

Ατομική Διπλωματική Εργασία

**SMART HOME APPLICATION -
WunderBar Controller**

Κυριάκος Πέτρου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2016

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

SMART HOME APPLICATION -

WunderBar Controller

Κυριάκος Πέτρου

Επιβλέπων Καθηγητής

Βάσος Βασιλείου

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

Μάιος 2016

Ευχαριστίες

Θερμές ευχαριστίες εκφράζω στο Δρ. Βάσο Βασιλείου για την ευκαιρία που δόθηκε να ασχοληθώ και να μελετήσω ένα τόσο ενδιαφέρον project, καθώς και την υποστήριξη που είχα εκ μέρους του καθ' όλη την διάρκεια υλοποίησης της διπλωματικής μου εργασίας. Θερμές ευχαριστίες εκφράζω επίσης στην οικογένεια μου για την υπομονή και την στήριξη τους σε όλο το φάσμα των σπουδών μου.

Περίληψη

Το έξυπνο γραφείο αποτέλεσε το γενικό θέμα της παρούσας διπλωματικής εργασίας. Η πρόκληση της υλοποίησης αυτοματισμών επετεύχθη με την χρήση μίας κεντρικής εφαρμογής με την ονομασία "το έξυπνο σπίτι" και με τη βοήθεια υλικό-τεχνικού εξοπλισμού WunderBar και Arduino. Στα πλαίσια της διαδικασίας υλοποίησης χρησιμοποιήθηκαν η γλώσσες προγραμματισμού JAVA, C και SQL. Επίσης έγινε χρήση του Relayr cloud με σκοπό την επιτέλεσή του ως διαμεσολαβητή στην επικοινωνία μεταξύ του εξοπλισμού και της εφαρμογής.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή	1
	1.1. Εισαγωγή στην έννοια του "Smart Home"- "Εξυπνο Σπίτι"	
	1.2. Θεωρίες που μελετήθηκαν	
	1.3. Ανάλυση απαιτήσεων και στόχοι της εφαρμογής	
	1.4. Επισκόπηση της Διπλωματικής Εργασίας- Σενάριο "Εξυπνο Γραφείο"	
Κεφάλαιο 2	Περιγραφή Υλικού	
	2.1 WunderBar	
	2.1.1 Εταιρικό Υπόβαθρο-Relayr- Επιτυχίες	
	2.1.2 Γιατί WunderBar	
	2.1.3 Project που ήδη υπάρχουν	
	2.1.4 Πως ξεκινάς με το WunderBar	
	2.1.5 Πως χρησιμοποιήθηκε στην εργασία	
	2.1.6 Ανάλυση αισθητήρων	
	2.2 Arduino UNO	
	2.2.1 Εταιρικό Υπόβαθρο	
	2.2.2 Γιατί Arduino- ARDUINO UNO	
	2.2.3 Πώς ξεκινάς με το ARDUINO UNO	
	2.2.4 Πως χρησιμοποιήθηκε στην εργασία	
Κεφάλαιο 3	Γνωστικό Υπόβαθρο	
	3.1. Επεξήγηση της έννοιας CLOUD	
	3.2. Relayr CLOUD	
	3.3. Developer Dashboard	
	3.3.1. Τι είναι το Dashboard	
	3.3.2. Relayr Developer Dashboard	
	3.4. MQTT	
	3.4.1. MQTT protocol	
	3.4.2. MQTT subscribe client	
	3.4.3. MQTT publish client	
	3.5. Threads	

Κεφάλαιο 4 **Ανάλυση-Σχεδίαση του GUI**.....

- 4.1. Ανάλυση της υλοποίησης του σεναρίου
- 4.2. Επικοινωνία μεταξύ WunderBar και Cloud
- 4.3. Επικοινωνία μεταξύ Arduino και Cloud
- 4.4. Επικοινωνία μεταξύ Cloud και Client
- 4.5. Επικοινωνία μεταξύ Client και Gui
- 4.6. Επικοινωνία μεταξύ Client και Database
- 4.7. Επικοινωνία μεταξύ Gui και Arduino
- 4.8. Επικοινωνία μεταξύ Gui και Database

Κεφάλαιο 5 **Παρουσίαση Λειτουργιών του GUI**.....

- 5.1. Είσοδος στην εφαρμογή
 - 5.1.1. Επικοινωνία μεταξύ Gui και Database
- 5.2. Κεντρική Σελίδα
 - 5.2.1. Καρτέλες Πληροφοριών
 - 5.2.2. Menu
 - 5.2.2.1. Readings
 - 5.2.2.1.1. Επικοινωνία μεταξύ Gui και Database
 - 5.2.2.2. Publish
 - 5.2.2.2.1. Επικοινωνία μεταξύ Gui και Arduino
 - 5.2.2.3. Rules
 - 5.2.2.3.1. Επικοινωνία μεταξύ Gui και Arduino
 - 5.2.2.4. Charts
 - 5.2.2.4.1. Επικοινωνία μεταξύ Gui και Database
- 5.3. About

Κεφάλαιο 6 **Συζήτηση**.....

- 6.1. Μελλοντικές επεκτάσεις της Εφαρμογής
- 6.2. Βελτιώσεις

Βιβλιογραφία

Κεφάλαιο 1

Εισαγωγή

- 1.1. Εισαγωγή στην έννοια του "Smart Home"- "Έξυπνο Σπίτι"
 - 1.2. Θεωρίες που μελετήθηκαν
 - 1.3. Ανάλυση απαιτήσεων και στόχοι της εφαρμογής
 - 1.4. Επισκόπηση της Διπλωματικής Εργασίας- Σενάριο "Έξυπνο Γραφείο"
-

1.1. Εισαγωγή στην έννοια του "Smart Home"- "Έξυπνο Σπίτι"

Μια καινούργια έννοια στο κόσμο του διαδικτύου. Με τον όρο έξυπνο σπίτι εννοούμε τους διάφορους αυτοματισμούς που μπορούμε να υλοποιήσουμε σε ένα σπίτι. Με το πάτημα ενός κουμπιού να μπορείς να ελέγξεις μια ηλεκτρική συσκευή ή ακόμα και μια διαδικασία που θα αλλάξουν το περιβάλλον του σπιτιού. Το πιο συναρπαστικό κομμάτι όμως το συναντάμε στον έλεγχο και στο πάτημα του κουμπιού από μακριά. Με την βοήθεια αισθητήρων μπορείς να καταγράψεις μετρήσεις όπως η θερμοκρασία, ο φωτισμός, η ηχορύπανση, η υγρασία και πολλών άλλων παραμέτρων που εμπίπτουν σε ένα οικιακό περιβάλλον. Δημιουργώντας ένα εσωτερικό δίκτυο αισθητήρων παρέχεται η δυνατότητα στον χρήστη να επιβλέπει τις μετρήσεις αυτές και να ενεργεί κατάλληλα στον χώρο του ώστε να μειώσει να αυξήσει αλλά και να τις αλλάξει. Προχωρώντας ένα βήμα παραπέρα με την χρήση του διαδικτύου οι μετρήσεις μπορούν να παρουσιαστούν στον χρήστη ακόμη και όταν βρίσκεται εκτός του σπιτιού. Κάπου εκεί εισέρχεται και η έννοια έξυπνο σπίτι. Ο χρήστης έχει την δυνατότητα να κάνει όλες αυτές τις αλλαγές με το πάτημα ενός κουμπιού. Χρησιμοποιώντας κάποιο πρόγραμμα είτε στο κινητό του είτε στον υπολογιστή του μπορεί να αλλάξει κάποια συνθήκη στο περιβάλλον ή να δημιουργήσει μια καινούργια. Ωστόσο όσο προχωρά η τεχνολογία ο χρήστης πλέον θέλει να κάνει την διαδικασία μόνο μια φορά και να γίνονται όλα αυτόματα. Εκεί πλέον το έξυπνο σπίτι πρέπει να γίνει ακόμη πιο έξυπνο. Το πρόγραμμα παίρνει τις μετρήσεις από τους αισθητήρες και παρεμβαίνει από μόνο του στο περιβάλλον. Προϋπόθεση όμως είναι ο χρήστης να καθορίσει τις στιγμές που θέλει το πρόγραμμα να δράσει από μόνο του. Στην διπλωματική αυτή δημιουργήθηκε ένα πρόγραμμα το οποίο υλοποιεί και με τους δύο τρόπους το Smart Home. Ο χρήστης είτε δρα από μόνος του στο περιβάλλον είτε

καθορίζει την στιγμή που το πρόγραμμα θα δράσει μόνο του. Οι ηλεκτρικές συσκευές που λαμβάνουν τις δράσεις αυτές στον κόσμο των δικτύων ονομάζονται Internet Of Things (IOT). Άρα ένα έξυπνο σπίτι μπορούμε να το ονομάσουμε σαν ένα σύνολο από Internet Of Things. Για τους σκοπούς της διπλωματικής εργασίας η εφαρμογή που υλοποιήθηκε έχει να κάνει με το "έξυπνο γραφείο" .

1.2. Θεωρίες που μελετήθηκαν

Η διπλωματική εργασία υλοποιήθηκε σε δύο μέρη. Το πρώτο μέρος αφορούσε την μελέτη διαφόρων πεδίων στην περιοχή των Δικτύων αλλά και της Πληροφορικής γενικότερα. Έγινε μελέτη πρωτοκόλλων επικοινωνίας, μελετήθηκε η έννοια του Cloud και η χρησιμότητα του στην εφαρμογή όπως επίσης και η έννοια του Client και με ποιό τρόπο θα φανεί χρήσιμος ένας Client. Μετέπειτα έγινε μελέτη για το ποιές συσκευές αλλά και για το πώς θα χρησιμοποιηθούν ως IOT. Επίσης έγινε μελέτη στο κατά πόσο θα ήταν χρήσιμο να υλοποιηθεί μια βάση δεδομένων και πως θα επικοινωνούσε η βάση αυτή με την εφαρμογή. Με τα αποτελέσματα των μελετών φτιάχτηκε και το σενάριο μάς -το οποίο θα αναλυθεί πιο κάτω- αλλά και οι συσκευές που θα χρησιμοποιηθούν. Οι συσκευές που αποφασίστηκε να χρησιμοποιηθούν είναι το WunderBar της Relayr και το Arduino. Έτσι και χρειάστηκε να γίνει μελέτη στις συσκευές αυτές.

Για το WunderBar μελετήθηκε ο τρόπος λειτουργίας του αλλά και ο τρόπος επικοινωνίας μαζί του ώστε να πάρουμε τα δεδομένα που χρειαζόμαστε . Για το Arduino μελετήθηκε το πως θα μας είναι χρήσιμο και το πώς θα εγκαταστήσουμε επικοινωνία μαζί του. Τέλος βάση και του σεναρίου μελετήθηκε ο τρόπος επικοινωνίας μεταξύ του WunderBar και του Arduino. Οι μελέτες για τις συσκευές έγιναν κυρίως από τις ιστοσελίδες τους αφού υπάρχουν αρκετά παραδείγματα που βοηθούν στην κατανόηση όλων των πεδίων γύρω από αυτό που χρειαζόμαστε.

1.3. Ανάλυση απαιτήσεων και στόχοι της εφαρμογής

Όπως αναλύθηκε πιο πάνω η εφαρμογή έχει στόχο την αλληλεπίδραση του χρήστη με το γραφείο αλλά και την δράση της εφαρμογής -από μόνη της - στο έξυπνο γραφείο. Οι απαιτήσεις που μπορεί κάποιος να έχει από την εφαρμογή αρχικά έχουν να κάνουν με την ασφάλεια και την αξιοπιστία των δεδομένων και της εφαρμογής γενικότερα. Αυτά αποτελούν και τις χαμηλού επιπέδου απαιτήσεις. Μια εφαρμογή που θα έδινε πρόσβαση στον οποιοδήποτε θα έδινε και την δυνατότητα κακόβουλης ενέργειας. Αυτό θα έφερνε αντίθετα αποτελέσματα και η εφαρμογή δεν θα ήταν χρήσιμη. Μια ακόμη σημαντική απαίτηση του

χρήστη από την εφαρμογή είναι η αξιοπιστία της. Οι δράσεις που ο χρήστης κάνει μέσω της εφαρμογής πρέπει να εκτελούνται χωρίς προβλήματα και η εφαρμογή να ανταποκρίνεται οποιαδήποτε στιγμή την χρειαστεί ο χρήστης. Επιπρόσθετα υπάρχουν και οι απαιτήσεις υψηλού επιπέδου. Προβλήματα θα προκαλούνταν ακόμα και αν τα δεδομένα που έπαιρνε η εφαρμογή ήταν λανθασμένα ή δυσανάγνωστα. Επίσης ένας απλός χρήστης το μόνο που χρειάζεται είναι η μια εφαρμογή που είναι εύκολα κατανοητή στον τρόπο λειτουργίας της και να είναι χρήσιμη. Αν κάναμε ανάλυση των απαιτήσεων σε γενικές γραμμές θα έδειχνε ότι χρειαζόμαστε μια εφαρμογή που θα είναι λειτουργική αξιόπιστη και ασφαλής.

1.4. Επισκόπηση της Διπλωματικής Εργασίας- Σενάριο "Έξυπνο Γραφείο"

Η παρούσα διπλωματική εργασία ασχολήθηκε με την δημιουργία ενός σεναρίου που αφορά το έξυπνο γραφείο. Όπως αναλύθηκε και πιο πάνω για να κάνουμε ένα σπίτι ή ένα δωμάτιο ή ένα γραφείο έξυπνο χρειαζόμαστε μερικούς αισθητήρες και ηλεκτρικές εγκαταστάσεις ώστε να ενώσουμε τις ηλεκτρικές μας συσκευές στο διαδίκτυο. Επίσης χρειαζόμαστε μια εφαρμογή που θα επιτρέπει δράσεις του χρήστη πάνω στις ηλεκτρικές συσκευές είτε μέσω του ίδιου δικτύου είτε μέσω απομακρυσμένου δικτύου. Επιπρόσθετα η εφαρμογή μας πρέπει να δίνει δεδομένα στον χρήστη είτε αυτά προέρχονται από τους αισθητήρες είτε εξυπηρετούν κάποιο ερώτημα του χρήστη. Το σενάριο που θα αναλύσουμε στην παρούσα διπλωματική εργασία έχει να κάνει με ένα έξυπνο γραφείο. Στο σενάριο υπάρχουν οι απαραίτητοι αισθητήρες που θα παίρνουν μετρήσεις της θερμοκρασίας, της υγρασίας, της φωτεινότητας, μέτρηση για απόσταση αντικειμένου από τον αισθητήρα, μέτρηση για το χρώμα αντικειμένου που πλησίασε, μέτρηση επιτάχυνσης και θέσης που βρίσκεται ο αισθητήρας. Οι αισθητήρες θα λαμβάνουν τις μετρήσεις και θα τις στέλνουν στο Cloud από εκεί η εφαρμογή θα λαμβάνει τις μετρήσεις μέσω κάποιου Client. Η εφαρμογή έχει την δυνατότητα να εμφανίζει να τις διαχειρίζεται και να φυλάει τις μετρήσεις στην βάση δεδομένων. Ο χρήστης με την σειρά του όταν θέλει να δράσει στις ηλεκτρικές συσκευές στέλνει το κατάλληλο αίτημα μέσω της εφαρμογής και η εφαρμογή χρησιμοποιεί ένα άλλο Client ώστε να στείλει το μήνυμα στο Cloud και το ενημερώνει το Arduino. Το Arduino χρησιμοποιεί την πληροφορία που λαμβάνει και ενεργοποιεί τα λαμπάκια. Τα λαμπάκια για τον σκοπό της εργασίας συμβολίζουν τις ηλεκτρικές συσκευές του γραφείου. Οι αισθητήρες που χρησιμοποιήθηκαν ανήκουν στη πλατφόρμα WunderBar η οποία μαζί με το Cloud ανήκουν στην εταιρία Relayr. Παρακάτω θα γίνει εκτενής ανάλυση του WunderBar και του Arduino.

Κεφάλαιο 2

Περιγραφή Υλικού

2.1 WunderBar

2.1.1 Εταιρικό Υπόβαθρο-Relayr- Επιτυχίες

2.1.2 Γιατί WunderBar

2.1.3 Project που ήδη υπάρχουν

2.1.4 Ανάλυση αισθητήρων

2.1.5 Πώς ξεκινάς με το WunderBar

2.1.6 Πως χρησιμοποιήθηκε στην εργασία.

2.2 Arduino UNO

2.2.1 Εταιρικό Υπόβαθρο

2.2.2 Γιατί Arduino- ARDUINO UNO

2.2.3 Πώς ξεκινάς με το ARDUINO UNO

2.2.4 Πως χρησιμοποιήθηκε στην εργασία

2.1 WunderBar

2.1.1 Εταιρικό Υπόβαθρο-Relayr- Επιτυχίες

Η relayr ιδρύθηκε στο Βερολίνο το 2013 και ο σκοπός ίδρυσης της είναι να προσφέρει λύσεις με λογισμικά, υλικό και συμβουλευτικές υπηρεσίες σε εταιρίες που είναι έτοιμες να αντιμετωπίσουν την βιομηχανική ψηφιοποίηση. Μέσα στην δράση της ανήκει και το Wunderbar το οποίο έφερε και τα πρώτα βραβεία της εταιρίας . Η relayr κατακτά την πρώτη θέση στο Cisco IoT Challenge, τον Οκτώβριο του 2014. Τον Ιανουάριο του 2015 παίρνει το επόμενο βραβείο στο διαγωνισμό Focus Digital Star Award, και τον Μάρτιο του 2015 παίρνει το επόμενο βραβείο στο διαγωνισμό CeBIT CODE _n Awards. Τέλος ακολουθεί το βραβείο στο διαγωνισμό FINAKI IT Innovation Awards τον Ιούνιο του 2015 (Relayr, 2016).

2.1.2 Γιατί WunderBar

Μέσα από την πρωτοποριακή του σχεδίαση το WunderBar προσφέρει άμεση πρόσβαση στο κόσμο του "Internet of Things". Το WunderBar αποτελείται από μια κεντρική

μονάδα, τον αισθητήρα θερμοκρασίας-υγρασίας, αισθητήρα φωτός-ακρίβειας, αισθητήρα γυροσκοπίου - επιτάχυνσης και αισθητήρα φωνής. Επίσης υπάρχει και μία μονάδα επέκτασης. Η μονάδα επέκτασης δίνει την δυνατότητα στον χρήστη του WunderBar να προσθέσει σε αυτό περισσότερους αισθητήρες. Στην μονάδα αυτή θα μπορούσε να γίνει η σύνδεση μεταξύ του WunderBar και του Arduino. Ωστόσο το Cloud της Relayr προσφέρει την δυνατότητα στον χρήστη να προσθέσει όποιο αισθητήρα ή συσκευή χρειάζεται εύκολα και γρήγορα κάνοντας απλά Prototype τον αισθητήρα ή την συσκευή στο Cloud. Επιπρόσθετα η εταιρία δημιούργησε Web Socket με το οποίο μπορεί κάποιος να συνδεθεί στο cloud και στον κάθε αισθητήρα ξεχωριστά ώστε να παρακολουθήσει τις μετρήσεις του. Το Web Socket δουλεύει δημιουργώντας subscribe Client ο οποίος συνδέεται με τον αισθητήρα για να λαμβάνει τις μετρήσεις αυτές. Στην περίπτωση που συνδέσαμε κάποια άλλη συσκευή εκτός από αισθητήρα μέσω του Web Socket έχουμε την δυνατότητα να δημιουργήσουμε publish Client οποίος θα στέλνει τα μηνύματα στην συσκευή αυτή. Η δουλειά που έκανε η εταιρία στον τομέα αυτό λύνει τα χέρια του χρήστη με αποτέλεσμα να κάνει πιο ελκυστικό το προϊόν της και να δίνει στο χρήστη ένα έτοιμο δίκτυο αισθητήρων για την δημιουργία έξυπνων σπιτιών.

2.1.3. Project που ήδη υπάρχουν

A/A	Programmers	Title	Hardware and Software components:	WebSite	About
1.	Andrew Steinheister	Octoblu and Wunderbar : Baby Monitor that texts you updates	<ul style="list-style-type: none"> • Octoblu account, • Relayr WunderBar 	https://www.hackster.io/andrewsteinheister/octoblu-and-wunderbar-baby-monitor-that-texts-you-updates-a2cb6d?ref=platform&ref_id=525_trending_offset=6	By using Relay's Wunderbar on Octoblu's IoT platform, it was easy to create a simple baby monitor that texts you updates about your baby.
2.	• Interface,	GreenBox	• Grow lights,	http://www.hackster.io/	The idea of the

	<p>hardware design by Hugo</p> <ul style="list-style-type: none"> • Coding, hardware design by Morris • Hardware buildup by Florian 		<ul style="list-style-type: none"> • Pc Fan • Arduino Mega 2560 R3 • Submersible • Water pump, • WunderBar, • Wall socket, • AC power cord inl • AC pc power cabl • Hydroponic • drip system, <ul style="list-style-type: none"> • 1 channel • 5V relay 	<p>kster.io/ain-t-no-party-like-a-unix-party/greenbox</p>	<p>project was to create a box which can automatically grow all sorts of plants, with as less user interaction as possible, similar to http://www.clickandgrow.com/ The Greenbox is designed to be a closed box to have a better control over the whole growing procedure. ES P8266 to connect it to the web.</p>
--	---	--	--	---	--

2.1.4 Ανάλυση αισθητήρων

Παρακάτω γίνεται εκτενής παρουσίαση των αισθητήρων. Αναλύεται ο τρόπος λειτουργίας τους αλλά και οι μηχανισμοί που αποτελούν τον κάθε αισθητήρα.

Αισθητήρας Επιτάχυνσης: Στον αισθητήρα αυτό υπάρχει το τσιπ MPU-6500 το οποίο κατασκευάζει η InvenSense. Στο τσιπ αυτό συνδυάζεται ένας αισθητήρας επιτάχυνσης 3 αξόνων και ένα γυροσκόπιο 3 αξόνων. Ο αισθητήρας επιτάχυνσης είναι μια

ηλεκτρομηχανική συσκευή που χρησιμοποιείται για να μετρήσει την φυσική επιτάχυνση ("G- δύναμη") σε αντίθεση με την επιτάχυνση(το ποσοστό της μεταβολής της ταχύτητας).

. Η φυσική επιτάχυνση έχει να κάνει με την ελεύθερη πτώση. Το MPU-6500 ενσωματώνει ένα μικρο-ηλεκτρο-μηχανικό σύστημα (MEMS) 3 αξόνων επιταχυνσιόμετρο, το οποίο αποτελείται από 3 αισθητήρες μέτρησης της επιτάχυνσης στον X, Y και Z άξονα. Η επιτάχυνση μπορεί να είναι είτε θετική είτε αρνητική, ανάλογα με την κατεύθυνση της απόκλισης.

Ο αισθητήρας σαν έξοδο παρουσιάζει τα αποτελέσματα του στην μορφή των 16-bit, η οποία στη συνέχεια μετατρέπεται είτε από την κύρια μονάδα ή στην περίπτωση όπου γίνεται απευθείας σύνδεση με το Bluetooth LE (ένα σενάριο στο οποίο μια εφαρμογή είναι άμεσα συνδεδεμένη με τον αισθητήρα, χωρίς τη μεσολάβηση του relayr Cloud) σε μια εφαρμογή στο κινητό τηλέφωνο.

Ο αισθητήρας παρουσιάζει τα αποτελέσματα με κλίμακα από -2G έως + 2G, από -4G έως + 4G, $\pm 8 G$, και $\pm 16G$. Η προεπιλεγμένη κλίμακα σύμφωνα με την οποία η κύρια μονάδα μετατρέπει τις τιμές, είναι η κλίμακα $\pm 2g$.

Αισθητήρας Γυροσκοπίου: Ένα γυροσκόπιο μετρά τη γωνιακή ταχύτητα κίνησης, δηλαδή την ταχύτητα της μεταβολής της γωνίας ενός αντικειμένου - στον X,Y,Z άξονα.

Η έξοδος ενός γυροσκοπίου παρουσιάζεται σε μοίρες ανά δευτερόλεπτο ($^{\circ} / \text{sec}$).

Το γυροσκόπιο περιλαμβάνει ένα εύρος τιμών προγραμματιζόμενο από το χρήστη και αποτελείται από τα ακόλουθα εύρη: ± 250 , ± 500 , ± 1000 και $2000 \pm ^{\circ} / \text{sec}$. Ωστόσο είναι προεπιλεγόμενο το εύρος $\pm 250 ^{\circ} / \text{sec}$.

Αισθητήρας θερμοκρασίας-υγρασίας: Στον αισθητήρα αυτό ενσωματώνεται το chip HTU21D το οποίο κατασκευάζεται από την Measurement Specialties™. Το HTU21D είναι ένας ψηφιακός αισθητήρας σχετικής υγρασίας με θερμοκρασία εξόδου. Η απεικόνιση των μετρήσεων γίνεται από 11 μέχρι 14 bits ενώ η έξοδος που παράγει ο αισθητήρας αφορά την θερμοκρασία δωματίου. Το εύρος θερμοκρασιών που μπορεί να πιάσει ο αισθητήρας είναι μεταξύ -40 και +125 °C με τυπική ακρίβεια $\pm 0.3 ^{\circ} \text{C}$. Στην περίπτωση όμως που ο αισθητήρας βρίσκεται συνδεδεμένος ακόμη στην κεντρική μονάδα οι ενδείξεις του αισθητήρα θα είναι 2-3 °C πιο ψηλές. Ο αισθητήρας υγρασίας βασίζεται σε ένα πολυμερές το οποίο είναι απορροφητικό νερού και αλλάζει την ηλεκτρική αγωγιμότητα του κυκλώματος που είναι συνδεδεμένο με αποτέλεσμα να

μετρά την ποσότητα του νερού που απορροφά. Η υγρασία εξαρτάται από τη θερμοκρασία του περιβάλλοντος που μετράται. Η ακρίβεια της υγρασία είναι $\pm 2\%$ RH στους 25°C (μεταξύ 20% και 80% RH).

Αισθητήρας Φωτός-Χρώματος-Απόστασης: Στον αισθητήρα αυτό ενσωματώνεται το τσιπ TCS3771 IC το οποίο κατασκευάζεται από την AMS AG. Το IC ενσωματώνει τον αισθητήρα χρώματος σύμφωνα με το σύστημα RGB αλλά και τον ανιχνευτή απόστασης σε μια μόνο συσκευή. Επιπλέον, στον αισθητήρα υπάρχει ένα άσπρο LED που υποστηρίζει υπέρυθρες για εντοπισμό αντικειμένων. Όταν ένα αντικείμενο τοποθετείται σε μια μετρήσιμη απόσταση από τον αισθητήρα, ο αισθητήρας απόστασης ανιχνεύει τον στόχο και ενεργοποιεί το άσπρο LED το οποίο βρίσκεται εγκατεστημένο σ' αυτόν. Κατά τη διάρκεια αυτής της περιόδου, τα 4 κανάλια του αισθητήρα χρώματος - κόκκινο, πράσινο μπλε και άσπρο - θα ανιχνεύσουν την ποσότητα του φωτός που αντανακλάται από το αντικείμενο και θα μετατρέψουν το σήμα σε ψηφιακή μορφή. Διαφορετικά αντικείμενα αντανακλούν το φως σε διαφορετικά μήκη κύματος και σε διαφορετικές ποσότητες. Με αυτό τον τρόπο λειτουργεί και το ανθρώπινο μάτι για να αντιλαμβάνεται τα χρώματα. Κατά παρόμοιο τρόπο, ο αισθητήρας χρώματος παράγει ένα σύνολο αξιών οι οποίες στη συνέχεια μεταφράζονται από τα Android, iOS SDKs και το Developer Dashboard σε χρώμα σύμφωνα με το μοντέλο χρωμάτων RGB. Για την ανίχνευση απόστασης χρησιμοποιείται το εξωτερικό InfraRed LED ώστε να εκπέμπει φως, και στη συνέχεια να μετρηθεί από τον ολοκληρωμένο ανιχνευτή φωτός η ποσότητα του ανακλώμενου φωτός.

Η ποσότητα του φωτός που ανιχνεύεται από την επιφάνεια αντανάκλασης χρησιμοποιείται για τον προσδιορισμό απόστασης του αντικειμένου από τον αισθητήρα. Ένα εσωτερικό πρόγραμμα οδήγησης(driver) του LED μπορεί να ρυθμιστεί ώστε να παρέχει σταθερό ρεύμα 12,5 mA, 25 mA, 50 mA ή 100 mA. Όσο υψηλότερο είναι το παρών ρεύμα, τόσο μεγαλύτερο είναι το εύρος ανίχνευσης, φθάνοντας περίπου τα 50 εκατοστά ανίχνευσης.

Αισθητήρας Ήχου : Ο αισθητήρα αυτός μετρά τη μέση τιμή θορύβου του περιβάλλοντος. Βασίζεται στο τσιπ SPU0410HR5H (το οποίο φτιάχνεται από την Knowles) αλλά και στο μικρόφωνο με τεχνολογία MEMS το οποίο συσχετίζει αναλογικά και ψηφιακά κυκλώματα. Το μικρόφωνο MEMS μετατρέπει τις μεταβολές της πίεσης του αέρα σε χαμηλή τάση αναλογικού σήματος, το οποίο στη συνέχεια ενισχύεται περίπου 100 φορές και διοχετεύεται σε έναν μετατροπέα. Ο μετατροπέας με την σειρά

του μετατρέπει το αναλογικό σήμα στον μέσο όρο και παρέχει μία τιμή εξόδου RMS που μπορεί να μετρηθεί από τον μετατροπέα αναλογικού σε ψηφιακό σήμα. Ο αισθητήρας θορύβου είναι ρυθμισμένος να είναι πιο ευαίσθητος σε κοινές πηγές θορύβου, όπως ανθρώπινες φωνές, μουσική και τους ήχους της κυκλοφορίας. Η τιμή RMS που μπορεί να πάρει είναι ανάμεσα σε 0-1024 και αντιπροσωπεύει το μέσο επίπεδο του θορύβου κατά τη διάρκεια μιας σύντομης χρονικής περιόδου. Ακολούθως το chip θα διαβάσει κάθε διάστημα μέτρησης (το προεπιλεγμένο χρονικό διάστημα είναι ένα δευτερόλεπτο) ώστε να παρουσιάσει την ένταση του ήχου . Ωστόσο η έξοδος που παρουσιάζει ο αισθητήρας δεν είναι σε ντεσιμπέλ επειδή χρειάζονται πιο πολύπλοκα εργαλεία για την μέτρηση σε ντεσιμπέλ. Ο αισθητήρας μας παρέχει ουσιαστικά μία μέτρηση που αντιπροσωπεύει την αλλαγή στο επίπεδο θορύβου κατά τη διάρκεια μιας χρονικής περιόδου. Η μέτρηση που παρουσιάζεται στην έξοδο αυξάνεται καθώς αυξάνεται το επίπεδο θορύβου και το αντίστροφο.

Αισθητήρας Υπέρυθρων: Ο αισθητήρα αυτός βασίζεται στο τσιπ SFH4441 το οποίο κατασκευάζεται από την OSRAM GmbH και είναι ένας υψηλής απόδοσης και υψηλής ισχύος πομπός υπερύθρων. Το IR Led που έχει ελέγχεται από το Nordic Chip και περιλαμβάνει την εφαρμογή του πρωτοκόλλου κώδικα NEC. Αυτό το πρωτόκολλο χρησιμοποιείται από πολλούς για τηλεκατευθυνόμενη διασύνδεση ηλεκτρονικών συσκευών, όπως τηλεοράσεις, Συστήματα Ήχου και LED φώτα όπως το IR LED του αισθητήρα. Χρησιμοποιώντας τα relayr SDK, είναι δυνατό να ελεγχθεί μια ποικιλία οικιακών συσκευών. Σε μελλοντικές εκδόσεις σχεδιάζεται να εφαρμοστούν πρόσθετα πρωτόκολλα IR ώστε χρησιμοποιηθούν από αυτή την ενότητα.

Μονάδα Γέφυρας: Το τσιπ αυτό περιλαμβάνει 8 σημεία γενικής χρήσης εισόδου / εξόδου (GPIOs), καθώς και μια υποδοχή 4 ακίδων. Χρησιμοποιείται για επέκταση του WunderBar και είναι συμβατό με Grove συσκευές-αισθητήρες από την SeeedStudios. Στο τσιπ περιλαμβάνονται δύο GPIOs από 3.3V σε 5V που μπορούν να χρησιμοποιηθούν ως ψηφιακές εισοδοί ή έξοδοι, ως I2Cs ή ως UART υποδοχές. Το firmware στη μονάδα έχει ως στόχο να ελέγξει κάποιες από τις ψηφιακές μονάδες της οικογένειας Seeed Grove, όπως η Solid State Relay, ή το Vibration Motor με μια απλή εντολή από τα SDKs ή από το Developer Dashboard στην της relayr.

Κυρίως Μονάδα-Master Module: Η κύρια μονάδα είναι η ιθύνων νους πίσω από το WunderBar . Είναι το μόνο δομημένο στοιχείο που είναι ικανό να συνδεθεί με Wifi και

είναι υπεύθυνο για την παράδοση των δεδομένων από τους αισθητήρες προς το cloud και από το cloud στους αισθητήρες.

Η κύρια μονάδα αποτελείται από τρία κύρια μέρη, κάθε μέρος είναι υπεύθυνο για μια διαφορετική πτυχή της συνολικής λειτουργικότητας της:

Η μονάδα Gainspan: Ενσωματώνοντας το τσιπ GS1500m, η μονάδα Gainspan είναι υπεύθυνη για την σύνδεση στο WiFi. Ξεκινά την επικοινωνία με το cloud της relayr, χρησιμοποιώντας το πρωτόκολλο MQTT / SSL και είναι υπεύθυνη για τη σύνδεση με το μεσίτη του MQTT.

Η μονάδα Kinetis: Ενσωματώνοντας τον ARM Cortex M4, το Kinetis K Series, που κατασκευάζεται από την Freescale, αποτελεί τον πυρήνα του τσιπ για την κύρια μονάδα. Αποθηκεύει και διατηρεί όλα τα κλειδιά άδειας τα οποία χρειάζονται για την επικοινωνία με το cloud και την επικοινωνία με τις μονάδες των αισθητήρων, τα οποία είναι μυστικά. Περιλαμβάνει μια υπολογιστική μονάδα που μετατρέπει τα δεδομένα που έρχονται από τις μονάδες των αισθητήρων σε σημαντικές αναγνώσεις. Επίσης, «μεταφράζει» τα δεδομένα των BLE σε JSON / MQTT - επιτρέπει τόσο στα SDK όσο και στο cloud να επικοινωνούν με τις μονάδες των αισθητήρων. Επικοινωνεί με τη μονάδα Gainspan καθοδηγώντας το για συνδεθεί με το WiFi χρησιμοποιώντας τους κατάλληλους κωδικούς. Τέλος, διατρέχει την Bootloader εφαρμογή που διαχειρίζεται την αναβάθμιση του firmware.

Η Κεντρική μονάδα: Ενσωματώνοντας το τσιπ nRF51822 αυτό το τμήμα είναι υπεύθυνο για όλη την επικοινωνία του Bluetooth χαμηλής ενέργειας (BLE), περιλαμβάνει σάρωση για διαθέσιμες συσκευές BLE και κάνει την σύνδεση με αυτά, καθώς και σάρωση για την υπηρεσία Onboarding κατά τη διάρκεια της διαδικασίας onboarding.

Τα LED της κύριας μονάδας: *WiFi LED:* Βρίσκεται στην άκρη της κύριας μονάδας, ανάμεσα στο συνδετήρα USB και το πηνίο που διακρίνεται με ένα K. Όταν το LED αυτό είναι ενεργοποιημένο, το Master module δεν είναι συνδεδεμένο με WiFi. Όταν αυτή το LED είναι σβηστό αυτό αποτελεί ένδειξη ότι η κύρια μονάδα έχει συνδεθεί επιτυχώς με το δίκτυο WiFi.

LED Φόρτισης μπαταρίας: βρίσκεται ανάμεσα στο κουμπί επαναφοράς και την μπαταρία IC. Δείχνει την κατάσταση του κυκλώματος φόρτισης της μπαταρίας. Αυτό το LED θα αναβοσβήνει μετά από 4 ώρες χρήσης, σε περίπτωση που δεν έχει συνδεθεί μπαταρία. Όταν η μπαταρία φορτίζεται αυτή η ενδεικτική λυχνία θα παραμείνει αναμμένη.

Η κύρια μονάδα λειτουργίας Δείκτης: Βρίσκεται δίπλα στο τσιπ Kinetis. Αναβοσβήνει μία φορά ανά δευτερόλεπτο κατά τη διάρκεια της λειτουργίας Onboarding υποδεικνύοντας ότι το WunderBar περιμένει να onboarded. Ενώ αναβοσβήνει συνεχώς αναφέροντας ότι η κύρια μονάδα είναι στο Device Firmware Update (DFU) κατάσταση bootloader και ότι είναι σε αναμονή για το νέο firmware. Αυτή η λειτουργία μπορεί να ξεκινήσει με ένα παρατεταμένο πάτημα στο κουμπί onboarding.

Σε περίπτωση που η κύρια μονάδα εντοπίσει κάποιο πρόβλημα κατά τη σύνδεση στο relayr Cloud πλατφόρμα, αυτή η ενδεικτική λυχνία θα είναι επίσης συνεχώς ενεργοποιημένη. Το LED σβήνει όταν η κύρια μονάδα είναι συνδεδεμένη με το cloud αυτό το LED παραμένει απενεργοποιημένο το μεγαλύτερο μέρος του χρόνου, και αναβοσβήνει για λίγο μόνο κάθε φορά που ένας αισθητήρας δημοσιεύει στοιχεία.

2.1.5 Πώς ξεκινάς με το WunderBar

Αρχικά ο χρήστης δημιουργεί λογαριασμό στην ιστοσελίδα της Relayr με δικό του όνομα χρήστη και κωδικό. Οι κωδικοί αυτοί χρησιμοποιούνται για αν ενωθεί στον λογαριασμό του ο χρήστης είτε μέσω του application είτε μέσω της ιστοσελίδας. Το WunderBar εκτός από την πρωτοποριακή κατασκευή του φτιάχτηκε ώστε ο χρήστης να μπορεί να το χρησιμοποιήσει άμεσα. Το μόνο που χρειάζεται να κάνει ο χρήστης είναι να ανεβάσει στους αισθητήρες το firmware τους. Αυτό πετυχαίνεται μέσα από application που φτιάχτηκε από την Relayr και γίνεται σε μερικά απλά βήματα. Το application προσφέρεται δωρεάν στο google play για χρήστες του Android και στο App Store για χρήστες IOS. Ωστόσο λόγω της κατασκευής του ο χρήστης είναι αναγκαίο να έχει Smartphone που προσφέρει το Bluetooth Low Energy (BLE). Αφού παρέχεται αυτή η δυνατότητα ο χρήστης μπορεί να κατεβάσει το application και να ρυθμίσει εύκολα το WunderBar. Η διαδικασία αυτή ονομάζεται onboarding . Το application μέσω του bluetooth αρχικά ενώνεται στο master module και στέλνει σε αυτό αρχικά τις ρυθμίσεις του wifi. Το wifi πρέπει να υποστηρίζει security type WPA2-Personal και ο router να δουλεύει στα 2.5GHZ. Ακολούθως το master module ενώνεται στο internet και το application αρχίζει να στέλνει τις ρυθμίσεις των αισθητήρων στο cloud. Το master module που βρίσκεται σε onboard κατάσταση παίρνει τις ρυθμίσεις από το cloud και τις στέλνει στους αισθητήρες ανάλογα με το είδος ρύθμισης. Όταν ρυθμιστούν όλοι οι αισθητήρες το onboarding τελειώνει και οι αισθητήρες αρχίζουν να στέλνουν τις μετρήσεις στο master module και εκείνο με την σειρά του τις προωθεί στο cloud. Ακολούθως ο χρήστης μπορεί να τις δει από το application του. Επιπρόσθετα ο χρήστης

μπορεί να δει τις μετρήσεις μέσα και από το Developer Dashboard το οποίο βρίσκεται στην ιστοσελίδα της Relayr. Η διαδικασία αυτή είναι μια από τις δύο διαδικασίες που μπορούν να γίνουν ώστε να γίνουν onboard οι αισθητήρες. Κάθε αισθητήρας μπορεί να ρυθμιστεί ξεχωριστά μέσω του BLE. Μπορεί οποιοσδήποτε να φτιάξει ένα application στο οποίο θα συνδέει το Smartphone του με το αισθητήρα που θέλει και να επικοινωνεί άμεσα μαζί του με την χρήση του BLE χωρίς την παρεμβολή του cloud. Για τον λόγο αυτό οι αισθητήρες μπορούν να αποκοπούν από το κεντρικό WunderBar και να παίρνουν την ενέργεια τους από μπαταρίες των 3V.

2.1.6 Πως χρησιμοποιήθηκε στην εργασία

Για τους σκοπούς της παρούσας διπλωματικής εργασίας το WunderBar ρυθμίστηκε από το application της Relayr και χρησιμοποιήθηκε ώστε να λαμβάνει τις μετρήσεις από τους αισθητήρες και να τις στέλνει στο cloud. Μετέπειτα χτίστηκε μια εφαρμογή η οποία περιλαμβάνει ένα MQTT Client ο οποίος κάνει subscribe στο cloud και περιμένει να πάρει τις μετρήσεις. Κάθε φορά που το master module στέλνει τις μετρήσεις στο cloud, τότε ο client τις λαμβάνει και τις μεταφέρει στην εφαρμογή. Για λόγους ασφάλειας ο client χρησιμοποιεί μοναδικό κωδικό token ο οποίος κάθε δύο εβδομάδες ανανεώνεται. Ο κωδικός αυτός βρίσκεται στην ιστοσελίδα της Relayr κάτω από την ενότητα των πληροφοριών του λογαριασμού χρήστη. Για την δημιουργία του client η Relayr προσφέρει την βιβλιοθήκη που χρησιμοποιεί ο client και μερικά παραδείγματα. Έτσι για την υλοποίηση του client χρησιμοποιήθηκε το jar αρχείο που περιέχει την βιβλιοθήκη αυτή. Οι αισθητήρες που χρησιμοποιήθηκαν ήταν ο αισθητήρας θερμοκρασίας/υγρασίας, ο αισθητήρας γυροσκοπίου, ο αισθητήρας φωτός και ο αισθητήρας ήχου. Επίσης χρησιμοποιήθηκε το cloud της Relayr για να υλοποιηθεί το prototype του arduino. Για το arduino θα γίνει εκτενής αναφορά στην επόμενη ενότητα.

2.2 Arduino UNO

2.2.1 Εταιρικό Υπόβαθρο

Το Arduino ιδρύθηκε από τους Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, και τον David Mellis. Η εταιρία αποτελείται από διάφορα τμήματα με το κάθε τμήμα να αποτελείται από εξειδικευμένο προσωπικό στον τομέα που ασχολείται το τμήμα. Οι αρχικές πλακέτες σχεδιάστηκαν από τον Massimo Banzi και David Cuartielles. Το πρώτο λογισμικό φτιάχτηκε από τον David Cuartielles και ήταν βασισμένο στο Wiring. Το Wiring είναι λογισμικό ανοικτού κώδικα για προγραμματισμό των microcontrollers. Ακολούθως η εταιρία συνεργάστηκε με την SparkFun Electronics, και μαζί δημιούργησαν το LilyPad Arduino και μετέπειτα η SparkFun ανέπτυξε τα Arduino Pro και Arduino Pro Mini. Στην συνέχεια φτιάχτηκε ένα πρόγραμμα πιστοποίησης όπου άλλες εταιρίες μπορούν να χρησιμοποιήσουν το arduino ακόμα και με επεξεργαστές που δεν υποστηρίζονταν με αυτό. Η ταυτότητα της μάρκας, της συσκευασίας και τα γραφικά δημιουργήθηκαν από την TODO (Arduino, 2016).

2.2.3 Γιατί Arduino

Το Arduino έχει χρησιμοποιηθεί σε χιλιάδες διαφορετικά προγράμματα και εφαρμογές. Το λογισμικό μπορεί να χρησιμοποιηθεί εύκολα ακόμα και από αρχάριους χρήστες και μπορεί να γίνει αρκετά ευέλικτο ώστε να ικανοποιήσει τους πιο προχωρημένους χρήστες. Επίσης το λογισμικό του είναι συμβατό με Mac, Windows και Linux. Το arduino μπορεί εύκολα να επεκταθεί με απλά κυκλώματα σε αισθητήρες και διάφορες άλλες πλακέτες. Επιπρόσθετα υποστηρίζει με την βοήθεια και των απαραίτητων shields την σύνδεση του στο internet κάτι που το βάζει στον κόσμο των IOT. Η δυνατότητα επέκτασης του φτάνει ακόμη και στο επίπεδο σύνδεσης του με ηλεκτρικές συσκευές μέσω των κατάλληλων κυκλωμάτων. Ωστόσο το Arduino υπάρχει σε πολλές εκδόσεις. Η πλακέτα που χρησιμοποιείται στην εργασία είναι το ARDUINO UNO. Η λέξη UNO προέρχεται από την Ιταλία και σημαίνει ένα. Το UNO δηλώνει την έκδοση του λογισμικού. Το Arduino UNO αποτελείται από πέντε αναλογικές υποδοχές - inputs και έξι ψηφιακές υποδοχές - inputs. Ακόμη υποστηρίζεται υποδοχή παροχής για ηλεκτροδότηση των 3,5 και 9V όπως επίσης και γείωση. Για τους σκοπούς της εργασίας χρειάστηκε το Arduino να συνδεθεί στο internet και για τον λόγο αυτό προστέθηκε ethernet shield. Το ethernet shield υποστηρίζει την δυνατότητα του web server κάτι που χρησιμοποιήθηκε στην εργασία.

Επίσης το ethernet shield μπορεί να δεχτεί εξωτερική μνήμη SD όπου μπορούν να αποθηκευτούν πληροφορίες και δεδομένα του web server. Το κάθε pin-υποδοχή του arduino ενώνεται με το ethernet shield και αυτό με την σειρά του έχει όλα τα pins που έχει και το arduino ώστε να μην ακυρώνονται τα pins και να παρέχεται η δυνατότητα δημιουργίας κυκλωμάτων ακόμη και με το ethernet shield ενωμένο στο arduino. Ως αποτέλεσμα έχουμε δύο πλακέτες ενωμένες την μια πάνω στην άλλη. Η πλακέτα που βρίσκεται πάνω είναι κάποιο shield και η πλακέτα που βρίσκεται κάτω είναι το arduino (Arduino, 2016).

2.2.4 Πώς ξεκινάς με το ARDUINO UNO

Το arduino διακρίνεται για τον εύκολο προγραμματισμό του. Ο προγραμματιστής προγραμματίζει το firmware του arduino ώστε να λειτουργά όπως αυτός το χρειάζεται. Ο κατασκευαστής μαζί με την πλακέτα παρέχει το καλώδιο σύνδεσης του arduino με τον Η/Υ και το καλώδιο σύνδεσης για απευθείας ηλεκτροδότηση του arduino από τον τοίχο. Το καλώδιο σύνδεσης με τον Η/Υ χρησιμοποιείται κυρίως για να ανεβάσει ο προγραμματιστής το firmware αλλά συνάμα αποτελεί και καλώδιο ηλεκτροδότησης της πλακέτας αντί του καλωδίου απευθείας σύνδεσης. Αρχικά ο προγραμματιστής κατεβάζει την εφαρμογή που περιέχει το περιβάλλον προγραμματισμού του arduino και η οποία βρίσκεται στην ιστοσελίδα του. Σε περίπτωση που ο προγραμματιστής είναι αρχάριος η εφαρμογή προσφέρει παραδείγματα τα οποία αφορούν μερικές λειτουργίες του arduino. Μέσα από την ιστοσελίδα υπάρχουν και παραδείγματα εκμάθησης του arduino αλλά και της εφαρμογής. Στην ιστοσελίδα μπορεί επίσης κάποιος να λάβει μέρος σε συζητήσεις αλλά και να υποβάλει ερωτήσεις σχετικά με την λειτουργία ή τον τρόπο προγραμματισμού. Η γλώσσα προγραμματισμού του arduino είναι η ίδια με την C και έχει δικές της βιβλιοθήκες που καλύπτουν όλες τις λειτουργίες του. Ωστόσο υπάρχει σταθερή δομή που πρέπει να χρησιμοποιήσει ο προγραμματιστής. Στο πρόγραμμα του πρέπει να χρησιμοποιήσει τις συναρτήσεις `setup()` και `loop()`. Κατά την συνάρτηση του `setup()` ο χρήστης αρχικοποιεί τις μεταβλητές του, ορίζει αν οι υποδοχές του είναι για είσοδο ή έξοδο πληροφοριών, για να δείξει έναρξη κάποιας βιβλιοθήκης και διάφορες άλλες αρχικοποιήσεις. Η συνάρτηση αυτή τρέχει μόνο μια φορά κάθε φορά που είτε το arduino ξεκινά την λειτουργία του είτε γίνεται επανεκκίνηση. Κατά την συνάρτηση `loop()` το πρόγραμμα εκτελείται σε διαδοχικούς βρόγχους. Χρησιμοποιείται για να

εκτελείται κώδικας που θέλει ο προγραμματιστής συνεχόμενα. Δίνει την δυνατότητα να αλληλεπιδρά το arduino ζωντανά με τις εντολές. Αφού ο προγραμματιστής τελειώσει με τον κομμάτι του προγραμματισμού χρησιμοποιεί το καλώδιο σύνδεσης με τον Η/Υ ώστε να ανεβάσει το firmware στο arduino. Η εφαρμογή πριν ανεβάσει τον κώδικα στο arduino εκτελεί τον μεταγλωττιστής και προηδοποιεί τον προγραμματιστή για τυχόν συντακτικά λάθη παραλήψεις και σταματά την διαδικασία. Για να γίνει σωστή η μεταγλώττιση πρέπει να καθοριστεί το είδος του arduino κάτι που γίνεται από το κουμπί του μενού "tools" στην εφαρμογή. Επιπρόσθετα παρέχεται η δυνατότητα στον προγραμματιστή να παρακολουθήσει την εκτέλεση του κώδικα του χρησιμοποιώντας το την κονσόλα που υπάρχει στην εφαρμογή. Η δυνατότητα αυτή ονομάζεται serial monitor. Ως αποτέλεσμα να γίνεται πιο εύκολη και αποτελεσματική η αντιμετώπιση των προβλημάτων (Arduino, 2016).

2.2.5 Πως χρησιμοποιήθηκε στην εργασία

Η παρούσα διπλωματική εργασία έχει σαν σενάριο για υλοποίηση το "Έξυπνο Γραφείο". Το arduino χρησιμοποιήθηκε για τον έλεγχο και την εκμετάλλευση των IOT συσκευών. Για τους σκοπούς όμως της διπλωματικής εργασίας υλοποιήθηκε απλά ένα μικρό κύκλωμα το οποίο περιέχει απλά μια λάμπα LED. Η λάμπα αυτή αναβοσβήνει όταν δέχεται εντολές από την εφαρμογή του "Έξυπνου Γραφείου". Οι εντολές που δέχεται έχουν να κάνουν με το άνοιγμα ή κλείσιμο των παραθυρόφυλλων, με το άνοιγμα ή κλείσιμο του κλιματιστικού και προειδοποιήσεις για ψηλά επίπεδα ηχορύπανσης. Η χρησιμότητα όμως της εφαρμογής προϋποθέτει απομακρυσμένο έλεγχο των IOT και αυτομάτως σύνδεση του arduino στο internet. Η σύνδεση στο internet έγινε με την βοήθεια του ethernet shield. Ωστόσο για την σύνδεση με το WunderBar κατά την φάση ανάπτυξης του λογισμικού που θα τρέξει στο arduino υλοποιήθηκε ο MQTT client-server που θα δίνει την δυνατότητα στον προγραμματιστή να στέλνει και να λαμβάνει μηνύματα με από και προς το arduino. Για την υλοποίηση του client προστέθηκαν οι κατάλληλες βιβλιοθήκες τις οποίες είχε αναπτύξει η εταιρία του arduino για τον σκοπό αυτό. Ακολούθως χρησιμοποιήθηκε το cloud της Relayr για να φτιαχτεί το πρωτότυπο στο cloud όπου θα γίνεται εισαγωγή των μηνυμάτων και η προώθηση τους στον κατάλληλο client. Κατά την διαδικασία δημιουργίας του πρωτότυπου δόθηκαν από το cloud ο κωδικός και το όνομα στο οποίο θα συνδεόταν ένας client αλλά και ο κωδικός του client. Τα στοιχεία αυτά επισυνάφτηκαν στον κώδικα του arduino και

χρησιμοποιήθηκαν στην δημιουργία του client. Ακολούθως έγινε η σύνδεση της εφαρμογής του "Έξυπνου Γραφείου" με το πρωτότυπο του arduino που φτιάχτηκε στο cloud. Στο σημείο αυτό αναπτύχθηκε ακόμα ένας client ο οποίος συνδέεται επίσης με το πρωτότυπο του arduino. Η τελική σύνδεση εμπεριέχει την σύνδεση του arduino μέσω client στο cloud και την σύνδεση του "Έξυπνου Γραφείου" στο πρωτότυπο του arduino μέσω client στο cloud. Για την σύνδεση του "Έξυπνου Γραφείου" με το cloud χρησιμοποιήθηκε ο Paho MQTT client και οι κωδικοί που δημιουργήθηκαν κατά την προτυποποίηση του arduino.

Κεφάλαιο 3

Γνωστικό Υπόβαθρο

3.1. Επεξήγηση της έννοιας Cloud Computing

3.1.1. Τι είναι το Cloud

3.1.2. Relayr CLOUD

3.2. Developer Dashboard

3.2.1. Ιστορικά σημειώματα για το Dashboard

3.2.2. Relayr Developer Dashboard

3.3. MQTT

3.3.1. Τι είναι το MQTT protocol

3.3.2. MQTT Topics

3.3.3. MQTT subscribe client

3.3.4. MQTT publish client

3.4. Threads

3.1. Επεξήγηση της έννοιας Cloud Computing

3.1.1. Τι είναι το Cloud Computing

Το Cloud Computing είναι η χρησιμοποίηση των βασισμένων στο internet τεχνολογιών για τον παροχή των υπηρεσιών και απεικόνιση τους στον internet. Οι τεχνολογίες αυτές βασιζόμενες στα διαγράμματα δικτύων των υπολογιστών υπάρχουν στο σύννεφο - cloud ώστε να εξυπηρετούν τις υπηρεσίες που προβάλλουν. Με βάση αυτό το cloud δίνει την

αίσθηση ότι υπάρχουν άπειροι υπολογιστικοί πόροι αξιοποίησης από τους χρήστες. Βάση όμως της εταιρίας και των υπηρεσιών που προβάλλονται στην κάθε περίπτωση ο χρήστης έχει την δυνατότητα αξιοποίησης όσο πιο πολλών πόρων με βάση πληρωμής. Ο χρήστης σύμφωνα με τις ανάγκες του πληρώνει για την χρησιμοποίηση των πόρων που χρειάζεται. Παράδειγμα του σεναρίου αυτού αποτελεί το Dropbox. Ο χρήστης σύμφωνα με τον χώρο που θα χρειαστεί στον cloud του Dropbox πληρώνει για να αυξηθεί η χωρητικότητα σ' αυτό. Επιπρόσθετα όταν η υπηρεσία αναπτύχθηκε μια φορά για τον χρήστη παραμένει χωρίς να χρειάζεται την μεσολάβηση και πάλι κάποιας τεχνολογίας για να παραχθεί. Η Cloud Computing εφαρμογή αναπτύσσεται με σκοπό την δημιουργία επικοινωνία μεταξύ του χρήστη και του cloud και βρίσκεται στο ψηλότερο επίπεδο στην κλίμακα των εφαρμογών διαδικτύου ενώ συνδυάζει εικονικούς πόρους και τους παρουσιάζει ως υπηρεσία πάνω από το internet. Το Application Layer αποτελεί το ψηλότερο επίπεδο στην αρχιτεκτονική των δικτύων και εκεί βρίσκονται όλες οι εφαρμογές. Ως αποτέλεσμα να τοποθετείται και το Cloud στο επίπεδο αυτό αφού είναι μια διαδικτυακή εφαρμογή. Στο επίπεδο αυτό ο χρήστης δεν χρειάζεται να έχει γνώσεις ούτε εμπειρία για την υποδομή και τις τεχνολογίες που τρέχουν από πίσω από το cloud εναντίων τον απασχολεί για το τι κάνει η εφαρμογή και όχι το πώς την κάνει ή με ποιά υποδομή υλοποιείται. Το Cloud Computing αποτελεί το φυσικό επακόλουθο της εξέλιξης της τεχνολογίας των πληροφοριών on-demand σε υπηρεσίες και προϊόντα. Όταν δημιουργήθηκε το Cloud Computing είχε σαν στόχο του να παρέχει μια αληθινή, πολύτιμη, και οικονομική συμβολή για υποδομές διαδικτύου. Κάτι που ενθάρρυνε την ανάπτυξη των υπηρεσιών διαδικτύου και λογισμικού αλλά και την έρευνα στην εικονική αναπαράσταση - "virtualization" -, το υπολογιστικό πλέγμα - "grid computing"- και άλλα πολλά βοηθητικά προγράμματα. Παρόλο που το Cloud Computing διακρίνεται από πολλά θετικά στοιχεία όπως όλα στον τομέα των δικτύων υπάρχουν εμπόδια στην πτυχή που αφορά την ασφάλεια του. Εμπόδια που αφορούν την προστασία της ιδιωτικής ζωής και νομικά θέματα. Ωστόσο το Cloud Computing είναι ένα σχετικά καινούργιο μοντέλο στην επιστήμη της πληροφορικής και κανείς δεν μπορεί να εκφραστεί με βεβαιότητα ότι θα καλυφθούν όλα τα επίπεδα ασφάλειας και με ποιόν τρόπο οι εφαρμογές ασφαλείας θα επιδρούν στο Cloud. Ως εκ τούτου το θέμα αυτό αποτελεί την υπ' αριθμό ένα ανησυχία των ειδικών σε σχέση με το Cloud Computing. Τα θέματα ασφαλείας εξάγονται μέσα από τις υπηρεσίες που παρέχει το κάθε Cloud Computing στους χρήστες και χωρίζονται σε τρεις κατηγορίες - το S.P.I. model ,η κατηγορία Software as a Service (SaaS) , η κατηγορία Platform as a Service (PaaS), και η κατηγορία Infrastructure as a Service

(IaaS). Ο χρήστης παίρνει πρόσβαση στο Cloud μέσα από τις εφαρμογές που παρέχει η εταιρία δημιουργίας του Cloud. Στο σημείο αυτό συναντούμε τα θέματα ασφάλειας (SaaS) αφού ο χρήστης μπορεί να έχει πρόσβαση στο Cloud μέσω οποιασδήποτε συσκευής αλλά η πρόσβαση αυτή γίνεται μέσω ενός thin client interface. Το βάρος της ασφάλειας στο SaaS αποκομίζεται στους διαχειριστές του Cloud . Επιπρόσθετα ο χρήστης έχει την δυνατότητα να αναπτύξει δικές του εφαρμογές ώστε να αποκτήσει πρόσβαση στο Cloud χωρίς να εγκαταστήσει οποιαδήποτε πλατφόρμα στην υποδομή του. Η δυνατότητα αυτή αν και δημιουργεί θέματα ασφαλείας τύπου PaaS σε αντίθεση με τα SaaS επιτυγχάνεται περισσότερος έλεγχος του πελάτη αλλά και μεγαλύτερη επεκτασιμότητα. Ακόμη μία υπηρεσία που προσφέρεται από ένα Cloud Computing είναι η παροχή υποδομής όπου ο χρήστης μπορεί να αναπτύξει και να εκτελέσει αυθαίρετο λογισμικό, το οποίο μπορεί να περιλαμβάνει λειτουργικά συστήματα και εφαρμογές. Αυτή η κατηγορία στο μοντέλο ανήκει στο IaaS και προσφέρει μεγαλύτερο έλεγχο του χρήστη από τις υπόλοιπες κατηγορίες. Παρακάτω θα αναλυθεί ο τρόπος λειτουργίας του Relayr Cloud αλλά και το μοντέλο που χρησιμοποιεί η εταιρία στο Cloud της (Alexandros Marinos, Gerard Briscoe, 2009) (Keiko Hashizume, 2013) (Vouk, 2008).

3.1.2. Relayr CLOUD

Η Relayr στην προσπάθειά της για πρόσβαση στον κόσμο των Internet Of Things (IOT) έκτισε το δικό της Cloud Computing. Στόχος της είναι να δημιουργήσει μια επικοινωνία Πολλών προς Πολλά -Any to Any - που αφορά συνδέσεις μεταξύ οποιασδήποτε υπηρεσίας , οποιουσδήποτε αισθητήρες και οποιαδήποτε υλικά μηχανήματα. Για υλοποίηση του στόχου της ανέπτυξε τις υπηρεσίες για να υποστηρίζουν πολλαπλά πρωτόκολλα, αλγόριθμους δρομολόγησης και προσαρμογές για API. Επίσης το Relayr Cloud υποστηρίζει ανάλυση δεδομένων με γραφικές παραστάσεις. Οι γραφικές παραστάσεις που παρουσιάζονται χωρίζονται είτε σε ζωντανές γραφικές παραστάσεις όπου ζωντανά δείχνουν την διακύμανση των μετρήσεων από τους αισθητήρες είτε προσαρμόζονται ανά ώρα είτε ανά πέντε ώρες είτε ανά μέρα είτε ανά εβδομάδα ή μήνα ή χρόνο. Για την σύνδεση στο Relayr Cloud είναι απαραίτητο να δημιουργήσεις το δικό σου λογαριασμό με τους δικούς σου κωδικούς και στοιχεία. Μέσα από την ιστοσελίδα που είναι η εφαρμογή του Relayr Cloud ο χρήστης έχει την δυνατότητα να επιλέξει μεταξύ πολλών ενοτήτων που αφορούν τόσο το επιχειρησιακό περιβάλλον που μπορεί να χρησιμοποιήσει ο χρήστης όσο και το περιβάλλον που αφορά του προγραμματιστές. Το

κομμάτι που αφορά τους προγραμματιστές ονομάζεται "Developer Dashboard" . Ο χρήστης έχει την επιλογή να επιλέξει τι ρόλο ή ρόλους θα ακολουθήσει στην χρησιμοποίηση του Relayr Cloud. Μια από τις επιλογές του χρήστη είναι να βάλει μπρός ,να υλοποιήσει και να προβάλλει τις ιδέες του κάνοντας αίτηση για εγγραφή στα εργαστήρια που κάνει η Relayr. Για τους σκοπούς της εργασίας δημιουργήθηκε λογαριασμός στην ιστοσελίδα και χρησιμοποιήθηκε για την ενότητα των προγραμματιστών. Αφού έγινε η ρύθμιση του WunderBar μέσω της εφαρμογής του android οι αισθητήρες και οι μετρήσεις τους εμφανίστηκαν στην ιστοσελίδα. Οι αισθητήρες WunderBar στέλνουν τις μετρήσεις στο master module και αυτό με την σειρά του τις προωθεί στο Cloud. Η σύνδεση γίνεται με το πρωτόκολλο MQTT και η επικοινωνία ασφαλίζεται με το TLS. Όπως αναλύθηκε πιο πάνω το Relayr Cloud προσφέρει γρήγορη σύνδεση άλλων συσκευών με την διαδικασία προτυποποίησης. Μέσα από το developer dashboard υπάρχει η επιλογή του Device ->"New Device" όπου οδηγεί στον σύνδεσμο που ο χρήστης επιλέγει την συσκευή που θέλει να κάνει προτυποποίηση στο Cloud. Αν και υπάρχουν πολλές επιλογές σχετικά με τα μοντέλα συσκευών δίνεται και η δυνατότητα επιλογής χωρίς μοντέλο. Εφόσον ο προγραμματιστής επιλέξει την συσκευή που χρειάζεται το Cloud δημιουργεί τα κωδικούς σύνδεσης και την ονομασία του client που πρέπει να συνδεθεί ο client του προγραμματιστή ώστε να έχει πρόσβαση στο πρότυπο. Η δυνατότητα προτυποποίησης ανήκει στην κατηγορία μοντέλου PaaS ενώ η δυνατότητα σύνδεσης στην κατηγορία SaaS. Συμπερασματικά το Relayr Cloud είναι ένα δυνατό εργαλείο για γρήγορη εισαγωγή συσκευών και εργασιών στον κόσμο των IOT (Relayr, 2016).

3.2. Developer Dashboard

3.2.1. Ιστορικά σημειώματα για το Dashboard

Το Dashboard δεν αποτελεί καινούργιο εργαλείο στην σύγχρονη εποχή της τεχνολογικής ανάπτυξης αλλά είναι η μετονομασία του Executive Information Systems (EISs) το οποίο αναπτύχθηκε το 1980. Παρόλο που υπήρχε από το 1980 οι πιθανότητες να το συναντούσε κάποιος είναι μειωμένες αφού δεν υπήρχαν πολλά τέτοια εργαλεία και όσα υπήρχαν αναπτύχθηκαν μόνο για τα γραφεία στελεχών. Το EISs όταν δημιουργήθηκε κύριος σκοπός του ήταν να παρουσιάζει ελάχιστες βασικές οικονομικές μετρήσεις κάτι που κάνει ευνόητο τον λόγο που μόνο στα γραφεία στελεχών υπήρχε. Κατα την διάρκεια των 1990s οι εταιρίες αποφάσισαν να ακολουθήσουν την τεχνολογία ώστε να

αντεπεξέλθουν στον τεράστιο όγκο πληροφοριών που κατακλύζονταν. Στην διάρκεια αυτών των χρόνων δόθηκε έμφαση στην συλλογή, διόρθωση, ενσωμάτωση και πρόσβαση στα δεδομένα τους στην προσπάθεια για απόκτηση δεδομένων με εγγύηση στην ακρίβεια, διαχρονικότητα και χρησιμότητα τους. Με το πέρασμα των χρόνων η ανάγκη για χρησιμοποίηση των δεδομένων αυτό και σε συνδυασμό με την ανάπτυξη της επιστήμης της πληροφορικής μεγάλες εταιρίες όπως η IBM και η Oracle εισήγαγα το όρο online analytics. Επιπρόσθετα στην ιδέα αυτή δημιουργήθηκε και η ανάγκη για δημιουργία αναλυτικών παραστάσεων όχι μόνο σε δεδομένα που αφορούσαν τα οικονομικά αλλά σε οποιαδήποτε δεδομένα χρειάζονταν παραστάσεις. Η πρόσβαση στα δεδομένα αυτά έγινε με την ανάπτυξη του EISs και μετονομασία του σε dashboard και υλοποίηση της ιδέας σε διαδικτυακό λογισμικό . Τα δεδομένα αναλύονται σε γραφικές παραστάσεις οι οποίες μέσα από κάποιο browser παρουσιάζονται στο χρήστη. Προχωρώντας στην εξέλιξη του dashboard διαπιστώνεται ότι δεν παρέμειναν μόνο στην δημιουργία αναλυτικών παραστάσεων αλλά πρόσθεσαν και άλλα εργαλεία σε αυτό. Επίσης το dashboard έγινε ένα σπουδαίο εργαλείο για το Cloud Computing. Μετέπειτα το dashboard ενσωματώνεται στο Cloud Computing και αποκτά σημαντικό ρόλο στην τεχνολογική ανάπτυξη μιας εταιρίας. Η Relayr δημιούργησε το δικό της dashboard που αφορά τους προγραμματιστές εφαρμογών για την εξυπηρέτηση του Cloud Computing της. Εφόσον το cloud έχει να κάνει με εικονική υποδομή τότε το dashboard μπορεί να χαρακτηριστεί σαν η εφαρμογή υλοποίησης των υπηρεσιών της εικονικής υποδομής (Few, 2006).

3.2.2. Relayr Developer Dashboard

Η Relayr επιδίωξε την παροχή υπηρεσιών όπως είναι οι ζωντανές και μακροχρόνιες γραφικές παραστάσεις των μετρήσεων που προέρχονται από το WunderBar και η προτυποποίηση πρόσθετων συσκευών. Αυτό οδήγησε στην δημιουργία του δικού της dashboard. Μέσα από το Developer dashboard της Relayr ο προγραμματιστής μπορεί να παρακολουθήσει τις μετρήσεις αλλά και τις γραφικές παραστάσεις των μετρήσεων από τους αισθητήρες. Το dashboard δημιουργήθηκε με γνώμονα την επικοινωνία του προγραμματιστή με την ομάδα ανάπτυξης στην Relayr. Ο προγραμματιστής μέσα από την ενότητα του Documentation μπορεί να επιλέξει την γλώσσα προγραμματισμού του client στον οποίο θα αναφέρεται η εφαρμογή που θα αναπτύξει. Κάθε επιλογή εκτρέπεται σε καινούργιο σύνδεσμο όπου περιγράφεται ο τρόπος με τον οποίο

αναπτύσσεται ο client στην γλώσσα αυτή. Η ομάδα ανάπτυξης συνεχώς μετατρέπει και βελτιώνει τον κώδικα που εκθέτει στις επιλογές αυτές δημιουργώντας έτσι μια συνεχής επικοινωνία με τον προγραμματιστή. Επίσης μέσα από το dashboard δημιουργήθηκε μια ενότητα με συζητήσεις και αμεσότερη επικοινωνία μεταξύ προγραμματιστή και της ομάδας ανάπτυξης λογισμικών στην Relayr. Η ενότητα αυτή χωρίζεται σε έξι κατηγορίες, την κατηγορία που αφορά ερωτήματα για τεχνική υποστήριξη, την κατηγορία που αφορά την υποστήριξη για παραλήψεις από την Relayr, την κατηγορία με θέματα που δεν ανήκουν σε κάποια άλλη κατηγορία, την κατηγορία που αφορά ερωτήματα για τις εφαρμογές που υλοποιήθηκαν για το Smart phone από την Relayr, την κατηγορία που αφορά ερωτήματα για την αποστολή WunderBar και την κατηγορία για την ανάλυση του WunderBar. Οι προγραμματιστές έχουν την δυνατότητα να επιβάλουν τις ερωτήσεις τους και ο/η υπεύθυνος της ομάδας σε συνδυασμό με όλη την ομάδα απαντούν στο ερώτημα.

3.3. MQTT

3.3.1. MQTT protocol

Το MQTT είναι τα αρχικά για τις λέξεις Message Queuing Telemetry Transport που σημαίνει Μεταφορά μηνυμάτων τηλεμετρίας σε ουρές. Το πρωτόκολλο αυτό χρησιμοποιείται σαν πρωτόκολλο δημοσίευσης και συνδρομής σε μηνύματα και είναι εξαιρετικά απλό και ελαφρύ πρωτόκολλο ανταλλαγής μηνυμάτων, σχεδιάστηκε για συγκεκριμένες συσκευές, για χαμηλά εύροι ζώνης, για αναξιόπιστα δίκτυα ή δίκτυα με υψηλό χρόνο αναμονής. Οι αρχές του σχεδιασμού είναι να ελαχιστοποιηθεί το εύρος ζώνης του δικτύου και τις απαιτήσεις των πόρων της συσκευής, ενώ προσπαθεί να διασφαλίσει την γενική αξιοπιστία αλλά και κάποιο βαθμό αξιοπιστίας της παράδοσης. Αποτελεί ιδανικό πρωτόκολλο για ανταλλαγή μηνυμάτων από μηχανή σε μηχανή (M2M) ή από IOT σε IOT (Internet Of Things) καθώς και για smart phone εφαρμογές όπου οι ταχύτητες δικτύου και η ισχύς της μπαταρίας τους είναι μικρό. Το MQTT δημιουργήθηκε από τον Dr Andy Stanford-Clark από την IBM, και τον Arlen Nipper από την Arcom (μετέπειτα έγινε Eurotech). Τον Νοέμβριο του 2011 οι IBM και Eurotech ανακοίνωσαν την από κοινού συμμετοχή τους στο M2M - Ομάδα Εργασίας του Eclipse - και έδωσαν δωρεά τον κώδικα του MQTT στο έργο Eclipse PAHO. Επίσης το MQTT χρησιμοποιεί συγκεκριμένη πόρτα επικοινωνίας, την 1883, η οποία είναι και η εγγεγραμμένη πόρτα χρησιμοποίησης του MQTT TCP/IP με πρωτόκολλο ασφαλείας το SSL. Η

κρυπτογράφηση στο δίκτυο αντιμετωπίζεται με το SSL ανεξάρτητα από το MQTT. Στο MQTT μπορεί να προστεθεί όνομα και κωδικός χρήστη για περαιτέρω ασφάλεια. Μετέπειτα αν ο προγραμματιστής χρειάζεται περισσότερη ασφάλεια μπορεί στον κώδικα του να προσθέσει αλγόριθμους κρυπτογράφησης οι οποίοι δεν ανήκουν στο πρωτόκολλο του MQTT. Υπάρχουν πολλές διαθέσιμες επιλογές για τους προγραμματιστές που θέλουν να φτιάξουν τον δικό τους MQTT client. Το MQTT χωρίζεται σε subscribe client και σε publish client όπου στην κάθε διαθέσιμη γλώσσα προγραμματισμού του client υπάρχουν και οι δύο κατηγορίες. Ο client που δημιουργήθηκε από την ομάδα προγραμματισμού του WunderBar και χρησιμοποιήθηκε στην εργασία είναι ο Eclipse Paho Java και η επιλογή του έγινε εφόσον αποφασίστηκε ότι η εφαρμογή "Smart Office" θα υλοποιηθεί στο Eclipse περιβάλλον. Έγινε χρήση τόσο του subscribe client όσο και του publish client. Ο subscribe client αποτελεί την επικοινωνία μεταξύ του cloud και της εφαρμογής για την αποστολή των μετρήσεων από το cloud στην εφαρμογή ενώ ο publish client αποτελεί την επικοινωνία μεταξύ της εφαρμογής και του cloud για την αποστολή μηνυμάτων από την εφαρμογή στο πρωτότυπο του arduino. Πιο κάτω θα γίνει εκτενής ανάλυση των subscribe/publish client (Brian O'Connell, 2007).

3.3.2. MQTT Topic

Το MQTT χρησιμοποιεί κάποιο κλειδί το οποίο προσδιορίζει το κανάλι που θα περνούν οι πληροφορίες των subscribe/publish clients. Ο subscribe client χρησιμοποιεί το κανάλι αυτό για να παίρνει τις πληροφορίες από τον publish client και ο publish client αντίθετα. Το κλειδί αυτό έχει την δική του καθορισμένη δομή και είναι δομημένο με ιεραρχικά επίπεδα χωρητικότητας . Οι διάφοροι τύποι μηνυμάτων επιτρέπουν στον client να καθορίζει στο topic διάφορους παραμέτρους ακρίβειας. Ωστόσο το MQTT επιβάλλει κάποιους κανόνες για τον χώρο που πρέπει να έχει το topic. Μια σωστή επιλογή είναι η αποφυγή μεγάλων topics επειδή όσο πιο μεγάλο είναι το topic τόσο πιο μεγάλη είναι η αναγκαία ταχύτητα του δικτύου. Παρόλο που είναι καλύτερο να χρησιμοποιούνται μικρά topics, ένα καλά σχεδιασμένο και πιο περιγραφικό topic που επιτρέπει την χρήση χαρακτήρων αναγνώρισης έχει σημαντικά οφέλη. Η κάθε εταιρία που θα επιδιώξει να δημιουργήσει τον δικό της MQTT client φτιάχνει και την δικής της ιεράρχηση χώρου για το topic. Η Relayr το κλειδί που χρησιμοποιεί είναι ο μοναδικός κωδικός του client και είναι η εξής "/v1/ <<client password>>/. Το MQTT συνδέεται στο cloud με την χρήση των κωδικών του χρήστη και ακολούθως εγκαθιστά το κανάλι που θα γίνει η ανταλλαγή

των πληροφοριών με την εισαγωγή του topic δομημένος όπως αναλύθηκε πιο πάνω (Brian O'Connell, 2007).

3.4.1. MQTT subscribe client

Ένας subscribe client χρησιμοποιεί το topic ώστε να ανοίξει το κανάλι επικοινωνίας με τον publish client. Εφόσον εγκατασταθεί το κανάλι ο subscriber περιμένει να δεχτεί τα μηνύματα από τον publisher και τα προσθέτει στην ουρά. Ο προγραμματιστής μπορεί να τυπώσει και να διαχειριστεί τα μηνύματα χρησιμοποιώντας την βιβλιοθήκη του client. Στην εργασία η βιβλιοθήκη που χρησιμοποιήθηκε αφορά το περιβάλλον **eclipse**. Από εκεί δημιουργείται ο subscribe client για τον κάθε αισθητήρα ξεχωριστά. Οι αισθητήρες στην περίπτωση αυτή είναι οι publisher και η εφαρμογή στο eclipse ο subscriber. Για την δημιουργία του καναλιού ο subscriber χρειάζεται τον μοναδικό κωδικό του publisher. Πολλοί από τους αισθητήρες συνδυάζουν περισσότερους από ένα αισθητήρα στην ίδια μονάδα και για τον λόγο αυτό οι μετρήσεις που φτάνουν στην ουρά του subscriber χωρίζονται και τυπώνονται στην κονσόλα του eclipse. Το WunderBar έχει προεπιλογή να παίρνει μετρήσεις και να τις στέλνει στο cloud κάθε 30 δευτερόλεπτα. Επειδή όμως ο κάθε client τρέχει σε διαφορετικό thread και το κάθε κομμάτι του κώδικα εκτελείται παράλληλα γίνεται κατορθωτή η τύπωση όλων των μετρήσεων στην κονσόλα. Ωστόσο υπάρχει πρόβλημα στην δημιουργία όλων των καναλιών ταυτόχρονα. Ο κάθε client επειδή τρέχει σε διαφορετικό thread αρχίζει την διαδικασία εγκατάστασης του καναλιού του με το cloud παράλληλα. Στην περίπτωση που υπάρχει κεντρικό σημείο στον κώδικα που καλεί όλους τους client μαζί, θα τρέξουν σε διαφορετικό thread και θα αρχίσουν μαζί την διαδικασία για εγκατάσταση του καναλιού τους πάνω σε μια γραμμή επικοινωνίας. Το cloud πρέπει να δημιουργήσει μια γραμμή για κάθε client ταυτόχρονα, ως αποτέλεσμα να έχουμε μια μη σταθερή κατάσταση στην εφαρμογή μας όπου υπάρχει υψηλό ποσοστό το cloud να μην καταφέρει να εξυπηρετήσει κάποιον από τους client λόγω σύγκρουσης ή ταυτοχρονισμού. Έτσι έγινε διαχωρισμός στον κάθε subscriber ώστε να αποφευχτού τα προβλήματα ταυτοχρονισμού. Τα μηνύματα που παίρνει ο κάθε client μετατρέπονται από τις κλάσεις της βιβλιοθήκης σε τύπο γραμματοσειράς κάτι που κάνει πιο εύκολη την διαχείρισή τους (Brian O'Connell, 2007).

3.4.2. MQTT publish client

Ο publish client είναι αυτός που θα στείλει τα δεδομένα στο κανάλι που εγκαταστάθηκε μεταξύ αυτού και του subscriber. Για την αποστολή των μηνυμάτων ο publisher χρησιμοποιεί συγκεκριμένους παραμέτρους. Η πρώτη παράμετρος είναι τύπου

γραμματοσειράς και χρησιμοποιείται από το cloud για να ταυτίσει τον publisher με κάποιον subscriber που ενδιαφέρεται για να εγκαταστήσει κανάλι επικοινωνίας με αυτόν. Η δεύτερη παράμετρος είναι το `MqttPayload` και περιλαμβάνει δεδομένα της εφαρμογής αλλά και επιγραφές των πρωτοκόλλων ωστόσο υπάρχει και πεδίο που καθορίζει το σημείο που αρχίζουν τα δεδομένα. Αυτό επιτρέπει την πρόσβαση στα χρησιμοποιημένα δεδομένα και δεν χρειάζεται να δημιουργηθούν ξανά. Η τρίτη παράμετρος είναι το `BYTE` και δουλειά της είναι να εκφράζει το Quality Of Service (QoS). Υπάρχουν τρεις εκδόσεις QoS και χαρακτηρίζονται με ένα από τους αριθμούς 0, 1 ή 2. Η πρώτη έκδοση είναι το QoS-0 το οποίο υποδηλώνει ότι publisher και το cloud ανταλλάσσουν μηνύματα χωρίς να λαμβάνουν μέτρα που να εξασφαλίζουν ότι τα μηνύματα παραδόθηκαν πέρα από αυτά που ισχύουν στο TCP/IP πρωτόκολλο. Η δεύτερη έκδοση είναι το QoS 1 το οποίο με την ανταλλαγή των μηνυμάτων εξασφαλίζει και την παράδοση τους αλλά το μήνυμα μπορεί να παραδοθεί περισσότερες από μία φορές. Η τρίτη έκδοση είναι το QoS 2 και είναι η έκδοση που δίνει την εντολή στο MQTT για αποστολή μηνύματος ακριβώς μια φορά. Όταν ένας publisher αποστέλλει ένα μήνυμα με QoS 2 σε ένα subscriber με QoS 1 τότε το μήνυμα θα παραδοθεί σε QoS 1 ενώ όταν σταλεί ένα μήνυμα με QoS 0 στον subscriber με QoS 1 τότε το μήνυμα θα σταλεί με QoS 0. Έτσι είναι ευθύνη του προγραμματιστή για την επιλογή του QoS για την οποία παίρνει την απόφαση αφού μελετήσει σωστά τα σενάρια που μπορούν να συμβούν μεταξύ subscriber και publisher. Η τέταρτη παράμετρος είναι τύπου Boolean η οποία με μία παράμετρος που παίρνει τις τιμές αληθής ή ψευδής ώστε να δείξει αν υπάρχει διατηρούμενη δημοσίευση. Μια διατηρούμενη δημοσίευση είναι το τελευταίο μήνυμα που στάλθηκε στον subscriber με το συγκεκριμένο topic. Αυτό επιτρέπει στους μετέπειτα subscribers να λαμβάνουν το πιο πρόσφατο μήνυμα που δόθηκε από τον publisher ακόμα και αν αυτό στάλθηκε πριν την μεταξύ τους σύνδεση. Εάν όμως η παράμετρος είναι ψευδής τότε οι subscribers που θα ενωθούν με τον publisher μετά που θα αποσταλεί το μήνυμα στο κανάλι θα πρέπει να περιμένουν το επόμενο μήνυμα. Ο publisher επίσης επιστρέφει ένα αναγνωριστικό ακέραιο αριθμό ώστε σε συνδυασμό με την μέθοδο `MqttAdvancedCallback` να γίνεται ανίχνευση της αποστολής του μηνύματος στο cloud. Η μέθοδος αυτή είναι η προέκταση της μεθόδου `Callback` και χρησιμοποιείται και στους subscribers ωστόσο η σημαντικότερη χρησιμότητα της βρίσκεται στους publishers. Η `Callback` μέθοδος χρησιμοποιεί μόνο μηνύματα με QoS 1 ή 2 και ο λόγος είναι επειδή στο QoS 0 δεν κάνει εξασφάλιση του απεσταλμένου μηνύματος. Η μέθοδος αυτή βρίσκεται στην βιβλιοθήκη και χρησιμοποιείται κατά την δημιουργία του publish client. Στην παρούσα εργασία ο

publish client χρησιμοποιήθηκε για την αποστολή μηνυμάτων από την εφαρμογή του "Έξυπνου Γραφείου" στο Arduino. Το arduino κάνει subscribe στο πρωτότυπο του arduino που βρίσκεται στο cloud ενώ η εφαρμογή κάνει publish στο πρωτότυπο του arduino. Έτσι το cloud γίνεται ο μεσάζον στην επικοινωνία μεταξύ της εφαρμογής και του arduino. Τα μηνύματα φεύγουν από την εφαρμογή με προορισμό το cloud και από το cloud μεταφέρονται στο arduino (Brian O'Connell, 2007).

3.5. Threads

Ένα πρόγραμμα με υψηλό χρόνο εκτέλεσης μπορεί να διασπαστεί σε κομμάτια τα οποία θα τρέχουν παράλληλα ώστε ο χρόνος εκτέλεσης του συνολικού προγράμματος να μειωθεί. Κάθε κομμάτι που τρέχει παράλληλα είναι και ένα thread. Κλάσεις από το Java API χρησιμοποιούν τα threads ώστε να εξυπηρετούν διάφορες διαδικασίες ως αποτέλεσμα οι προγραμματιστές που χρησιμοποιούν την java να κάνουν και χρήση των threads χωρίς όμως να το καταλαβαίνουν. Αρχικά όταν δημιουργήθηκαν τα threads στόχος τους ήταν να κάνουν διάσπαση του κώδικα για να μειώσουν την πολυπλοκότητα ενός προγράμματος. Υπάρχουν όμως περιπτώσεις προγραμμάτων που δεν επιτυγχάνουν στην παράλληλη εκτέλεση και αυτά είναι προγράμματα που μεταχειρίζονται διαδικασίες που έχουν πολλές εξαρτήσεις μεταξύ τους. Αυξημένη χρήση των threads συναντούμε στα desktop computers και σε διεπαφές γραφικών επειδή εκεί ο χρήστης χρειάζεται περισσότερη επίδοση. Για την java πλέον τα threads αποτελούν σημαντικό παράγοντα και ο λόγος είναι το γεγονός ότι στην java δεν υπάρχει η έννοια της ασύγχρονης εισόδου και εξόδου. Ένα ακόμα σημαντικό πλεονέκτημα των threads είναι η χρησιμοποίηση πολλαπλών CPUs. Με την ραγδαία ανάπτυξη των πολλαπλών CPU και του μικρού κόστους τους αλλά και η χρησιμοποίηση βιβλιοθηκών στα λειτουργικά συστήματα που τρέχουν σε threads έκαναν τον προγραμματισμό σε threads σημαντικό παράγοντα για καλύτερη εκμετάλλευση των μηχανών που υποστήριζαν αυτά. Στην εργασία τα threads χρησιμοποιήθηκαν για τον παραλληλισμό των client αλλά και για να τρέχει το πρόγραμμα που θα στέλνει μηνύματα στο arduino αυτόματα και χωρίς την εντολή του χρήστη. Η ανάγκη για να τρέχουμε όλους τους client ταυτόχρονα και να στέλνουν όλοι τις μετρήσεις τους οδήγησαν στην χρησιμοποίηση των threads (Scott Oaks, 2004).

Κεφάλαιο 4

Ανάλυση-Σχεδίαση του GUI

- 4.1. Γενικό υπόβαθρο σεναρίου.
 - 4.2. Επικοινωνία μεταξύ WunderBar και Cloud
 - 4.3. Επικοινωνία μεταξύ Arduino και Cloud
 - 4.4. Επικοινωνία μεταξύ Cloud και Client
 - 4.5. Επικοινωνία μεταξύ Client και Gui
 - 4.6. Επικοινωνία μεταξύ Client και Database
 - 4.7. Επικοινωνία μεταξύ Gui και Database
 - 4.8. Επικοινωνία μεταξύ Gui και Arduino
-

4.1. Γενικό υπόβαθρο σεναρίου

Η εφαρμογή του "Έξυπνου Γραφείου" έγινε για την υλοποίηση του παρακάτω σεναρίου. Η εφαρμογή έχει σαν σκοπό την δημιουργία ενός αυτοματοποιημένου χώρου εργασίας που θα μπορεί ο χρήστης της να ελέγχει το περιβάλλον στο οποίο εργάζεται. Το σενάριο χρησιμοποιεί το WunderBar για να παίρνει τις μετρήσεις από τους αισθητήρες, το Arduino για να παραβαίνει στις ηλεκτρικές συσκευές του γραφείου, το Cloud της Relayr σαν μεσάζον στην επικοινωνία του WunderBar με την εφαρμογή και το Arduino. Επίσης γίνεται χρήση βάσης δεδομένων όπου αποθηκεύονται οι μετρήσεις που λαμβάνονται από τους αισθητήρες και τέλος δημιουργήθηκε η εφαρμογή του Έξυπνου Γραφείου για να λαμβάνει τις μετρήσεις, να τις παρουσιάζει σε γραφικές παραστάσεις στον χρήστη, να τις διαχειρίζεται και για επικοινωνεί με το arduino. Ο χρήστης μέσα από την εφαρμογή θα μπορεί να παρακολουθήσει τις μετρήσεις ζωντανά και να μάθει πως λειτουργεί και ποιές τεχνολογίες χρησιμοποιεί ο κάθε αισθητήρας. Για την είσοδο του ο χρήστης θα χρησιμοποιεί συγκεκριμένους κωδικούς εισόδου. Αυτό παραθέτει και περισσότερη ασφάλεια στα δεδομένα του χρήστη της

εφαρμογής. Η εφαρμογή ενώνεται με την βάση δεδομένων όπου από εκεί γίνεται ο έλεγχος της εισόδου στην εφαρμογή με μοναδικούς κωδικούς που καταχωρεί ο χρήστης. Επίσης η βάση δεδομένων χρησιμοποιείται για την αποθήκευση των μετρήσεων από τους αισθητήρες. Σε κάποιο σημείο έξω από την εφαρμογή τρέχει άλλη εφαρμογή η οποία δημιουργεί τους clients, ένα subscribe client για κάθε αισθητήρα. Οι clients εγκαθιστούν κανάλι επικοινωνίας με τους αισθητήρες.

3.2. Επικοινωνία μεταξύ WunderBar και Cloud

Το WunderBar αποτελείται από πέντε μονάδες - πλακέτες αλλά υπάρχουν και πλακέτες που υποστηρίζουν περισσότερους από ένα αισθητήρες. Ο κάθε αισθητήρας λαμβάνει τις ανάλογες μετρήσεις δωματίου και τις στέλνει στην κεντρική πλακέτα με την χρήση του BLE. Η κεντρική πλακέτα στέλνει τις μετρήσεις αυτές στο Cloud Computing που δημιούργησε η Relayr. Η κεντρική μονάδα είναι η μόνη μονάδα που ενώνεται στο internet. Η αποστολή των μετρήσεων γίνεται με το MQTT protocol και η ασφάλεια της επικοινωνίας επιτυγχάνεται με το πρωτόκολλο SSL . Η Relayr δημιούργησε το δικό της Android Application το οποίο χρησιμοποιείται για την αρχική ρύθμιση του WunderBar. Εφόσον το WunderBar ρυθμιστεί αρχίζει να στέλνει τις μετρήσεις στο Cloud. Η συχνότητα αποστολής των μετρήσεων στο Cloud κυμαίνεται περίπου στο μισό λεπτό.

4.3. Επικοινωνία μεταξύ Arduino και Cloud

Όπως αναλύθηκε και πιο πάνω το arduino χρησιμοποιήθηκε σαν subscriber. Για τον προγραμματισμό του Arduino χρησιμοποιήθηκε η εφαρμογή Arduino Genuino. Αρχικά δημιουργήθηκε το πρωτότυπο του Arduino στο Cloud. Κατά την δημιουργία του πρωτότυπου παράχθηκαν ο μοναδικός κωδικός συσκευής, ο μοναδικός κωδικός χρήστη στο MQTT, ο μοναδικός κωδικός πρόσβασης, ο μοναδικός κωδικός του client , το topic και το όνομα του server που θα γίνει η ένωση. Ο κωδικός της συσκευής χρησιμοποιείται για να γνωρίζει ο μεσάζον την συσκευή που θέλει ο subscriber να ενωθεί. Ο κωδικός του χρήστη χρησιμοποιείται για να αποφασίσει το cloud σε πιο λογαριασμό θα βρεθεί η συσκευή. Επειδή ο κωδικός της συσκευής είναι μοναδικός, ο κωδικός του χρήστη είναι ο ίδιος με τον κωδικό της συσκευής. Ο κωδικός πρόσβασης χρησιμοποιείται από το MQTT πρωτόκολλο για να αποκτήσει πρόσβαση στην συσκευή. Ο κωδικός του client αφορά το αναγνωριστικό του client. Για την εγκατάσταση της επικοινωνίας μεταξύ του cloud και του Arduino το MQTT

πρέπει να ξέρει τις δύο άκρες του καναλιού που θα εγκαταστήσει. Η μια άκρη είναι ο subscriber και η άλλη άκρη είναι ο publisher. Ωστόσο στην περίπτωση αυτή υπάρχει και μεσάζον στην επικοινωνία αυτή η οποία είναι το Cloud της Relayr. Άρα η μια άκρη είναι το Cloud και η άλλη άκρη είναι ο subscriber που στην περίπτωση μας είναι το Arduino. Το topic χρησιμοποιείται για να ξέρει το MQTT σε ποία συσκευή θα κάνει subscribe ο client και η δομή του είναι όπως περιγράφηκε πιο πάνω. Αφού θέλουμε να κάνουμε subscribe στην συσκευή ο κωδικός του client που θα χρησιμοποιηθεί στο topic είναι και ο κωδικός της συσκευής. Το όνομα του server χρησιμοποιείται για να ξέρει το MQTT σε ποίο server θα κάνει αίτηση. Οι κωδικοί αυτοί χρησιμοποιούνται στο λογισμικό που θα φορτωθεί στο arduino ώστε να φτιαχτεί η σύνδεση μεταξύ του πρωτότυπου και του arduino. Το πρόγραμμα που θα τρέξει στο Arduino δημιουργείται ένας ethernet Client ώστε ο client να δημιουργηθεί στο ethernet shield. Ακολούθως δημιουργείται μια πλήρης ρυθμισμένη μεταβλητή τύπου client. Ο τύπος client είναι σαν ένα είδος πίνακα και παίρνει σαν παραμέτρους τους κωδικούς που αναλύθηκαν πιο πάνω. Στην μέθοδο setup γίνεται η αρχικοποίηση του Ethernet Client και στην μέθοδο του loop γίνεται η επανάκτηση του client. Λόγο του ότι το arduino είναι subscriber μέσα στο χρησιμοποιείται η μέθοδος callback η οποία καθορίζει την θύρα στην οποία βρίσκεται ενωμένο το LED, λαμβάνει τα μηνύματα και τα διαχειρίζεται.

4.4. Επικοινωνία μεταξύ Cloud και Client

Το WunderBar εφόσον ενωθεί με το Cloud αρχίζει την διαδικασία αποστολής των μετρήσεων του στο Cloud. Οι μετρήσεις για να φανούν όμως χρήσιμες πρέπει να παρουσιαστούν στον χρήστη και να επεξεργαστούν. Στην ενότητα αυτή θα αναλυθεί ο τρόπος με τον οποίο γίνεται παραλαβή των μετρήσεων από το Cloud στην εφαρμογή. Η αποστολή από το Cloud στην εφαρμογή θα γίνει με την χρήση MQTT Client. Ο Client που χρειάζεται σ' αυτή την περίπτωση είναι subscriber ο οποίος θα ανοίγει κανάλι επικοινωνίας με το Cloud και θα περιμένει την αποστολή των μετρήσεων από τους αισθητήρες. Η ομάδα προγραμματισμού της Relayr δημιούργησε την βιβλιοθήκη java-sdk-1.1.0.jar και χρησιμοποιώντας τις κλάσεις της βιβλιοθήκης δημιουργούνται οι clients. Ο κάθε client που θα δημιουργηθεί θα αφορά την επικοινωνία του με κάποια μονάδα αισθητήρων με το Cloud να έχει τον ρόλο του μεσάζον. Για τον λόγο αυτό ο Client ενώνεται στο Cloud και παίρνει από εκεί τις μετρήσεις. Για την ασφάλεια σύνδεσης στο Cloud είναι αναγκαίο να χρησιμοποιηθεί το μοναδικό αναγνωριστικό του λογαριασμού στην Relayr. Αυτό βρίσκεται στο Dashboard της Relayr κάτω από την ενότητα με τα στοιχεία λογαριασμού και είναι το Token. Η υλοποίηση του

Client αποτελείται από τρεις κλάσεις. Η πρώτη κλάση καλεί τον Client, η κλάση που ξεκινά τον Client και η κλάση που κάνει το ψάξιμο των συσκευών στο Cloud. Το Token καταχωρείται στον κώδικα και στο σημείο που γίνεται η αρχικοποίηση του Token στην κλάση που καλεί το Client. Στην κλάση που ξεκινά ο Client καταχωρείται ο κωδικός της συσκευής. Η κάθε συσκευή και επιπρόσθετα το πρωτότυπο της συσκευής έχουν το δικό τους μοναδικό κωδικό αναγνώρισης. Έτσι όταν θα ενωθεί ο Client με το Cloud για να δημιουργηθεί το κανάλι επικοινωνίας χρειάζεται το πρωτότυπο στο οποίο θα κάνει subscribe ο Client. Με την εγκατάσταση του καναλιού οι μετρήσεις αρχίζουν να φτάνουν στον Client όπου εκεί γίνεται ο διαχωρισμός τους. Μια πλακέτα στο WunderBar μπορεί να έχει περισσότερους από ένα αισθητήρες αλλά οι μετρήσεις που θα στείλει θα έχουν το δικό τους αναγνωριστικό ώστε όταν φτάσουν στον Client να μπορεί να γίνει ο διαχωρισμός τους. Η κάθε μέτρηση που φτάνει στον Client περιλαμβάνει την δική της ονομασία κάτι που κάνει πιο εύκολο τον διαχωρισμό τους.

4.5. Επικοινωνία μεταξύ Client και Gui

Πιο πάνω έγινε περιγραφή της επικοινωνίας του Client με το Cloud. Αφού επιτύχει η σύνδεση της επικοινωνίας αυτής μετέπειτα οι μετρήσεις πρέπει να παρουσιαστούν και να διαχειριστούν αναλόγως από τον χρήστη. Το Gui έχει το δικό του κεντρικό πρόγραμμα που απ' εκεί οδηγείται σ' όλες τις διαδικασίες ο χρήστης του. Έτσι και οι Client ενώνονται σε κάποιο σημείο με την κεντρική σελίδα. Παρ' όλα αυτά οι Client τρέχουν στο δικό τους Thread. Επειδή η εφαρμογή θέλει να παρουσιάσει τις μετρήσεις στο χρήστη αλλά να κρατήσει την λειτουργικότητας της και για τις υπόλοιπες διαδικασίες ο κάθε Client πρέπει να τρέχει χωρίς να επηρεάζει τόσο την λειτουργία του όσο και την δική του λειτουργία. Κάθε μισό λεπτό ο αισθητήρας αποστέλλει μετρήσεις και ο client τις λαμβάνει και μετρήσεις αυτές παρουσιάζονται στην εφαρμογή. Ο χρήστης με την σειρά του θέλει να βλέπει τις μετρήσεις αυτές χωρίς να σταματούν. Για να συμβεί αυτό πρέπει ο Client να τρέχει συνεχώς αλλά καθώς τρέχει ταυτόχρονα ο χρήστης να μπορεί να προχωρήσει και σε άλλες διαδικασίες στην εφαρμογή. Η λέξη ταυτόχρονα αμέσως οδηγεί στην δημιουργία των Thread. Κάθε Client τρέχει στο δικό του Thread και μέσα από την κεντρική σελίδα του Gui καλείται το Thread ώστε να τρέξει ο Client και να γίνει παρουσίαση των μετρήσεων. Η διαδικασία καλέσματος του Thread γίνεται μόνο μια φορά αλλά το Thread τρέχει από πίσω και κάθε φορά που παίρνει μετρήσεις ο Client αλλάζει τα πεδία στα οποία ανήκει η κάθε μέτρηση που λήφθηκε. Επιπρόσθετα η εφαρμογή προσφέρει την δυνατότητα αυτόνομης ανταπόκρισης στο

περιβάλλον του γραφείου. Ο χρήστης καθορίζει κάποια όρια σε συγκεκριμένες καταστάσεις όπως ψηλές θερμοκρασίες και η εφαρμογή πρέπει με βάση τα όρια αυτά να στέλνει από μόνη της μήνυμα στο Arduino ώστε αυτό με την σειρά του να κάνει τις απαραίτητες κινήσεις. Η διαδικασία απόφασης αφορά τα άμεσα δεδομένα τα οποία θα τα πάρει από κάποιο από τους Client. Η λειτουργία όμως αυτής της διαδικασίας δεν πρέπει να εμποδίζει την λειτουργία του υπόλοιπου προγράμματος αλλά να τρέχει στο παρασκήνιο και να ενεργά από μόνο της. Για τους λόγους αυτούς χρησιμοποιήθηκε καινούργιο Thread στο οποίο θα εκτελείται η διαδικασία αυτή και θα καλείται από την κεντρική σελίδα αφού επιλεγούν από τον χρήστη τα όρια.

4.6. Επικοινωνία μεταξύ Client και Database

Εν συνεχεία με την προηγούμενη ενότητα οι Clients εκτός από τις λειτουργίες που ειπώθηκαν πιο πάνω εκτελούν και λειτουργίες αποθήκευσης των μετρήσεων σε βάση δεδομένων. Οι Clients για τους σκοπούς της εργασίας χτίστηκαν στο περιβάλλον Eclipse αλλά με την κατάλληλη υποδομή θα μπορούσε να χτιστεί σε κάποιο server και να τρέχουν ξεχωριστά από τα υπόλοιπα προγράμματα. Παρ' όλα αυτά σύμφωνα με το σενάριο στόχος της εργασίας είναι η αποθήκευση των μετρήσεων σε βάση δεδομένων ώστε να κρατείται ιστορικό. Σαν μια αναπαράσταση των Client σε server δημιουργήθηκαν σε ξεχωριστό έργο όλοι όπου στόχος τους είναι η προώθηση των μετρήσεων στην βάση δεδομένων. Για τον σκοπό αυτό δημιουργήθηκε μια βάση δεδομένων στο εργαλείο SQL Management Studio στην οποία αποθηκεύονται οι μετρήσεις ανά λεπτό. Με απλά Query και κάνοντας χρήση της βιβλιοθήκης που περιέχουν κλάσεις από το JDBC οι Clients προωθούν τις μετρήσεις στην Βάση. Οι μετρήσεις λαμβάνονται σε τύπο γραμματοσειρά και μετατρέπονται στον κατάλληλο τύπο όπου θα γίνει και η αποθήκευση τους στην βάση. Η βάση δεδομένων αποτελείται από τους πίνακες Gyroscope, Light, Microphone, Thermometer και User. Ο πίνακας Gyroscope έχει τα πεδία Angular Speed, Acceleration Day και Hour. Ο πίνακας Light έχει τα πεδία Proximity, Color, Luminosity, Day και Hour. Ο πίνακας Microphone έχει τα πεδία Noise Level, Day και Hour. Ο πίνακας Thermometer έχει τα πεδία Temperature, Humidity, Day και Hour. Ενώ ο User περιέχει τα πεδία Username και Password και αφορά τους χρήστες της εφαρμογής. Κάθε εγγραφή στους πίνακες Gyroscope, Thermometer, Microphone και Light αποτελεί είναι και μια μέτρηση με την ημερομηνία και ώρα που έγινε η καταγραφή της. Στον πίνακα User καταγράφονται οι χρήστες στους οποίους θα δίνεται πρόσβαση στην εφαρμογή. Υπάρχει αρχικός χρήστης ώστε κάποιος που θα χρησιμοποιήσει

την εφαρμογή ή την βάση δεδομένων για πρώτη φορά να μπορεί να έχει πρόσβαση σ' αυτήν και μετέπειτα να μπορεί να αλλάξει ή να προσθέσει κάποιον άλλο χρήστη. Ο Client λαμβάνει μετρήσεις κάθε μισό λεπτό ωστόσο επειδή δεν είναι απαραίτητος τόσο μεγάλος αριθμός μετρήσεων οι Client αποθηκεύουν μία μέτρηση κάθε 120 μετρήσεις. Η 120η μέτρηση επιτυγχάνει την αποθήκευση μιας μέτρησης κάθε ένα λεπτό.

4.7. Επικοινωνία μεταξύ Gui και Database

Σε συνέχεια με την προηγούμενη ενότητα και αφού έγινε αποθήκευση των δεδομένων στην βάση η εφαρμογή προσφέρει την δυνατότητα παρουσίασης του ιστορικού ανά μέρα ανά μήνα και ανά χρόνο. Ο χρήστης μπορεί να παρακολουθήσει τις μετρήσεις και να βγάλει τα συμπεράσματα του. Μέσα από τα συμπεράσματα που θα εξάγει μπορεί να λάβει μέτρα ώστε να μειωθούν ή να αυξηθούν οι τιμές των μετρήσεων. Στο σημείο αυτό θα κάνει χρήση ενός από τα Thread που αναλύθηκαν πιο πάνω. Ο χρήστης μπορεί να περιηγηθεί στην κεντρική σελίδα και να μεταφερθεί στην σελίδα που θα καθορίσει τα όρια που θα λαμβάνουν οι μετρήσεις. Μόλις μια μέτρηση περάσει το κάτω ή πάνω όριο τότε η εφαρμογή αντιδρά από μόνη της και στέλνει το κατάλληλο μήνυμα στο arduino. Χρησιμοποιώντας την βιβλιοθήκη sqljdbc4.jar επιτυγχάνεται η σύνδεση μεταξύ της εφαρμογής και της βάσης δεδομένων και δίνεται η δυνατότητα της εξαγωγής των μετρήσεων από την βάση και της εισόδου τους στην εφαρμογή. Η σύνδεση αυτή γίνεται πάνω από το HTTP πρωτόκολλο. Εκτός από την παρουσίαση των δεδομένων σαν απλές μετρήσεις προσφέρεται επίσης η αναπαράσταση γραφικών παραστάσεων. Με την χρήση της βιβλιοθήκης jfreechart-1.0.19.jar δημιουργείται καινούργιο πλαίσιο στο οποίο δίνεται η δυνατότητα να γίνει αλλαγή στα ονόματα των στηλών, δίνεται το όνομα και το είδος της γραφικής παράστασης. Οι μετρήσεις που χρησιμοποιούνται στην γραφική παράσταση αφορά το ιστορικό που βρίσκεται φυλαγμένο στην βάση δεδομένων. Αφού έγινε εφικτή η σύνδεση της εφαρμογής με την βάση με ένα απλό query τα δεδομένα εισέρχονται στην εφαρμογή και μετατρέπονται σε γραφικές παραστάσεις. Για τους σκοπούς της εργασίας το query που εισάγει τις μετρήσεις βρίσκει και τον μέσο όρο κάθε μέρας. Έτσι στην γραφική παράσταση έχουμε στον άξονα των χ τις μέρες και στον άξονα των ψ τον μέσο όρο ανά μέρα. Όπως προαναφέρθηκε εκτός από τους πίνακες που κρατούνται οι μετρήσεις στην βάση δεδομένων υπάρχει και ο πίνακας που κρατά τους χρήστες που θα έχουν πρόσβαση στην εφαρμογή. Ένας χρήστης για να μεταφερθεί στην κύρια σελίδα της εφαρμογής πρέπει να έχει πρόσβαση με λογαριασμό στην εφαρμογή. Οι

λογαριασμοί βρίσκονται στην βάση δεδομένων στον πίνακα User. Η πρώτη σελίδα που θα παρουσιαστεί στον χρήστη ανοίγοντας την εφαρμογή είναι η σελίδα που του ζητά να βάλει τους κωδικούς του ώστε να αποκτήσει πρόσβαση στην κύρια σελίδα της εφαρμογής. Η πρόσβαση δίνεται ύστερα από έλεγχο στην βάση δεδομένων. Αν υπάρχουν οι κωδικοί σύνδεσης που έδωσε ο χρήστης τότε θα επιτραπεί η είσοδος και θα μεταφερθεί στην κύρια σελίδα του προγράμματος αλλιώς θα παραμείνει στην ίδια σελίδα μέχρι να δώσει σωστούς κωδικούς. Αυτό επιτυγχάνει περισσότερο έλεγχο στην πρόσβαση των δεδομένων του χρήστη και επιπρόσθετα αυξάνει την γενική ασφάλεια της εφαρμογής.

4.8. Επικοινωνία μεταξύ Gui και Arduino

Στις πιο πάνω ενότητες αναλύθηκε ο τρόπος με τον οποίο θα γίνεται αλληλεπίδραση με Arduino , Client και Cloud. Το Arduino θα αποτελεί τον subscriber στην επικοινωνία αυτή και το Cloud τον μεσάζοντα και κάποιος πρέπει να διατηρεί τον ρολό του publisher. Το Gui θα διαδραματίζει τον ρόλο του publisher στην επικοινωνία μεταξύ του Gui και Arduino. Το Gui θα στέλνει τα κατάλληλα μηνύματα στο πρωτότυπο του Cloud και από εκεί θα προωθούνται στον subscriber, το Arduino. Το Arduino με την σειρά του θα λαμβάνει τα μηνύματα αυτά και θα ενεργοποιεί το LED. Μέσα στην εφαρμογή υπάρχουν δύο επιλογές για εγκατάσταση του καναλιού μεταξύ του Gui με το Arduino. Ο χρήστης έχει την δυνατότητα είτε να στείλει το μήνυμα στο Arduino είτε χειροκίνητα είτε να καθορίσει το όριο στο οποίο θα γίνει η αποστολή του μηνύματος. Μελετώντας το ιστορικό των μετρήσεων οι οποίες παρουσιάζονται στην επικοινωνία μεταξύ Gui και Database ο χρήστης μπορεί χειροκίνητα να στείλει το κατάλληλο μήνυμα στο Arduino. Επιπρόσθετα παρέχεται η δυνατότητα το μήνυμα να αποστέλλεται από την εφαρμογή αυτόματα. Η χρήστης καθορίζει ένα πάνω ή κάτω όριο στο οποίο θέλει να μην ξεπερνούν οι μετρήσεις του και η εφαρμογή στέλνει το μήνυμα αυτόνομα στο arduino ώστε να μειώσει ή να αυξήσει τις μετρήσεις αυτές.

Κεφάλαιο 5

Παρουσίαση Λειτουργιών του GUI

5.1. Είσοδος στην εφαρμογή

5.1.1. Επικοινωνία μεταξύ Gui και Database

5.2. Κεντρική Σελίδα

5.2.1. Καρτέλες Πληροφοριών

5.2.2. Menu

5.2.2.1. Readings

5.2.2.1.1. Επικοινωνία μεταξύ Gui και Database

5.2.2.2. Publish

5.2.2.2.1. Επικοινωνία μεταξύ Gui και Arduino

5.2.2.3. Rules

5.2.2.3.1. Επικοινωνία μεταξύ Gui και Arduino

5.2.2.4. Charts

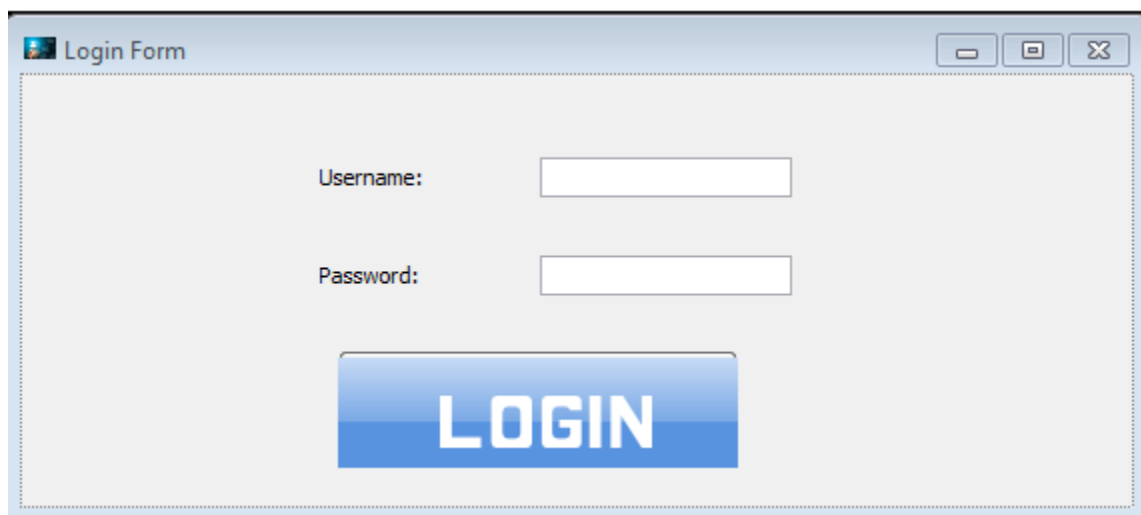
5.2.2.4.1. Επικοινωνία μεταξύ Gui και Database

5.3. About

5.4. Είσοδος στην εφαρμογή

5.4.1. Επικοινωνία μεταξύ Gui και Database

Η Αρχική σελίδα της εφαρμογής αφορά την επικοινωνία μεταξύ Gui και Database. Ο χρήστης καλείται να κάνει login. Η παρακάτω εικόνα δείχνει το πως θα φαίνεται η αρχική σελίδα. Ο χρήστης βάζει τους κωδικούς του στα κενά πεδία με πρώτο το όνομα χρήστη και δεύτερο τον κωδικό του.

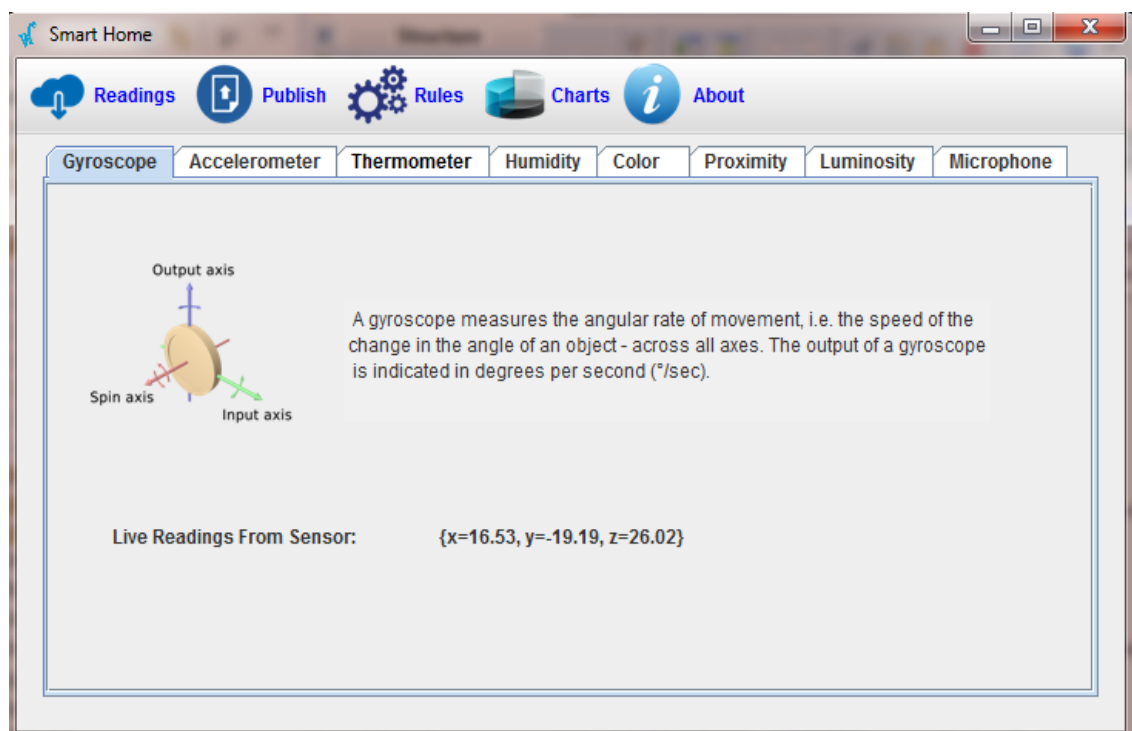


The image shows a screenshot of a software window titled "Login Form". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is light gray and contains two text input fields. The first field is labeled "Username:" and the second is labeled "Password:". Below these fields is a large, blue, rectangular button with the word "LOGIN" written in white, bold, capital letters.

5.2. Κεντρική Σελίδα

5.2.1. Καρτέλες Πληροφοριών

Αφού οι κωδικοί είναι σωστοί το πρόγραμμα μεταφέρεται στην αρχική σελίδα. Η αρχική σελίδα αποτελείται από ενότητες. Η μια ενότητα είναι το Menu και η δεύτερη χωρίζεται σε οχτώ παράθυρα. Στο κάθε παράθυρο περιγράφεται η λειτουργία του κάθε αισθητήρα όπως επίσης γίνεται και η ζωντανή παρουσίαση των μετρήσεων τους. Η εικόνα παρακάτω δείχνει τόσο την ενότητα των παραθύρων όσο και του μενού. Για την παρουσίαση των μετρήσεων γίνεται χρήση της επικοινωνίας του Client με το Gui όπως περιγράφηκε και πιο πάνω. Το πρώτο παράθυρο περιγράφει την λειτουργία του γυροσκοπίου και παρουσιάζει τις μετρήσεις του. Το δεύτερο παράθυρο περιγράφει την λειτουργία του Επιταχυνσιόμετρου και παρουσιάζει τις μετρήσεις του. Στο τρίτο παράθυρο γίνεται περιγραφή του Θερμόμετρου και παρουσίαση των μετρήσεων του. Στο τέταρτο παράθυρο περιγράφεται ο αισθητήρας υγρασίας. Παρακάτω περιγράφονται οι αισθητήρες του χρώματος της απόστασης και της φωτεινότητας οι οποίοι ανήκουν στην ίδια μονάδα ,υλικό. Στο τελευταίο παράθυρο περιγράφεται το μικρόφωνο. Για να μεταφερθεί στα παράθυρα ο χρήστης και να δει τις πληροφορίες - παρόμοιες με αυτές στη πιο κάτω εικόνα - φτάνει να πατήσει στην καρτέλα του αισθητήρα που χρειάζεται.



5.2.2. Menu

5.2.2.1. Readings

Στο μενού ανήκουν όλες οι εξωτερικές συνδέσεις. Μια από τις συνδέσεις αυτές είναι η σύνδεση με την βάση δεδομένων. Αυτό το κομμάτι υλοποιείται στο Readings. Στην ενότητα αυτή υλοποιείται η επικοινωνία μεταξύ της βάσης δεδομένων και του Gui. Οι μετρήσεις φυλάγονται στην βάση δεδομένων κάθε ένα λεπτό. Το πρόγραμμα δίνει την δυνατότητα στον χρήστη να αντλήσεις αυτές τις μετρήσεις και να τις κατηγοριοποιήσει σε μετρήσεις ανά μέρα, ανά μήνα και ανά χρονιά. Ο χρήστης επιλέγει ένα από τα παιδιά με τα οποία θέλει να κατηγοριοποιήσει τα δεδομένα του και αναλόγως εμφανίζονται τα κατάλληλα πεδία στα οποία και θα καταχωρήσει τις τιμές με τις οποίες θα γίνει η επιλογή των δεδομένων που θέλει να του εμφανιστούν. Η παρακάτω εικόνα δείχνει τις τιμές που αποθηκεύτηκαν στην βάση δεδομένων από την μονάδα του γυροσκοπίου και του επιταχυνσιόμετρου για το έτος 2016.

AngularSpeed	Acceleration	Day
{x=11.12, y=-21.27, z=40.88}		2016-04-08
	{x=0.98, y=-0.11, z=0.13}	2016-04-08
{x=10.24, y=-19.47, z=59.28}		2016-04-08
	{x=0.98, y=-0.12, z=0.13}	2016-04-08
{x=15.54, y=-27.59, z=70.26}		2016-04-08
	{x=0.98, y=-0.11, z=0.13}	2016-04-08
{x=14.95, y=-25.88, z=72.93}		2016-04-08
	{x=0.98, y=-0.11, z=0.12}	2016-04-08
{x=12.8, y=-21.69, z=71.54}		2016-04-08
	{x=0.98, y=-0.11, z=0.13}	2016-04-08
{x=16.57, y=-17.72, z=30.51}		2016-04-08
	{x=0.98, y=-0.11, z=0.12}	2016-04-08
{x=15.19, y=-17.09, z=14.64}		2016-04-08

Year

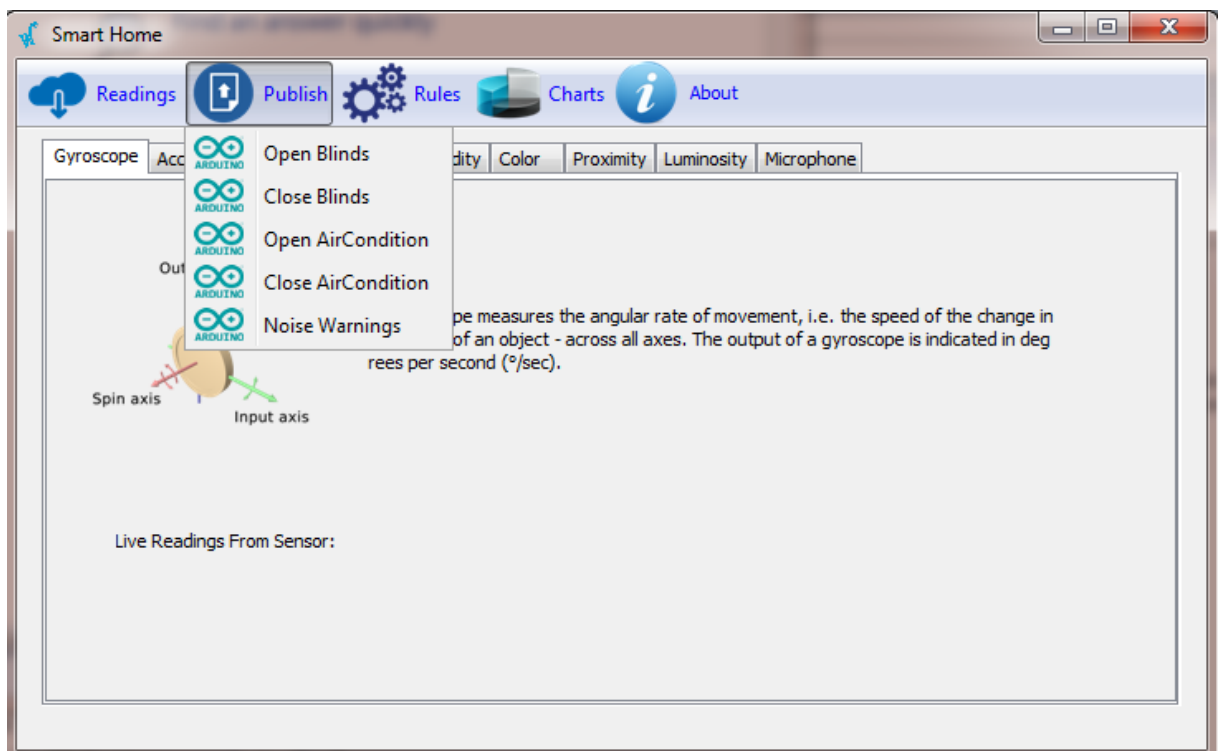
Find By Day

Find By Month

Find By Year

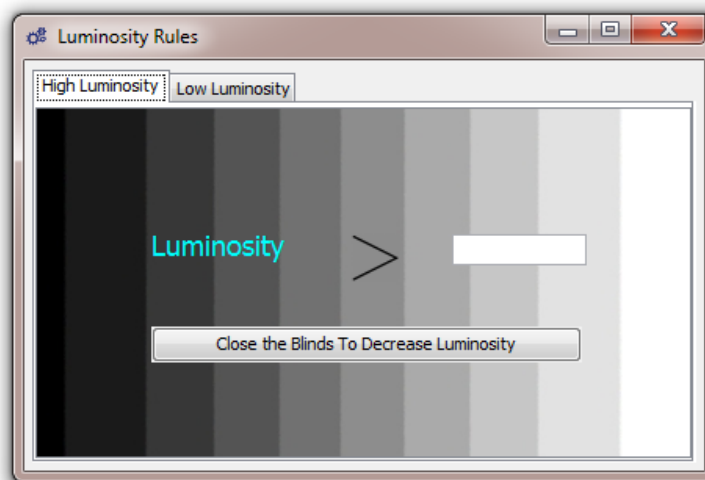
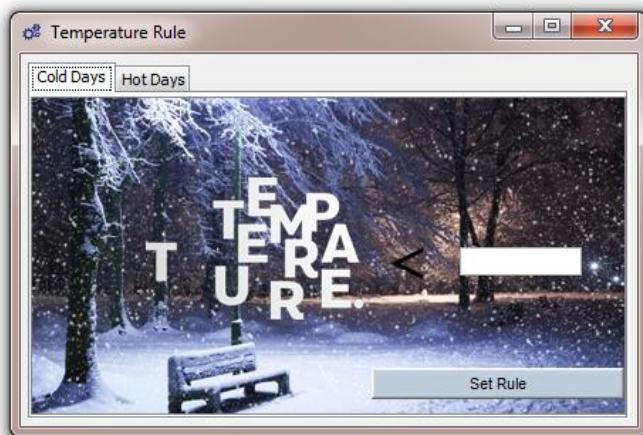
5.2.2.2. Publish

Η επόμενη ενότητα στο μενού είναι το publish. Στην ενότητα αυτή υλοποιείται η επικοινωνία μεταξύ arduino και Gui. Ο χρήστης της εφαρμογής αφού μελετήσει τις μετρήσεις που θα αντλήσει στα Readings μπορεί με το πάτημα ενός κουμπιού να δώσει εντολή στο arduino για να αλλάξει είτε την θερμοκρασία είτε την υγρασία είτε την φωτεινότητα του δωματίου είτε για να στείλει ειδοποίηση για ψηλά επίπεδα θορύβου. Πατώντας στο κουμπί του μενού Publish ανοίγει η λίστα με τα κουμπιά Open Blinds, Close Blinds, Open AirCondition, Close AirCondition και Noise Warnings. Τα πρώτα δύο κουμπιά με το πάτημα τους δίνουν εντολή στο arduino να ανοίξει και να κλείσει τις περσίδες αντίστοιχα με σκοπό την μείωση ή την αύξηση της φωτεινότητας του δωματίου. Τα επόμενα δύο κουμπιά στέλνουν το μήνυμα στο arduino να ανοίξει ή να κλείσει το κλιματιστικό αντίστοιχα το κάθε κουμπί με σκοπό την μείωση ή αύξηση της θερμοκρασίας και της υγρασίας. Το τελευταίο κουμπί στέλνει το μήνυμα στο arduino να ανάψει την φωνητική προειδοποίηση για αυξημένο θόρυβο στο δωμάτιο. Η ανάγκη για αλλαγή των συνθηκών στο δωμάτιο με το πάτημα απλώς ενός κουμπιού είναι και ο λόγος για τον οποίο έγινε τόσο απλή η αλληλεπίδραση του χρήστη. Η παρακάτω εικόνα δείχνει το πού βρίσκουμε το κουμπί publish και το πώς θα ανοίξει η λίστα μετά το πάτημα του publish.



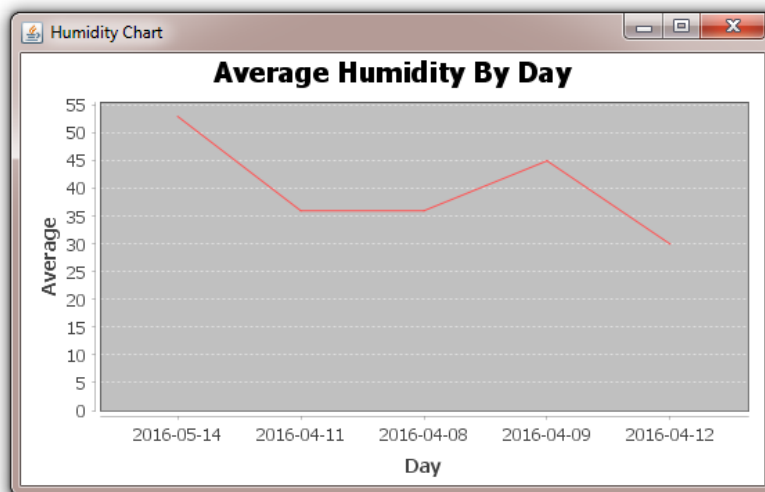
5.2.2.3. Rules

Στο επόμενο κουμπί του μενού γίνεται η υλοποίηση της αυτόνομης ενέργειας της εφαρμογής στο δωμάτιο. Όπως αναλύθηκε και πιο πάνω η εφαρμογή επιδρά στο δωμάτιο ώστε να αλλάξει μερικά δεδομένα. Ωστόσο για να επιδράσει χρειάζεται κάποιες ρυθμίσεις - παραμέτρους από τον χρήστη της. Οι παράμετροι αυτοί έχουν να κάνουν με τα όρια στα οποία θέλει ο χρήστης από την εφαρμογή να ενεργά αυτόνομα. Καθορίζει τα κάτω ή πάνω όρια μετρήσεων είτε της θερμοκρασίας είτε της υγρασίας είτε της φωτεινότητας και η εφαρμογή αναλόγως στέλνει από μόνη της τα ίδια μηνύματα με αυτά που μπορεί ο χρήστης να στείλει στο κουμπί publish. Η εφαρμογή κάνει την δουλειά του χρήστη εξυπηρετώντας τον. Πιο κάτω οι εικόνες με την σειρά Θερμοκρασία, Υγρασία και Φωτεινότητα δείχνουν το περιβάλλον του καθενός στο οποίο γίνεται καθορισμός των ορίων.

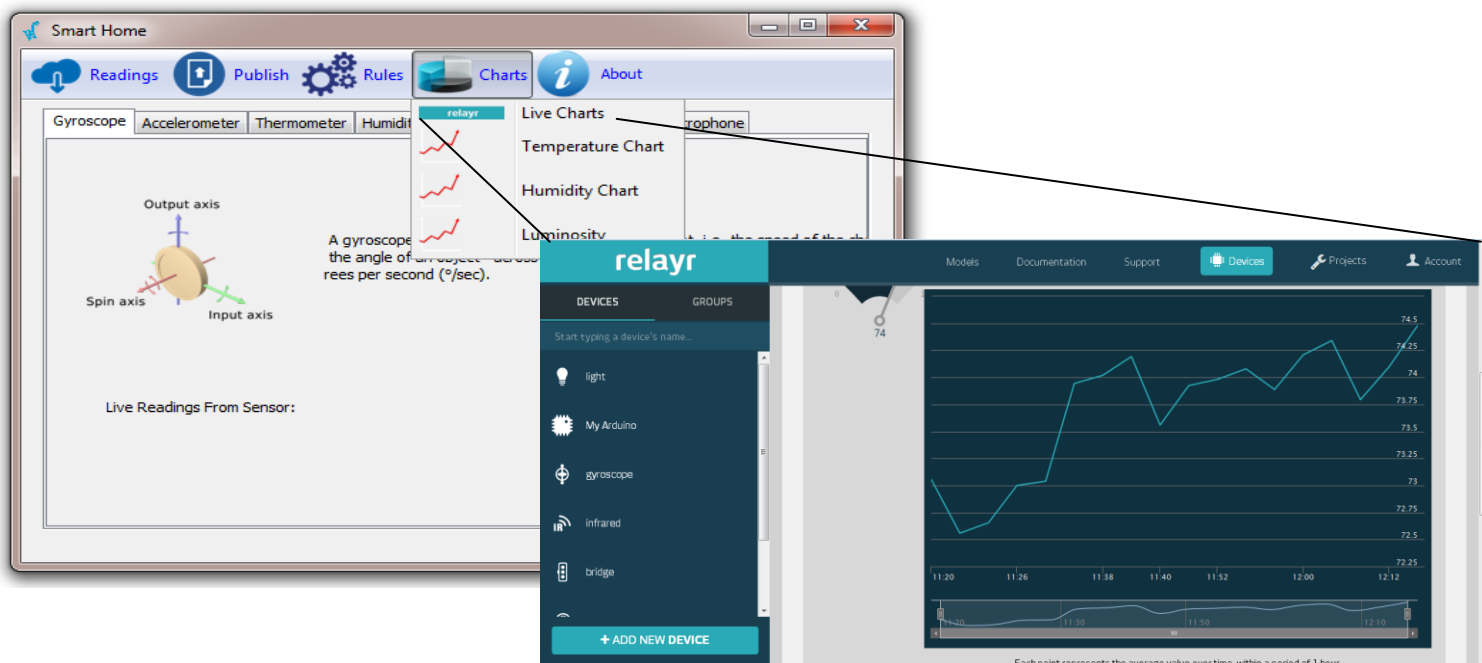


5.2.2.4. Chart

Το τελευταίο κουμπί στο μενού που αφορά και την τελευταία εξωτερική σύνδεση της εφαρμογής. Η εφαρμογή προσφέρει την δυνατότητα στον χρήστη να δει τις μετρήσεις του σε γραφικό περιβάλλον. Η εφαρμογή επικοινωνεί με την βάση δεδομένων και ανακτά όλες τις μετρήσεις. Ακολούθως δημιουργείται η γραφική παράσταση. Οι γραφικές παραστάσεις που προσφέρονται στην εφαρμογή είναι για την υγρασία, την θερμοκρασία, και την φωτεινότητα. Στο άξονα των ψ οι μετρήσεις σε μέσο όρο της κάθε μέρας ενώ στον άξονα των χ βρίσκονται οι μέρες. Στην παρακάτω εικόνα φαίνεται η γραφική παράσταση του μέσου όρου υγρασίας της κάθε μέρας.



Ωστόσο η Relayr μέσω της διαδικτυακής της κονσόλας υποστηρίζει και ζωντανές γραφικές παραστάσεις. Έτσι και η εφαρμογή δίνει την δυνατότητα στον χρήστη να μεταφερθεί μέσα από τον browser του στην συγκεκριμένη ιστοσελίδα και να παρακολουθήσει τις μετρήσεις ζωντανά.



5.3. About

Στο κουμπί αυτό γίνεται σύντομη περίληψη της εφαρμογής ώστε ο χρήστης να ξέρει πως να χρησιμοποιήσει την εφαρμογή. Επίσης δίνονται περαιτέρω πληροφορίες τόσο για την επικοινωνία του χρήστη με τον δημιουργό όσο και με την εταιρία Relayr. Τέλος παρουσιάζονται οι δικαιούχοι κάθε ενότητας για αποφυγή υποκλοπής κώδικα.

Κεφάλαιο 6

Συζήτηση

6.1. Μελλοντικές επεκτάσεις της Εφαρμογής

6.2. Βελτιώσεις

6.1. Μελλοντικές επεκτάσεις της Εφαρμογής

Η εφαρμογή κτίστηκε στα πλαίσια Διπλωματικής εργασίας και έγινε σε συγκεκριμένο σενάριο και ύστερα από συνεννόηση με τον καθηγητή. Το σενάριο αυτό αν και είναι μικρό αλλά ολοκληρωμένο λειτουργικά μπορεί να επεκταθεί ακόμη περισσότερο. Οι επεκτάσεις που μπορούν να γίνουν αφορά περισσότερο την λειτουργικότητα της εφαρμογής. Η εφαρμογή μπορεί να γίνει ακόμη πιο έξυπνη. Επεκτάσεις που μπορούν να γίνουν είναι η φωνητική εντολή του χρήστη, η φωνητική απάντηση της εφαρμογής στον χρήστη αλλά και η πιο γρήγορη επεξεργασία των δεδομένων της. Με εγκατάσταση ιδιού προγράμματος ο χρήστης θα δίνει εντολή στην εφαρμογή να ξεκινήσει, να ανοίξει κάποιο παράθυρο ή καρτέλα αλλά και να θέσει ερωτήματα στην εφαρμογή. Η εφαρμογή με την σειρά της θα κάνει την κατάλληλη έρευνα και θα παρουσιάσει είτε φωνητικά είτε ανοίγοντας τα δεδομένα σε καινούργιο παράθυρο. Αυτό θα επιφέρει ακόμη περισσότερη ευκολία χρήσης αλλά και θα γίνει πιο ελκυστική η εφαρμογή. Επίσης στο έργο θα μπορούσε να γίνει αλλαγή στο arduino με πρόγραμμα φωνητικής αλληλεπίδρασης.

6.2. Βελτιώσεις

Για τους σκοπούς της Διπλωματικής εργασίας έγινε μια απλή αναπαράσταση του πώς παίρνει τις εντολές το arduino με την χρήση ενός LED για κάθε εντολή. Ωστόσο για την εισαγωγή της εφαρμογής σε ένα πραγματικό σενάριο χρειάζονται περισσότερο υλικό από αυτό που χρησιμοποιήθηκε. Για τις ηλεκτρικές συνδέσεις μεταξύ arduino και ηλεκτρικών συσκευών χρειάζεται και ανάλογες αντιστάσεις αλλά και το ανάλογο relay που θα βοηθήσει στις συνδέσεις.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Alexandros Marinos, Gerard Briscoe, 2009. *Community Cloud Computing*. © Springer International Publishing AG, Τόμος 5931 , pp. 472-484.
2. Alexandros Marinos, G. B., n.d. *Community Cloud Computing*. [Online]
Available at: http://link.springer.com/chapter/10.1007/978-3-642-10665-1_43#page-1
[Accessed 5 03 2016].
3. Arduino, 2016. *Arduino cc*. [Ηλεκτρονικό]
Available at: <https://www.arduino.cc/>
[Πρόσβαση 21 4 2016].
4. Brian O'Connell, A. S.-C. , M. G., 2007. *IBM developer works*. [Ηλεκτρονικό]
Available at: <http://www.ibm.com/developerworks/lotus/library/expeditor-mqtt/>
[Πρόσβαση 2 5 2016].
5. Few, S., 2006. Information Dashboard Design: The effective visual communication of data. *O'reilly*, pp. 1-37.
6. Keiko Hashizume, D. G. R. E. F.-M. E. B. F., 2013. An analysis of security issues for cloud computing. *Journal of Internet Services and Applications*, 4(5).
7. Relayr, 2016. *Relayr*. [Ηλεκτρονικό]
Available at: <http://docs.relayr.io/>
[Πρόσβαση 2016].
8. Scott Oaks, H. W., 2004. *Java Threads*. 3 επιμ. s.l.:O'reilly.
9. Vouk, M. A., 2008. Cloud Computing – Issues, Research and Implementations. *Journal of Computing and Information Technology*, 4(16), p. 235–246.