεδομένων σε ιστοσελίδα. Η παρουσίαση μπορεί να γίνεται σε διαγράμματα και χάρτες, ενώ η ενημέρωση μπορεί να γίνεται για καθορισμένα διαστήματα χρόνο ή σε πραγματικό χρόνο.

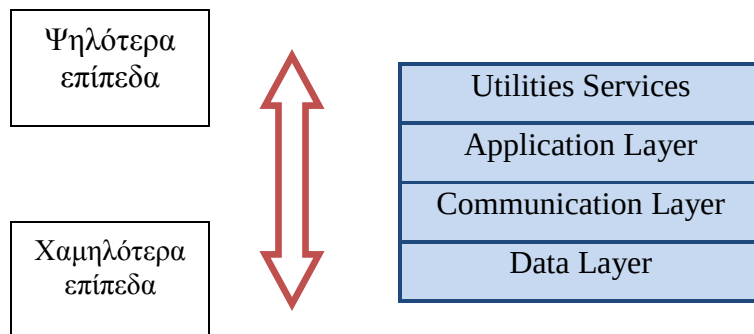
- Εξυπηρετητής:

1. Διασύνδεση μέσω Bluetooth με αισθητήρες.
2. Ανάκτηση δεδομένων από τους αισθητήρες.
3. Αποθήκευση των δεδομένων σε Βάση Δεδομένων.
4. Ανάκτηση δεδομένων από την Βάση Δεδομένων.

5. Επεξεργασία των δεδομένων από τους αισθητήρες για την παραγωγή συμπερασμάτων.
6. Επεξεργασία των δεδομένων από τους πελάτες και εξαγωγή συμπερασμάτων.
7. Συμπίεση και αποστολή πακέτων. Η συμπίεση γίνεται για να μειώσει το χρόνο και την καθυστέρηση μεταφοράς των δεδομένων, όπως επίσης και για εξοικονόμηση ενέργειας.
8. Παραλαβή και αποσυμπίεση πακέτων από κάποιο πελάτη.
9. Παρουσίαση των δεδομένων σε ιστοσελίδα. Η παρουσίαση μπορεί να γίνεται σε διαγράμματα και χάρτες, ενώ η ενημέρωση μπορεί να γίνεται για καθορισμένα διαστήματα χρόνο ή σε πραγματικό χρόνο.

3.4.2 Ανάλυση Αρχιτεκτονικής

Η πλατφόρμα ενδιάμεσου λογισμικού Feel The World Linux δίνει την δυνατότητα εγκατάστασης και χρήσης της και στην πλευρά του πελάτη, αλλά και στην πλευρά του εξυπηρετητή. Αυτό έχει σαν αποτέλεσμα οι κλάσεις και οι μέθοδοι που καθορίζουν και τα επίπεδα που εμπεριέχονται στην πλατφόρμα, να είναι ενιαίες. Τα πιο σημαντικά επίπεδα είναι το Application Layer, το Data Layer, το Utilities Services και το Communication Layer.



Σχήμα 3.3: Επίπεδα Feel The World Linux.

- Application Layer: είναι το επίπεδο στο οποίο εμπεριέχονται κλάσεις και μέθοδοι, οι οποίες βοηθούν στην διαχείριση των δεδομένων του χρήστη. Υπάρχουν κλάσεις που αναπαριστούν τα δεδομένα που λαμβάνονται από τους αισθητήρες. Κάποιοι αισθητήρες αναπαριστώνται με εξειδικευμένους τύπους δεδομένων. Υπάρχουν κλάσεις που επεξεργάζονται τα δεδομένα και κλάσεις για την παρουσίαση των πληροφοριών. Επίσης, υπάρχουν βοηθητικές κλάσεις για να καλούνται τα δεδομένα.
- Data Layer: είναι ένα από τα χαμηλότερα επίπεδα της πλατφόρμας. Είναι υπεύθυνο για την υλοποίηση των δομών των αισθητήρων. Οι δομές αυτές χρησιμοποιούνται στην συλλογή των δεδομένων καθώς και στην αποθήκευσή τους. Η δομή που χρησιμοποιείται σε αυτό το επίπεδο αφορά τους εξωτερικούς αισθητήρες της Sensaris, οι οποίοι συνδέονται μέσω Bluetooth. Επίσης, σε αυτό το επίπεδο γίνεται η ρύθμιση της ταχύτητας και της διάρκειας συλλογής των δεδομένων, όπως επίσης και η προτεραιότητα του κάθε τύπου δεδομένων. Οι επιλογές προτεραιότητας είναι:
 1. Very_High
 2. High
 3. Normal
 4. Low
 5. Very_Low

Η δομή των εξωτερικών αισθητήρων καλύπτει τις ανάγκες μιας εφαρμογής για συλλογή δεδομένων από τους ακόλουθους 17 διαφορετικούς τύπους δεδομένων:

1. Επιτάχυνση
2. Θερμοκρασία περιβάλλοντος
3. Βαρύτητα
4. Γυροσκόπιο
5. Φως
6. Γραμμή επιτάχυνσης
7. Μαγνητικό πεδίο

8. Προσανατολισμός
9. Πίεση
10. Προσέγγιση
11. Υγρασία
12. Περιστροφή
13. Θερμοκρασία
14. GPS
15. Τοποθεσία δικτύου
16. Μικρόφωνο
17. Κάμερα

Σε αυτό το επίπεδο υπάρχει και το πακέτο υλοποίησης Βάσης Δεδομένων. Η βάση δεδομένων είναι MySQL[19], και εμπεριέχονται έτοιμες μέθοδοι για την δημιουργία, την διαγραφή και την αρχικοποίηση βάσης δεδομένων. Επιπλέον, υπάρχει κλάση που αναπαριστά τα δεδομένα της βάσης δεδομένων και υπάρχει και κλάση για το χειρισμό τους, δηλαδή την εισαγωγή τους στη βάση δεδομένων, την ανάκτηση τους και την διαγραφή τους. Επίσης, δίνεται η δυνατότητα στο χρήστη για μετατροπή της βάσης δεδομένων σε αρχείο και αποστολής του αρχείου σε εξυπηρετητή.

- Communication Layer: σε αυτό το επίπεδο υπάρχουν οι βοηθητικές κλάσεις και μέθοδοι που υποστηρίζουν την αποστολή και την παραλαβή των δεδομένων. Επίσης, πολύ σημαντική είναι η επικοινωνία με τον εξυπηρετητή, μέσω προκαθορισμένων socket και port. Σε αυτό το επίπεδο γίνεται ο έλεγχος ύπαρξης διαθέσιμου δικτύου για αποστολή των δεδομένων.
- Utilities Services: το πιο ψηλό επίπεδο. Είναι το πιο κοντινό προς το χρήστη και είναι με αυτό που έχει την μεγαλύτερη αλληλεπίδραση. Εμπεριέχονται κλάσεις και μέθοδοι που βοηθούν στην υλοποίηση εφαρμογών, μειώνοντας τον κόπο και τον χρόνο υλοποίησης του προγραμματιστή, καθώς και τις γραμμές κώδικα που θα χρειαστεί να γράψει.

Σε αυτό το επίπεδο υπάρχουν οι κλάσεις και οι μέθοδοι για την δημιουργία αρχείων .txt και .csv για την αποθήκευση των δεδομένων. Αυτές οι κλάσεις είναι χρήσιμες σε περίπτωση που δεν είναι δυνατή η δημιουργία βάσης δεδομένων, αλλά και στην περίπτωση όπου τα δεδομένα προορίζονται για αποστολή.

Ακόμα ένα πολύ σημαντικό εργαλείο αυτού του επιπέδου είναι οι κλάσεις που αφορούν την σύνδεση Bluetooth, την μοναδική σύνδεση που έχουμε με τους αισθητήρες. Με αυτές τις κλάσεις γίνεται εύκολα και αφαιρετικά η ανίχνευση και σύνδεση με τους αισθητήρες με πολύ λίγο κώδικα.

Τέλος, για αυτό το επίπεδο εξίσου σημαντική είναι και η κλάση για την συμπίεση και αποσυμπίεση των αρχείων που θα αποσταλούν. Με αυτό τον τρόπο εξοικονομούμε χρόνο αποστολής αλλά και ενέργειας του ηλεκτρονικού υπολογιστή και του αισθητήρα.

Κεφάλαιο 4

Υλοποίηση Συστήματος

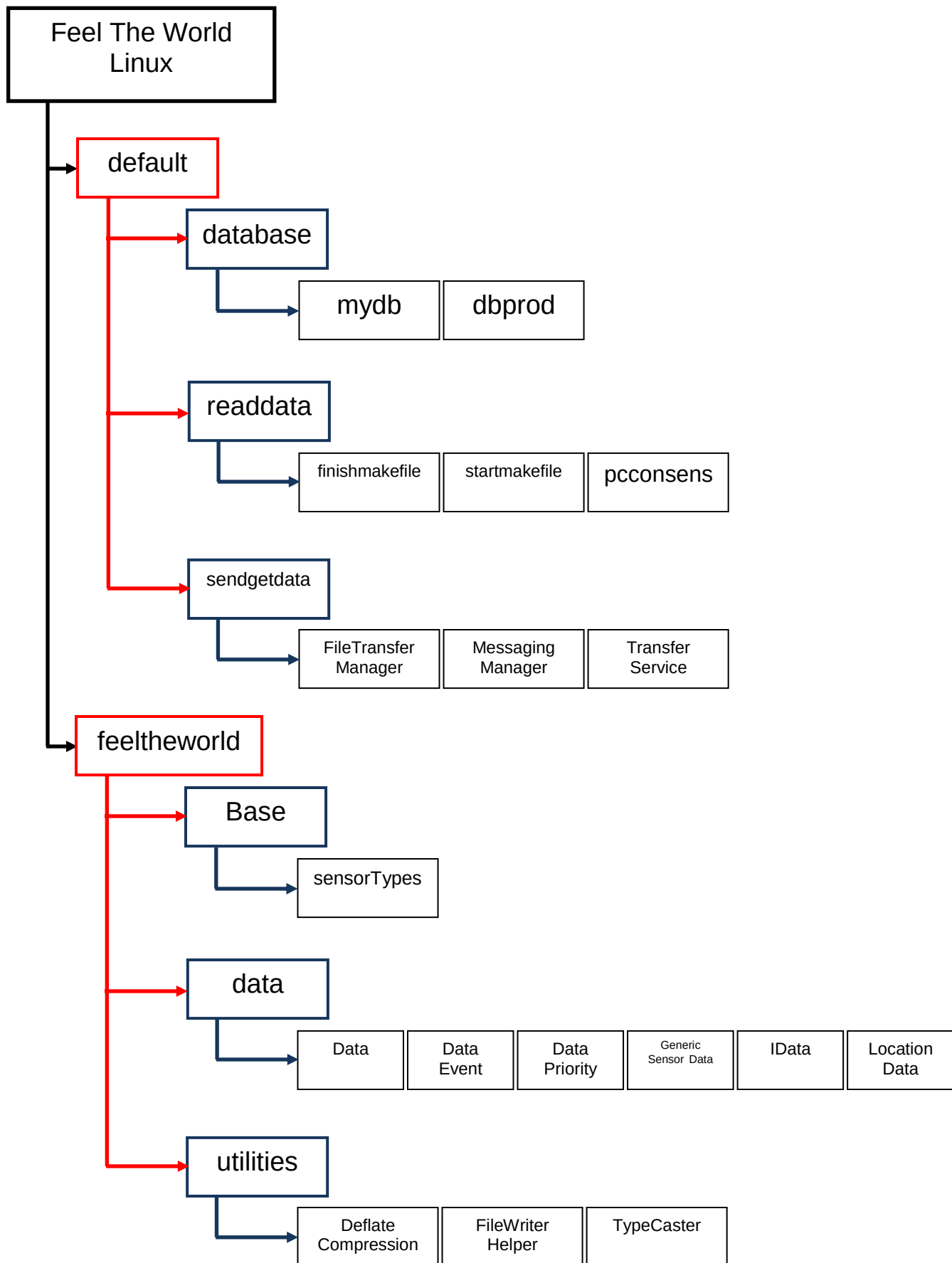
4.1 Διάγραμμα κλάσεων	27
4.2 Λεπτομέρειες Υλοποίησης	33

Σε αυτό το κεφάλαιο, περιγράφετε ο τρόπος υλοποίησης της πλατφόρμας ενδιάμεσου κώδικα. Αναφέρονται το διάγραμμα κλάσεων και οι λεπτομέρειες της υλοποίησης. Παρουσιάζονται όλα τα πακέτα και οι κλάσεις, με τις μεθόδους τους. Ακολουθεί εκτενέστερη περιγραφή των κλάσεων που αναπαριστούν την βάση δεδομένων, την ανάγνωση των δεδομένων από τους αισθητήρες και για τις προτεραιότητες. Στην περιγραφή γίνεται αναφορά σε κομμάτια κώδικα σημαντικά στην υλοποίηση.

4.1 Διάγραμμα κλάσεων

Το διάγραμμα κλάσεων είναι η σχηματική απεικόνιση της πλατφόρμας ενδιάμεσου κώδικα Feel The World Linux. Αποτελείτε από δύο κύρια πακέτα, το default και το feeltheworld. Το κάθε ένα εμπεριέχει κλάσεις και μεθόδους για διαφορετικούς σκοπούς. Το default χωρίζεται σε 3 υποκατηγορίες. Το database, το readdata και το senddata. Επίσης, το feeltheworld χωρίζεται σε 3 υποκατηγορίες. Το base, το data και το utilities.

Ο λόγος που η πλατφόρμα χωρίζεται σε κατηγορίες και υποκατηγορίες είναι η ευκολία χρήσης της. Ο χρήστης, δηλαδή ο προγραμματιστής, μπορεί να χρησιμοποιήσει πιο εύκολα κάποια κλάση ή μέθοδο. Στην περίπτωση που ο χρήστης θέλει να προσθέσει κλάσεις ή μεθόδους, μπορεί να κατηγοριοποιήσει τις προσθήκες του καλύτερα. Ακόμη πιο σημαντική είναι η περίπτωση που ο χρήστης θέλει να αντικαταστήσει κάποια μέθοδο ή να την ενισχύσει (@overwrite). Τέλος, η κατηγοριοποίηση της πλατφόρμας παίζει πολύ σημαντικό ρόλο στην γρήγορη αποσφαλμάτωση.



Εικόνα 4.1 Διάγραμμα κλάσεων

FeelTheWorldLinux: Πλατφόρμα Ενδιάμεσου Λογισμικού. Χωρίζεται σε δύο κατηγορίες.

1. Default: η κατηγορία αυτή ασχολείται με την υλοποίηση της βάσης δεδομένων, όπως επίσης και την διαχείριση της. Εμπεριέχει πακέτα για το σχηματισμό της βάσης και των πινάκων, ενώ υπάρχουν και οι κλάσεις για την μετατροπή των δεδομένων σε τύπους δεδομένων που μπορούν να αποθηκευτούν στην βάση δεδομένων. Επίσης εμπεριέχει τα πακέτα για την δημιουργία σύνδεσης με τους αισθητήρες, την διακοπή της σύνδεσης αλλά και την ανάγνωση των δεδομένων.
- ❖ Database: Πακέτο για την υλοποίηση βάσης δεδομένων. Εμπεριέχει δύο κλάσεις για την διαχείριση των δεδομένων και της Βάσης Δεδομένων.
- Mydb: Δημιουργεί την βάση δεδομένων MySQL με όνομα που επιλέγει ο χρήστης. Ακολουθώς, δημιουργεί τον πίνακα των δεδομένων με όνομα TB και στήλες id (integer) και data(String). Το id δίνεται αυτόματα από την κλάση ενώ το data είναι τα δεδομένα που δίνονται από τον χρήστη. Εμπεριέχει μεθόδους για την εισαγωγή των δεδομένων, την ανεύρεση τους και την επιστροφή τους αν υπάρχουν. Η αναζήτηση γίνεται είτε με το data είτε με το id. Επίσης, υπάρχουν μέθοδοι για την μετατροπή της βάσης σε αρχείο κειμένου (*.txt), για να μπορεί να γίνει η αποστολή όλων των δεδομένων της Βάσης Δεδομένων. Ανάλογα, υπάρχει και η μέθοδος για μετατροπή του αρχείου κειμένου σε Βάση Δεδομένων. Υπάρχουν μέθοδοι για την διαγραφή ενός αντικειμένου με βάση το id ή το data. Τέλος, υπάρχουν μέθοδοι για την επαναφορά της Βάσης Δεδομένων ή και την ολοκληρωτική διαγραφή της.
 - Dbprod: η κλάση που αντιπροσωπεύει το αντικείμενο που εισάγεται ή εξάγεται από την Βάση Δεδομένων. Ο χρήστης μπορεί να δώσει τιμές κατά την αρχικοποίηση του αντικειμένου, αλλά του δίνεται η δυνατότητα να αλλάξει ή να παρουσιάσει την τιμή με έτοιμες μεθόδους.

❖ **Readdata**: πακέτο για την ανάγνωση των δεδομένων από τους αισθητήρες. Η αλλαγή από την πλατφόρμα Android στην πλατφόρμα Linux είχε σαν αποτέλεσμα την αλλαγή στον τρόπο ανάγνωσης των δεδομένων από τους αισθητήρες. Πλέον δεν υπάρχουν ενσωματωμένοι αισθητήρες, ενώ εξακολουθούμε να έχουμε ένωση με Bluetooth με τους εξωτερικούς. Σε αντίθεση με την πλατφόρμα Android, η πλατφόρμα Linux δεν εμπεριέχει έτοιμες κλάσεις για την ένωση μέσω Bluetooth. Ο τρόπος ανάγνωσης των δεδομένων έχει ως εξής: καλείτε το `Startmakefile`, το οποίο τρέχει εντολές σε `terminal[20]` για την αντιστοίχιση του αισθητήρα και την δημιουργία ένωσης. Ακολούθως, ο αισθητήρας ξεκινά να στέλνει τα δεδομένα τα οποία αποθηκεύονται σε αρχείο κειμένου, το οποίο αντιπροσωπεύει την συσκευή. Στη συνέχεια μπορούμε να διαβάσουμε το αρχείο το οποίο κάνει αυτόματα ανανέωση όταν ο αισθητήρας στείλει κάποιο αποτέλεσμα. Τέλος, οποιαδήποτε στιγμή θέλουμε να τερματίσουμε την ένωση, τρέχουμε το αρχείο `Finishmakefile`. Σε αυτό το πακέτο υπάρχουν τρεις κλάσεις:

- **Finishmakefile**: η κλάση που αναλαμβάνει τον τερματισμό της σύνδεσης και της ανάγνωσης δεδομένων από τους αισθητήρες. Λειτουργεί στέλνοντας την εντολή `echo χχχχχχ| sudo -S sh -c "killall "cat ""` στο `terminal`, σκοτώνοντας τις διεργασίες που αφορούν την ένωση με τους αισθητήρες.
- **Pcconsens**: διαβάζει τα δεδομένα σε πραγματικό χρόνο από το αρχείο και αποθηκεύει τα στοιχεία σε διαφορετικές βάσεις. Πριν την αποθήκευση, επεξεργάζεται τα δεδομένα από τους αισθητήρες, μετατρέποντας τα σε δεκαδικούς αριθμούς για πιο εύκολη εισαγωγή και και άμεση χρήση.
- **Startmakefile**: η κλάση η οποία τρέχοντας την εντολή `echo χχχχχχ| sudo -S rfcomm bind /dev/rfcomm0 00:07:80:46:E0:C7` στο `terminal`, ξεκινά την σύνδεση μεταξύ υπολογιστή και αισθητήρα. Ο αισθητήρας συλλέγει τα δεδομένα και τα στέλνει στη συσκευή. Η συσκευή λαμβάνει τα δεδομένα και τα αποθηκεύει σαν εγγραφή σε αρχείο. Ακολούθως, το αρχείο χρησιμοποιείται για να παίρνουμε τα δεδομένα.

- ❖ **Sendgetdata:** πακέτο για την αποστολή των δεδομένων σε εξυπηρετητή. Χρησιμοποιείται για την δημιουργία του αρχείου με τα δεδομένα και επίσης εμπεριέχει τις κλάσεις για τη παραλαβή ανάλογου αρχείου από τον εξυπηρετητή. Δημιουργεί την ένωση με τον εξυπηρετητή. Σε αυτό το πακέτο υπάρχουν τρεις κλάσεις:
 - **fileTransferManager:** δημιουργία του αρχείου με βάση τον τύπο επιλογής του χρήστη. Παρέχει τις μεθόδους για την αποστολή των αρχείων σε εξυπηρετητή, είτε τα αρχεία είναι συμπιεσμένα είτε όχι. Επίσης, έχει τις μεθόδους για παραλαβή του αρχείου.
 - **MessagingMananger:** αποτελεί την κλάση για την υλοποίηση πελάτη-εξυπηρετητή (client-server) ένωση. Εμπεριέχει μεθόδους για την δημιουργία ένωσης μέσω socket, id address και port. Έχει την ικανότητα για ανταλλαγή μηνυμάτων μεταξύ του εξυπηρετητή και του πελάτη. Τα μηνύματα μπορεί να είναι είτε συμπιεσμένα είτε όχι.
 - **TransferService:** η κλάση που εργάζεται στο παρασκήνιο με σκοπό την αποστολή των δεδομένων. Εμπεριέχει τις μεθόδους για τις προτεραιότητες. Ελέγχει τις προτεραιότητες των αρχείων και στέλλει αυτό με την πιο ψηλή, ενώ περνά ο χρόνος μειώνει σταδιακά την προτεραιότητα ούτως ώστε να γίνει αποστολή και άλλων αρχείων με πιο χαμηλή προτεραιότητα. Αν φτάσει κάποιο αρχείο με 2 επίπεδα πιο ψηλή προτεραιότητα από αυτή που βρίσκεται, τότε αποστέλλετε αυτό και ακολούθως συνεχίζει από εκεί που είχε σταματήσει. Υπάρχουν πέντε επίπεδα προτεραιότητας, τα οποία καθορίζουν και την σημαντικότητα του αρχείου.
- 2. **Feeltheworld:** η κατηγορία που αφορά την αναπαράσταση των δεδομένων, από τους αισθητήρες. Επίσης, ασχολείται με την συμπίεση και την αποσυμπίεση των αρχείων που θα σταλούν στον εξυπηρετητή.

- ❖ Base: εμπεριέχει την κλάση για την αρχικοποίηση και τον καθορισμό των αισθητήρων.
 - sensorTypes: η κλάση που χρησιμοποιείτε στον καθορισμό συγκεκριμένου ακέραιου αριθμού για κάθε αισθητήρα από το 1 μέχρι το 19. Ο λόγος αυτής της ανάθεσης είναι για την ευκολία καθορισμού και αναγνώρισης του κάθε αισθητήρα κατά την διαχείριση του και την παραλαβή των δεδομένων που διαβάζει και αποστέλλει.
- ❖ Data: πακέτο που ασχολείτε με τον καθορισμό των δεδομένων. Περιέχει έξι κλάσεις:
 - Data: η κλάση, η οποία χρησιμοποιείται από τον χρήστη για τον καθορισμό των δεδομένων κάποιου αισθητήρα. Χρησιμοποιείτε στον διαχωρισμό των δεδομένων που λαμβάνονται από κάθε αισθητήρα.
 - DataEvent: χρησιμοποιείται στον χειρισμό των Data κάθε αισθητήρα.
 - DataPriority: η κλάση, η οποία καθορίζει τους πέντε τύπους προτεραιότητας των δεδομένων. Οι πέντε τύποι είναι: Very_High, High, Normal, Low and Very_Low.
 - GenericSensorData, IData: κλάσεις, η οποίες χρησιμοποιούνται από τον χρήστη για τον καθορισμό των δεδομένων κάποιου αισθητήρα.
 - LocationData: κλάση, η οποία χρησιμοποιείτε από τον χρήστη για τον καθορισμό των δεδομένων αισθητήρα που σχετίζεται με την γεωγραφική θέση (GPS).

❖ Utilities: πακέτο που ασχολείται με την συμπίεση και αποσυμπίεση δεδομένων, με την δημιουργία αρχείου με τα δεδομένα και με την ανεύρεση του τύπου του αισθητήρα. Περιέχει τρεις κλάσεις:

- DeflateCompression: η κλάση που αναλαμβάνει την συμπίεση και την αποσυμπίεση μιας συμβολοσειράς. Δέχεται σαν είσοδο την συμβολοσειρά και το μέγεθος των δεδομένων και επιστρέφει την συμπιεσμένη ή την αποσυμπιεσμένη συμβολοσειρά.
- FileWriterHelper: η κλάση που αναλαμβάνει την δημιουργία αρχείου κειμένου και την εγγραφή σε αυτό των δεδομένων που διαβάζονται από ένα αισθητήρα. Ο αισθητήρας καθορίζεται από ένα ακέραιο, ενώ τα δεδομένα από μία λίστα με IData.
- TypeCaster: κλάση που επιστρέφει τον τύπο του αισθητήρα όταν λάβει σαν είσοδο ένα ακέραιο αριθμό από το 1 μέχρι το 19.

4.2 Λεπτομέρειες Υλοποίησης

Στην υλοποίηση της πλατφόρμας πολύ σημαντικό κομμάτι αποτελεί η επικοινωνία μέσω Bluetooth. Για να μπορέσουμε να την επιτύχουμε, στέλνουμε εντολές στο terminal, οι οποίες υλοποιούν την ένωση με τον αισθητήρα, και ακολούθως, διαβάζουμε τις εισαγωγές του αισθητήρα.

Με τον πιο κάτω κώδικα, μέσω της JAVA στέλνεται εντολή στο terminal (make), η οποία εκτελείται. Ακολούθως, διαβάζουμε τα μηνύματα που παρουσιάζει το terminal. Η συσκευή ψάχνει διαθέσιμους αισθητήρες μέσω Bluetooth και όταν βρει αυτούς που ψάχνει, δημιουργεί σύνδεση.

Pcconsens.java

```

try {

    p = Runtime.getRuntime().exec("make");

    stdInput = new BufferedReader(new InputStreamReader(
        p.getInputStream()));

    while ((line1 = stdInput.readLine()) != null)
        System.out.println(line1);

} catch (Exception e1) {
    e1.printStackTrace();
}

```

Η εντολή «make» διαβάζει και εκτελεί όλες τις γραμμές που βρίσκονται κάτω από το all, ενώ το «make clean» εκτελεί μόνο τις εντολές κάτω από το clean.

Η εντολή «hcitool scan» ψάχνει και επιστρέφει όλες τις συσκευές που εκπέμπουν Bluetooth. Η εντολή «rfcomm bind /dev/rfcomm0 00:07:80:46:E0:C7» υλοποιεί την ένωση με τον αισθητήρα με την κατάλληλη διεύθυνση. Αν δεν βρεθεί ο κατάλληλος αισθητήρας ή δεν μπορέσει να γίνει η σύνδεση, αρχίζει ξανά η ίδια διαδικασία. Οι εντολές «echo χχχχχχ» και «sudo -S» χρησιμοποιούνται για να δώσουν πρόσβαση υπερχρήστη και να επιτραπεί η χρήση των υπόλοιπων εντολών. Μόλις επιτύχει την σύνδεση, τα δεδομένα από τους αισθητήρες καταγράφονται σε αρχείο. Για κάθε αισθητήρα υπάρχει ξεχωριστό αρχείο. Στην συνέχεια το πρόγραμμα μπορεί να διαβάζει το αρχείο, ενώ εισάγονται δεδομένα από τους αισθητήρες. Τέλος, η εντολή «-c "killall "cat ""» σκοτώνει κάθε διεργασία που έχει σχέση με τον αισθητήρα τερματίζοντας έτσι την διαδικασία.

Makefile

all:

```

    hcitool scan
    echo χχχχχχ| sudo -S rfcomm bind /dev/rfcomm0 00:07:80:46:E0:C7

```

clean:

```

    echo χχχχχχ| sudo -S sh -c "killall "cat ""

```


Στη συνέχεια, παρουσιάζεται ο κώδικας όπου τα δεδομένα του αισθητήρα που έχουν καταγραφεί στο αρχείο, τοποθετούνται σε βάση δεδομένων. Για κάθε στοιχείο που διαβάζουν οι αισθητήρες δημιουργούμε ξεχωριστή βάση δεδομένων. Για την κάθε βάση δίνουμε ξεχωριστό όνομα και μέσω του κατασκευαστή έχουμε την ικανότητα για πρόσβαση στα δεδομένα της. Ακολουθώντας, διαβάζουμε τα δεδομένα που υπάρχουν στο αρχείο γραμμή γραμμή. Η κάθε γραμμή αποτελεί μία εγγραφή του αισθητήρα για ένα στοιχείο. Επεξεργαζόμαστε την γραμμή και κρατάμε το στοιχείο και την τιμή την οποία μετατρέπουμε σε πραγματικό αριθμό. Στην συνέχεια, με βάση το στοιχείο γίνεται εισαγωγή της τιμής στη βάση δεδομένων.

```
mydb dbuv = new mydb("uv");
mydb dbo = new mydb("o");
mydb dbh = new mydb("hum");

try {
    in = new BufferedReader(new FileReader("/dev/rfcomm0"));
    String s1;

    while ((s1 = in.readLine()) != null) {

        .....

        dbh.adddata(data1, "hum", as);
        dbuv.adddata(data1, "uv", bs);
        dbo.adddata(data1, "o", cs);
    }
}
```

Τέλος, ο τερματισμός της διαδικασίας ανάγνωσης των δεδομένων από τον αισθητήρα γίνεται και πάλι με εντολή στο terminal. Εκτελώντας την εντολή make clean, εκτελείται ο κώδικας κάτω από το clean στο Makefile, με αποτέλεσμα κάθε διεργασία που έχει σχέση με τον αισθητήρα και την σύνδεση, να τερματίζεται.

```
try {
    p = Runtime.getRuntime().exec("make clean");

    stdInput = new BufferedReader(new InputStreamReader(
        p.getInputStream()));
}
```

```

        while ((line = stdInput.readLine()) != null)
            System.out.println(line);

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

```

Για την δημιουργία και χρήση της βάσης δεδομένων θα πρέπει να έχουμε τις πιο κάτω εντολές. Στην αρχή καλείται η κλάση για να ενεργοποιηθεί η mysql. Ακολούθως δημιουργείται ένωση με την βάση δεδομένων χρησιμοποιώντας το username και το password. Τέλος, χρησιμοποιώντας το αντικείμενο της ένωσης με την βάση, δημιουργείται ένα Statement για να χρησιμοποιηθεί στην εκτέλεση των επερωτήσεων. Ακολούθως, οι επερωτήσεις γράφονται σε συμβολοσειρά και εκτελούνται μέσω του Statement.

Ένωση:

```

Connection con;
Statement st;

Class.forName("com.mysql.jdbc.Driver");
con = DriverManager.getConnection("jdbc:mysql://localhost:3306/",
                                   "root", "cr7");
st = con.createStatement();

```

Εκτέλεση επερώτησης:

```

String s = "INSERT into TB VALUES('" + id + "','" + data.getData()
           + "') ";

int a = st.executeUpdate(s);
if (a == 0) {
    System.out.println("PLM sti isagwgi tis vasis");
    return;
}

```

Τερματισμός Βάσης Δεδομένων:

```

con.close();

```

Προτεραιότητες αρχείων:

```
int next_file() {
    String name = "";
    DataPriority dp = DataPriority.Very_High;
    DataFile df;
    int i = 0;

    while (!(filenames_stored.isEmpty())) {
        i = 0;

        while (i < filenames_stored.size()) {
            df = filenames_stored.get(i);
            if ((dp == df.getDataPriority()) && !(df.isSent())) {
                return i;
            }
            i++;
        }
        dp = reduce(dp);
    }
    return 0;
}
```

Η πιο πάνω μέθοδος ορίζει την σειρά με την οποία θα διαβαστούν τα αρχεία με βάση την προτεραιότητα την οποία έχει το κάθε ένα. Ενόσω υπάρχουν αρχεία η πιο πάνω μέθοδος εργάζεται. Το διάβασμα των αρχείων ξεκινά από αυτά με την πιο ψηλή προτεραιότητα. Εάν ένα αρχείο έχει πιο ψηλή προτεραιότητα ή ίση με την προτεραιότητα που εξετάζουμε αυτή τη στιγμή, τότε διαβάζεται. Σε οποιαδήποτε άλλη περίπτωση παραμένει στην λίστα και η διαδικασία συνεχίζεται. Εάν δεν υπάρχουν άλλα αρχεία με αυτή την προτεραιότητα τότε μειώνεται με την χρήση της μεθόδου reduce και η διαδικασία ξανά ξεκινά.

Δύσκολα σημεία και επίλυση:

Στην υλοποίηση της Πλατφόρμας Ενδιάμεσου Λογισμικού αντιμετωπίσαμε αρκετές δυσκολίες και επιλύσαμε αρκετά προβλήματα που είχαν προκύψει. Το πιο σημαντικό από αυτά ήταν και η σύνδεση με τους αισθητήρες. Ακολουθήθηκαν αρκετές μέθοδοι και στάδια μέχρι την τελική επίλυση.

Σε πρώτη φάση, η σύνδεση γινόταν χειροκίνητα μέσω του terminal και το αρχείο με τις εγγραφές δημιουργούνταν ξεχωριστά. Αλλά για να αρχίσει η ανάγνωση από το αρχείο, θα έπρεπε να σταματήσει η σύνδεση. Για το λόγο αυτό, το διάβασμα γινόταν σε πακέτα, δηλαδή διαβάζονταν δεδομένα για κάποιο χρονικό διάστημα και στη συνέχεια τερματιζόταν η σύνδεση για να εκτελεστεί η διαδικασία επεξεργασίας και αποθήκευσης των δεδομένων. Όλη η διαδικασία επαναλαμβανόταν.

Στη συνέχεια, η σύνδεση μέσω terminal γινόταν με άλλες εντολές, επιτρέποντας έτσι την παράλληλη επεξεργασία του αρχείου των εγγραφών. Παράλληλα με τις εντολές στο terminal έτρεχε το πρόγραμμα για να επεξεργαστεί και να εισάγει τα δεδομένα στην Βάση Δεδομένων.

Ακολούθως, έγιναν αλλαγές για την αυτοματοποίηση της σύνδεσης με τους αισθητήρες, προσθέτοντας κώδικα που εκτελούσε τις εντολές μέσω του προγράμματος. Στη συνέχεια, το αποτέλεσμα δεν ήταν το αναμενόμενο. Οι εντολές δεν εκτελούνταν λόγω μη εγκεκριμένων αδειών(permission denied) από το λειτουργικό σύστημα. Η λύση ήταν η δημιουργία ενός αρχείου makefile το οποίο θα εμπεριέχει τις εντολές που θα εκτελούνται στο terminal, ενώ η εκτέλεση του προγράμματος γινόταν από τον φάκελο user/bin για να διατηρεί όλες τις άδειες του λειτουργικού συστήματος.

Τέλος, και οι δύο διαδικασίες, της δημιουργίας σύνδεσης και του διαβάσματος του αρχείου, επιβάλλεται να εκτελούνται παράλληλα. Για αυτό το λόγο χρησιμοποιήθηκε κώδικας πολλαπλών νημάτων (multi thread). Τα δύο κομμάτια κώδικα αποτελούν τις κλάσεις που εκτελούνται παράλληλα. Με αυτό το τρόπο γίνεται η ανάγνωση των δεδομένων από τους αισθητήρες και η καταγραφή τους σε αρχείο, παράλληλα με την διαδικασία της ανάκτησης των δεδομένων από το αρχείο και την επεξεργασία και αποθήκευσή τους.

Κεφάλαιο 5

Δοκιμή – Αξιολόγηση

5.1 Εφαρμογές που αναπτύχθηκαν	40
5.1.1 Sense Me	40
5.1.2 Smart Home	46
5.2 Μεθοδολογία Αξιολόγησης	52
5.3 Μετρικές Αξιολόγησης	52

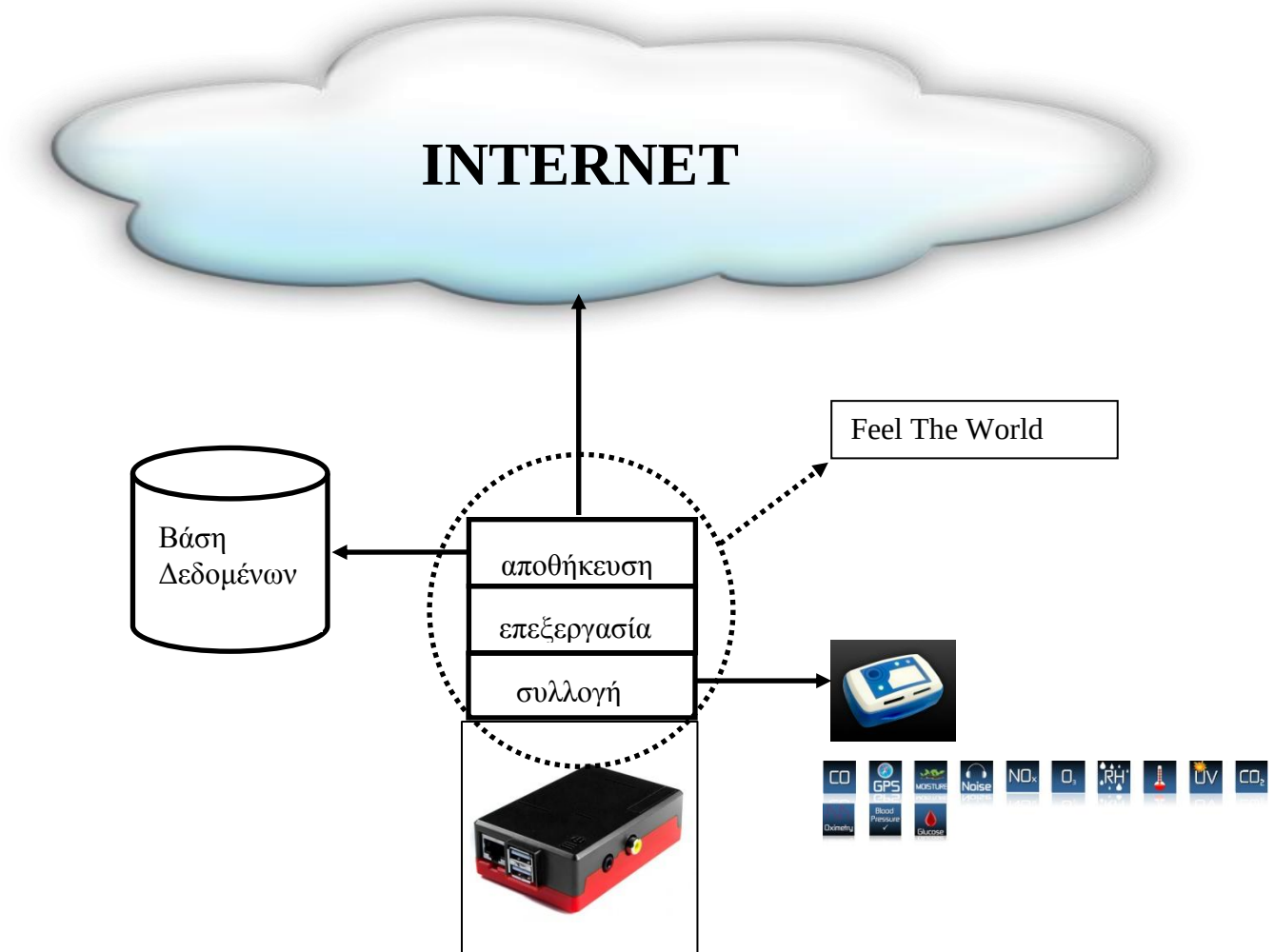
Σε αυτό το κεφάλαιο, γίνεται χρήση της πλατφόρμας ενδιαμέσου λογισμικού, για την δημιουργία δύο συστημάτων. Σκοπός της δημιουργίας των συστημάτων είναι η αξιολόγηση της πλατφόρμας. Παρουσιάζονται η μεθοδολογία που ακολουθήθηκε, όπως επίσης και οι μετρικές. Η αξιολόγηση γίνεται με σκοπό την ανάδειξη των δυνατοτήτων της πλατφόρμας, αλλά και επιβεβαίωσης της χρησιμότητας της σε συστήματα που αφορούν απεικόνιση και διαχείριση δεδομένων πραγματικού χρόνου από αισθητήρες.

5.1 Εφαρμογές που αναπτύχθηκαν

Για την πειραματική αξιολόγηση της πλατφόρμας ενδιάμεσου λογισμικού Feel The World Linux, έχουν αναπτυχθεί δύο συστήματα. Το πρώτο αφορά την ανάγνωση και απεικόνιση των δεδομένων από αισθητήρα σε διαγράμματα και σε χάρτη, ενώ το δεύτερο σύστημα αποτελεί την ψηφιακή μοντελλοποίηση ενός έξυπνου σπιτιού με αυτοματοποιημένες διεργασίες.

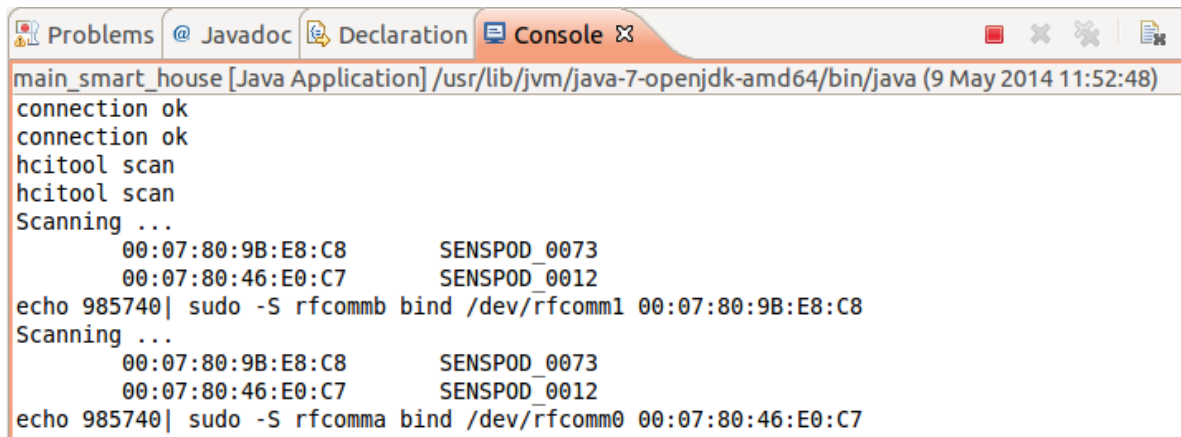
5.1.1 Sense Me

Η εφαρμογή Sense Me έχει αναπτυχθεί με σκοπό την αξιολόγηση της πλατφόρμας ενδιάμεσου λογισμικού, αλλά και για την παρατήρηση των δεδομένων που συλλέγουν οι αισθητήρες. Χρησιμοποιείται για την αναγνώριση του περιβάλλοντος, αλλά και για την ενημέρωση των χρηστών για την περιοχή μέσω Διαδικτύου. Αποτελείται από τρία κομμάτια. Το πρώτο αφορά την συλλογή των δεδομένων, το δεύτερο την επεξεργασία των δεδομένων και το τρίτο την αποθήκευση των δεδομένων σε Βάση Δεδομένων. Επίσης, έχει αναπτυχθεί ιστοσελίδα για την παρουσίαση των δεδομένων στο Διαδίκτυο. Ακολουθεί η αρχιτεκτονική της εφαρμογής και ο τρόπος λειτουργίας της. Όπως φαίνεται και στην εικόνα 5.1, η εφαρμογή έχει αναπτυχθεί πάνω στην πλατφόρμα ενδιάμεσου λογισμικού.



Εικόνα 5.1 Αρχιτεκτονική Εφαρμογής

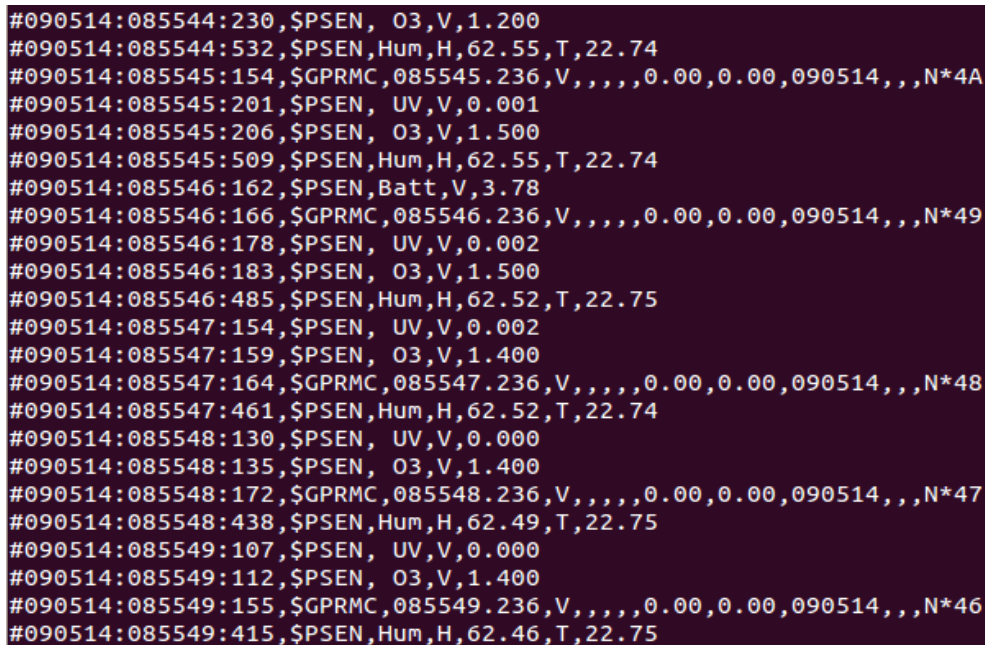
Η εφαρμογή εγκαθίσταται σε κάποια χαμηλού κόστους συσκευή με λειτουργικό σύστημα Linux, όπως για παράδειγμα το Raspberry PI, και ακολούθως την ενεργοποιούμε. Η εφαρμογή θα δημιουργήσει σύνδεση με τους αισθητήρες που εμφανίζονται. Ακολούθως, θα αρχικοποιήσει την βάση δεδομένων και θα ξεκινήσει η συλλογή των δεδομένων. Στην συνέχεια θα επεξεργαστεί τα δεδομένα και αφού τα αποθηκεύσει στη βάση δεδομένων θα απεικονιστούν σε διαγράμματα σε ιστοσελίδα.



```
main_smart_house [Java Application] /usr/lib/jvm/java-7-openjdk-amd64/bin/java (9 May 2014 11:52:48)
connection ok
connection ok
hcitool scan
hcitool scan
Scanning ...
    00:07:80:9B:E8:C8      SENSPOD_0073
    00:07:80:46:E0:C7      SENSPOD_0012
echo 985740| sudo -S rfcomm bind /dev/rfcomm1 00:07:80:9B:E8:C8
Scanning ...
    00:07:80:9B:E8:C8      SENSPOD_0073
    00:07:80:46:E0:C7      SENSPOD_0012
echo 985740| sudo -S rfcomm bind /dev/rfcomm0 00:07:80:46:E0:C7
```

Εικόνα 5.2 Ένωση με τους αισθητήρες και αρχικοποίηση της Βάσης Δεδομένων

Μέσω της πλατφόρμας γίνεται η αρχικοποίηση της βάσης δεδομένων, όπως παρατηρούμε από την εικόνα 5.2. Στη συνέχεια, διαβάζονται τα δεδομένα από τους αισθητήρες. Η εικόνα 5.3 παρουσιάζει τα δεδομένα που διαβάζονται από τους αισθητήρες. ακολούθως επεξεργάζονται τα δεδομένα, έτσι ώστε να παραμείνει μόνο η τιμή του στοιχείου που παρακολουθούμε.



```
#090514:085544:230,$PSEN, 03,V,1.200
#090514:085544:532,$PSEN,Hum,H,62.55,T,22.74
#090514:085545:154,$GPRMC,085545.236,V,,,,,0.00,0.00,090514,,,N*4A
#090514:085545:201,$PSEN, UV,V,0.001
#090514:085545:206,$PSEN, 03,V,1.500
#090514:085545:509,$PSEN,Hum,H,62.55,T,22.74
#090514:085546:162,$PSEN,Batt,V,3.78
#090514:085546:166,$GPRMC,085546.236,V,,,,,0.00,0.00,090514,,,N*49
#090514:085546:178,$PSEN, UV,V,0.002
#090514:085546:183,$PSEN, 03,V,1.500
#090514:085546:485,$PSEN,Hum,H,62.52,T,22.75
#090514:085547:154,$PSEN, UV,V,0.002
#090514:085547:159,$PSEN, 03,V,1.400
#090514:085547:164,$GPRMC,085547.236,V,,,,,0.00,0.00,090514,,,N*48
#090514:085547:461,$PSEN,Hum,H,62.52,T,22.74
#090514:085548:130,$PSEN, UV,V,0.000
#090514:085548:135,$PSEN, 03,V,1.400
#090514:085548:172,$GPRMC,085548.236,V,,,,,0.00,0.00,090514,,,N*47
#090514:085548:438,$PSEN,Hum,H,62.49,T,22.75
#090514:085549:107,$PSEN, UV,V,0.000
#090514:085549:112,$PSEN, 03,V,1.400
#090514:085549:155,$GPRMC,085549.236,V,,,,,0.00,0.00,090514,,,N*46
#090514:085549:415,$PSEN,Hum,H,62.46,T,22.75
```

Εικόνα 5.3 Τα δεδομένα του αισθητήρα


```

main_smart_house [Java Application] /usr/lib/jvm/java-7-openjdk-amd64/bin/java (9 May 2014 12:05:48)
----- OFF Lamp-----
bat: 3.77 float: 3.77
----- battery full -----

teliki wra: 09:06:49
----- OFF Lamp-----
uv: 0.000 float: 0.0
----- ON Lamp Garden-----
INSERT into TB VALUES('20','0.000')

teliki wra: 09:06:49
----- OFF Lamp-----
o3: 1.500 float: 1.5
----- OFF Fum-----
INSERT into TB VALUES('17','1.500')

teliki wra: 09:06:49
----- OFF Lamp-----
hum: 61.25 23.75 float: 42.5
----- ON Window-----
INSERT into TB VALUES('11','61.25 23.75')

teliki wra: 09:06:50
----- OFF Lamp-----
o3: 1.500 float: 1.5
----- OFF Fum-----
INSERT into TB VALUES('18','1.500')

```

Εικόνα 5.4 Επεξεργασία και εισαγωγή των δεδομένων στη Βάση Δεδομένων

Ακολούθως, οι τιμές μετατρέπονται σε εγγραφές και αποθηκεύονται στη Βάση Δεδομένων. Τέλος, έχει αναπτυχθεί ιστοσελίδα που μπορούν να επισκέπτονται όλοι οι χρήστες του Διαδικτύου. Εμπεριέχει πληροφορίες για τους αισθητήρες και το Raspberry PI. Παρουσιάζει τα δεδομένα σε διαγράμματα Google Charts[24] και Google Maps[23] σε πραγματικό χρόνο.

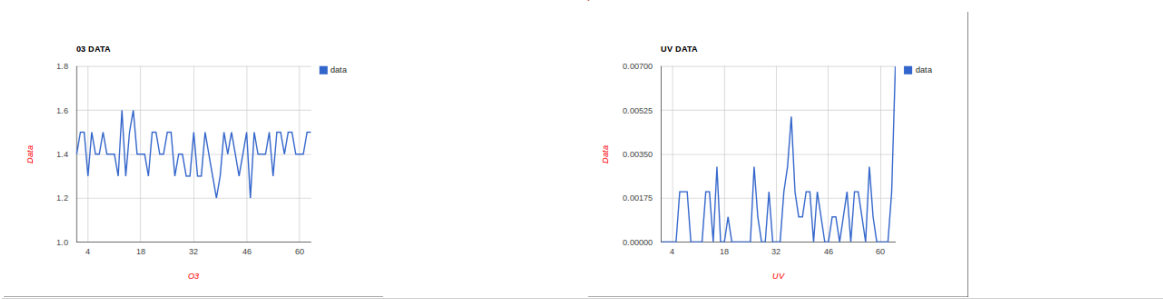
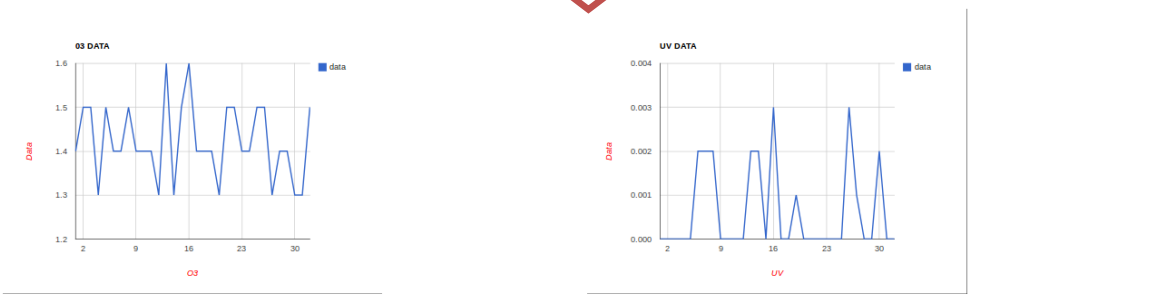
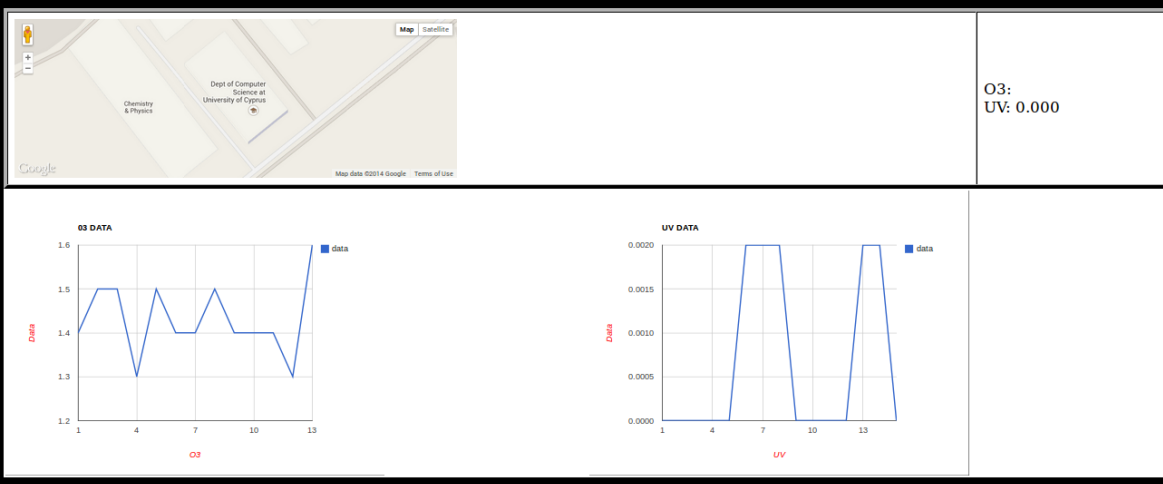


Εικόνα 5.5 Ιστοσελίδα

Εικόνα 5.6 Ιστοσελίδα

Feel The World

Ομάδα Εργασίας: Δρ. Γιώργος Πάλλης, Χαρίτων Ευσταθιάδης, Χρυσοβαλάντης Περιάνδρου



Εικόνα 5.7 Διαγράμματα παρουσίασης δεδομένων.

Η ιστοσελίδα αποτελείται από αρκετές σελίδες και υπερσυνδέσμους, βοηθώντας τον χρήστη να μετακινηθεί σε σημαντικές ιστοσελίδες, ή ενημέρωση του σχετικά με τα εργαλεία που χρησιμοποιούνται. Στην σελίδα data παρουσιάζονται σε πραγματικό χρόνο τα

δεδομένα που διαβάζονται από τους αισθητήρες. Χρησιμοποιώντας κώδικα html δημιουργείται η σελίδα. Ακολούθως μέσω κώδικα php δημιουργείται σύνδεση με την βάση δεδομένων, ανακτώντας τα στοιχεία που εμπεριέχονται σε αυτή. Στην συνέχεια, με τη χρήση javascript και ajax δημιουργούνται τα διαγράμματα και αναπαριστώνται τα δεδομένα.

Η αναπαράσταση των δεδομένα γίνεται για να μπορεί ο χρήστης να παρατηρεί την εξέλιξη των στοιχείων και να ενημερώνεται για τις αλλαγές που παρατηρούνται από τους αισθητήρες. Πολύ σημαντική είναι η απεικόνιση των δεδομένων σε πραγματικό χρόνο. Αυτό δίνει την ικανότητα στον αναλυτή να δημιουργήσει εφαρμογές, με την χρήση της πλατφόρμας, οι οποίες να ασχολούνται με επίλυση προβλημάτων, στα οποία χρειάζεται άμεση εκτέλεση κάποιας λειτουργίας σε περίπτωση που ο αισθητήρας δώσει τιμή πάνω από το κατώφλι. Για παράδειγμα, τέτοιες εφαρμογές είναι εφαρμογές που ασχολούνται με συναγερμούς για πυρασφάλεια ή προστασία από ληστεία. Εφαρμογές που ασχολούνται με αναπαράσταση δεδομένων σε πραγματικό χρόνο είναι και οι εφαρμογές που ασχολούνται με την οικονομία και την διακύμανση τις αξίας μετοχών ή τις ισοτιμίας νομισμάτων.

Η εφαρμογή Sense Me είναι πολύ σημαντική και χρήσιμη. Είναι σημαντική γιατί αποδεικνύει πως μπορούν να συλλέγονται και να παρουσιάζονται σε πραγματικό χρόνο τα δεδομένα από τους αισθητήρες, ενώ είναι χρήσιμη γιατί μπορεί να λύση αρκετά προβλήματα. Για παράδειγμα, μπορεί να παρουσιάζει την ακτινοβολία μιας περιοχής, εξάγοντας συμπεράσματα για την ηλιοφάνεια και την θερμοκρασία, προσελκύοντας εταιρείες, οι οποίες ασχολούνται με την ηλιακή ενέργεια, ή εταιρείες που ασχολούνται με τον τουρισμό.

5.1.2 Smart Home

Η εφαρμογή Smart Home έχει αναπτυχθεί με σκοπό την αξιολόγηση της πλατφόρμας ενδιάμεσου λογισμικού, αλλά και την πρακτική χρήση της πλατφόρμας σε πραγματικό πρόβλημα και την αυτοματοποίηση λειτουργιών ενός σπιτιού. Χρησιμοποιείται για την αναγνώριση του περιβάλλοντος ενός σπιτιού και την αυτοματοποίηση κάποιων λειτουργιών, όπως το άνοιγμα των παραθύρων και την αύξηση της έντασης της φωνής μιας τηλεόρασης. Αποτελείται από τέσσερα κομμάτια. Το πρώτο αφορά την συλλογή των δεδομένων, το δεύτερο την επεξεργασία των δεδομένων, το τρίτο την ενεργοποίηση ή απενεργοποίηση κάποιων λειτουργιών ανάλογα με τις ενδείξεις από τους αισθητήρες και το τέταρτο την αποθήκευση των δεδομένων σε Βάση Δεδομένων.

Κάποιες ενδεικτικές λειτουργίες που αναπτύχθηκαν είναι οι εξής:

- Άναμμα/Σβήσιμο λάμπας – ακτινοβολία
- Άνοιγμα/Κλείσιμο εξωτερικού παραθύρου – ώρα
- Άνοιγμα/Κλείσιμο παραθύρου – υγρασία
- Ενεργοποίηση/Απενεργοποίηση συναγερμού – όζον
- Ανέβασμα/Κατέβασμα έντασης της τηλεόρασης – θόρυβος

Λάμπα: ανάβει όταν η τιμή από τον αισθητήρα για την φωτεινότητα είναι κάτω από το κατώφλι, που σημαίνει ότι δεν είναι ικανοποιητικό το φως.

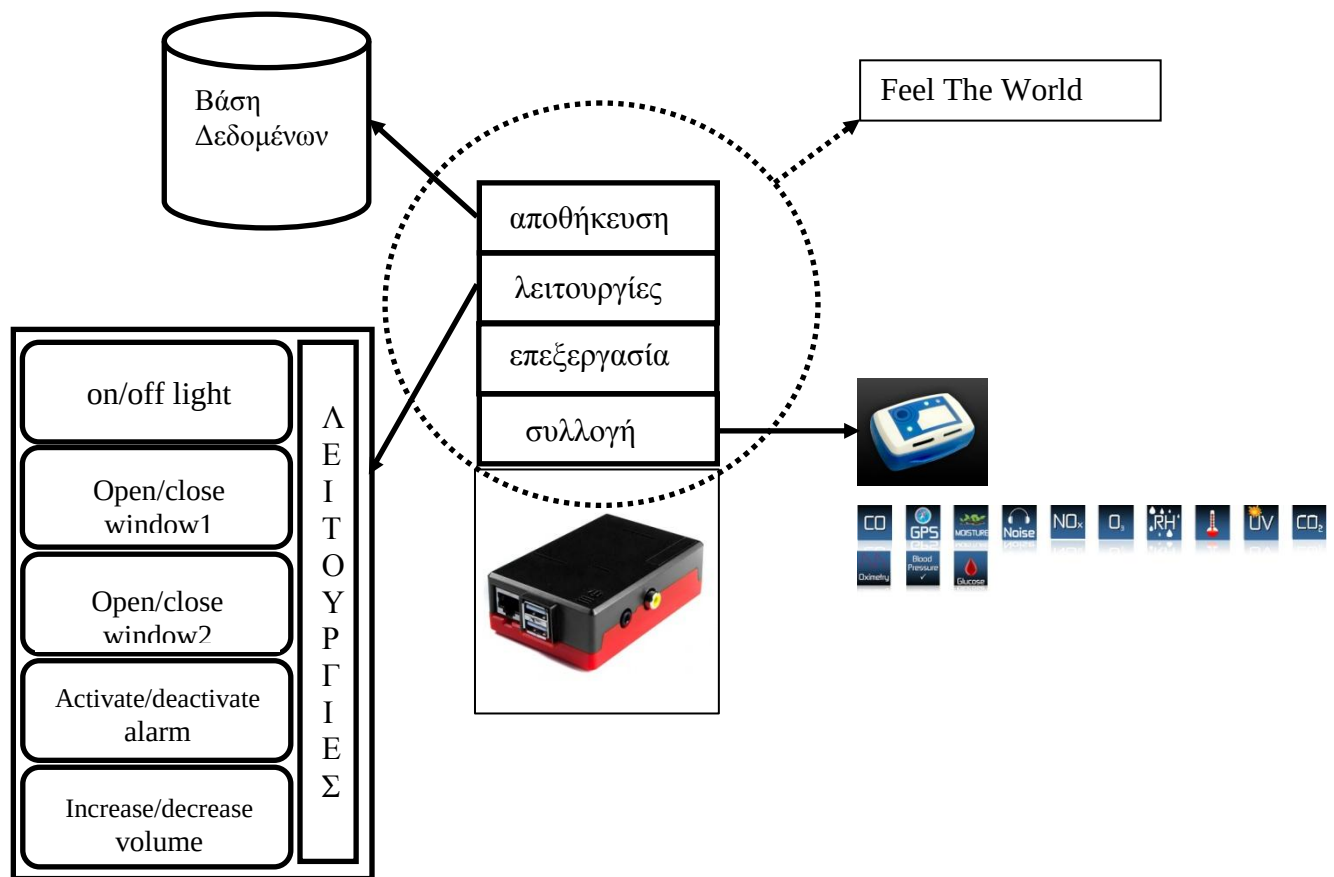
Εξωτερικό παράθυρο: ρυθμίζεται με την ώρα και είναι ανοικτό κατά τη μέρα και κλειστό κατά την νύχτα.

Παράθυρο: ανοίγει όταν η τιμή από τον αισθητήρα για την υγρασία είναι πιο πάνω από το κατώφλι, που σημαίνει ότι υπάρχει αρκετή υγρασία στο δωμάτιο.

Συναγερμός: ενεργοποιείται όταν η τιμή από τον αισθητήρα για το όζον είναι πιο πάνω από το κατώφλι, που σημαίνει ότι υπάρχει αυξημένη ποσότητα όζον στο δωμάτιο.

Ένταση τηλεόρασης: εφόσον υπάρχει θόρυβος σημαίνει δεν παρακολουθείται η τηλεόραση, άρα μειώνει την ένταση για να μην ενοχλεί. Όσο πιο μεγάλος ο θόρυβος τόσο πιο πολύ θα χαμηλώνει η ένταση.

Ακολουθεί η αρχιτεκτονική της εφαρμογής και ο τρόπος λειτουργίας της. Όπως φαίνεται και στην εικόνα 5.8, η εφαρμογή έχει αναπτυχθεί πάνω στην πλατφόρμα ενδιάμεσου λογισμικού.



Εικόνα 5.8 Αρχιτεκτονική Εφαρμογής

Η εφαρμογή εγκαθίσταται σε κάποια χαμηλού κόστους συσκευή με λειτουργικό σύστημα Linux, όπως για παράδειγμα το Raspberry PI, και ακολούθως την ενεργοποιούμε. Η

εφαρμογή θα δημιουργήσει ένωση με τους αισθητήρες που εμφανίζονται. Ακολούθως, θα αρχικοποιήσει την βάση δεδομένων και θα ξεκινήσει η συλλογή των δεδομένων. Θα εμφανίσει τις οθόνες που αναπαριστούν το σπίτι με τις λειτουργίες και θα αρχίσει να ελέγχει τις τιμές και να τις αναπαριστά. Τέλος, αποθηκεύει τις τιμές για να μπορεί στο μέλλον να εξάγει και συμπεράσματα.

Στην εικόνα 5.9 παρουσιάζεται η εύρεση των αισθητήρων και η δημιουργία σύνδεσης με το πρόγραμμα. Στη συνέχεια αρχικοποιείται η Βάση Δεδομένων και ξεκινά η ανάγνωση δεδομένων από τους αισθητήρες.

```
main_smart_house [Java Application] /usr/lib/jvm/java-7-openjdk-amd64/bin/java (9 May 2014 21:23:49)
00:07:80:9B:E8:C8      SENSP0D_0073
00:07:80:46:E0:C7      SENSP0D_0012
echo 985740| sudo -S rfcomm bind /dev/rfcomm3 00:07:80:46:E0:C7
connection ok
connection ok
connection ok
teliki wra: 18:24:27
----- OFF Lamp-----
uv: 0.002 float: 0.002
----- ON Lamp Garden-----
INSERT into TB VALUES('1','0.002')

teliki wra: 18:24:28
----- OFF Lamp-----
uv: 0.000 float: 0.0
----- ON Lamp Garden-----
INSERT into TB VALUES('2','0.000')
```

Εικόνα 5.9 Ένωση με τους αισθητήρες και αρχικοποίηση της Βάσης Δεδομένων

Στην εικόνα 5.10 παρουσιάζονται τα δεδομένα που διαβάζονται από τους αισθητήρες. Στη συνέχεια, τα δεδομένα που διαβάζονται από τους αισθητήρες επεξεργάζονται, έτσι ώστε να παραμείνει μόνο η τιμή του στοιχείου που παρακολουθούμε. Ακολούθως, οι τιμές μετατρέπονται σε εγγραφές και αποθηκεύονται στη Βάση Δεδομένων.

```
#090514:181735:242,$PSEN, 03,V,1.200
#090514:181735:539,$PSEN,Hum,H,51.29,T,25.15
#090514:181736:069,$GPRMC,181735.689,V,,,,,0.00,0.00,090514,,,N*4A
#090514:181736:210,$PSEN, UV,V,0.000
#090514:181736:215,$PSEN, 03,V,1.300
#090514:181736:516,$PSEN,Hum,H,51.29,T,25.15
#090514:181737:061,$GPRMC,181736.689,V,,,,,0.00,0.00,090514,,,N*49
#090514:181737:189,$PSEN, UV,V,0.002
#090514:181737:193,$PSEN, 03,V,1.400
#090514:181737:492,$PSEN,Hum,H,51.29,T,25.15
#090514:181738:069,$GPRMC,181737.689,V,,,,,0.00,0.00,090514,,,N*48
#090514:181738:163,$PSEN, UV,V,0.000
#090514:181738:168,$PSEN, 03,V,1.300
#090514:181738:469,$PSEN,Hum,H,51.29,T,25.14
#090514:181739:069,$GPRMC,181738.689,V,,,,,0.00,0.00,090514,,,N*47
#090514:181739:142,$PSEN, 03,V,1.500
#090514:181739:147,$PSEN, UV,V,0.002
#090514:181739:445,$PSEN,Hum,H,51.25,T,25.14
#090514:181740:061,$GPRMC,181739.689,V,,,,,0.00,0.00,090514,,,N*46
#090514:181740:115,$PSEN,Batt,V,3.77
#090514:181740:120,$PSEN, 03,V,1.500
#090514:181740:125,$PSEN, UV,V,0.002
#090514:181740:422,$PSEN,Hum,H,51.26,T,25.15
```

Εικόνα 5.10 Τα δεδομένα του αισθητήρα

```
main_smart_house [Java Application] /usr/lib/jvm/java-7-openjdk-amd64/bin/java (9 May 2014 21:25:46)
INSERT into TB VALUES('17','0.000')

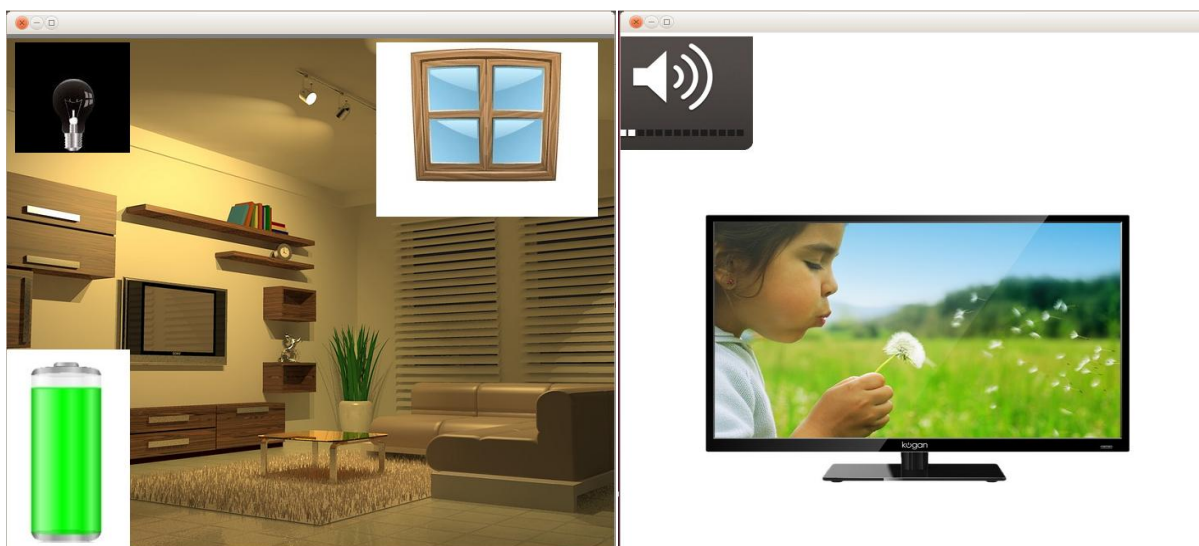
teliki wra: 18:26:46
----- OFF Lamp-----
hum: 51.33 25.47 float: 38.4
----- OFF Window-----
INSERT into TB VALUES('11','51.33 25.47')

teliki wra: 18:26:47
----- OFF Lamp-----
bat: 3.76 float: 3.76
----- battery full -----

teliki wra: 18:26:47
----- OFF Lamp-----
uv: 0.000 float: 0.0
----- ON Lamp Garden-----
INSERT into TB VALUES('18','0.000')
```

Εικόνα 5.11 Επεξεργασία και εισαγωγή των δεδομένων στη Βάση Δεδομένων

Στην εικόνα 5.11 μπορούμε να παρατηρήσουμε την εισαγωγή των δεδομένων στη βάση δεδομένων, όπως επίσης και την ενεργοποίηση κάποιας λειτουργίας. Τέλος, έχει αναπτυχθεί γραφική διεπαφή για την απεικόνιση των λειτουργιών του έξυπνου σπιτιού. Η ανάπτυξη έγινε στην Java με την χρήση της βιβλιοθήκης JFrame[25]. Πιο κάτω παρουσιάζονται οι διεπαφές και οι αλλαγές τους κατά την αλλαγή των τιμών των αισθητήρων.



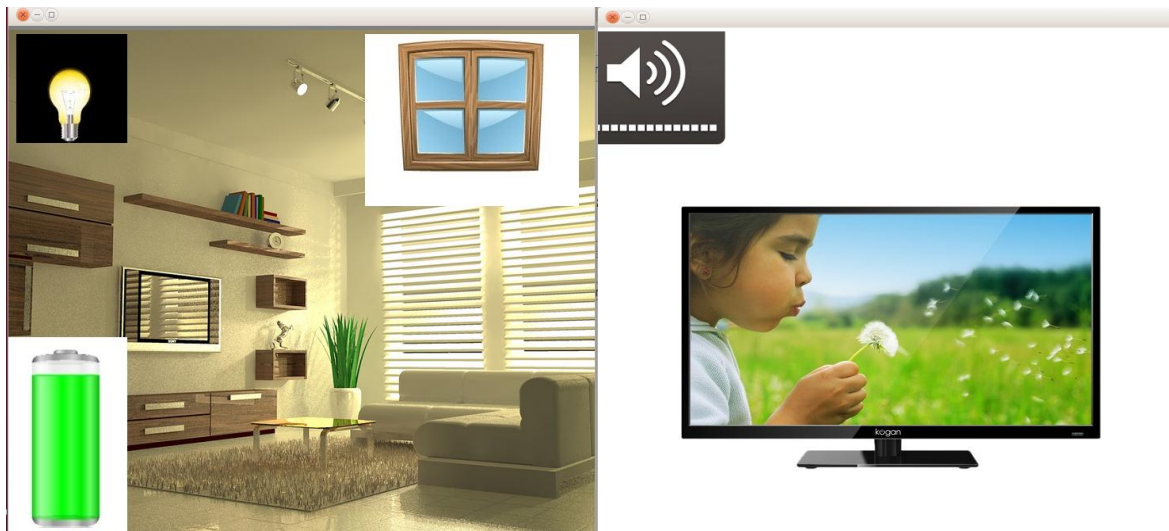
Εικόνα 5.12 Αρχικές διεπαφές

Στην εικόνα 5.12 απεικονίζονται οι αρχικές εικόνες, όταν όλες οι λειτουργίες είναι απενεργοποιημένες.



Εικόνα 5.13 Αλλαγή των τιμών και των διεπαφών

Σε αυτή την εικόνα παρουσιάζονται όλοι οι αισθητήρες ενεργοποιημένοι. Μπορούμε να παρατηρήσουμε την λάμπα που είναι αναμμένη και το παράθυρο που είναι ανοικτό. Αυτό γίνεται γιατί οι τιμές του αισθητήρα για την ακτινοβολία-φωτεινότητα μέσα στο δωμάτιο είναι πιο κάτω από το κατώφλι το οποίο έχουμε ορίσει, ενώ η τιμή για την υγρασία είναι πιο πάνω από το κατώφλι. Επίσης, εμφανίζεται το εικονίδιο για τον κίνδυνο για το λόγο ότι το ποσοστό του όζον στην ατμόσφαιρα έχει μεγαλώσει. Ακολούθως παρατηρούμε ότι το εξωτερικό παράθυρο είναι ανοικτό λόγω της ώρας. Δηλαδή αν είναι μέρα τότε ανοίγει για να φωτιστεί το δωμάτιο, ενώ αν είναι νύχτα κλείνει. Τέλος παρατηρούμε ότι η ένταση της τηλεόρασης έχει ανέβει μιας και δεν αφουγκράζεται ο αισθητήρας κάποια ομιλία.



Εικόνα 5.14 Αλλαγή των τιμών και των διεπαφών

Στην εικόνα 5.14 παρουσιάζονται κάποιες άλλες λειτουργίες όπως το κλειστό παράθυρο. Αυτό συμβαίνει για το λόγο ότι ο αισθητήρας διαβάζει τιμές για την υγρασία που είναι πιο μικρές από το κατώφλι το οποίο έχουμε ορίσει. Ακολούθως παρατηρούμε ότι το εικονίδιο για το όζον έχει εξαφανιστεί, αφού η τιμή του πλέον είναι φυσιολογική. Τέλος μπορούμε να παρατηρήσουμε ότι παραμένει ανοικτό το παράθυρο λόγω ώρας αλλά η φωτεινότητα στο δωμάτιο είναι χαμηλή με αποτέλεσμα να παραμένει αναμμένη η λάμπα. Όσο αφορά την τηλεόραση η ένταση της έχει ανέβει στο 100% γιατί δεν υπάρχει καθόλου θόρυβος.

5.2 Μεθοδολογία Αξιολόγησης

Η μεθοδολογία αξιολόγησης που ακολουθήθηκε είναι η πειραματική αξιολόγηση. Λόγω του ότι η πλατφόρμα μπορεί να αξιολογηθεί μόνο στη χρήση και στην εκμετάλλευση των κλάσεων και των μεθόδων που προσφέρει, κάναμε χρήση της βιβλιοθήκης για να αναπτύξουμε τις δύο πιο κάτω εφαρμογές:

1. senseme: εφαρμογή εγκατεστημένη σε Raspberry Pi, η οποία συλλέγει δεδομένα από αισθητήρες και τα παρουσιάζει σε ιστοσελίδα.
2. smart home: εφαρμογή εγκατεστημένη σε ηλεκτρονικό υπολογιστή που ελέγχει τις λειτουργίες ενός σπιτιού, με βάση τις μετρήσεις από τους αισθητήρες.

5.3 Μετρικές Αξιολόγησης

Για την αξιολόγηση της πλατφόρμας χρησιμοποιήθηκαν οι παρακάτω μετρικές:

1. Γραμμές κώδικα: πολύ σημαντική μετρική στην ανάπτυξη κάποιου προγράμματος είναι οι γραμμές κώδικα που χρειάζεται να γράψει ο προγραμματιστής. Για παράδειγμα αν χρειάζεται να γίνει δημιουργία και εισαγωγή μιας συμβολοσειράς σε βάση δεδομένων χρειάζονται οι εξής γραμμές:

Χωρίς τη χρήση της πλατφόρμας:

```
Class.forName("com.mysql.jdbc.Driver");
con = DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "cr7");
st = con.createStatement();
boolean a = false;
boolean b = false;
try {
    a = st.execute("CREATE DATABASE " + "hum");
} catch (Exception e) {}
if (a == true) {
    return;}
try {
    b = st.execute("CREATE TABLE TB ( id INTEGER not NULL, data VARCHAR(300))");
```

```

} catch (Exception e) {}
if (a == true) {
    return;}
String s = "INSERT into TB VALUES(" + id + "," + line+ ")" ";
int a = st.executeUpdate(s);
if (a == 0) {
    return;}
con.close();

```

Γραμμές κώδικα: 20

Χρήση της πλατφόρμας:

```

mydb dbh = new mydb("hum");
String line;
Int as=1;
dbprod data1 = new dbprod(as, line);
dbh.adddata(data1, "hum", as);

```

Γραμμές κώδικα: 5

Γενικά χρειάστηκαν 200 γραμμές για την δημιουργία του smart home και 115 γραμμές για το senseme με τη χρήση της πλατφόρμας, που σε συνδυασμό με το πιο πάνω απλό παράδειγμα παρουσιάζεται η απλότητα και η ευκολία στην δημιουργία μιας εφαρμογής με τη χρήση της πλατφόρμας. Για την δημιουργία των πιο πάνω εφαρμογών χωρίς την χρήση της πλατφόρμας χρειάστηκαν 940 και 680 γραμμές αντίστοιχα.

2. Παράλληλη εκτέλεση: ένα μεγάλο πλεονέκτημα της πλατφόρμας είναι η παράλληλη εκτέλεση, δηλαδή η χρήση ταυτόχρονα πολλών αισθητήρων. Η πλατφόρμα δίνει την δυνατότητα χρήσης της από πολλούς αισθητήρες ταυτόχρονα, έτσι μπορεί να καλύψει πολλά είδη εφαρμογών. Χωρίς την χρήση της πλατφόρμας η παράλληλη εκτέλεση μπορεί να υλοποιηθεί μόνο με την εκτέλεση εντολών στη γραμμή εντολών του terminal πράγμα αδύνατο κατά την εκτέλεση της εφαρμογής στο Raspberry PI. Ένα πλεονέκτημα που προσδίδει η παράλληλη εκτέλεση είναι και ο όγκος δεδομένων, αφού μπορούμε να παραλάβουμε και να επεξεργαστούμε πολύ περισσότερα δεδομένα από τους αισθητήρες. Έτσι, δεν υπάρχει χρόνος αδράνειας μεταξύ των μηνυμάτων από τους

αισθητήρες και έχουμε καλύτερη απεικόνιση των δεδομένων. Ο κάθε αισθητήρας στέλνει μία τιμή float κάθε δευτερόλεπτο. Στην περίπτωση που δεν έχουμε την πλατφόρμα ο όγκος των δεδομένων ανά λεπτό θα ήταν $60 * 8 \text{ bytes} = 480 \text{ bytes} / \text{s}$. Με την χρήση της πλατφόρμας έχουμε $480 \text{ bytes} / \text{s}$ από κάθε αισθητήρα. Για παράδειγμα στην περίπτωση του Smart home, στο οποίο λαμβάνουμε δεδομένα από δύο αισθητήρες, θα είχαμε $960 \text{ bytes} / \text{s}$, δηλαδή το διπλάσιο όγκο δεδομένων.

3. Μνήμη: η μνήμη του Raspberry PI καθορίζεται από την εξωτερική μνήμη που θα προσθέσουμε. Ανεξάρτητα με την εξωτερική μνήμη υπάρχουν δύο βασικοί τρόποι για αποθήκευση. Αυτοί είναι σε αρχείο κειμένου και σε βάση δεδομένων. Η υλοποίηση της πλατφόρμας έχει την ικανότητα και για τα δύο. Πολύ πιο οικονομική για την μνήμη είναι η βάση δεδομένων. Για την υλοποίηση της βάσης χρησιμοποιήθηκε μόνο ένας πίνακας με δύο στήλες, μία για την ταυτότητα και μια για τα δεδομένα. Οι τύποι που χρησιμοποιήθηκαν είναι ακέραιος για την ταυτότητα και συμβολοσειρά για τα δεδομένα. Έτσι, χρησιμοποιήθηκαν μόνο τα απαραίτητα για την δημιουργία και την χρήση της βάσης δεδομένων, πράγμα που καθιστά την βάση δεδομένων βέλτιστη. Για την επιλογή της συμβολοσειράς αντί του δεκαδικού αριθμού για τα δεδομένα, λάβαμε υπόψη ότι ένας δεκαδικός αριθμός τριών δεκαδικών ψηφίων αποτελείται από 8 bytes ενώ μια συμβολοσειρά από 5 bytes .
4. Πραγματικός χρόνος απεικόνισης δεδομένων: οι σύγχρονες εφαρμογές που αφορούν το Διαδίκτυο των Πραγμάτων, έχουν σαν κύρια απαίτηση την απεικόνιση των δεδομένων σε πραγματικό χρόνο. Ένα από τα κύρια συστατικά της πλατφόρμας είναι και η ανάγνωση των δεδομένων σε πραγματικό χρόνο. Τα δεδομένα αυτά παίρνονται από Bluetooth. Αυτή η τεχνολογία δεν είναι πολύ προηγμένη στις συσκευές που τρέχουν Linux, σε αντίθεση με τις συσκευές Android που είναι κατασκευασμένες να έχουν αλληλεπίδραση σε πραγματικό χρόνο με Bluetooth. Αυτό έχει σαν αποτέλεσμα να δημιουργείται πρόβλημα. Μέσω των κλάσεων της πλατφόρμας γίνεται η ένωση και η αλληλεπίδραση με τους αισθητήρες σε πραγματικό χρόνο. Αυτό επιτυγχάνεται με εντολές μέσα από την Java που εκτελούνται στο terminal σαν υπερχρήστης.

Χωρίς την πλατφόρμα, η πιο πιθανή χρήση των αισθητήρων θα γινόταν σε πακέτα δεδομένων, δηλαδή θα γινόταν η σύνδεση, θα διαβάζονταν δεδομένα για κάποιο χρονικό διάστημα και ακολούθως θα τερματιζόταν η σύνδεση και θα διαβάζονταν τα δεδομένα με κάποια καθυστέρηση. Στη συνέχεια, αυτή η διαδικασία συνεχίζεται.

Τα πλεονεκτήματα που προσφέρει η πραγματική απεικόνιση των δεδομένων είναι:

- Εξοικονόμηση χρόνου και μπαταρίας αφού απαιτείται μόνο μία σύνδεση. Κάθε σύνδεση απαιτεί 10-15 δευτερόλεπτα, ενώ απαιτεί και περίπου το 0.3% από την μπαταρία του ηλεκτρονικού υπολογιστή. Άρα, για πεντάλεπτα πακέτα θα χρειαζόμασταν 2-3 λεπτά σε δημιουργία ένωσης και 3 - 4.5% μπαταρία για μία ώρα συλλογής δεδομένων
(Χρόνος δημιουργίας ένωσης: Εκτέλεση 1: 11s, εκτέλεση 2: 9s, εκτέλεση 3: 10s, εκτέλεση 4: 10s, εκτέλεση 5: 14s, εκτέλεση 6: 14s, εκτέλεση 7: 15s, εκτέλεση 8: 15s, εκτέλεση 9: 12s, εκτέλεση 10: 11s.)
- Χρήση των δεδομένων για άμεσες αλλαγές στην κατάσταση των πραγμάτων, δηλαδή ενεργοποίηση συναγερμού
- Άμεση παρακολούθηση μέσω διαγραμμάτων, στοιχείων που διαβάζουν οι αισθητήρες

Λειτουργία για μία ώρα		
	Χρήση Πλατφόρμας	Χωρίς τη χρήση πλατφόρμας (5λεπτα πακέτα)
Αριθμός συνδέσεων	1	12
Χρόνος για συνδέσεις	10 - 15 sec	2 – 3 min
Απαιτήσεις μπαταρίας	0.3 %	3 – 4.5 %

Συνολική Αξιολόγηση		
	Χρήση Πλατφόρμης	Χωρίς τη χρήση πλατφόρμης (5λεπτα πακέτου)
Γραμμές κώδικα Sense Me	115	680
Γραμμές κώδικα Smart Home	200	940
Όγκος Δεδομένων Smart Home (bytes/s)	960	480
Τρόποι Αποθήκευσης	Βάση Δεδομένων, Αρχείο	Αρχείο
Κόστος ανά έγγραφη (bytes)	9	12
Αριθμός συνδέσεων	1	12
Χρόνος για συνδέσεις	10 - 15 sec	2 – 3 min
Απαιτήσεις μπαταρίας	0.3 %	3 – 4.5 %

Κεφάλαιο 6

Συμπεράσματα – Μελλοντική Εργασία

6.1 Συμπεράσματα	58
6.2 Μελλοντική εργασία	59

Στο κεφάλαιο αυτό παρουσιάζονται χρήσιμα συμπεράσματα, τα οποία έχουν εξαχθεί από την ολοκλήρωση της παρούσας ατομικής διπλωματικής εργασίας. Ακολούθως, παραθέτονται ιδέες και προτάσεις σχετικά με μελλοντικές επεκτάσεις και αλλαγές σε σχέση με την πλατφόρμα ενδιάμεσου λογισμικού Feel The World Linux, που μπορούν να υλοποιηθούν σε μελλοντικές εργασίες.

6.1 Συμπεράσματα

Το διαδίκτυο των πραγμάτων είναι κάτι το καινούργιο στις μέρες μας, αλλά τείνει να καθιερωθεί και να αλλάξει το τρόπο λειτουργίας του διαδικτύου, των ηλεκτρονικών υπολογιστών και γενικά όλων των ηλεκτρονικών συσκευών, αλλά και το τρόπο ζωής. Με την χρήση αισθητήρων κάθε στιγμή. Με την διασύνδεση των πάντων με το διαδίκτυο. Με την συνεχείς αύξηση των έξυπνων συσκευών. Με την συνεχή εξέλιξη της τεχνολογίας στο τομέα των επεξεργαστών, της μνήμης και τη μείωση του μεγέθους των ηλεκτρονικών εξαρτημάτων, γίνεται απαραίτητη η ανάπτυξη εφαρμογών που να καλύπτουν τις καινούργιες ανάγκες των χρηστών. Αυτός ήταν και ο λόγος της δημιουργίας της πλατφόρμας ενδιάμεσου λογισμικού, ούτως ώστε η δουλειά του προγραμματιστή να γίνεται πιο εύκολη και πιο σύντομη. Ο κύριος στόχος της πλατφόρμας ήταν να απευθύνεται σε προγραμματιστές, που μπορεί και να μην έχουν μεγάλη προγραμματιστική εμπειρία, για να τους διευκολύνει στην ανάπτυξη εφαρμογής που χειρίζεται δεδομένα από αισθητήρες. Πολύ σημαντική ήταν και η επιλογή χαμηλού κόστους συσκευές, όπως επίσης και για την επιλογή ενός εκ των δωρεάν λειτουργικών συστημάτων.

Αρχικά, έγινε μελέτη της πλατφόρμας Android, κατανοώντας το τρόπο δημιουργίας και λειτουργίας των εφαρμογών στα έξυπνα τηλέφωνα. Ακολούθως, έγινε εξονυχιστική έρευνα και μελέτη της πλατφόρμας ενδιάμεσου λογισμικού Feel The World, η οποία είχε κατασκευαστεί σε προηγούμενη διπλωματική εργασία. Η πλατφόρμα είναι αναπτυγμένη σε λειτουργικό σύστημα Android. Στη συνέχεια, έγινε βελτίωση της πλατφόρμας με δύο επιπλέον πακέτα. Το πρώτο αφορούσε την δημιουργία, διαχείριση και έλεγχο βάσης δεδομένων, ενώ το δεύτερο αφορούσε τις προτεραιότητες των δεδομένων και των αισθητήρων.

Ακολούθησε η μετατροπή της πλατφόρμας από Android σε Linux. Αφού έγιναν οι απαραίτητες αλλαγές και προσθήκες, ακολούθησε η αξιολόγηση της. Για το λόγο αυτό, αναπτύχθηκαν δύο εφαρμογές. Η μία αφορά την συλλογή δεδομένων από τους αισθητήρες και τη παρουσίαση τους σε διαγράμματα σε ιστοσελίδα. Η δεύτερη αποτελεί την ψηφιακή υλοποίηση ενός έξυπνου σπιτιού και την αυτοματοποίηση κάποιων λειτουργιών του.

Η δημιουργία της πλατφόρμας αποτελεί μια πρωτοποριακή κίνηση αφού δεν υπάρχει κάτι παρόμοιο στην αγορά. Με αυτή την κίνηση προσδοκούμε στην εισαγωγή στα δίκτυα αισθητήρων, συσκευές χαμηλού κόστους, όπως και το Raspberry PI. Επιπλέον, η πλατφόρμα μπορεί να χρησιμοποιηθεί για εφαρμογές σε συσκευές χαμηλού κόστους, που δεν μπορούσαν να γίνουν μέχρι σήμερα λόγω τεχνικών δυσκολιών. Δυσκολιών που επιλύονται με την χρήση της πλατφόρμας. Τέτοιες εφαρμογές είναι συστήματα συναγερμού, συστήματα ελέγχου δωματίων νοσοκομείου, πραγματικού χρόνου παρουσιάσεις δεδομένων και λείψεις αποφάσεων και άλλα.

Η εκπόνηση αυτής της διπλωματικής εργασίας μου έχει προσφέρει αρκετές εμπειρίες και εφόδια για την υπόλοιπη μου ζωή. Ο καλός προγραμματισμός και η τυπικότητα στις παραδόσεις είναι ένα όπλο που θα χρειαστώ και στα επόμενα χρόνια, είτε στο μεταπτυχιακό πρόγραμμα σπουδών είτε στην δουλειά που θα εργάζομαι. Για να ολοκληρωθεί η εργασία, έχω έρθει σε επαφή με αρκετά θέματα, τα οποία μου ήταν άγνωστα, ενώ τα περισσότερα αποτελούν καινούργια θέματα που απασχολούν αυτή τη στιγμή τα μεγάλα ερευνητικά κέντρα του εξωτερικού. Έχω έρθει σε επαφή με έννοιες, όπως το έξυπνο σπίτι και το Διαδίκτυο των Πραγμάτων, το οποίο αποτελεί την εξέλιξη του Διαδικτύου, η οποία έγινε λόγω του τρόπου χρήσης του. Έχω επεξεργαστεί τα έξυπνα τηλέφωνα, μία συσκευή που έχει κατακλύσει την αγορά και έχει αλλάξει το τρόπο ζωής μας. Μαθαίνοντας το Android API, μπορώ να δημιουργήσω εφαρμογές για κινητές συσκευές. Επιπλέον, έχω έρθει σε επαφή με εργαλεία όπως αισθητήρες και το Raspberry PI. Έχω δημιουργήσει ιστοσελίδες, κάνοντας χρήση εργαλείων όπως το Google Charts και το Google Maps. Τέλος, πολύ σημαντικές ήταν οι γνώσεις που απόκτησα στον προγραμματισμό, αφού έχω κάνει εκτεταμένη χρήση του εργαλείου Eclipse και της γλώσσας προγραμματισμού Java.

6.2 Μελλοντική εργασία

Σε αυτή τη διπλωματική εργασία παρουσιάζουμε μια πλήρως λειτουργήσιμη πλατφόρμα ενδιάμεσου λογισμικού. Πετυχαίνοντας όλους τους αρχικούς μας στόχους, αλλά και την

ενσωμάτωση αρκετών ιδεών που προέκυψαν κατά την διαδικασία υλοποίησης, έχουμε ένα εργαλείο στην υλοποίηση εφαρμογών. Ένα εργαλείο που βοηθά τον προγραμματιστή να δημιουργήσει μια εφαρμογή που χειρίζεται αισθητήρες, σε πολύ μικρό χρονικό διάστημα, με πολύ λίγες γραμμές κώδικα, αλλά και με πολύ λίγες προγραμματιστικές γνώσεις.

Παρόλα αυτά, μπορούν να γίνουν πολλές προσθήκες και βελτιστοποιήσεις στη πλατφόρμα. Αν και ο καλύτερος τρόπος να αναγνωριστούν αδυναμίες και ελλείψεις της πλατφόρμας θα ήταν να χρησιμοποιηθεί για την ανάπτυξη αρκετών εφαρμογών, εντούτοις, με την ανάπτυξη των δύο εφαρμογών για την αξιολόγηση, μπορέσαμε να έχουμε μια πρώτη γεύση από τις δυνατότητες της πλατφόρμας, αλλά και των αδυναμιών της.

Μια πολύ σημαντική προσθήκη θα ήταν ένα πακέτο με κλάσεις και μεθόδους που αφορούν κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων. Τα δεδομένα που διαβάζονται από τους αισθητήρες αποτελούν ευαίσθητα προσωπικά δεδομένα των χρηστών. Δεδομένα τα οποία μας δίνουν λεπτομέρειες σε σχέση με το σπίτι ή το τόπο διαμονής του χρήστη, ενώ μπορεί να χρειαστεί να τα επεξεργαστούμε και να τα στείλουμε σε κάποιο εξυπηρετητή. Έτσι, θα ήταν πολύ καλύτερα αν τα δεδομένα κρυπτογραφούνταν, ούτως ώστε ο χρήστης να νιώθει ασφάλεια.

Τέλος, σε μεταγενέστερο στάδιο, θα μπορούσε να γίνει η προσπάθεια να μπορεί η πλατφόρμα να χειρίζεται και δεδομένα από άλλους αισθητήρες πέραν της εταιρείας Sensaris, κυρίως με χαμηλότερη τιμή. Με επιπλέον, κλάση για την ένωση και την επεξεργασία θα ήταν μια προσθήκη που θα έκανε ακόμα πιο προσιτή την πλατφόρμα στους κατασκευαστές εφαρμογών.

Βιβλιογραφία

- [1] Έξυπνα τηλέφωνα, <http://en.wikipedia.org/wiki/Smartphone>
- [2] Internet of Things, http://en.wikipedia.org/wiki/Internet_of_Things
- [3] Kevin Ashton, http://en.wikipedia.org/wiki/Kevin_Ashton,
<http://www.rfidjournal.com/articles/view?4986>, <https://www.rfidjournal.com/purchase-access?type=Article&id=169&r=%2Farticles%2Fview%3F169%2F2>
- [4] Auto-ID Labs, http://en.wikipedia.org/wiki/Auto-ID_Labs
- [5] Linux, <http://www.ubuntu.com/>
- [6] Raspberry PI, <http://www.raspberrypi.org/>,
http://en.wikipedia.org/wiki/Raspberry_Pi
- [7] Bluetooth, <http://www.bluetooth.com/Pages/Bluetooth-Home.aspx>,
<http://en.wikipedia.org/wiki/Bluetooth>
- [8] Wi-fi, <http://en.wikipedia.org/wiki/Wi-Fi>
- [9] M. Mun, P. Boda, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, and R. West, “PEIR, the personal environmental impact report, as a platform for participatory sensing systems research”, Proceedings of the 7th international conference on Mobile systems, applications, and services - Mobisys '09, Wroclaw, Poland: 2009, p. 55.
- [10] Ορισμός ΔτΠ, <http://www.eoc.org.cy/index.php?id=3598>

- [11] In the real world, things matter more than ideas. By Kevin Ashton,
<http://www.rfidjournal.com/articles/view?4986>,
<http://www.rfidjournal.com/articles/view?4986#sthash.7QrjnmYq.dpuf>
- [12] Smart home, http://en.wikipedia.org/wiki/Home_automation
- [13] Google Android, [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [14] SQLite, <http://www.sqlite.org/lang.html>,
<http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html>
- [15] Sensaris, <http://www.sensaris.com/>
- [16] Eclipse, <http://www.eclipse.org/>
- [17] API, http://en.wikipedia.org/wiki/Application_programming_interface
- [18] Global Positioning System, http://en.wikipedia.org/wiki/Global_Positioning_System
- [19] MySQL, <http://www.mysql.com/>,
http://www.w3schools.com/php/php_mysql_intro.asp
- [20] Ubuntu terminal, <https://help.ubuntu.com/community/UsingTheTerminal>
- [21] Java, <http://www.java.com/en/>, <http://docs.oracle.com/javase/7/docs/api/>
- [22] Andoid SDK, Android SDK, <http://developer.android.com/sdk/index.html>
- [23] Google Maps, <https://developers.google.com/maps/documentation/android/?hl=EL>,
<https://developers.google.com/maps/documentation/javascript/tutorial?hl=EL>

- [24] Google Charts, <https://developers.google.com/chart/interactive/docs/reference>
- [25] JFrame, <http://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html>
- [26] R. Ballagas, F. Memon, R. Reiners, and J. Borchers, “iStuff mobile”, Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07, San Jose, California, USA: 2007, p. 1107.
- [27] H. Lu, J. Yang, Z. Liu, N.D. Lane, T. Choudhury, and A.T. Campbell, “The Jigsaw continuous sensing engine for mobile phone applications”, Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems - SenSys '10, Zurich, Switzerland: 2010, p. 71.
- [28] P. Mohan, V.N. Padmanabhan, and R. Ramjee, “Nericell”, Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08, Raleigh, NC, USA: 2008, p. 323.
- [29] Campaignr, <http://urban.cens.ucla.edu/technology/campaignr>
- [30] Sympian OS, <http://en.wikipedia.org/wiki/Symbian>
- [31] Mac OS, http://en.wikipedia.org/wiki/Mac_OS
- [32] PDA, http://en.wikipedia.org/wiki/Personal_digital_assistant
- [33] Theophilos Phokas, Hariton Efstathiades, George Pallis, Marios D. Dikaiakos. Feel The World: A Mobile Framework for Participatory Sensing. 10th International Conference on Mobile Web Information Systems (MobiWIS 2013), August 2013, Paphos, Cyprus

Παράρτημα Α

Στο παράρτημα αυτό επισυνάπτεται το API (Application Programming Interface) της πλατφόρμας ενδιάμεσου λογισμικού Feel The World Linux. Με την χρήση αυτού του API ο προγραμματιστής μπορεί να κάνει χρήση της πλατφόρμας.

mydb

Constructor Summary	
mydb (java.lang.String name)	
Method Summary	
void	adddata (dbprod <Type> data, java.lang.String name, int id) Add data to database
dbprod	findProduct (java.lang.String data, java.lang.String name) Find data from database
dbprod	findProduct (int id, java.lang.String name) Find id from database
void	db2txt (java.lang.String filename, java.lang.String name) Converts a database to a text file
void	txt2db (java.lang.String nameoffile) Converts a text file to a database
boolean	deleteProduct (java.lang.String data, java.lang.String name) Delete data from database
boolean	deleteProduct (int id, java.lang.String name) Delete data from database
void	resetdatabase (java.lang.String name) Reset database
void	deletedatabase (java.lang.String name) Delete database
Methods inherited from class java.lang.Object	
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

dbprod<type>

Constructor Summary	
	dbprod ()
	dbprod (int id, type data)
	dbprod (type data, int id)
	dbprod (type data)
Method Summary	
void	setID (int id) Set id to product
int	getID () Get id from product
void	setData (type data) Set data to product
type	getData () Get data from product

pcconsens

Method Summary	
void	pconwithsens () Connect pc with sensors

startmakefile

Method Summary	
void	startcon () Start the connection pc - sensors

finishmakefile

Method Summary	
void	finishcon () Stop the connection pc - sensors

TransferService

Method Summary	
static boolean	insertFile (java.lang.String filename, DataPriority priority) Insert file
int	Next file () Choose the next file
DataPriority	Reduce (DataPriority a) Reduce the level of priority
Methods inherited from class java.lang.Object	
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

DataFile

Constructor Summary	
DataFile (Boolean isSent, java.lang.String filename, DataPriority priority)	
Method Summary	
boolean	isSent () Return TRUE if is sent
void	setSent (boolean isSent) Set isSet
java.lang.String	getFilename () Return the name of the file
DataPriority	getDataPriority () Return the priority of the data
void	setDataPriority (DataPriority dataPriority) Set the data priority
void	setFilename (java.lang.String filename) Set the name of the file

TypeCaster

Method Summary	
Static java.lang.String	typeToString (int sensor_type) Converts type to String

Data <type>

Constructor Summary	
Data (type value, java.lang.String timestamp, int sensorType)	
Method Summary	
java.lang.String	toString () Converts data to String
int	getType () Return type