

**ΑΕΙ ΠΕΙΡΑΙΑ ΤΤ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

## **Έξυπνο Σπίτι – SmartHome**

Σχεδίαση και ανάπτυξη πλακέτας Arduino για την διαχείριση έξυπνων σπιτιών με χρήση εφαρμογής Android

**Μαρίνα Αρντίτ**

**Μπικίκου Ευγενία**

**Εισηγητής: Έλληνας Ιωάννης, Καθηγητής**

**ΑΙΓΑΛΕΩ**

**ΔΕΚΕΜΒΡΙΟΣ 2015**







**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Έξυπνο Σπίτι – Smart Home**

**Μαρίνα Αρντίτ     Α.Μ:36165**

**Μπικίκου Ευγενία   Α.Μ:37094**

**Εισηγητής:**

**Έλληνας Ιωάννης**

**Εξεταστική Επιτροπή:**

**Ημερομηνία εξέτασης: .....**



## ΕΥΧΑΡΙΣΤΙΕΣ

Σε αυτό το σημείο και ολοκληρώνοντας τον κύκλο σπουδών μας, με την παρούσα πτυχιακή εργασία, στο Τμήμα Ηλεκτρονικών Υπολογιστικών Συστημάτων του ΑΤΕΙ Πειραιά, θα θέλαμε να εκφράσουμε τις ευχαριστίες μας σε ορισμένους ανθρώπους που μας στήριξαν με τον τρόπο τους.

Αρχικά θα θέλαμε να ευχαριστήσουμε τους γονείς μας, για την αμέριστη συμπαράσταση που μας έδειξαν όλο αυτό το χρονικό διάστημα και εν συνεχεία τον επιβλέπων καθηγητή μας κ. Ιωάννη Έλληνα, για την άριστη συνεργασία την οποία είχαμε, καθώς επίσης και για τη συνεχή καθοδηγούμενη επίβλεψη της πτυχιακής εργασίας, καθ' όλη τη διάρκεια της εκπόνησής της.

Τέλος θα θέλαμε να εκφράσουμε τις ευχαριστίες όλους τους καθηγητές του τμήματος, που φρόντισαν να μας μεταλαμπαδεύσουν την τεχνογνωσία εκείνη, ώστε να μπορέσουμε να ανταπεξέλθουμε στις σημερινές ανάγκες της κοινωνίας της πληροφορικής.





## ΠΕΡΙΛΗΨΗ

Τα έξυπνα συστήματα μπορούν να ελέγχουν εκτός από τις ηλεκτρολογικές και μηχανολογικές εγκαταστάσεις, οικιακές συσκευές και συσκευές πολυμέσων (multimedia), δημιουργώντας ένα ενοποιημένο σύστημα. Στις τελευταίες εντάσσονται οι συσκευές τηλεπικοινωνιών, τα ηχοσυστήματα αλλά και οι τηλεοράσεις του σπιτιού. Συνδυάζοντας όλες αυτές τις ανεξάρτητες, αρχικά, εγκαταστάσεις σε μία κοινή βάση, αποκτάται πλήρης έλεγχος της οικίας ο οποίος μπορεί να διεξαχθεί ακόμα και από μακριά.

Στα πλαίσια της πτυχιακής εργασίας δημιουργήθηκε μια εφαρμογή Android για τη διαχείριση ενός έξυπνου σπιτιού που ελέγχεται από ένα μικροεπεξεργαστή Arduino. Για το έξυπνο σπίτι, υλοποιήθηκε μία πλακέτα Arduino η οποία δέχεται «εντολές» ενεργοποίησης και απενεργοποίησης των οικιακών συσκευών, μέσω εφαρμογής Android.

Αρχικά γίνεται μία εισαγωγή στα «smart homes» και δίνεται ένας ορισμός αυτών. Έπειτα παρουσιάζονται ορισμένα από τα πλεονεκτήματα των έξυπνων σπιτιών, που αφορούν στην εξοικονόμηση ενέργειας, την ασφάλεια και την διευκόλυνση της καθημερινότητας των χρηστών. Στη συνέχεια της εργασίας, γίνεται μία εισαγωγή στα Arduino και Android και στα βασικά τους στοιχεία. Τέλος παρουσιάζεται η υλοποίηση του συστήματος (πλακέτας Arduino και εφαρμογής Android).

### Λέξεις Κλειδιά

Έξυπνο Σπίτι, Arduino, Android, Διαχείριση Πριζών και των Φώτων, Διαχείριση Συστήματος Συναγερμού, Διαχείριση Κεντρικής Θέρμανσης, Διαχείριση Θερμοσίφωνα



## Summary

Smart systems are able to control not only electrical and mechanical facilities, but also home compliances and multimedia devices, creating a united system. Multimedia devices consist of telecommunication devices, sound systems and home TV sets. Combining all those independent facilities on a common base, you gain full control of a house even from a far distance.

An Android application has been developed based on the purpose of this paper to prove how efficient the management of a 'smart home' is. In order to support the idea of the 'smart home', gateway was partially implemented as an interface between the application and those manageable devices.

Initially, an introduction based on the main characteristics of a "smart home" takes place which focuses on their scientific definition. Afterwards, the advantages of the "smart homes" are analyzed in terms of energy saving, safety and improving the everyday life of their residents by giving specific examples of the amenities that the latter enjoy during their accommodation in a "smart home". Thereinafter, an introduction to the fundamental elements of the Android application takes place and which technologies were implemented in order to design this system.

## Keywords

Smart home, Arduino, Android, Jack and lights management, Alarm system management, Central heating management, Water heater management.



## ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ.....	vii
ΠΕΡΙΛΗΨΗ .....	ix
Summary .....	xi
ΠΕΡΙΕΧΟΜΕΝΑ .....	xiii
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ.....	xv
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....	xvxi
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ ΣΤΑ SMART HOMES .....	<b>Error! Bookmark not defined.</b>
1.1 Πλεονεκτήματα των Smart Homes.....	<b>Error! Bookmark not defined.</b>
1.2 Έλεγχος Smart Homes .....	<b>Error! Bookmark not defined.</b>
1.3 Σκοπός και δομή της εργασίας.....	<b>Error! Bookmark not defined.</b>
ΚΕΦΑΛΑΙΟ 2: ΕΙΣΑΓΩΓΗ ΣΤΟ ARDUINO .....	<b>Error! Bookmark not defined.</b>
2.1 Τί είναι το Arduino;.....	<b>Error! Bookmark not defined.</b>
2.2 Το Arduino Mega 2560r3.....	<b>Error! Bookmark not defined.</b>
2.2.1 Είσοδοι - Έξοδοι.....	<b>Error! Bookmark not defined.</b>
2.2.2 Τροφοδοσία.....	<b>Error! Bookmark not defined.</b>
2.2.3 Ενσωματωμένα κουμπιά και LED .....	<b>Error! Bookmark not defined.</b>
2.3 Arduino IDE και σύνδεση με τον υπολογιστή .....	<b>Error! Bookmark not defined.</b>
2.4 Γλώσσα προγραμματισμού .....	<b>Error! Bookmark not defined.</b>
ΚΕΦΑΛΑΙΟ 3: ΕΙΣΑΓΩΓΗ ΣΤΟ ANDROID .....	<b>Error! Bookmark not defined.</b>
3.1 Τί είναι το Android;.....	<b>Error! Bookmark not defined.</b>
3.2 Δομικά στοιχεία αρχιτεκτονικής .....	<b>Error! Bookmark not defined.</b>
3.3 Ανατομία μιας Εφαρμογής Android.....	<b>Error! Bookmark not defined.</b>
3.4 Εφαρμογή App Inventor .....	<b>Error! Bookmark not defined.</b>
3.4.1 Ιστορική αναδρομή του App Inventor.....	<b>Error! Bookmark not defined.</b>
3.4.2 Προαπαιτούμενα για τη χρήση του App Inventor .....	<b>Error! Bookmark not defined.</b>

3.4.3 Περιβάλλον του App Inventor .....	<b>Error! Bookmark not defined.</b>
3.4.4 Συναρτήσεις του App Inventor .....	<b>Error! Bookmark not defined.</b>
ΚΕΦΑΛΑΙΟ 4: ΥΛΟΠΟΙΗΣΗ ΕΡΓΑΣΙΑΣ .....	<b>Error! Bookmark not defined.</b> <u>3</u>
4.1 Υλοποίηση πλακέτας Android .....	<b>Error! Bookmark not defined.</b> <u>3</u>
4.1.1 Κομμάτια πλακέτας .....	<b>Error! Bookmark not defined.</b> <u>3</u>
4.1.2 Βασικά κομμάτια κώδικα για το Arduino .....	39
4.2 Κώδικας Εφαρμογής Android .....	<b>Error! Bookmark not defined.</b> <u>0</u>
4.2.1 Εμφάνιση επιλογών του Σαλονιού .....	<b>Error! Bookmark not defined.</b> <u>2</u>
4.2.2 Φως .....	<b>Error! Bookmark not defined.</b> <u>2</u>
4.2.3 Πρίζες .....	<b>Error! Bookmark not defined.</b> <u>4</u>
ΚΕΦΑΛΑΙΟ 5: MANUAL ΕΦΑΡΜΟΓΗΣ .....	<b>Error! Bookmark not defined.</b> <u>5</u>
5.1 Εγκατάσταση εφαρμογής .....	<b>Error! Bookmark not defined.</b> <u>5</u>
5.2 Είσοδος στην εφαρμογή .....	<b>Error! Bookmark not defined.</b> <u>7</u>
5.3 Ρύθμιση Σαλονιού .....	49
5.4 Ρύθμιση Διαδρόμου .....	<b>Error! Bookmark not defined.</b> <u>3</u>
5.5 Ρύθμιση Κουζίνας .....	<b>Error! Bookmark not defined.</b> <u>5</u>
5.6 Ρύθμιση Δωματίων .....	59
5.7 Ρύθμιση Τουαλέτας .....	73
5.8 Ρυθμιση Θερμοκρασίας .....	76
5.9 Ρύθμιση Ασφάλειας .....	77
5.10 Ρύθμιση Υπενθύμησης .....	80
ΠΑΡΑΡΤΗΜΑ Α: ΚΩΔΙΚΑΣ ARDUINO .....	83
ΠΑΡΑΡΤΗΜΑ Β: BLOCKS ΤΟΥ ANDROID .....	87
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	93

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.1: Πλακέτα Arduino Mega .....	7
Εικόνα 2.2: Είσοδοι και έξοδοι του Arduino .....	9
Εικόνα 2.3: Τροφοδοσία Arduino.....	10
Εικόνα 2.4: Arduino IDE.....	<b>Error! Bookmark not defined.</b> 2
Εικόνα 2.5: Μεταγλωττιστής Arduino IDE.....	<b>Error! Bookmark not defined.</b> 2
Εικόνα 2.6: Βασική δομή προγράμματος Arduino.....	16
Εικόνα 3.1: Αρχιτεκτονική του Android.....	18
Εικόνα 3.2: Επιφάνεια εργασίας App Inventor .....	26
Εικόνα 3.3: Επιφάνεια προγραμματισμού του App Inventor .....	27
Εικόνα 3.4: Προσομοιωτής του App Inventor .....	28
Εικόνα 3.5: Control blocks.....	29
Εικόνα 3.6: Logic blocks.....	29
Εικόνα 3.7: Math blocks .....	30
Εικόνα 3.8: Text blocks .....	30
Εικόνα 3.9: Lists blocks.....	31
Εικόνα 3.10: Color blocks .....	31
Εικόνα 3.11: Variable blocks .....	32
Εικόνα 3.12: Procedure blocks .....	32
Εικόνα 4.1: Arduino Mega.....	<b>Error! Bookmark not defined.</b> 3
Εικόνα 4.2: Arduino Nano 328 .....	33
Εικόνα 4.3: Audio Sensor.....	33
Εικόνα 4.4: MQ2 GAS Sensor .....	34
Εικόνα 4.5: DHT11 Temp-Humidity Sensor .....	34
Εικόνα 4.6: PIR Sensor.....	34
Εικόνα 4.7: RGB LEDS .....	34
Εικόνα 4.8: Bluetooth Connector .....	35

## Έξυπνο Σπίτι - Smarthome

Εικόνα 4.9: Radio TEA5767.....	35
Εικόνα 4.10: Ultra Sonic Sensor.....	35
Εικόνα 4.11: Capacitors .....	35
Εικόνα 4.12: Motor SShield .....	36
Εικόνα 4.13: LDR.....	36
Εικόνα 4.14: Real Time Clock DS1307+18B20 .....	36
Εικόνα 4.15: Micro Switches .....	37
Εικόνα 4.16: Dimmer .....	37
Εικόνα 4.17: Digital POT .....	37
Εικόνα 4.18: Trasmiter Receiver 477Hz.....	37
Εικόνα 4.19: Keypad 4x4 .....	38
Εικόνα 4.20: Lever Shiifter.....	38
Εικόνα 4.21: WiFi Connection .....	38
Εικόνα 4.22: Relay Shield.....	38
Εικόνα 4.23: Relay Board.....	39
Εικόνα 4.24: Καλώδια.....	39
Εικόνα 4.25: Block κώδικα Interface Σαλονιού (1).....	42
Εικόνα 4.26: Block κώδικα Interface Σαλονιού (2).....	43
Εικόνα 4.27: Block κώδικα ενεργοποίησης λάμπας.....	<b>Error! Bookmark not defined.</b> 3
Εικόνα 4.28: Block κώδικα απενεργοποίησης λάμπας .....	44
Εικόνα 4.29: Block κώδικα ενεργοποίησης Πρίζας .....	44
Εικόνα 4.30: Block κώδικα απενεργοποίησης Πρίζας.....	44
Εικόνα 5.1: Μενού .....	45
Εικόνα 5.2: Ρυθμίσεις.....	45
Εικόνα 5.3: Ασφάλεια.....	45
Εικόνα 5.4: Φάκελος Αποθήκευσης Εφαρμογών.....	45
Εικόνα 5.5: Έγκριση Εγκατάστασης (1) .....	46
Εικόνα 5.6: Έγκριση Εγκατάστασης (2) .....	46



Εικόνα 5.7: Εγκατάσταση Εφαρμογής.....	46
Εικόνα 5.8: Ολοκλήρωση Εγκατάστασης.....	46
Εικόνα 5.9: Μενού Συσκευής Android.....	47
Εικόνα 5.10: Ενεργοποίηση Bluetooth κατά την έναρξη της εφαρμογής.....	47
Εικόνα 5.11: Είσοδος στην εφαρμογή.....	48
Εικόνα 5.12: Εισαγωγή Username και Password.....	48
Εικόνα 5.13: Είσοδος στην εφαρμογή.....	48
Εικόνα 5.14 και 5.15: Κυρίως οθόνη-Πίνακας Ελέγχου.....	49
Εικόνα 5.16: Επιλογή Σαλονιού.....	49
Εικόνα 5.17: Μενού του Σαλονιού.....	<b>Error! Bookmark not defined.0</b>
Εικόνα 5.18: Χρήση Dimmer.....	<b>Error! Bookmark not defined.0</b>
Εικόνα 5.19: On/Off Λάμπας.....	<b>Error! Bookmark not defined.1</b>
Εικόνα 5.20: On/Off Κρυφού φωτισμού.....	<b>Error! Bookmark not defined.1</b>
Εικόνα 5.21: Επιλογή RGB Κρυφού φωτισμού.....	<b>Error! Bookmark not defined.1</b>
Εικόνα 5.22: On/Off Πριζών Σαλονιού.....	<b>Error! Bookmark not defined.2</b>
Εικόνα 5.23: On/Off Hi-Fi.....	<b>Error! Bookmark not defined.2</b>
Εικόνα 5.24: Διαχείριση Hi-Fi.....	<b>Error! Bookmark not defined.2</b>
Εικόνα 5.25: On/Off Air condition.....	53
Εικόνα 5.26: Επιλογή διαδρόμου.....	53
Εικόνα 5.27: Μενού Διαδρόμου.....	54
Εικόνα 5.28: On/Off στα φώτα του Διαδρόμου.....	54
Εικόνα 5.29: On/Off Πριζών του Διαδρόμου.....	55
Εικόνα 5.30: Επιλογή Κουζίνας.....	55
Εικόνα 5.31: Μενού Κουζίνας.....	56
Εικόνα 5.32: On/Off στα φώτα της Κουζίνας.....	56
Εικόνα 5.33: On/Off στο ψυγείο της Κουζίνας.....	57
Εικόνα 5.34: On/Off Φούρνου της Κουζίνας.....	57
Εικόνα 5.35: On/Off Απορροφητήρα της Κουζίνας.....	58

Εικόνα 5.36: On/Off Απορροφυτήρα της Κουζίνας.....	58
Εικόνα 5.37: Επιλογή Δωματίων .....	59
Εικόνα 5.38: Επιλογή Υπνοδωματίου.....	59
Εικόνα 5.39: Επιλογή Υπνοδωματίου.....	59
Εικόνα 5.40: Μενού Δωματίου (1) .....	59
Εικόνα 5.41: Μενού Δωματίου (2) .....	59
Εικόνα 5.42: Dimmer Δωματίου.....	60
Εικόνα 5.43: On/Off στο Φως του Δωματίου.....	60
Εικόνα 5.44: On/Off Πορτατίφ του Δωματίου .....	61
Εικόνα 5.45: On/Off στις Πρίζες του Δωματίου .....	61
Εικόνα 5.46: On/Off στο Air του Δωματίου.....	62
Εικόνα 5.47: Διαχείριση Ρολό του Δωματίου.....	62
Εικόνα 5.48: Διαχείριση Τέντας του Δωματίου .....	63
Εικόνα 5.49: Πίσω στο Μενού Υπνοδωματίων .....	63
Εικόνα 5.50: Επιλογή Υπνοδωματίου.....	63
Εικόνα 5.51: Επιλογή Υπνοδωματίου.....	64
Εικόνα 5.52/Εικόνα 5.53: Μενού Παιδικού Δωματίου (1) και (2) .....	64
Εικόνα 5.54: Dimmer Παιδικού Δωματίου.....	64
Εικόνα 5.55: On/Off στο Φως του Παιδικού Δωματίου.....	65
Εικόνα 5.56: On/Off στο Πορτατίφ του Παιδικού Δωματίου.....	65
Εικόνα 5.57: On/Off στις Πρίζες του Παιδικού Δωματίου .....	66
Εικόνα 5.58: On/Off στο condition του Παιδικού Δωματίου.....	66
Εικόνα 5.59: Διαχείριση Ρολό του Παιδικού Δωματίου .....	67
Εικόνα 5.60: Διαχείριση Τέντας του Παιδικού Δωματίου.....	67
Εικόνα 5.61: Πίσω στο Μενού Παιδικού Δωματίου .....	68
Εικόνα 5.62: Επιλογή Υπνοδωματίου.....	68
Εικόνα 5.63: Επιλογή Υπνοδωματίου.....	68
Εικόνα 5.64: Μενού Ξενώνα (1) .....	68

## Έξυπνο Σπίτι - Smarthome

Εικόνα 5.65: Μενού Ξενώνα (2).....	68
Εικόνα 5.66: Dimmer Ξενώνα .....	69
Εικόνα 5.67: On/Off στο Φως του Ξενώνα.....	69
Εικόνα 5.68: On/Off στο Πορτατίφ του Ξενώνα .....	70
Εικόνα 5.69: On/Off στις του Ξενώνα .....	70
Εικόνα 5.70: On/Off στο Φως του Ξενώνα.....	71
Εικόνα 5.71: Διαχείριση Ρολό του Ξενώνα .....	71
Εικόνα 5.72: Διαχείριση Τέντας του Ξενώνα .....	72
Εικόνα 5.73: Πίσω στο Μενού Ξενώνα .....	72
Εικόνα 5.74: Επιλογή Υπνοδωματίου.....	72
Εικόνα 5.75: Επιλογή Τουαλέτας .....	73
Εικόνα 5.76: Μενού Τουαλέτας .....	73
Εικόνα 5.77: On/Off στο Φως της Τουαλέτας.....	74
Εικόνα 5.78: On/Off στις Πρίζες της Τουαλέτας .....	74
Εικόνα 5.79: Επιλογή Θερμοσίφωνα.....	75
Εικόνα 5.80: Μενού Θερμοσίφωνα .....	75
Εικόνα 5.81: Ρύθμιση ώρας θερμοσίφωνα.....	75
Εικόνα 5.82: Επιλογή Θερμοκρασίας.....	76
Εικόνα 5.83: Interface Θερμοκρασίας .....	76
Εικόνα 5.84: Επιλογή Ασφάλειας.....	77
Εικόνα 5.85: Interface Ασφάλειας .....	77
Εικόνα 5.86: Διαχείριση Συναγερμού .....	78
Εικόνα 5.87: Έλεγχος γενικής θερμοκρασίας.....	78
Εικόνα 5.88: Έλεγχος Ανθρώπινης παρουσίας.....	79
Εικόνα 5.89: Έλεγχος διαρροής αερίου .....	79
Εικόνα 5.90: Έλεγχος Πριζών του χώρου .....	80
Εικόνα 5.91: Επιλογή Υπενθύμισης.....	80
Εικόνα 5.92: Μενού Υπενθύμισης .....	81

Έξυπνο Σπίτι - Smarthome

Εικόνα 5.93: Πληκτρολόγηση Υπενθύμισεις .....**Error! Bookmark not defined.**1

Εικόνα 5.94: Καταχωρημένη Υπενθύμιση .....**Error! Bookmark not defined.**1

Εικόνα 5.95: Διαγραφή Υπενθύμισεις..... 82

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 2.1: Χαρακτηριστικά του Arduino Mega2560.....	8
Πίνακας 2.2: Βασικά σημεία του κώδικα .....	13



## ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ ΣΤΑ SMART HOMES

Επισήμως η λέξη «έξυπνο» (smart) χρησιμοποιήθηκε για αναφορά σε τεχνολογικά επιτεύγματα πρώτη φορά κατά τη δεκαετία του 70. Αναφερόταν σε στρατιωτικά προϊόντα, όπως βόμβες ή πυραύλους που καθοδηγούσαν τον εαυτό τους προς το στόχο («έξυπνες» βόμβες). Κατά την τεχνολογική άνθηση της δεκαετίας του 80 η λέξη «έξυπνο» απέκτησε άλλες προεκτάσεις: αναφερόταν σε συσκευές που εμπεριείχαν μικροσίπ, όπως οι υπολογιστές και οι προηγμένες οικιακές συσκευές. Βέβαια αυτό άλλαξε με την πάροδο του χρόνου και πλέον σήμερα δεν αποκαλούμε έναν σύγχρονο υπολογιστή «έξυπνο», παρόλο που οι σημερινοί υπολογιστές είναι εκθετικά ισχυρότεροι από εκείνους της δεκαετίας του 80.

Σήμερα οι ορισμοί ποικίλουν και είναι περισσότερο τεχνολογικά προσανατολισμένοι. Ένας ορισμός για την έννοια του έξυπνου σπιτιού είναι ο εξής: «μια κατοικία που διαθέτει συσκευές, φωτισμό, θέρμανση, κλιματισμό, υπολογιστές, τηλεοράσεις, συστήματα ψυχαγωγίας, συστήματα ασφαλείας κλπ, που είναι σε θέση να επικοινωνούν μεταξύ τους και να ελέγχονται από τηλεχειρισμό ή Η/Υ από οποιοδήποτε σημείο στον κόσμο, χρησιμοποιώντας το διαδίκτυο ή το τηλέφωνο ή Bluetooth». Δηλαδή, το έξυπνο σπίτι «smart home» είναι εκείνο που είναι εξοπλισμένο με μια ποικιλία από τεχνολογικά εργαλεία που αυτοματοποιούν την ζωή στο σπίτι. Αυτά τα εργαλεία ποικίλουν ανάλογα με τις ανάγκες και τις προτιμήσεις των κατοίκων. Γενικά υπάρχουν διάφορες τεχνικές που χρησιμοποιούνται στην τεχνολογία των έξυπνων σπιτιών, με σκοπό να προσφέρουν άνεση, ασφάλεια και εξοικονόμηση ενέργειας και χρημάτων στους ενοίκους. Θεωρητικά κάτι που μπορεί να ελέγχεται εξ' αποστάσεως και αυτόματα, θεωρείται ότι είναι μια μορφή του οικιακού αυτοματισμού.

Οι έξυπνες εγκαταστάσεις αλληλεπιδρούν με το περιβάλλον χρησιμοποιώντας ένα μέσο επικοινωνίας με τη βοήθεια του οποίου ανταλλάσσουν δεδομένα προκειμένου να διεξάγουν κάποιες λειτουργίες όπως για παράδειγμα, να ενεργοποιήσουν το φωτισμό ενός χώρου ή να ρυθμίσουν τη θερμοκρασία. Έξυπνα συστήματα εγκαθίστανται και σε εμπορικές εφαρμογές όπου αναφέρονται με τον όρο αυτοματισμοί κτηρίων «building automation».

Τα έξυπνα συστήματα μπορούν να ελέγχουν εκτός από τις ηλεκτρολογικές και μηχανολογικές εγκαταστάσεις, οικιακές συσκευές και συσκευές πολυμέσων

(multimedia) δημιουργώντας ένα ενοποιημένο σύστημα. Στις τελευταίες εντάσσονται οι συσκευές τηλεπικοινωνιών, τα ηχοσυστήματα αλλά και οι τηλεοράσεις του σπιτιού. Συνδυάζοντας όλες αυτές τις ανεξάρτητες, αρχικά, εγκαταστάσεις σε μία κοινή βάση αποκτάται πλήρης έλεγχος της οικίας ο οποίος μπορεί να διεξαχθεί ακόμα και από μακριά.

Ένα χαρακτηριστικό των έξυπνων σπιτιών, είναι ότι τα ίδια περιφερειακά χρησιμοποιούνται για πολλές χρήσεις. Παράδειγμα επ' αυτού, είναι ότι οι αισθητήρες παρουσίας μπορούν να χρησιμοποιηθούν για τον έλεγχο του φωτισμού και του συστήματος θέρμανσης αλλά χρησιμεύουν και για το σύστημα του συναγερμού. Ένα άλλο παράδειγμα αφορά στις οθόνες των τηλεοράσεων, οι οποίες μπορούν να προβάλλουν και την εικόνα της θυροτηλεόρασης.

Το έξυπνο σπίτι είναι μία μακροχρόνια επένδυση η οποία πρέπει να αποδίδει, ακόμα και αφού έχουν παρέλθει αρκετά χρόνια από την κατασκευή του και να μπορεί να ακολουθεί τις συνεχώς μεταβαλλόμενες ανάγκες καθώς διαπιστώνουμε ότι στο έξυπνο σπίτι του 21ου αιώνα, μας καλωσορίζουν μία πληθώρα δυνατοτήτων όπως είναι οι έξυπνες οικιακές συσκευές, που παίρνουν από μόνες τους πρωτοβουλίες, τα έξυπνα έπιπλα με αίσθηση του περιβάλλοντος, τα κυκλώματα ασφαλείας που ελέγχονται από απόσταση και τα οικιακά συστήματα που κατανοούν τη διάθεσή μας και ενεργούν αναλόγως.

## 1.1 Πλεονεκτήματα των Smart Homes

Τα πλεονεκτήματα που προκύπτουν από τον αποτελεσματικό συντονισμό των συστημάτων αφορούν στη διευκόλυνση της καθημερινότητας των χρηστών. Η βελτίωση της ποιότητας ζωής των ενοίκων, έπειτα από κατάλληλο προγραμματισμό του συστήματος, συνοδεύεται από εξοικονόμηση της καταναλισκόμενης ενέργειας και κατ' επέκταση και από εξοικονόμηση χρημάτων. Επίσης, τα έξυπνα συστήματα είναι δυνατό να εξασφαλίσουν ασφαλέστερες συνθήκες διαβίωσης. Ας ρίξουμε λοιπόν στο σημείο αυτό μια συνοπτική ματιά στα οφέλη που ένα σημερινό έξυπνο σπίτι μπορεί να παρέχει για να καταλάβουμε περί τίνος πρόκειται.

Ένα θεμελιώδες όφελος από τη χρήση της τεχνολογίας του έξυπνου σπιτιού είναι η **εξοικονόμηση ενέργειας**. Με την κατάλληλη πρόβλεψη για τις ενεργειακές ανάγκες κάθε συσκευής έχουμε βελτιστοποίηση στη διαχείριση ενέργειας. Η διαχείριση ενέργειας αποτελεί προτεραιότητα στο σχεδιασμό έξυπνων σπιτιών. Στόχος της σχεδίασης είναι η

Μαρίνα Αρντίτ – Μπικίκου Ευγενία



μείωση της κατανάλωσης ενέργειας στο ελάχιστο δυνατό χωρίς να θυσιάζεται η άνεση των ατόμων που ζουν μέσα σε αυτό. Όταν μιλάμε για εξοικονόμηση στο έξυπνο σπίτι, δεν πρόκειται απλά για το ηλεκτρικό ρεύμα. Αντιθέτως, πρόκειται για εξοικονόμηση νερού, πετρελαίου, αερίου, θερμότητας και φυσικά χρόνου! Το κλειδί της υπόθεσης είναι η χρήση των διαφόρων ενεργειακών πόρων του σπιτιού, ακριβώς στην ποσότητα που χρειάζεται, ακριβώς όταν χρειάζεται και ακριβώς στο μέρος που χρειάζεται. Για παράδειγμα, η κατανάλωση ενέργειας μειώνεται με τον αυτόματο έλεγχο των θερμαντικών σωμάτων. Εφόσον η θερμοκρασία δωματίου φτάσει σε κάποιο επιθυμητό επίπεδο, τα θερμαντικά σώματα απενεργοποιούνται αυτόματα. Ένας άλλος τρόπος για την αποφυγή άσκοπης κατανάλωσης ενέργειας είναι η απενεργοποίηση της θέρμανσης όταν είναι ανοιχτά τα παράθυρα.

Ένα ακόμη σπουδαίο όφελος που προκύπτει από την ενσωμάτωση της τεχνολογίας του έξυπνου σπιτιού σε κάθε οικία είναι η **ποιότητα ζωής**. Πολλοί μάλιστα είναι εκείνοι που λόγω της παρεχόμενης άνεσης συναρπάζονται με την ιδέα του έξυπνου σπιτιού. Άνεση σε αυτή την περίπτωση σημαίνει ένα σπίτι, που όχι μόνο φροντίζει τον εαυτό του αλλά και τον ένοικο. Από τις θεαματικές πλέον λύσεις ψυχαγωγίας και διασκέδασης έως την αυτόματη εκτέλεση καθημερινών επαναλαμβανόμενων σεναρίων ενεργειών μέσα στο σπίτι (π.χ. πλύσιμο πιάτων). Ο ένοικος, μέσω οποιουδήποτε τονικού τηλεφώνου, σταθερού ή κινητού ή μέσω του internet, μπορεί να χειριστεί τις κύριες λειτουργίες της κατοικίας κατά τη διάρκεια απουσίας του. Έτσι, έχει τη δυνατότητα να ανάψει το θερμοσίφωνα λίγο πριν φτάσει σπίτι του και να ρυθμίσει τη θερμοκρασία του σπιτιού. Επίσης, μπορεί να προγραμματίσει αυτοματοποιημένο πότισμα κατά τη διάρκεια μακράς απουσίας. Η ποιότητα ζωής του χρήστη είναι το ζητούμενο και το αποτέλεσμα.

Ακόμα ένα από τα πλεονεκτήματα της τεχνολογίας του έξυπνου σπιτιού, είναι η **ασφάλεια**. Με την τεχνολογία του χτες τα ατυχήματα μέσα στο σπίτι ήταν κάτι παραπάνω από εύκολο να διαδραματιστούν. Χαρακτηριστικό παράδειγμα είναι το πόσο εύκολο είναι να πάθει κάποιος ηλεκτροπληξία σε ένα κατάλυμα παλαιών προδιαγραφών λόγω βλάβης του κυκλώματος ή απλά λόγω απροσεξίας. Στο έξυπνο σπίτι τα πράγματα είναι διαφορετικά. Ηλεκτρική ισχύς παρέχεται μόνο στις πρίζες που έχουν κάποια συσκευή συνδεδεμένη και με την οποιαδήποτε αποτυχία του κυκλώματος ή με το πρώτο βραχυκύκλωμα η παροχή ισχύος μπλοκάρει αυτόματα για την ασφάλεια του χρήστη. Επίσης τα σύγχρονα συστήματα προσφέρουν τη δυνατότητα παρακολούθησης της κατοικίας. Έτσι, ο ιδιοκτήτης έχει τη δυνατότητα, όχι μόνο να παρακολουθεί από όλες τις τηλεοράσεις του σπιτιού την εικόνα που καταγράφουν οι κάμερες, αλλά και

ενημερώνεται για την κατάσταση της οικίας κατά την απουσία του μέσω φωτογραφιών στο κινητό του. Σε περίπτωση που ενεργοποιηθούν οι αισθητήρες συναγερμού λόγω παραβίασης, υπάρχει η δυνατότητα αυτόματης καταγραφής εικόνων. Πέρα από αυτό, μπορεί να υπάρχουν ανιχνευτές αερίου, διαρροής νερού, καπνού και άλλων επικίνδυνων καταστάσεων όπως είναι η παραβίαση μιας κλειδαριάς. Αν ένας από αυτούς τους ανιχνευτές εντοπίσει ό,τι ύποπτο, τότε ενεργοποιείται ο αντίστοιχος συναγερμός και κλείνουν αυτόματα οι συσκευές που πρέπει ή ξεκινάνε άλλα προκαθορισμένα σενάρια ενεργειών και ο ιδιοκτήτης μπορεί να ενημερώνεται αν προκύψει κάτι έκτακτο όπως πυρκαγιά ή διαρροή νερού κατά την απουσία του.

Συνοπτικά τα πλεονεκτήματα ενός «έξυπνου σπιτιού είναι τα εξής:

- ✓ **ΑΝΕΣΗ**
  - Διαχείριση:
  - Φωτισμός
  - Ρολά/Τέντες
  - Θερμοκρασία
  - Αυτοματισμοί
  - Μετάδοση ήχου
- ✓ **ΕΠΙΚΟΙΝΩΝΙΑ**
  - Διαχείριση Πολυμέσων
  - Έλεγχος εισόδου
  - Ενδοεπικοινωνία
- ✓ **ΑΣΦΑΛΕΙΑ**
  - Επιτήρηση
  - Τεχνικός Συναγερμός
  - Συναγερμός

## 1.2 Έλεγχος Smart Homes

Οι δυνατότητες χειρισμού που εξασφαλίζονται χάρη στην ευελιξία του συστήματος είναι πολλαπλές και επιτρέπουν στον ιδιοκτήτη να προγραμματίζει το σπίτι, ώστε να επιτελεί συγκεκριμένες λειτουργίες με απλούς χειρισμούς. Για τον έλεγχο του συστήματος μπορεί να χρησιμοποιηθεί οθόνη αφής. Το έξυπνο σπίτι παρέχει την δυνατότητα στους ενοίκους του να ενεργούν εύκολα όταν βρίσκονται μέσα στο σπίτι.

Ορισμένες από τις λειτουργίες που διευκολύνουν τους ιδιοκτήτες κατά την παραμονή τους στο σπίτι αναφέρονται παρακάτω:

- ✓ Σενάρια φωτισμού: Επιλογή ενός από διάφορα φώτα του χώρου και ρύθμισης επιθυμητής έντασης και απόχρωσης.
- ✓ Λειτουργία ελέγχου ασφάλειας του χώρου (πχ συναγερμός).
- ✓ Λειτουργία υπενθύμισης μέσω αποστολής μηνύματος sms στο κινητό.
- ✓ Λειτουργία ελέγχου «έξυπνων συσκευών» του χώρου (θερμοσίφωνα, aircondition, ραδιοφώνου).

Οι ένοικοι θα έχουν την δυνατότητα μέσω της εφαρμογής να έχουν ενημέρωση για διάφορες πληροφορίες του χώρου τους.

Πιο συγκεκριμένα, οι χρήστες μπορούν να:

- ✓ ενημερώνονται σχετικά με την εσωτερική θερμοκρασία και να τη ρυθμίζουν αναλόγως
- ✓ ενημέρωση για τα φώτα του χώρου που είναι ανοιχτά και ρύθμιση τους
- ✓ ενεργοποίηση και απενεργοποίηση διαφόρων πριζών του χώρου και των αντίστοιχων συσκευών που είναι εκεί συνδεδεμένες
- ✓ ενεργοποίηση και απενεργοποίηση θερμοσίφωνα την επιθυμητή ώρα και ημερομηνία

### 1.3 Σκοπός και δομή της εργασίας

Στα πλαίσια της πτυχιακής εργασίας, δημιουργήθηκε μια πλακέτα Arduino που αντιπροσωπεύει το εσωτερικό σύστημα του «έξυπνου σπιτιού» και μία εφαρμογή Android για τη διαχείριση της πλακέτας μέσω Bluetooth.

Η εφαρμογή σχεδιάστηκε για smartphones που τρέχουν ANDROID και έχει τις παρακάτω λειτουργίες:

- ✓ Σύνδεση με τα στοιχεία του χρήστη
- ✓ Διαχείριση φωτισμού:
  - Ενεργοποίηση/απενεργοποίηση στα φώτα των δωματίων
  - Ρύθμιση έντασης τους
- ✓ Διαχείριση συστήματος συναγερμού:
  - Ενεργοποίηση/απενεργοποίηση του συναγερμού
- ✓ Διαχείριση θέρμανσης:

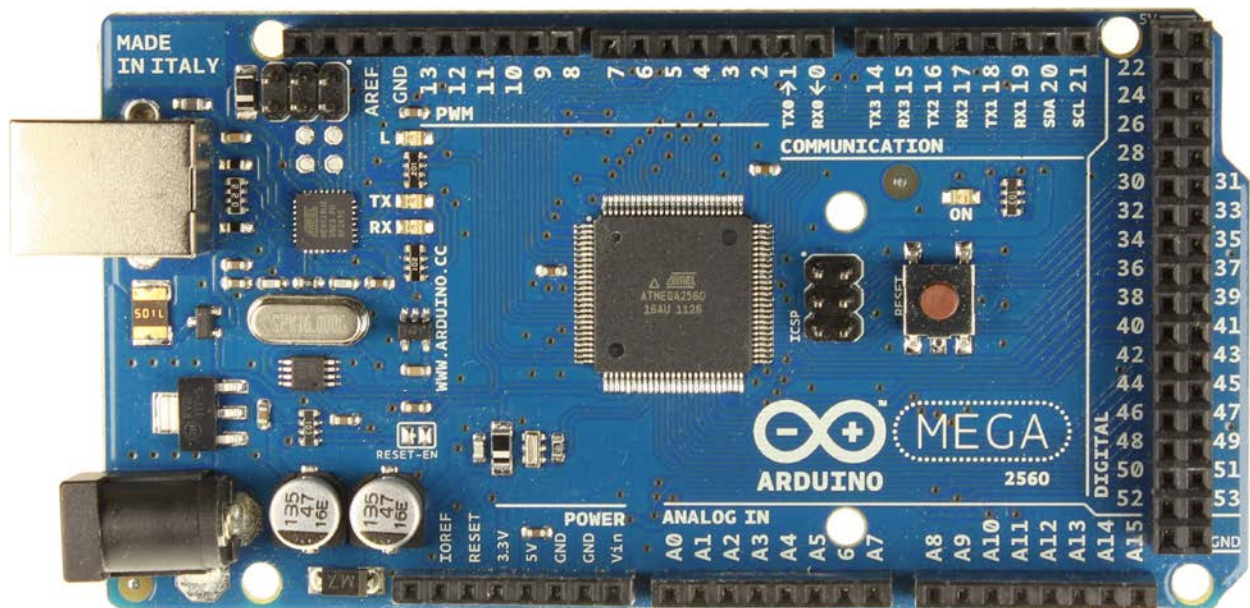
- Ενεργοποίηση/απενεργοποίηση θέρμανσης
- Ενεργοποίηση/απενεργοποίηση θέρμανσης με χρονοδιακόπτη (on/off timer)
- ✓ Διαχείριση θερμοσίφωνα
  - Ενεργοποίηση/απενεργοποίηση θερμοσίφωνα
  - Ενεργοποίηση/απενεργοποίηση θερμοκρασίας με χρονοδιακόπτη (on/off timer)
- ✓ Ενημέρωση θερμοκρασίας εσωτερικού και εξωτερικού χώρου
- ✓ Υπενθύμιση με αποστολή του επιθυμητού μηνύματος με sms στο κινητό
- ✓ Διαχείριση συσκευών του χώρου (πχ ραδιόφωνο, φούρνος, ψυγείο και συσκευές που βρίσκονται πάνω στις ελεγχόμενες πρίζες)

## ΚΕΦΑΛΑΙΟ 2: ΕΙΣΑΓΩΓΗ ΣΤΟ ARDUINO

Στο κεφάλαιο αυτό θα δούμε το θεωρητικό υπόβαθρο του Arduino όπως επίσης και πληροφορίες για την κατασκευή της πλακέτας του «έξυπνου σπιτιού»

### 2.1 Τί είναι το Arduino;

Το Arduino είναι μία πλακέτα που έχει ενσωματωμένο μικροελεγκτή (microcontroller) τύπου AVR της ATMEL. Η πλακέτα αποτελείται από μία απλή μητρική που έχει ενσωματωμένο τον μικροελεγκτή και διάφορες εισόδους και εξόδους (inputs and outputs). Είναι πλατφόρμα «ανοικτού κώδικα» (open source code) με ευέλικτη και εύκολη χρήση του λογισμικού (software) και υλικό (hardware).



Εικόνα 2.1: Πλακέτα Arduino Mega

Στην ουσία, πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα. Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται

δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου.

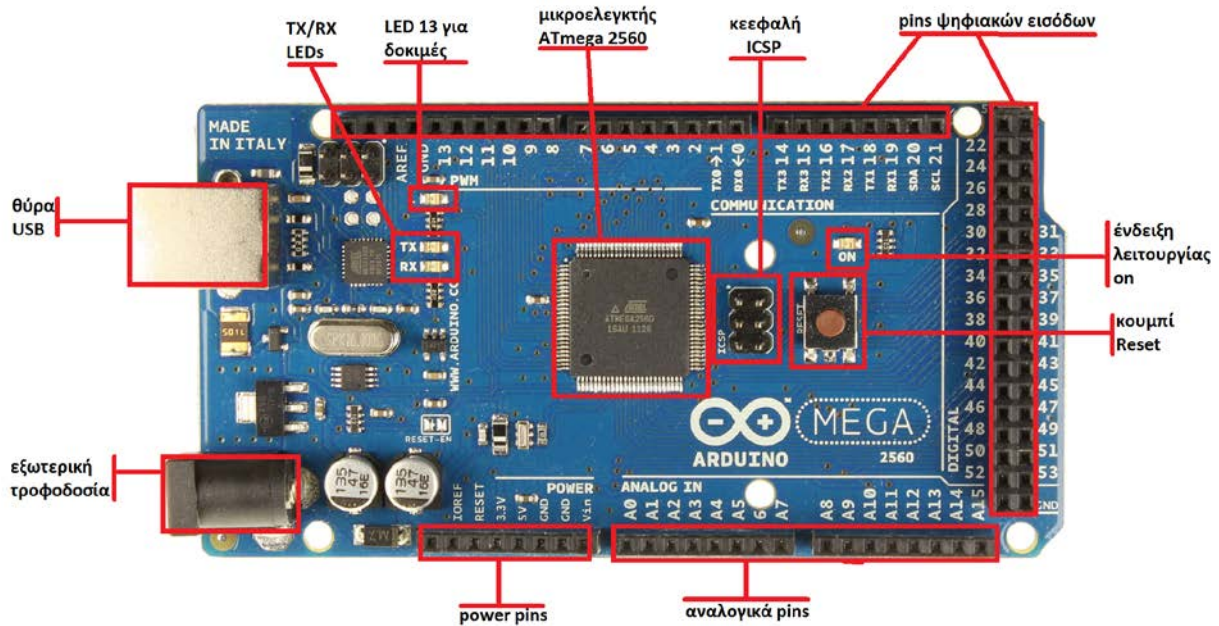
## 2.2 Το Arduino Mega 2560r3

Το **Arduino Mega 2560r3** (Εικόνα 2.1) είναι μία πλακέτα με βάση τον μικροελεγκτή ATmega2560. Έχει 54 pins ψηφιακών εισόδων/εξόδων (digital input/output pins) (εκ των οποίων οι 16 μπορούν να χρησιμοποιηθούν ως PWM), 15 αναλογικών εισόδων, 4 UARTs, μία θύρα USB, μία υποδοχή τροφοδοσίας, κεφαλίδα ICSP(ICSP header) και ένα κουμπί επαναφοράς των εργοστασιακών ρυθμίσεων (reset button). Αναλυτικότερα τα χαρακτηριστικά του βρίσκονται στον παρακάτω πίνακα.

Μικροελεγκτή	ATmega2560
Τάση λειτουργίας	5V
Τάση εισόδου (συνιστάται)	7-12V
Τάση εισόδου (όριο)	6-20V
Digital I / O Καρφίτσες	54 (εκ των οποίων 15 παρέχει έξοδο PWM)
Αναλογική Είσοδος Pins	16
DC ρεύματος ανά I/O Pin	20 Ma
DC ρεύμα για 3.3V Pin	50 Ma
Μνήμη Flash	256 KB εκ των οποίων 8 KB που χρησιμοποιούνται από bootloader
SRAM	8 KB
EEPROM	4 KB
Ταχύτητα ρολογιού	16 MHz

Πίνακας 2.1: Χαρακτηριστικά του Arduino Mega2560

## 2.2.1 Είσοδοι - Έξοδοι



Εικόνα 2.2: Είσοδοι και έξοδοι του Arduino

Το Arduino διαθέτει

### ✓ **Σειριακή Θύρα:**

Χρησιμοποιείται για επικοινωνία του υπολογιστή ή άλλων συσκευών με το Arduino. Όλες οι πλακέτες έχουν τουλάχιστον μία σειριακή θύρα. Επικοινωνεί με τον υπολογιστή μέσω USB θύρας, όπως επίσης και με τις ψηφιακές ακίδες (digital pins) 0 (RX) 1 (TX). Κατά συνέπεια δεν μπορεί να χρησιμοποιηθεί ταυτόχρονα η USB λειτουργία με τις ακίδες RX/TX.

### ✓ **Ψηφιακές Ακίδες (Digital pins)**

Οι ακίδες αυτές μπορούν να χρησιμοποιηθούν και ως είσοδοι και ως έξοδοι, σαν προεπιλογή λειτουργούν ως είσοδοι. Κατά την διάρκεια του προγραμματισμού του Arduino θέτοντας τα σε κατάσταση HIGH/LOW ώστε να επιτραπεί ή όχι η το ρεύμα στην συγκεκριμένη ακίδα. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA.

### ✓ **Αναλογικές Ακίδες (Analog pins)**

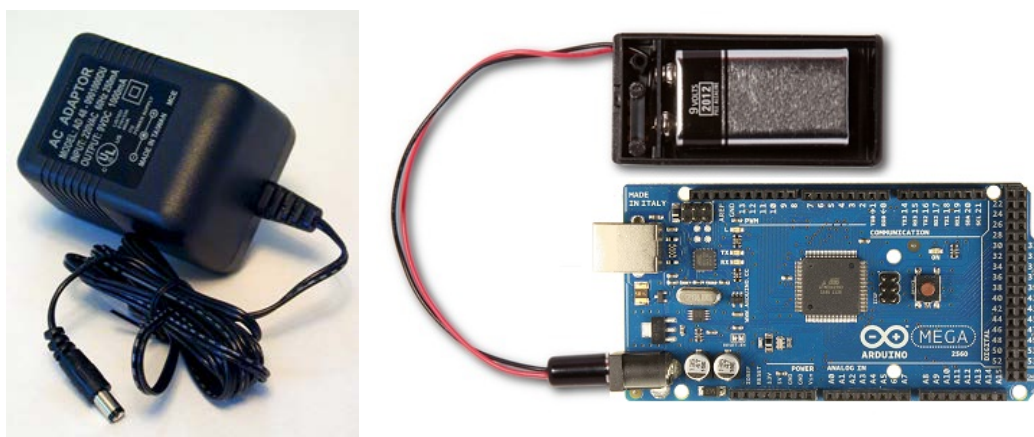
Οι μικροελεγκτές Atmega που χρησιμοποιούνται για την πλατφόρμα Arduino περιέχουν έναν ενσωματωμένο αναλογικό-σε-ψηφιακό μετατροπέα 6 καναλιών.



Ο μετατροπέας διαθέτει ανάλυση 10 bit, επιστρέφοντας ακέραιους από 0 έως 1023. Ενώ η κύρια λειτουργία της αναλογικής ακίδας είναι να διαβάζει αναλογικούς αισθητήρες. Οι αναλογικές ακίδες έχουν επίσης όλες τις λειτουργίες των γενικών ακίδων εισόδου/εξόδου.

### 2.2.2 Τροφοδοσία

Το Arduino τροφοδοτείται είτε από εξωτερική τροφοδοσία που παρέχεται είτε μιας υποδοχής των 2.1mm (θετικός πόλος στο κέντρο) που βρίσκεται στην κάτω αριστερή γωνία του Arduino είτε απευθείας από την θύρα USB του υπολογιστή (Εικόνα 2.3). Η επιλογή της πηγής γίνεται αυτόματα από το αναπτυσσόμενο. Ως εξωτερική τροφοδοσία ορίζεται είτε μια μπαταρία, είτε μετασχηματιστής των 9Volt από 220V. Η μπαταρία μπορεί να συνδεθεί στις υποδοχές του Arduino Vin και GND όπου τοποθετούνται ο θετικός πόλος και ο αρνητικός αντίστοιχα. Από την άλλη αν τροφοδοτηθεί με μετασχηματιστή απλά πρέπει να τοποθετηθεί το βύσμα στην υποδοχή που υπάρχει θετικό πόλο στο κέντρο.



Εικόνα 2.3: Τροφοδοσία Aduino

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



### 2.2.3 Ενσωματωμένα κουμπιά και LED

Το Arduino περιλαμβάνει

✓ **Δικόπτη *micro-switch* (RESET)**

Χρησιμοποιείται για να επαναφέρει στις εργοστασιακές ρυθμίσεις την πλακέτα.

✓ **4 Ενδείξεις LED**

○ **POWER**

ένδειξη λειτουργίας

○ **RX και TX**

χρησιμοποιούνται ως ένδειξη λειτουργίας της σειριακού interface, καθώς ανάβουν όταν υπάρχει αποστολή ή λήψη δεδομένων μέσω USB.

○ **L**

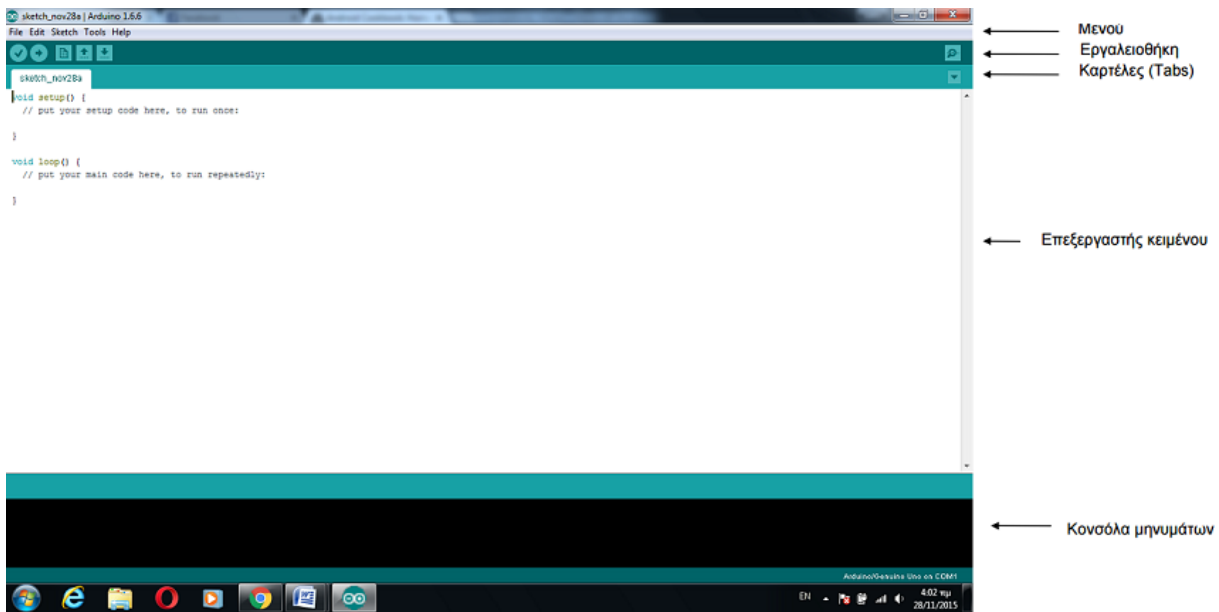
είναι συνδεδεμένο με το pin 13 εργοστασιακά, έτσι που και αν δεν του έχουμε φορτώσει κάποιο πρόγραμμα στο Arduino αυτό θα ανάψει θέτοντας την τιμή 1 στο πρόγραμμα

### 2.3 Arduino IDE και σύνδεση με τον υπολογιστή

Το Arduino IDE (Εικόνα 2.4) είναι η εφαρμογή με την οποία μπορούμε να δημιουργήσουμε/τροποποιήσουμε ένα πρόγραμμα για το Arduino.

Το Arduino είναι βασισμένο σε Java και συγκεκριμένα περιέχει:

- ✓ ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων,
- ✓ αρκετά έτοιμα παραδείγματα,
- ✓ μερικές έτοιμες βιβλιοθήκες για προέκταση της γλώσσας
- ✓ τον compiler για την μεταγλώττιση των προγραμμάτων (sketch),
- ✓ μία σειριακή οθόνη (serial monitor) που παρακολουθεί τις επικοινωνίες της σειριακής θύρα (USB), αναλαμβάνει να στείλει αλφαριθμητικά της επιλογής σας στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για την εκσφαλμάτωση (debugging) των sketch
- ✓ και τη δυνατότητα να κατεβάσουμε το μεταγλωττισμένο sketch στο Arduino.



Εικόνα 2.4: Arduino IDE



Εικόνα 2.5: Μεταγλωττιστής Arduino IDE

## 2.4 Γλώσσα προγραμματισμού

Το Arduino Mega μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη C++ με κάποιες μετατροπές). Λόγω του ότι η Wiring βασίζεται στην C/C++ ισχύουν οι ίδιες συναρτήσεις και βρόγχοι (functions and loops) που έχει και η C/C++.

Ακολουθεί πίνακας με τις πιο σημαντικές σταθερές και την επεξήγησή τους:

Όρισμα	Είδος	Τύπος	Παράμετροι	Περιγραφή
LOW	Σταθερά	Int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
HIGH	Σταθερά	Int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
INPUT	Σταθερά	Int	-	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
OUTPUT	Σταθερά	Int	-	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
pinMode	Εντολή	-	( <i>pin, mode</i> )	Καθορίζει αν το συγκεκριμένο ψηφιακό <i>pin</i> θα είναι <i>pin</i> εισόδου ή <i>pin</i> εξόδου ανάλογα με την τιμή που δίνεται στην παράμετρο <i>mode</i> (INPUT ή OUTPUT).
digitalWrite	Εντολή	-	( <i>pin, pinstatus</i> )	Θέτει την κατάσταση <i>pinstatus</i> (HIGH ή LOW) στο συγκεκριμένο ψηφιακό <i>pin</i> .
digitalRead	Συνάρτηση	Int	( <i>pin</i> )	Επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού <i>pin</i> (0 για LOW και 1 για HIGH) εφόσον αυτό είναι <i>pin</i> εισόδου.
analogReference	Εντολή	-	( <i>type</i> )	Δέχεται τις τιμές DEFAULT, INTERNAL ή EXTERNAL στην παράμετρο <i>type</i> για να καθορίσει την τάση αναφοράς ( $V_{ref}$ ) των αναλογικών εισόδων (5V, 1.1V ή η εξωτερική τάση με την οποία τροφοδοτείται το <i>pin</i> AREF)

analogRead	Συνάρτηση	Int	(pin)	Επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο pin αναλογικής εισόδου στην κλίμακα 0 ως $V_{ref}$ .
analogWrite	Εντολή	-	(pin, value)	Θέτει το συγκεκριμένο ψηφιακό pin σε κατάσταση ψευδοαναλογικής εξόδου (PWM). Η παράμετρος value καθορίζει το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην κλίμακα από 0 ως 255 (π.χ. με value 127, το πλάτος του παλμού είναι ίσο με μισή περίοδο).
noInterrupts	Εντολή	-	()	Σταματά προσωρινά την λειτουργία όλων των interrupts
Interrupts	Εντολή	-	()	Επαναφέρει την λειτουργία των interrupt που διακόπηκε προσωρινά από μια εντολή noInterrupts.
Millis	Συνάρτηση	unsigned long	()	Μετρητής που επιστρέφει το χρονικό διάστημα σε ms από την στιγμή που άρχισε η εκτέλεση του προγράμματος. Λάβετε υπόψη ότι λόγω του τύπου μεταβλητής (unsigned long δηλ. 32bit) θα γίνει overflow σε $2^{32}ms$ δηλαδή περίπου σε 50 μέρες, οπότε ο μετρητής θα ξεκινήσει πάλι από το μηδέν.

Delay	Εντολή	-	( <i>time</i> )	Σταματά προσωρινά την ροή του προγράμματος για <i>time</i> ms. Η παράμετρος <i>time</i> είναι unsigned long (από 0 ως 2 <sup>32</sup> ). Σημειώστε ότι παρά την προσωρινή παύση, συναρτήσεις των οποίων η εκτέλεση ενεργοποιείται από interrupt θα εκτελεστούν κανονικά κατά την διάρκεια μιας delay.
attachInterrupt	Εντολή	-	( <i>interrupt, function, triggermode</i> )	Θέτει σε λειτουργία το συγκεκριμένο <i>interrupt</i> , ώστε να ενεργοποιεί την συνάρτηση <i>function</i> , κάθε φορά που ικανοποιείται η συνθήκη που ορίζεται από την παράμετρο <i>triggermode</i> :  LOW (ενεργοποίηση όταν η κατάσταση του pin που αντιστοιχεί στο συγκεκριμένο interrupt γίνει LOW)
detachInterrupt	Εντολή	-	( <i>interrupt</i> )	Απενεργοποιεί το συγκεκριμένο <i>interrupt</i> .
Serial.begin	Μέθοδος κλάσης	-	( <i>datarate</i> )	Θέτει τον ρυθμό μεταφοράς δεδομένων του σειριακού interface (σε baud)
Serial.print	Μέθοδος κλάσης	-	( <i>data</i> )	Διοχετεύει τα δεδομένα <i>data</i> για αποστολή μέσω του σειριακού interface. Η παράμετρος <i>data</i> μπορεί να είναι είτε αριθμός είτε αλφαριθμητικό.

Πίνακας 2.2: Βασικά σημεία του κώδικα

Βασική δομή προγράμματος: // Ενσωματώσεις βιβλιοθηκών, δηλώσεις μεταβλητών...

```
void setup()  
{  
  // ...  
}
```



**Εκτελείται ΜΟΝΟ μία φορά κατά την εκκίνηση του sketch**

```
void loop()  
{  
  // ...  
}
```



**Βασικός κορμός προγράμματος επαναλαμβάνεται συνέχεια**

Εικόνα 2.6: Βασική δομή προγράμματος Arduino

## ΚΕΦΑΛΑΙΟ 3: ΕΙΣΑΓΩΓΗ ΣΤΟ ANDROID

Στο κεφάλαιο αυτό παρουσιάζεται το θεωρητικό υπόβαθρο για το Android, καθώς επίσης και τα βασικά κομμάτια κώδικα της εφαρμογής AppInventor που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής μέσω της οποίας χειριζόμαστε την πλακέτα του Arduino.

### 3.1 Τί είναι το Android;

Το λειτουργικό σύστημα Android αποτελεί ένα εργαλείο το οποίο σχεδιάστηκε για να βοηθήσει στην εξέλιξη των συσκευών κινητής τηλεφωνίας, μετατρέποντάς τα ουσιαστικά σε έναν μικρό υπολογιστή τσέπης. Εκτός από το Operating System, το middleware, θα συναντήσουμε και εφαρμογές κορμού (core application). Μέσω της γλώσσας Java και του Android SDK που παρέχει τα απαραίτητα εργαλεία και τα API's, ένας σύγχρονος προγραμματιστής μπορεί να δημιουργήσει εφαρμογές πολύ υψηλών απαιτήσεων και αποδόσεων.

Το DVM (Dalvik Virtual Machine) έρχεται να ολοκληρώσει με επιτυχία την πλατφόρμα Android καθώς πρόκειται για έναν εξομοιωτή κινητής συσκευής με όλα τα χαρακτηριστικά ενός πραγματικού τηλεφώνου, στον οποίο ο προγραμματιστής βλέπει άμεσα τα αποτελέσματα της δουλειάς του.

Η αρχιτεκτονική του Android φαίνεται στο αμέσως παρακάτω σχήμα:



Εικόνα 3.1: Αρχιτεκτονική του Android

### 3.2 Δομικά στοιχεία αρχιτεκτονικής

#### Linux Kernel (Πυρήνας Linux)

Το Android είναι βασισμένο στον πυρήνα του Linux Kernel για να ανταποκρίνεται σε όλες τις ανάγκες με ταχύτητα και ευελιξία προσεγγίζοντας έτσι ακόμα περισσότερο την εικόνα ενός μικρού υπολογιστή με περιβάλλον Windows.

#### Android Runtime

Στην περιοχή του Runtime βρίσκονται οι βιβλιοθήκες. Δομημένες κατάλληλα περιέχουν όλα τα απαραίτητα εργαλεία για την ανάπτυξη οποιασδήποτε εφαρμογής, όπως για παράδειγμα επίλυσης μαθηματικών εξισώσεων κ.α., που έχουν χρησιμοποιηθεί σε όλες τις γλώσσες προγραμματισμού. Επίσης συναντάμε την Virtual Machine που απαρτίζεται από καταχωρητές και τρέχει κλάσεις οι οποίες μεταγλωττίζονται από Java compiler. Το Linux Kernel μπορεί να εκτελέσει πολλαπλά στιγμιότυπα της Dalvik VM, ενώ παράλληλα παρέχει λειτουργικότητα και για δευτερεύουσες εφαρμογές, όπως νήματα (threads) και χαμηλού επιπέδου διαχείριση μνήμης.



## Βιβλιοθήκες

Οι βιβλιοθήκες του Android, περιλαμβάνουν ένα σύνολο από C/C++ βιβλιοθήκες, που χρησιμοποιούνται από διάφορα δομικά στοιχεία του συστήματος. Αυτές διατίθενται στους προγραμματιστές/developers μέσω του Android application framework. Μερικές από αυτές είναι οι παρακάτω:

- ✓ System C Libraries: Προέρχονται από την BSD (Berkeley Software Distribution) του προτύπου συστήματος βιβλιοθήκης C (libc) και είναι συντονισμένες για ενσωμάτωση σε Linux-based συσκευές.
- ✓ Media Libraries: Οι βιβλιοθήκες αυτές υποστηρίζουν δυνατότητες εγγραφής και playback πολλών γνωστών τύπων ήχου, εικόνας και video, όπως MPEG4, H.264, MP3, AAC, AMR, JPEG, PNG.
- ✓ Surface Manager: Βοηθά στην πρόσβαση στα subsystems της εφαρμογής.
- ✓ WebKit: Μηχανή για μοντέρνο web browser.
- ✓ SGL: Μηχανή για 2D γραφικά.
- ✓ 3D Libraries: Βιβλιοθήκες βασισμένες στα OpenGL ES 1.0 APIs.
- ✓ SQLite: μια ευέλικτη και ισχυρή βάση δεδομένων διαθέσιμη σε όλες τις εφαρμογές.

## Application Framework

Σχεδόν όλες οι εφαρμογές του Android αποτελούνται από μια γκάμα γραφικών και services τα οποία ανταποκρίνονται στις απαιτήσεις του χρήστη και είναι γραφικά εργαλεία όπως: ActivityViews, grids, lists, textViews, editIntroductionTexts, Spinners, Buttons, έναν ενσωματωμένο web browser ακόμα και MapView.

Οι Content Providers χρησιμοποιούνται για να επιτρέπουν στις εφαρμογές να έχουν πρόσβαση σε δεδομένα άλλων εφαρμογών (όπως π.χ. στις επαφές του κινητού τηλεφώνου) ή να μοιράζονται με άλλες εφαρμογές τα δικά τους δεδομένα.

Επιπλέον υπάρχουν: Οι Resource Manager, Activity Manager και Notification Manager, οι οποίοι διαχειρίζονται την κατάσταση των εφαρμογών και παρέχουν κατά τη διάρκεια της λειτουργίας του κινητού χρήσιμες πληροφορίες όπως π.χ. ότι υπάρχει διαθέσιμο δίκτυο wi-fi εντός εμβέλειας, προειδοποίηση για κάποια πληροφορία ή ενός νέου e-mail στο λογαριασμό σας.

## Applications

Μια σειρά από εφαρμογές βρίσκονται στο υψηλότερο επίπεδο της αρχιτεκτονικής του Android λογισμικού και περιλαμβάνει email client, SMS/MMS εφαρμογή, ημερολόγιο, έναν web browser, χάρτες και εφαρμογές επί αυτών, επαφές (contacts) κ.α. Όλες οι εφαρμογές είναι γραμμένες σε γλώσσα προγραμματισμού Java.

### 3.3 Ανατομία μιας Εφαρμογής Android

Υπάρχουν τέσσερα δομικά blocks από τα οποία αποτελείται μια Android εφαρμογή: Activity, Intent Receiver, Service, Content Provider. Δεν χρειάζεται κάθε εφαρμογή να έχει και τα τέσσερα αυτά συστατικά, αλλά σίγουρα χρειάζεται κάποιον συνδυασμό από αυτά.

#### AndroidManifest

Όταν ο προγραμματιστής αποφασίσει ποια στοιχεία του χρειάζονται για να αναπτύξει την εφαρμογή, αυτά πρέπει να καθοριστούν και να καταγραφούν σε ένα αρχείο που ονομάζεται AndroidManifest.xml. Στο αρχείο αυτό, δηλώνονται τα δομικά blocks που θα χρησιμοποιηθούν και ποιες δυνατότητες και προδιαγραφές θα εξυπηρετούν.

Σε κάθε εφαρμογή πρέπει να υπάρχει το αρχείο AndroidManifest.xml, το οποίο δημιουργείται αυτόματα όταν ξεκινάμε καινούργιο project μίας Android application. Το αρχείο αυτό περιέχει βασικές πληροφορίες σχετικά με την εφαρμογή, τις οποίες το λειτουργικό σύστημα πρέπει να γνωρίζει προτού τρέξει οποιοδήποτε άλλο κώδικα. Οι σημαντικότερες από αυτές τις πληροφορίες περιγράφονται παρακάτω:

- ✓ Η ονομασία του πακέτου της Java της εφαρμογής (Java package).
- ✓ Η έκδοση της εφαρμογής (π.χ. 1.0, 1.4.2, 2.7 κλπ).
- ✓ Η ελάχιστη έκδοση του λειτουργικού συστήματος Android που απαιτεί η εφαρμογή (min sdk version). Για παράδειγμα αν έχει δηλωθεί min sdk version ο αριθμός 7, που ισοδυναμεί με την έκδοση Android 2.1, τότε η εφαρμογή θα μπορεί εκτελεστεί σε συσκευές με έκδοση Android μεγαλύτερη ή ίση της 2.1.
- ✓ Το όνομα της εφαρμογής, καθώς και το εικονίδιό της.
- ✓ Οι άδειες που απαιτούνται για να εκτελεστούν ορισμένες λειτουργίες της εφαρμογής.
- ✓ Ιδιαίτερο στοιχείο το οποίο δηλώνεται στο αρχείο Manifest είναι οι άδειες πρόσβασης της εφαρμογής. Στο Android οι διάφοροι πόροι της συσκευής

(δίκτυο, εξωτερική κάρτα μνήμης, GPS) προστατεύονται και για να μπορεί κάποια εφαρμογή να τους χρησιμοποιήσει πρέπει να έχει δηλωθεί η αντίστοιχη άδεια στο αρχείο αυτό.

Για παράδειγμα, αν η εφαρμογή μας χρησιμοποιεί την κάμερα της συσκευής, θα πρέπει να δηλώνεται και αντίστοιχη άδεια. Αν θέλουμε να έχουμε πρόσβαση στο διαδίκτυο από την εφαρμογή μας, θα πρέπει να δηλώνεται η αντίστοιχη άδεια. Όμοια ισχύουν για το αν θέλουμε να αποθηκεύσουμε αρχεία στην κάρτα SD, αν θέλουμε να στείλουμε μηνύματα SMS, αν θέλουμε να πραγματοποιήσουμε τηλεφωνικές κλήσεις κλπ. Αν δε δηλώσουμε τις άδειες που απαιτούνται για τις διάφορες λειτουργίες της εφαρμογής, τότε θα εμφανίζεται σφάλμα και η εφαρμογή δε θα λειτουργεί. Οι άδειες αυτές περιγράφονται στο χρήστη τη στιγμή που εγκαθιστά την εφαρμογή, και πρέπει να συμφωνήσει με αυτές για να ολοκληρωθεί η εγκατάσταση.

### **Activity**

Η Activity είναι το πιο σύνηθες από τα τέσσερα άλλα δομικά στοιχεία. Ένα activity σχετίζεται με μια διαφορετική οθόνη της εφαρμογής. Κάθε activity είναι μια ξεχωριστή κλάση που κάνει extend την βασική κλάση Activity του Android. Η κλάση αυτή εμφανίζει στην οθόνη ένα interface χρήστη που συντίθεται από Views και απαντά σε διάφορα γεγονότα (events). Για παράδειγμα, μια εφαρμογή γραπτών μηνυμάτων μπορεί να έχει μια οθόνη που να δείχνει μια λίστα με τις επαφές στις οποίες μπορεί να σταλεί το συγκεκριμένο μήνυμα, μια δεύτερη οθόνη για εγγραφή του μηνύματος στη δεδομένη επαφή και άλλες οθόνες που να κάνουν ρυθμίσεις των μηνυμάτων ή να δείχνουν το ιστορικό αποστολής μηνυμάτων. Κάθε μία από τις οθόνες αυτές θα υλοποιούνταν ως μια activity. Η εναλλαγή μεταξύ των διαφορετικών "οθονών" ισοδυναμεί με έναρξη μιας καινούριας activity. Σε μερικές περιπτώσεις μια Activity μπορεί να επιστρέψει μια τιμή σε μια προηγούμενη activity.

Όταν μια νέα οθόνη ανοίγει, η προηγούμενη μπαίνει σε αναμονή λειτουργίας και τοποθετείται σε στοίβα με τις πρόσφατα ανολοκλήρωτες activities. Ο χρήστης μπορεί να πλοηγηθεί προς τα πίσω και να δει τα προηγούμενα ανοιχτά screens. Τα screens αυτά μπορεί επίσης να επιλεγεί να διαγραφούν από τη στοίβα ιστορικού όταν καταλαμβάνουν χώρο και δεν χρειάζεται να μένουν εκεί.

Το λογισμικό Android χρησιμοποιεί μια ειδική κλάση που ονομάζεται Intent για το "πέρασμα" από activity σε activity (οθόνη σε οθόνη). Τα δύο πιο σημαντικά στοιχεία της αρχιτεκτονικής λειτουργίας του intent είναι η ενέργεια και τα δεδομένα επί των οποίων

θα ενεργήσει. Τυπικές τιμές για μια δράση/ενέργεια είναι οι MAIN, VIEW, PICK, EDIT κ.α. Τα δεδομένα εκφράζονται μέσω ενός URI (Uniform Resource Indicator). Για παράδειγμα, για να δούμε ένα site στον browser , θα πρέπει να δημιουργήσουμε ένα Intent με τη δράση ACTION και με τα δεδομένα να καθορίζονται από το website-uri.

Υπάρχει επίσης μια σχετική κλάση που ονομάζεται IntentFilter. Αν το intent είναι μία αίτηση για να γίνει κάτι, το intent filter περιγράφει ποια και πόσα τέτοια intents μπορεί να χειριστεί μια activity.

### **Intent Receiver**

Ένας intent receiver χρειάζεται όταν ο προγραμματιστής της εφαρμογής θέλει να χρησιμοποιήσει κώδικα μέσα στην εφαρμογή του, που θα εκτελείται όταν συμβαίνει ένα εξωτερικό γεγονός, για παράδειγμα όταν χτυπά το τηλέφωνο ή όταν ένα ασύρματο δίκτυο γίνεται διαθέσιμο. Οι intent receivers δεν προβάλλουν κάποιο interface χρήστη, αλλά προβάλλουν Notifications για να ειδοποιήσουν τον χρήστη, εάν κάτι σημαντικό λαμβάνει χώρα. Οι Intent Receivers δηλώνονται και αυτοί στο AndroidManifest.xml.

### **Services**

Ένα service είναι τμήμα κώδικα που εκτελείται χωρίς κάποιο interface χρήστη. Ένα καλό παράδειγμα service είναι ο media player που παίζει τραγούδια από μια λίστα. Σε μια εφαρμογή media player, είναι λογικό να υπάρχουν διάφορες οθόνες, άρα και πολλές activities, όπου ο χρήστης θα μπορεί να επιλέξει τραγούδια και να τα ακούσει. Παρόλα αυτά το playback δε θα πρέπει να χειρίζεται από μια activity, γιατί ο χρήστης περιμένει να μπορεί να περιηγηθεί στη λίστα τραγουδιών του, χωρίς το τραγούδι που ακούει εκείνη τη στιγμή να σταματήσει . Σε αυτή την περίπτωση, η κύρια activity του media player θα ξεκινήσει να εκτελεί ένα service στο background, οπότε ο χρήστης θα μπορεί να κάνει pause, rewind κλπ.

### **Content Providers**

Οι εφαρμογές μπορούν να αποθηκεύσουν τα δεδομένα τους σε αρχεία, στη βάση δεδομένων SQLite, σε preferences ή χρησιμοποιώντας οποιονδήποτε άλλο μηχανισμό που τους παρέχει αυτή τη δυνατότητα. Ένας content provider επίσης είναι χρήσιμος εάν θέλουμε τα δεδομένα μιας εφαρμογής να γίνουν διαθέσιμα και σε άλλες εφαρμογές. Ένας content provider είναι μια κλάση που υλοποιεί ένα standard set από μεθόδους, οι οποίες επιτρέπουν σε άλλες εφαρμογές να αποθηκεύουν και να ανακτούν τον τύπο δεδομένων που χειρίζεται ο content provider.

## Resources

Στα resources μιας εφαρμογής ορίζεται το layout των activities, οι διάφορες εικόνες και λεκτικά που χρησιμοποιούνται στα activities. Σε κάθε activity αντιστοιχεί ένα Layout αρχείο, το οποίο περιγράφει τη θέση των διάφορων αντικειμένων στην οθόνη.

Το layout αρχείο είναι ένα αρχείο XML. Στην πράξη το αρχείο αυτό διαμορφώνεται από κατάλληλους γραφικούς editors που προσφέρονται από ολοκληρωμένα περιβάλλοντα ανάπτυξης όπως το Eclipse.

Στην ενότητα για την διεπαφή χρήστη αναφέρθηκε ότι η διάταξη των γραφικών στοιχείων δηλώνεται σε αρχεία xml. Συγκεκριμένα στο project της εφαρμογής μας υπάρχει ο φάκελος res/layout/ στον οποίο τοποθετούμε όλα τα αρχεία xml που αφορούν στο user interface της εφαρμογής μας (το res προέρχεται από το resources).

Έστω ότι η εφαρμογή μας περιέχει τρεις διαφορετικές οθόνες (δηλαδή τρεις διαφορετικές activities), με κάθε μία να χρησιμοποιεί το δικό της user interface. Τοποθετούμε λοιπόν τα αρχεία myfirstscreen.xml, mysecondscreen.xml και mythirdscreen.xml στον παραπάνω φάκελο.

Όμως η τρίτη οθόνη της εφαρμογής μας εμφανίζει τρεις εικόνες όπως το παράδειγμα παραπάνω, άρα θέλουμε να χρησιμοποιήσουμε διαφορετική διάταξη για το landscape mode της τρίτης οθόνης. Όλα τα αρχεία xml που αφορούν στο user interface και συγκεκριμένα στην περίπτωση του landscape mode, τοποθετούνται στο φάκελο res/layout-land/ (δηλαδή layout-landscape). Άρα, λοιπόν, δημιουργούμε ένα ακόμα αρχείο xml με το ίδιο ακριβώς όνομα mythirdscreen.xml το οποίο όμως τοποθετούμε στο φάκελο res/layout-land/. Η εντολή που χρησιμοποιείται για να δηλωθεί το αρχείο xml που θα χρησιμοποιηθεί σε μία activity είναι η setContentView(). Όταν θα εκτελεστεί η εφαρμογή μας και συγκεκριμένα η εντολή setContentView() το λειτουργικό σύστημα Android θα ελέγξει τον προσανατολισμό της συσκευής μας και θα επιλέξει το κατάλληλο αρχείο xml. Αν βρισκόμαστε στην πρώτη οθόνη σε προσανατολισμό portrait και αλλάξουμε προσανατολισμό σε landscape, τότε το Android θα ελέγξει αν υπάρχει το αρχείο myfirstscreen.xml στον φάκελο res/layout-land/. Δε θα το βρει όμως διότι δεν το έχουμε ορίσει, άρα θα χρησιμοποιήσει το προεπιλεγμένο αρχείο το οποίο βρίσκεται στο φάκελο res/layout/ και τελικά η διάταξη των στοιχείων θα είναι κοινή και για τους δύο προσανατολισμούς.

Τα ίδια ισχύουν και για τη δεύτερη οθόνη. Στην τρίτη όμως οθόνη, όταν αλλάξουμε προσανατολισμό από portrait σε landscape τότε θα βρεθεί το αρχείο mythirdscreen.xml

Μαρίνα Αρντίτ – Μπικίκου Ευγενία

στο φάκελο `res/layout-land/` άρα τελικά θα χρησιμοποιηθεί αυτό για τη διάταξη των γραφικών στοιχείων. Φυσικά όταν επιστρέψουμε πάλι σε `portrait mode` θα αλλάξει και η διάταξη.

Στο παράδειγμα που ακολουθεί, ορίζεται ένα `Textview` με αναγνωριστικό `textView1`, το οποίο είναι ένα `block` κειμένου. Δεξιά από αυτό (`android: layout_toRightOf = "@+id/textView1"`), ορίζεται ένα πεδίο στο οποίο μπορεί ο χρήστης να εισάγει κείμενο, με αναγνωριστικό `editText1`.

## Handlers

Ένας `Handler` επιτρέπει την αποστολή και επεξεργασία μηνυμάτων και εκτελέσιμων αντικείμενων που σχετίζονται με το `MessageQueue` ενός νήματος. Κάθε αντικείμενο τύπου `Handler` συνδέεται με ένα νήμα και την ουρά μηνυμάτων αυτού του νήματος

Υπάρχουν δύο κύριες χρήσεις για ένα `Handler`:

- ✓ ο χρονοπρογραμματισμός μηνυμάτων και `runnables` ώστε να εκτελεστούν σε κάποιο σημείο στο μέλλον
- ✓ η τοποθέτηση στην ουρά μιας ενέργειας που πρέπει να εκτελεστεί σε ένα διαφορετικό νήμα από το τρέχον

Η δεύτερη περίπτωση έχει ιδιαίτερο ενδιαφέρον, καθώς για λόγους καλής λειτουργίας, χρονοβόρες διαδικασίες που πρέπει να τρέξουν σε κάποιο `activity`, τοποθετούνται σε διαφορετικό `thread`. Αν για παράδειγμα με το πάτημα ενός πλήκτρου πρέπει να σταλεί κάτι στο διαδίκτυο και η απάντηση να ενημερώσει κάποιο `block` στο `activity`, τότε η διαδικασία γίνεται

μέσω ενός νέου νήματος. Προκειμένου όμως το νήμα να μπορεί να ενημερώσει το `activity` που το δημιούργησε θα πρέπει να χρησιμοποιήσει ένα `handler`.

## 3.4 Εφαρμογή App Inventor

Το `App Inventor for Android` είναι μία διαδικτυακή εφαρμογή πηγαίου κώδικα που μπορούμε να δημιουργήσουμε απλές, αλλά και πιο σύνθετες, εφαρμογές για `Android` με απλό τρόπο. Η πλατφόρμα έγινε διαθέσιμη αρχικά από την `Google`, από τα τέλη του `Μαρτίου` του `2012` φιλοξενείται και συντηρείται από το `MIT Center For Mobile Learning`.

### 3.4.1 Ιστορική αναδρομή του App Inventor

Η εφαρμογή δόθηκε προς διάθεση στις 15 Δεκεμβρίου του 2010. Η ομάδα του App Inventor είχε επικεφαλής τους Hal Abelson, Mark Friendman. Στο δεύτερο εξάμηνο του 2011 η Google παραχώρησε την πλατφόρμα στο MIT για την δημιουργία του The MIT Center for Mobile Learning (πλατφόρμα εκμάθησης κινητής τηλεφωνίας). Η έκδοση του MIT ξεκίνησε το Μάρτιο του 2012.

Στις 6 Δεκεμβρίου του 2013 το MIT κυκλοφόρησε την έκδοση App Inventor 2, μετονομάζοντας την αρχική «App Inventor Classic». Σημαντικές διαφορές είναι:

- ✓ Το Block Editor στην αρχική έκδοση «έτρεχε» σε ξεχωριστό παράθυρο, χρησιμοποιώντας τη βιβλιοθήκη «Open Blocks» για τον προγραμματισμό. Στην νέα έκδοση το αντικαταστάθηκε με το «Blockly», που «τρέχει» το Block Editor στο πρόγραμμα περιήγησης (browser).
- ✓ Η δεύτερη έκδοση δίνει την δυνατότητα σε πραγματικό χρόνο τον εντοπισμό σφαλμάτων για τις συνδεδεμένες συσκευές μέσω Wifi και όχι μόνο USB.

Το Μάιο του 2014 υπήρχαν 1,9 εκατομμύρια εγγεγραμμένοι χρήστες σε 195 χώρες, για 4,7 εκατομμύρια εφαρμογές.

### 3.4.2 Προαπαιτούμενα για τη χρήση του App Inventor

Για να μπορέσει να παίξει το App Inventor προαπαιτούνται τα εξής:

- ✓ Λειτουργικό Σύστημα (Operating System):
  - Windows
  - Linux
  - Macintosh
- ✓ Φυλλομετρητής (Browser):
  - Mozilla Firefox ver 3.6 ή νεότερη
  - Apple Safari ver 5.0 ή νεότερη
  - Google Chrome ver 4.0 ή νεότερη
  - Microsoft Internet Explorer 7 ή νεότερη
- ✓ Java 1.6 ή νεότερη
- ✓ Εγκατάσταση του App Inventor Setup Software
- ✓ Λογαριασμό στην Google (gmail)

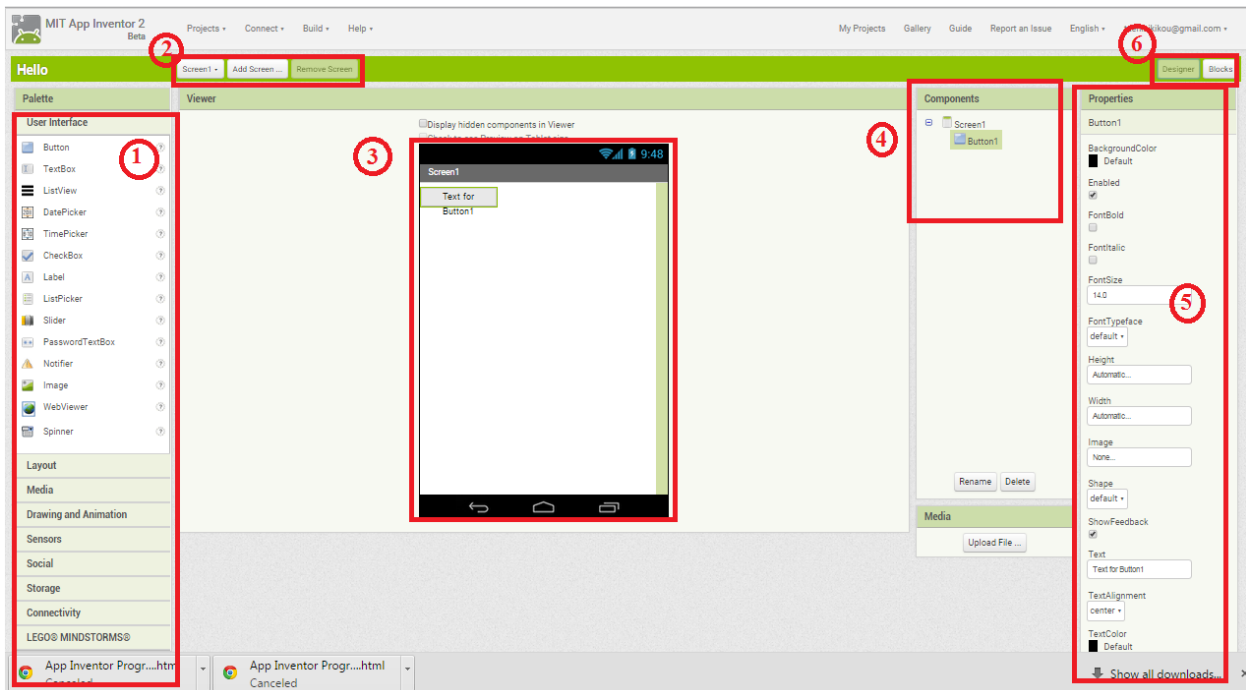
### 3.4.3 Περιβάλλον του App Inventor

Εφόσον κάνουμε είσοδο (login) στην πλατφόρμα θα εμφανιστεί η τελευταία εφαρμογή με την οποία εργαστήκαμε στην προηγούμενη είσοδο μας.

Στα αριστερά βρίσκονται οι καρτέλες με ομαδοποιημένα τα διάφορα εργαλεία που μπορούμε να σύρουμε και να τα βάλουμε στην εφαρμογή μας (Εικόνα 3.2 το 1, πχ κουμπιά (buttons), ετικέτες (labels) και άλλα). Στο κέντρο εμφανίζεται ένα κινητό όπου μπορούμε να τοποθετήσουμε διάφορα αντικείμενα και να δούμε πως θα φαίνονται στο κινητό μας(Εικόνα 3.2 το 3). Πάνω αριστερά έχει 3 επιλογές (Εικόνα 3.2 το 2): το πρώτο έχει την λίστα με όλα τα interfaces που έχουμε δημιουργήσει και ακριβώς δίπλα το κουμπί «Προσθήκης» νέου interface και τέλος το κουμπί διαγραφής interface. Πάνω δεξιά έχει 2 κουμπιά που μας δίνουν την δυνατότητα να περιηγούμαστε ανάμεσα στα παράθυρα της σχεδίασης και των block διαγραμμάτων προγραμματισμού (Εικόνα 3.2 το 6).

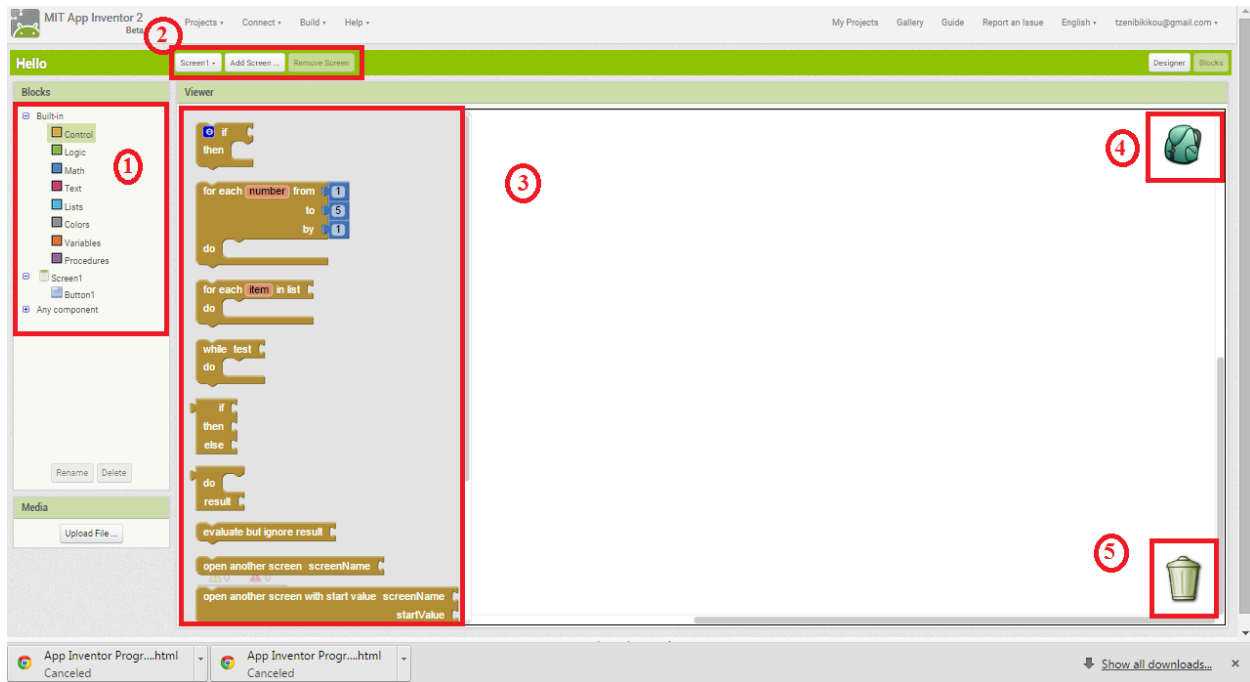
Τέλος στα δεξιά υπάρχουν δύο στήλες:

- ✓ Components: όπου μας δείχνει σε μορφή διαγράμματος όλα τα αντικείμενα που έχουν τοποθετηθεί πάνω στην οθόνη (Εικόνα 3.2 το 4).
- ✓ Properties: εμφανίζει όλες τις ιδιότητες του κάθε φορά επιλεγμένου αντικειμένου (Εικόνα 3.2 το 5).



Εικόνα 3.2: Επιφάνεια εργασίας App Inventor

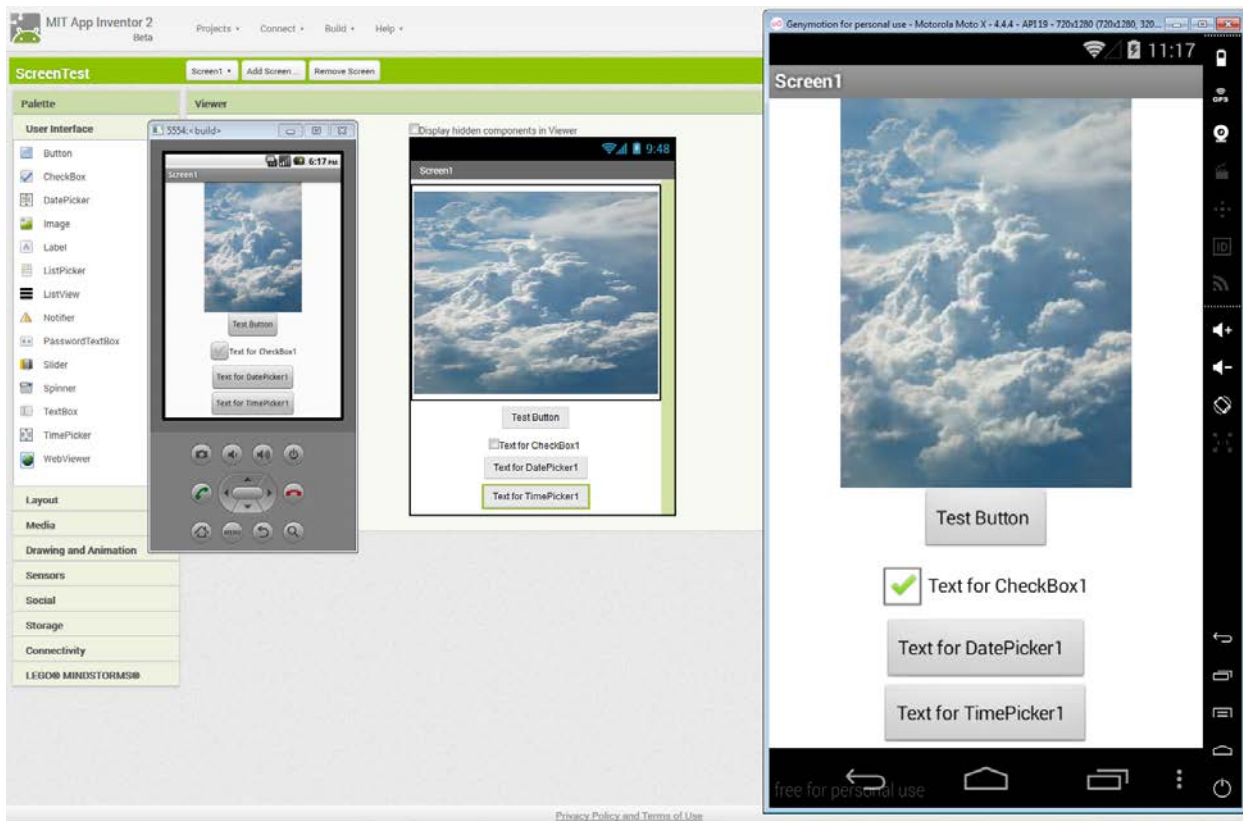




Εικόνα 3.3: Επιφάνεια προγραμματισμού του App Inventor

Στην επιφάνεια προγραμματισμού του App Inventor βλέπουμε στα δεξιά είναι ο χώρος όπου διαμορφώνουμε το σενάριο-πρόγραμμα για τις λειτουργίες της εφαρμογής (Εικόνα 3.3 1 και 3). Πάνω αριστερά έχουμε 3 δυνατότητες: να επιλέξουμε ποιού interface θέλουμε να επεξεργαστούμε τον κώδικα, να προσθέσουμε νέο κώδικα και τέλος να διαγράψουμε κάποιο κώδικα (Εικόνα 3.3 το 2). Κάτω δεξιά έχει έναν κάδο ανακύκλωσης (recycling bin Εικόνα 3.3 5) και στα αριστερά είναι όλα τα έτοιμα εργαλεία για την κατασκευή του σεναρίου.

Τέλος, το App Inventor και προσομοιωτή (emulator) όπου μπορούμε να δούμε το πώς θα παίζει η εφαρμογή και στο κινητό.

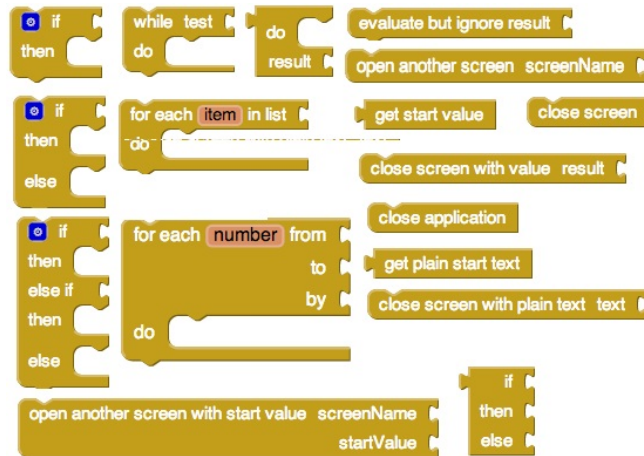


Εικόνα 3.4: Προσομοιωτής του App Inventor

### 3.4.4 Συναρτήσεις του App Inventor

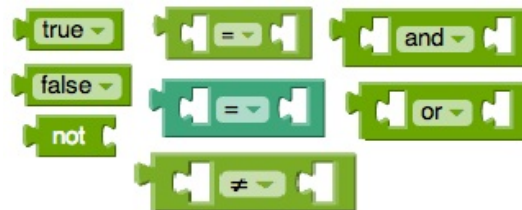
Το App Inventor χωρίζει τις διαθέσιμες συναρτήσεις σε κατηγορίες (Εικόνα 3.3 το 1):

- ✓ **Control (Ελέγχου):** Περιλαμβάνει συναρτήσεις ελέγχου όπως: if & if else, for, while, if then else, do, αξιολόγηση αποτελεσμάτων (evaluate but ignore result), άνοιγμα νέας οθόνης (open another screen), άνοιγμα νέας οθόνης με αρχικές τιμές, (open another screen with start value), αρχικοποίηση τιμών (get start value), κλείσιμο οθόνης (close screen), κλείσιμο οθόνης με τιμές, (close screen with value), κλείσιμο της εφαρμογής (close application), μορφοποίηση αρχικού κειμένου (get plain start text), κλείσιμο οθόνης με μορφοποιημένο κείμενο (close screen with plain text)



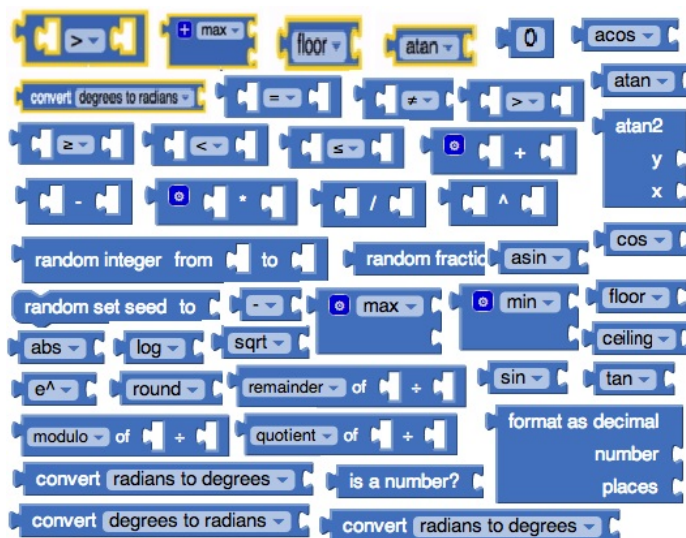
Εικόνα 3.5: Control blocks

- ✓ **Logic (Λογικές):** Περιλαμβάνει λογικές συναρτήσεις όπως: αληθές (**true**), ψευδές (**false**), όχι (**not**), ισότητα (**=**), ανισότητα (**≠**), λογικό ΚΑΙ (**and**), λογικό Ή (**or**).



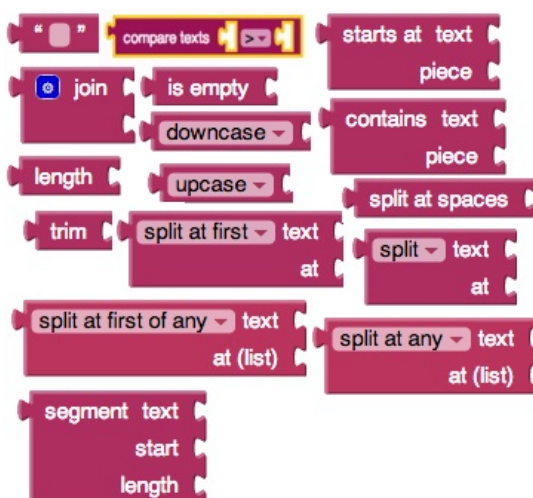
Εικόνα 3.6: Logic blocks

- ✓ **Math (Μαθηματικές):** Περιλαμβάνει μαθηματικές πράξεις, όπως: 0 (βασικός αριθμός), ίσον (**=**), διάφορο ίσον (**≠**), μεγαλύτερο (**>**), μεγαλύτερο ή ίσον (**≥**), μικρότερο (**<**), μικρότερο ή ίσον (**≤**), συν (**+**), πλην (**-**), επί (**\***), δια (**/**), εκθέτης (**^**), τυχαίος ακέραιος (**random integer**), τυχαίος αριθμός από το 0 έως το 1 (**random fraction**), επαναλαμβανόμενες αλληλουχίες τυχαίων αριθμών (**random set seed to**), ελάχιστο (**min**), μέγιστο (**max**), τετράγωνο (**sqrt**), απόλυτο (**abs**), λογάριθμος (**log**), φυσικός αριθμός e σε δύναμη (**e^**), περίπου (**round**), με δεκαδικά ψηφία (**floor**), μόντουλο (**modulo**), **remainder**, **quotient**, ημίτονο (**sin**), συνημίτονο (**cos**), εφαπτομένη (**tan**), τόξο ημιτόνου (**asin**), τόξο συνημιτόνου (**acos**), τόξο εφαπτομένης (**atan**), **atan2**, μετατροπή ακτίνων σε μοίρες (**convert radians to degrees**), μοίρες σε ακτίνια (**convert degrees to radians**), μορφοποίηση δεκαδικών (**format as a decimal**), αναγνώριση αριθμού (**is a number**), μετατροπή σε αριθμό (**convert number**).



Εικόνα 3.7: Math blocks

- ✓ **Text (Κειμένου):** Περιλαμβάνει συναρτήσεις για την επεξεργασία κειμένων, όπως: αλυσίδα χαρακτήρων (*string*), σύνδεση εισόδων από χαρακτήρες (*join*), μήκος κειμένου (*length*), έλεγχος ύπαρξης χαρακτήρων (*is empty?*), σύγκριση κειμένων (*compare texts*), κόψιμο αλυσίδας χαρακτήρων (*trim*), μεαροπή σε κεφαλαία (*upcase*), μετατροπή σε πεζά (*downcase*), αρχικό γράμμα (*starts at*), έλεγχος αν υπάρχει κάποιος χαρακτήρας (*contains*), διάσπαση κειμένου (*split at first*, *split at first of any*, *split*, *split at any*, *split at spaces*), εξαγωγή μέρους του κειμένου (*segment*), αντικατάσταση κειμένου (*replace all*).



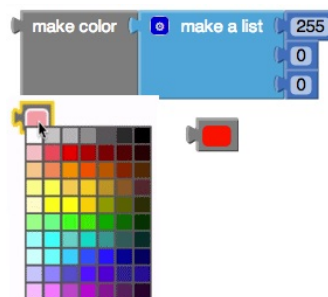
Εικόνα 3.8: Text blocks

- ✓ **Lists (Λίστας):** Περιλαμβάνει συναρτήσεις λίστας, όπως: δημιουργία κενής λίστας (create empty list), δημιουργία λίστας (make a list), προσθήκη περιεχομένου στην λίστα (add items to list), εύρεση αν υπάρχει στην λίστας (is in list), μήκος λίστας (length of list), έλεγχος αν είναι κενή μία λίστα (is list empty), τοποθέτηση τυχαίου αντικειμένου σε λίστα (pick a random item), εύρεση σε λίστα (index in list), επιλογή αντικειμένου από την λίστα (select list item), εισαγωγή αντικειμένου στην λίστα (insert list item), αντικατάσταση αντικειμένου στην λίστα (replace list item), διαγραφή αντικειμένου από την λίστα (remove list item), προσάρτηση σε λίστα (append to list), αντιγραφή σε λίστα (copy list), έλεγχος για το αν είναι λίστα (is a list?), διαχείριση λίστας από και σε csv (comma-separated value) (list to csv row, list to csv table, list from csv row, list from csv table).



Εικόνα 3.9: Lists blocks

- ✓ **Colors (Χρωμάτων):** Περιλαμβάνει συναρτήσεις επεξεργασίας χρωμάτων, όπως: χρώμα που επιθυμούμε (a color box), δημιουργία χρώματος (make color), διάσπαση χρώματος (split color).



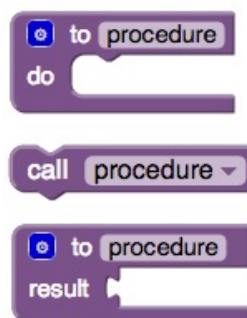
Εικόνα 3.10: Color blocks

- ✓ **Variables (Μεταβλητών):** Περιλαμβάνει συναρτήσεις διαχείρισης μεταβλητών, όπως: αρχικοποίηση μίας καθολικής μεταβλητής (initialize global name to), να πάρει τιμή από μία μεταβλητή (get), να θέσουμε μία τιμή σε μία μεταβλητή (set), αρχικοποίηση μίας τοπικής μεταβλητής (initialize local name to in (do), initialize local name to in (return)).



Εικόνα 3.11: Variable blocks

- ✓ **Procedures (Διαδικασιών):** Περιλαμβάνει συναρτήσεις διαχείρισης διαδικασιών όπως: συλλογή ακολουθιών από μπλοκς σε ένα γκρουπ (procedure do), συλλογή ακολουθιών από μπλοκς σε ένα γκρουπ βγάζοντας και το αποτέλεσμα τους (procedure result).



Εικόνα 3.12: Procedure blocks



## ΚΕΦΑΛΑΙΟ 4: ΥΛΟΠΟΙΗΣΗ ΕΡΓΑΣΙΑΣ

Σε αυτό το κεφάλαιο θα δούμε την υλοποίηση της εργασίας. Θα δούμε τα εξαρτήματα που χρησιμοποιήθηκαν για την κατασκευή του κυκλώματος, αλλά και την υλοποίηση και τα βασικότερα κομμάτια της εφαρμογής Smart Home.

### 4.1 Υλοποίηση πλακέτας Android

#### 4.1.1 Κομμάτια πλακέτας

##### Arduino Mega 2560



Εικόνα 4.1: Arduino Mega

Το Arduino Mega έχει το μικροελεγκτή ATmega328 που το κάνει να είναι και η "καρδιά" του κυκλώματος που ελέγχει όλες τις εισόδους και εξόδους του.

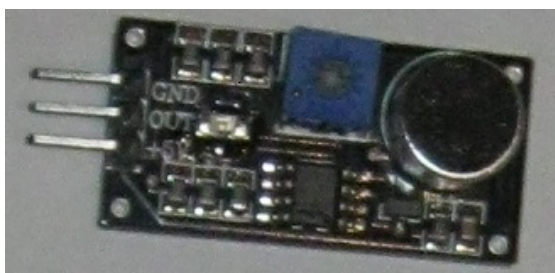
##### Arduino Nano 328



Εικόνα 4.2: Arduino Nano 328

Το Arduino Nano 328 είναι μια μικρή, πλήρης πλακέτα που έχει τον μικροελεγκτή ATmega328. Τροφοδοτείται μέσω mini B USB.

##### Audio Sensor



Εικόνα 4.3: Audio Sensor

Ο ανιχνευτής ήχου αφουγκράζεται τους εξωτερικούς ήχους και τους δημιουργεί ως εισόδους για την πλακέτα του Arduino ώστε να τις επεξεργαστεί και να δώσει τις αντίστοιχες εξόδους κάθε φορά.

### MQ2 GAS Sensor



Εικόνα 4.4: MQ2 GAS Sensor

Οι MQ2 αισθητήρες αερίου χρησιμοποιούν ένα μικρό καλοριφέρ με ένα ηλεκτροχημικό αισθητήρα. Αυτά είναι ευαίσθητα και χρησιμοποιούνται για την ανίχνευση αερίων σε περίπτωση διαρροής. Η έξοδος του είναι αναλογική και μπορεί να συνδεθεί σε μία αναλογική είσοδο του Arduino.

### DHT11 Temp-Humidity Sensor



Εικόνα 4.5: DHT11 Temp-Humidity Sensor

Ψηφιακός αισθητήρα θερμοκρασίας και υγρασίας. Ανιχνεύει την θερμοκρασία και την υγρασία του περιβάλλοντα χώρου και τα αποτελέσματα τα βγάζει σε ψηφιακή έξοδο.

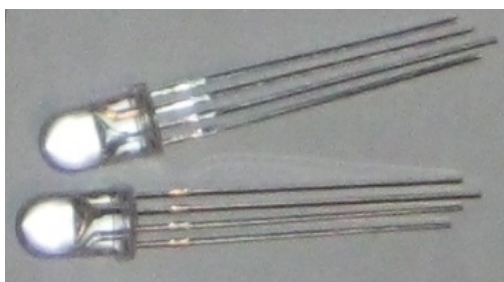
### PIR Sensor



Εικόνα 4.6: PIR Sensor

Είναι αισθητήρες κίνησης. Οι αισθητήρες αυτοί είναι κατασκευασμένοι από ένα πυροηλεκτρικό αισθητήρα, ο οποίος μπορεί να ανιχνεύσει επίπεδα υπέρυθρης ακτινοβολίας (όλα εκπέμπουν υπέρυθρη ακτινοβολία, όσο πιο θερμό τόσο περισσότερη ακτινοβολία εκπέμπει).

### RGB Leds

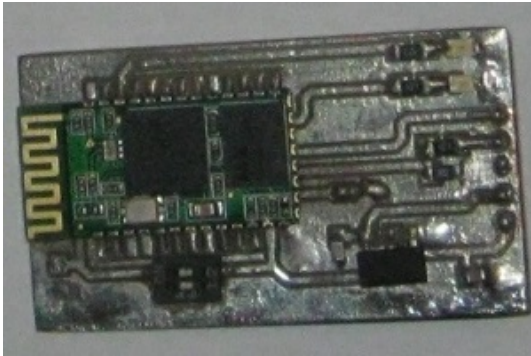


Εικόνα 4.7: RGB LEDS

Led φωτάκια τεχνολογίας RGB (red-green-blue)



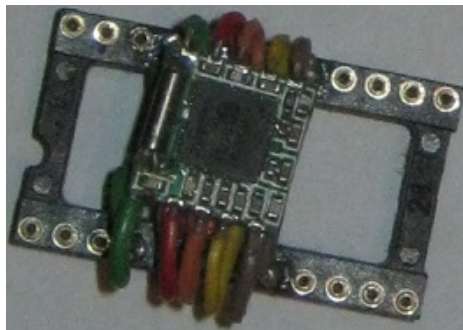
### Bluetooth Connector



Εικόνα 4.8: Bluetooth Connector

Είναι μία συσκευή Bluetooth η οποία επιτρέπει στο κύκλωμα να συνδεθεί με την Android συσκευή και να δέχεται αλλά και να στέλνει δεδομένα από και προς την εφαρμογή Smart Home.

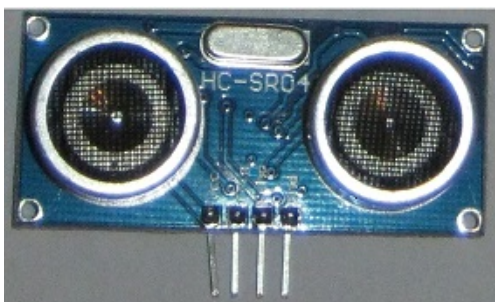
### Radio TEA5767



Εικόνα 4.9: Radio TEA5767

Το TEA5767 είναι ένας δέκτης FM, που ελέγχεται μέσω I2C οθόνες. Το ραδιόφωνο μπορεί να συντονιστεί με Ευρώπη, ΗΠΑ και ιαπωνικές συχνότητες FM.

### Ultra Sonic Sensor



Εικόνα 4.10: Ultra Sonic Sensor

Ο HC-SR04 είναι ένας αισθητήρας υπερήχων που μπορεί και υπολογίζει την απόσταση που έχει ένα αντικείμενο. Είναι εξαιρετικής ακρίβειας.

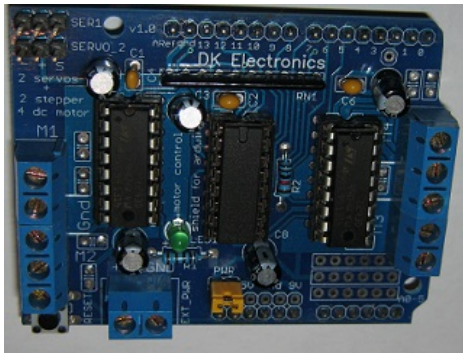
### Capacitors



Εικόνα 4.11: Capacitors

Ο πυκνωτής (capacitor) χρησιμοποιείται για την προσωρινή αποθήκευση ηλεκτρικής ενέργειας σε ένα ηλεκτρικό πεδίο

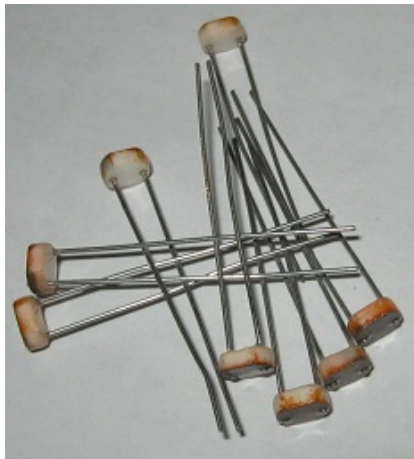
### Motor Shield



Εικόνα 4.12: Motor Shield

Το Motor Shield επιτρέπει τον έλεγχο της ταχύτητας και της κατεύθυνσης 2 κινητήρων συνεχούς ρεύματος, ξεχωριστά.

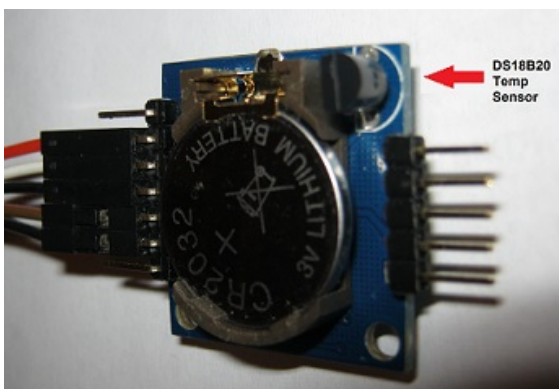
### LDR



Εικόνα 4.13: LDR

Το LDR είναι αισθητήρας φωτός, μπορεί «καταλάβει» αν έχει φως η σκοτάδι και να στείλει την αντίστοιχη έξοδο στο Arduino.

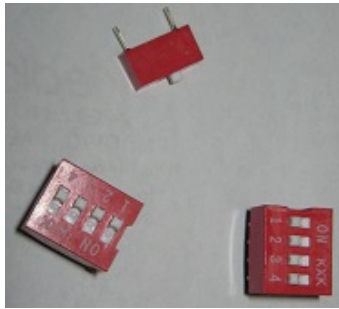
### Real Time Clock DS1307+DS18B20



Εικόνα 4.14: Real Time Clock DS1307+18B20

Το DS13007 είναι ένα σειριακό ρολόι πραγματικού χρόνου. Είναι σε δυαδικό κώδικα και παρέχει πληροφορίες δευτερολέπτων, λεπτών, ώρας και ημερομηνίας. Το DS18B20 είναι ένα ψηφιακό θερμομέτρο που παρέχει μετρήσεις θερμοκρασίας σε βαθμούς Κελσίου αλλά και ειδοποιήσεις.

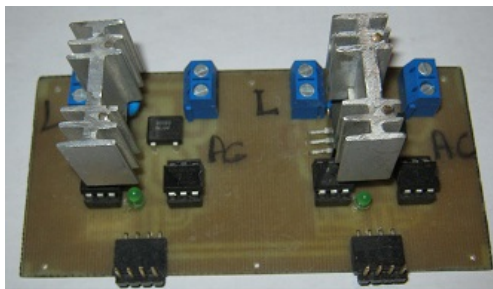
### Micro switches



Εικόνα 4.15: Micro Switches

Διακόπτες εισόδων Arduino. Τοποθετούνται στο κύκλωμα και μπορούμε χειροκίνητα να δώσουμε τιμές 1 και 0 για εισόδους.

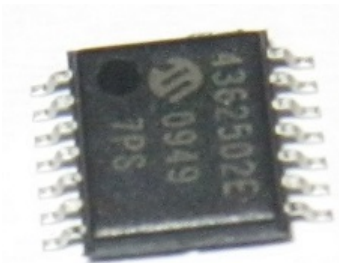
### Dimmer



Εικόνα 4.16: Dimmer

Το Dimmer ελέγχει την ένταση της φωτεινότητας ενός Led

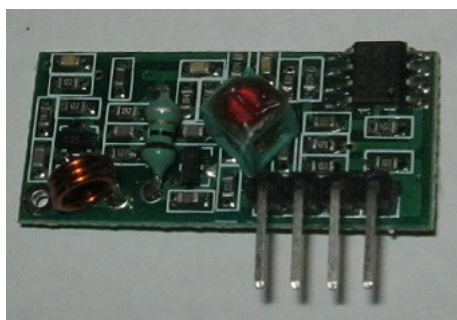
### Digital POT



Εικόνα 4.17: Digital POT

Το Ψηφιακό ποτενσιόμετρο, μας δίνει την δυνατότητα να ελέγχουμε την αντίσταση ενός κυκλώματος χωρίς να είναι απαραίτητο να τη ρυθμίζουμε χειροκίνητα.

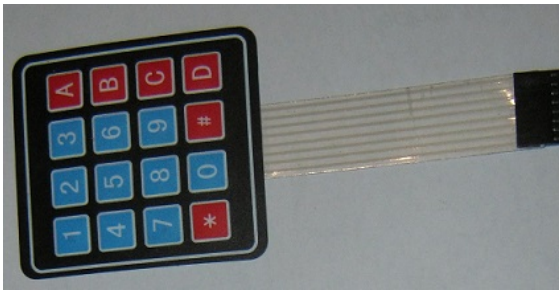
### Transmitter Receiver 477Hz



Εικόνα 4.18: Transmitter Receiver 477Hz

Το Transmitter Receiver είναι πομπός δέκτης σαν το Bluetooth σε συχνότητες 477Hz

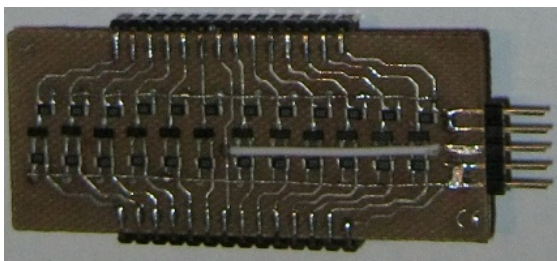
### Keypad 4x4



Εικόνα 4.19: Keypad 4x4

Είναι ένα πληκτρολόγιο 16 χειροκίνητων κουμπιών που προσφέρουν μία εύκολη στον χειρισμό έργων με μικροελεγκτές.

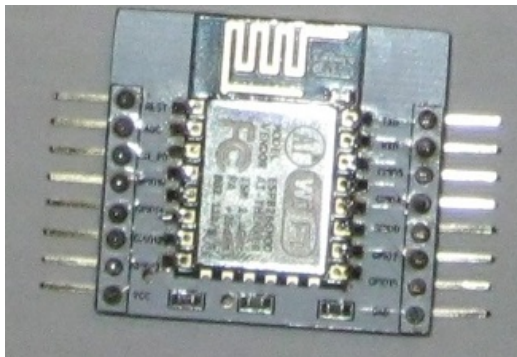
### Lever Shifter



Εικόνα 4.20: Lever Shifter

Μεταγάγει την τάση από 3,3V σε 5V και αντίστροφα.

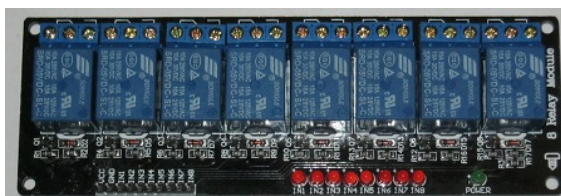
### WiFi Connection



Εικόνα 4.21: WiFi Connection

Δίνει πρόσβαση σε WiFi σύνδεση του Arduino με διαφορετικές συσκευές.

### Relay Shield

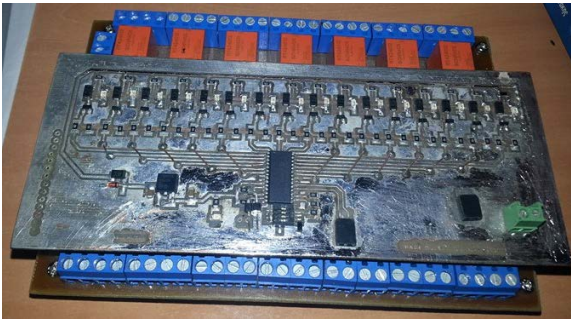


Εικόνα 4.22: Relay Shield

Το Relay Shield είναι μία «έξυπνη» μονάδα συμβατή με το Arduino και ελέγχει για υψηλή τάση.



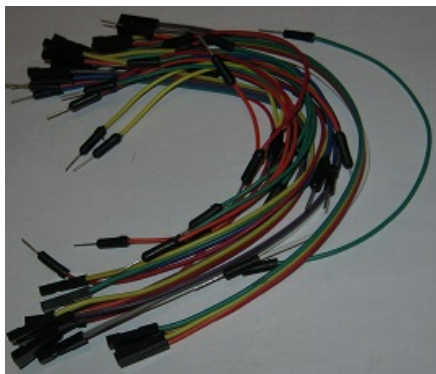
## Relay Board



Εικόνα 4.23: Relay Board

Το Relay Board έχει I2C πρωτόκολλο επικοινωνίας. Χρησιμοποιεί τον εγκέφαλο MCP23017. Έχει την δυνατότητα για interrupts

## Καλώδια



Εικόνα 4.24: Καλώδια

Καλώδια για την συνδεσμολογία.

### 4.1.2 Βασικά κομμάτια κώδικα για το Arduino

Στην παράγραφο που ακολουθεί αναλύονται κομμάτια κώδικα για τις βασικότερες λειτουργίες της πλακέτας, όπως φως, dimmer, πρίζες).

- ✓ Αρχικά έχουμε εισάγει τις βιβλιοθήκες που απαιτούνται για την λειτουργία όλου του κυκλώματος:

```
#include <Wire.h> // I2C βιβλιοθήκη
#include "RTClib.h" // RTC βιβλιοθήκη
RTC_DS1307 RTC; // RTC Module
#include <OneWire.h> // OneWire βιβλιοθήκη
#include <DallasTemperature.h> // DS18B20 library
#include "SoftwareSerial.h"
#include <SPI.h>
#include <TEA5767.h>
#include <Button.h>
#include "Adafruit_MCP23017.h"
#define DS18B20_PIN 3 // αρχικοποίηση pin
DS18B20
```

```

OneWire oneWire(DS18B20_PIN); // OneWire αναφορά
DallasTemperature sensors(&oneWire); // αρχικοποίηση DS18B20

```

- ✓ **Φως/Πρίζες/Διακόπτες:** Παρακάτω παρατήθεται ο βασικός κώδικας για τον έλεγχο στα φώτα, τις πρίζες και τις συσκευές όπως κουζίνα.

```

//Εισαγωγή απαραίτητων βιβλιοθηκών
#include <Wire.h>
#include "Adafruit_MCP23017.h"

//Δημιουργία μεταβλητών
Adafruit_MCP23017 mcp;
char rx_byte = 0;

void setup() {
  Serial.begin(9600);
  //Χρήση της αρχικής διεύθυνσης 0
  mcp.begin();
  mcp_Function_pinMode();
}

void loop() {
  //Είναι ο χαρακτήρας διαθέσιμος?
  if (Serial.available() > 0) {
    rx_byte = Serial.read();
    switch (rx_byte) {

  case 'A':
    mcp.readOutput(0);
    //Φως Off
    if (mcp.readOutput(0) == HIGH) {
      mcp.digitalWrite(0,LOW);
      Serial.println("LED is OFF");
    } else {
      //Φως On
      mcp.digitalWrite(0, HIGH);
      Serial.println("LED is ON");
    }
    break;

  case 'B':
    mcp.readOutput(1);
    //Φως Off
    if (mcp.readOutput(1) == HIGH) {
      mcp.digitalWrite(1,LOW);
      Serial.println("LED is OFF");
    } else {
      //Φως On
      mcp.digitalWrite(1, HIGH);
      Serial.println("LED is ON");
    }
    break;
  case 'C':
    mcp.readOutput(2);
    // Πρίζα Off
    if (mcp.readOutput(2) == HIGH) {
      mcp.digitalWrite(2,LOW);
      Serial.println("Jack is OFF");
    }

```

```

    } else {
        // Πρίζα On
        mcp.digitalWrite(2, HIGH);
        Serial.println("Jack is ON");
    }
    break;
    case 'D':
        mcp.digitalWrite(3, HIGH);
        //Κουζίνα On
        Serial.println("LED is ON");
        break;
    case 'd':
        mcp.digitalWrite(3, LOW);
        // Κουζίνα Off
        Serial.println("LED is Off");
        break;

    default:
        Serial.println(" ");
        break;
}
// end: switch (rx_byte)
}
// end: if (Serial.available() > 0)
}

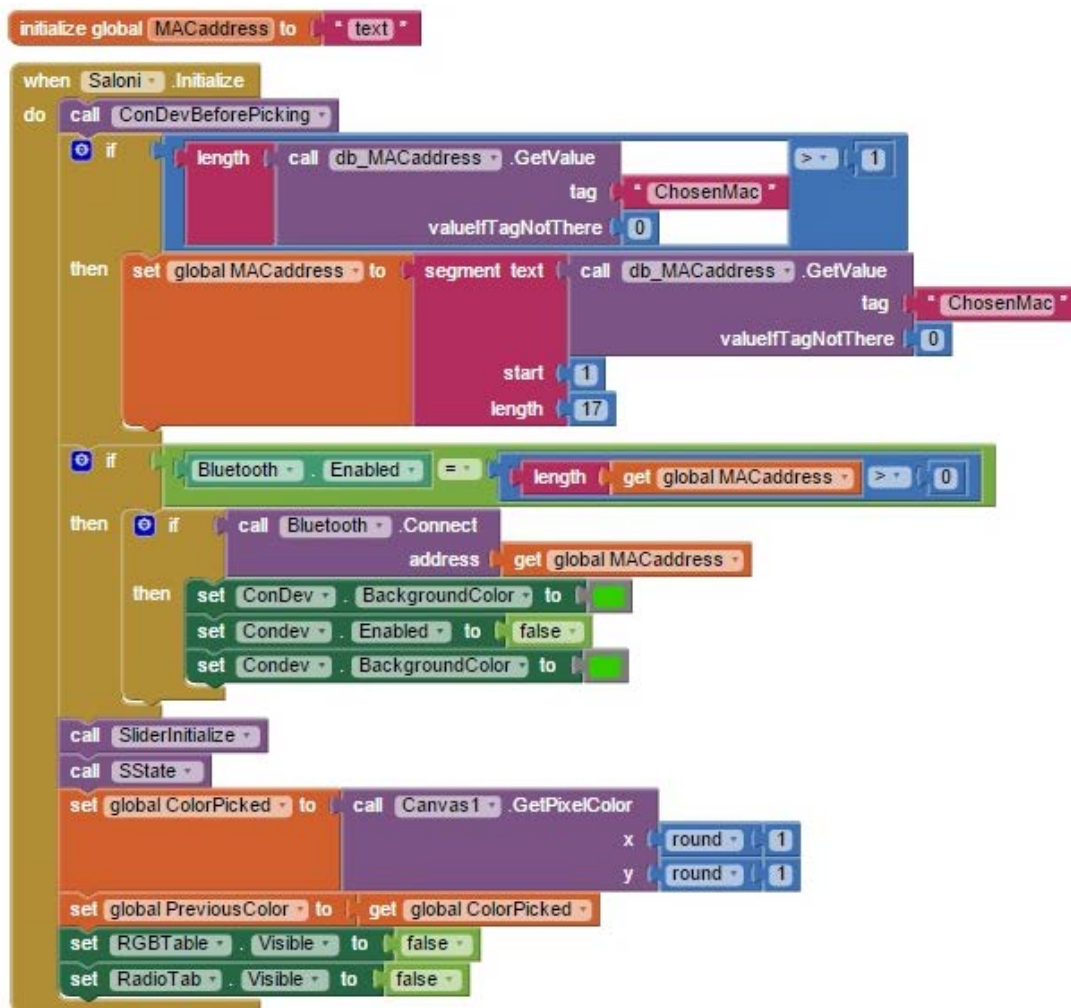
//Δηλώνεται η κατάσταση του κάθε pin ξεχωριστά
void mcp_Function_pinMode(){
    //το κάθε pin πρέπει να δηλωθεί ξεχωριστά ώστε να διαβάζουμε
την
    κατάσταση του
    mcp.pinMode(0, OUTPUT);
    mcp.pinMode(1, OUTPUT);
    mcp.pinMode(2, OUTPUT);
    mcp.pinMode(3, OUTPUT);
}
}

```

## 4.2 Κώδικας Εφαρμογής Android

Όπως αναφέρθηκε και στο 3<sup>ο</sup> Κεφάλαιο για την κατασκευή της εφαρμογής Smart Home χρησιμοποιήθηκε η πλατφόρμα *AppInventor*, που με απλό και γρήγορο τρόπο μας επιτρέπει να σχεδιάσου με το interface της εφαρμογής, καθώς επίσης και τον κώδικα που τρέχει από πίσω σε μορφή μπλοκ-διαγράμματος (block diagram) Στην ενότητα που ακολουθεί παρουσιάζονται τα βασικότερα κομμάτια του κώδικα.

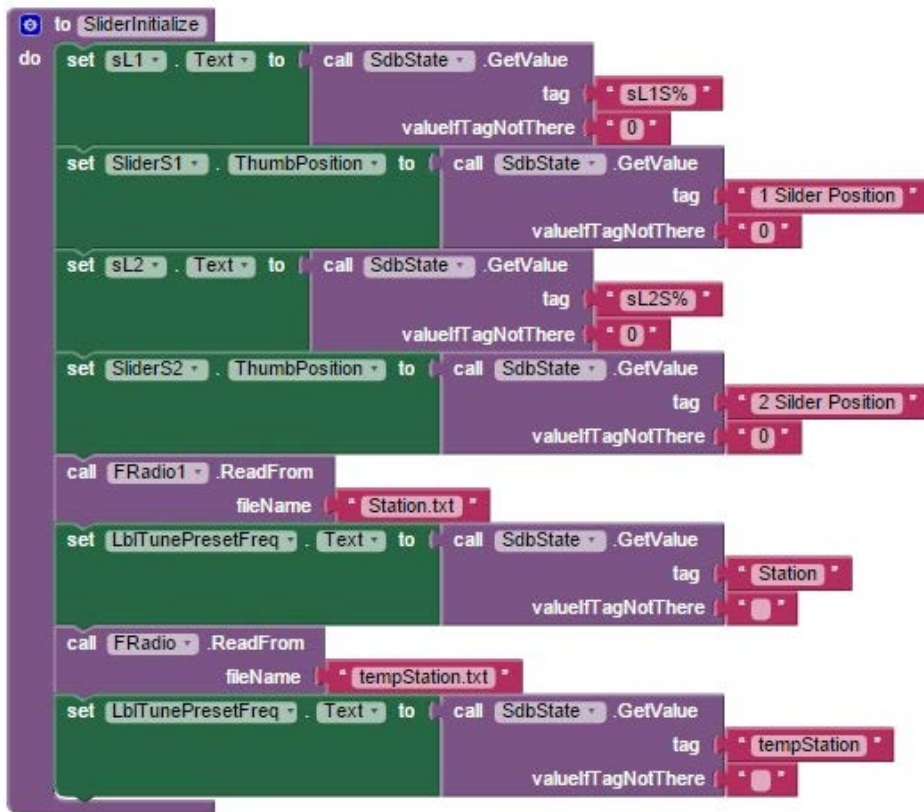
### 4.2.1 Εμφάνιση επιλογών του Σαλονιού



Εικόνα 4.25: Block κώδικα Interface Σαλονιού (1)

Κατά την είσοδο στο «Σαλόني» η εφαρμογή αναζητά στην βάση δεδομένων την επιθυμητή MAC του Bluetooth. Στην συνέχεια ελέγχει αν το Bluetooth του κινητού συνδέθηκε με το Arduino και πρασινίζει το αντίστοιχο εικονίδιο στο interface, αλλά και αρχικοποιεί όλο το interface της εφαρμογής.



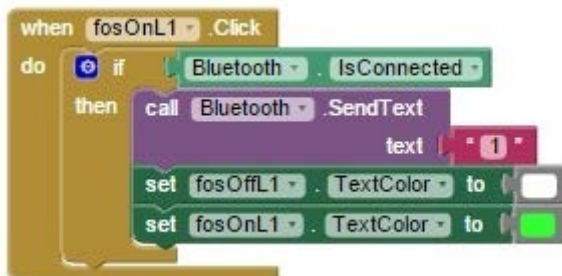


Εικόνα 4.26: Block κώδικα Interface Σαλονιού (2)

Κατά την αρχικοποίηση του Σαλονιού εμφανίζονται τα κείμενα με τα με τα αντίστοιχα κουμπιά (slider buttons, buttons) και αποκρύπτει το interface του RGB πίνακα και του HiFi και εφόσον επιλεγθούν από τον χρήστη τότε θα εμφανιστούν και θα «κρυφτούν» τα μη επιθυμητά κείμενα και κουμπιά.

#### 4.2.2 Φως

##### Ενεργοποίηση λάμπας

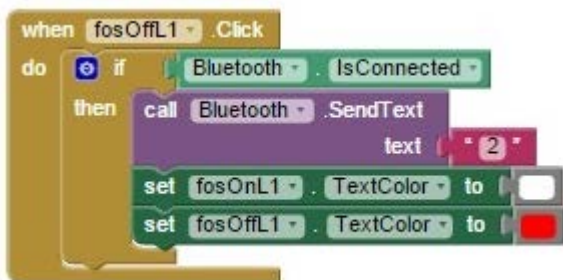


Εικόνα 4.27: Block κώδικα ενεργοποίησης λάμπας

Για να μπορέσει να ανάψει το φως εκτελούνται οι εξής διαδικασίες:

Μόλις πατηθεί το κουμπί **on** η εφαρμογή ελέγχει αν το Bluetooth είναι συνδεδεμένο με την πλακέτα εάν **NAI** τότε στέλνει το κείμενο **“1”** στην πλακέτα του Arduino και τέλος αλλάζει το χρώμα του κειμένου στο interface στιγμιαία σε γκρι και αμέσως μετά σε πράσινο.

### Απενεργοποίηση λάμπας



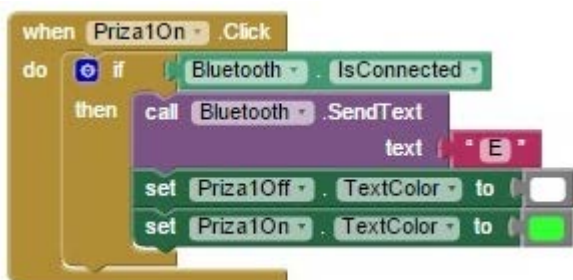
Εικόνα 4.28: Block κώδικα απενεργοποίησης λάμπας

Για να μπορέσει να σβήσει το φως εκτελούνται οι εξής διαδικασίες:

Μόλις πατηθεί το κουμπί **off** η εφαρμογή ελέγχει αν το Bluetooth είναι συνδεδεμένο με την πλακέτα εάν **ΝΑΙ** τότε στέλνει το κείμενο **"2"** στην πλακέτα του Arduino και τέλος αλλάζει το χρώμα του κειμένου στο interface στιγμιαία σε γκρι και αμέσως μετά σε κόκκινο.

## 4.2.3 Πρίζες

### Ενεργοποίηση Πρίζας

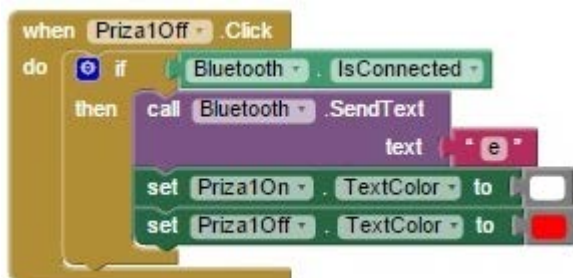


Εικόνα 4.29: Block κώδικα ενεργοποίησης Πρίζας

Για να μπορέσει να ενεργοποιήσει μία πρίζα εκτελούνται οι εξής διαδικασίες:

Μόλις πατηθεί το κουμπί **on** η εφαρμογή ελέγχει αν το Bluetooth είναι συνδεδεμένο με την πλακέτα εάν **ΝΑΙ** τότε στέλνει το κείμενο **"E"** στην πλακέτα του Arduino και τέλος αλλάζει το χρώμα του κειμένου στο interface στιγμιαία σε γκρι και αμέσως μετά σε πράσινο.

### Απενεργοποίηση Πρίζας



Εικόνα 4.30: Block κώδικα απενεργοποίησης Πρίζας

Για να μπορέσει να απενεργοποιήσει μία πρίζα εκτελούνται οι εξής διαδικασίες:

Μόλις πατηθεί το κουμπί **off** η εφαρμογή ελέγχει αν το Bluetooth είναι συνδεδεμένο με την πλακέτα εάν **ΝΑΙ** τότε στέλνει το κείμενο **"e"** στην πλακέτα του Arduino και τέλος αλλάζει το χρώμα του κειμένου στο interface στιγμιαία σε γκρι και αμέσως μετά σε κόκκινο.

## ΚΕΦΑΛΑΙΟ 5: MANUAL ΕΦΑΡΜΟΓΗΣ

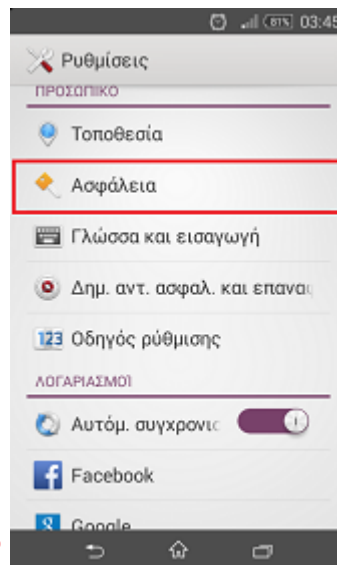
### 5.1 Εγκατάσταση εφαρμογής

Αρχικά χρειάζεται να κατεβάσουμε την εφαρμογή στην Android Συσκευή μας. Στην συνέχεια ακολουθήστε τα παρακάτω βήματα για την εγκατάσταση:

1. Πατήστε «Μενού» -> «Ρυθμίσεις» -> «Ασφάλεια» και τσεκάρτε την επιλογή «Άγνωστες πηγές»



Εικόνα 5.1: Μενού

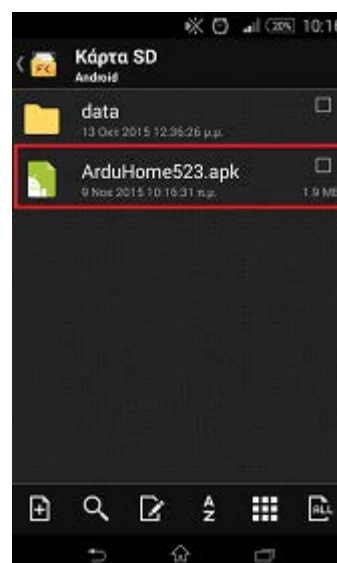


Εικόνα 5.2: Ρυθμίσεις



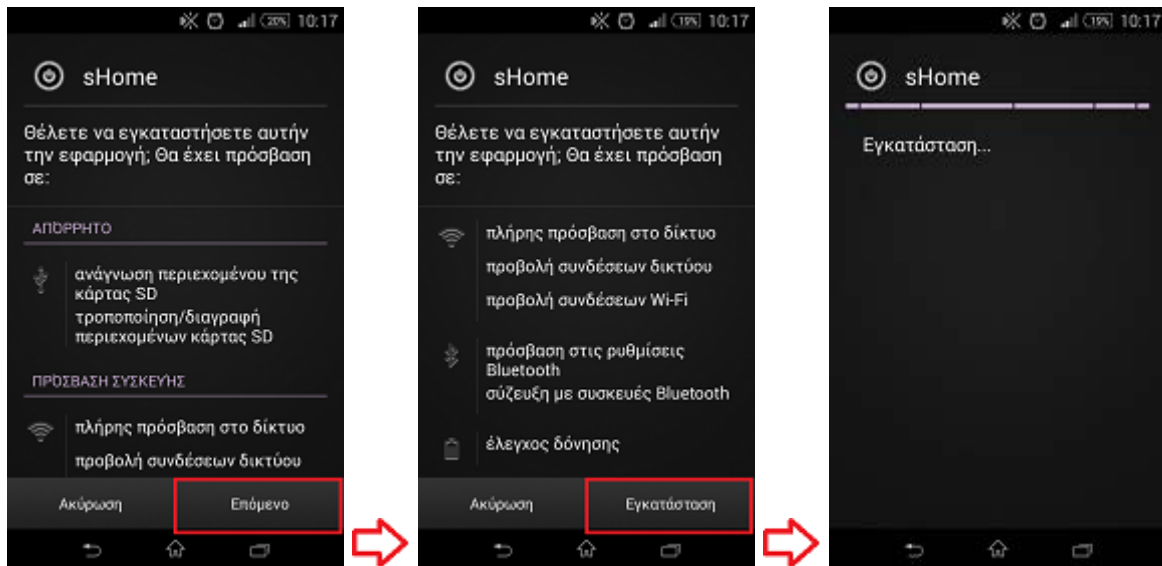
Εικόνα 5.3: Ασφάλεια

2. Πηγαίνεται στον φάκελο που είναι αποθηκευμένο το αρχείο ArduHome523.apk.



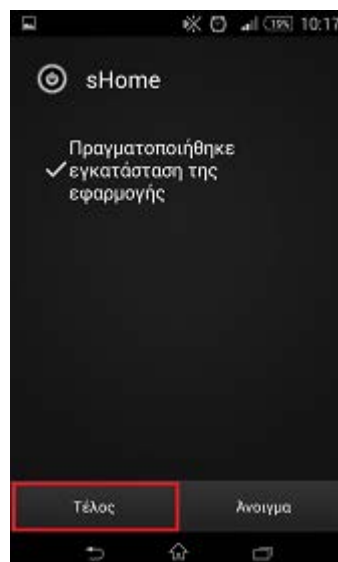
Εικόνα 5.4: Φάκελος Αποθήκευσης Εφαρμογών

3. Πατήστε πάνω του και θα εμφανιστεί η παρακάτω οθόνη και πατήστε «**Επόμενο**» και στην συνέχεια «**Εγκατάσταση**»:



Εικόνα 5.5: Έγκριση Εγκατάστασης (1) Εικόνα 5.6: Έγκριση Εγκατάστασης (2) Εικόνα 5.7: Εγκατάσταση Εφαρμογής

4. Μόλις ολοκληρωθεί η εγκατάσταση θα εμφανιστεί η παρακάτω εικόνα και πατήστε «**Τέλος**».



Εικόνα 5.8: Ολοκλήρωση Εγκατάστασης

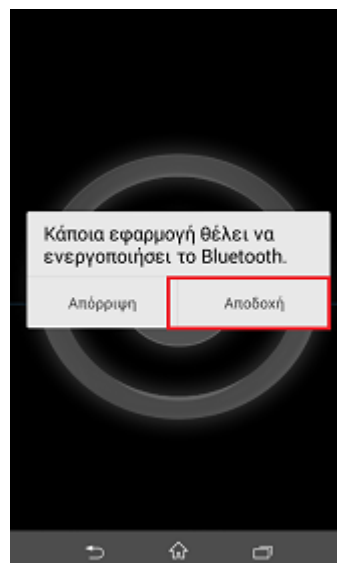
## 5.2 Είσοδος στην εφαρμογή

1. Για να ανοίξετε την εφαρμογή μπείτε στο «**Μενού**» και βρείτε το εικονίδιο της εφαρμογής.



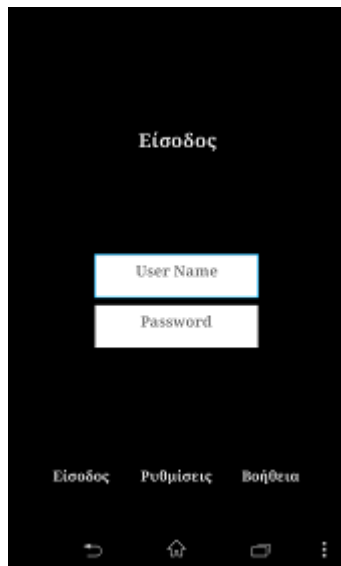
Εικόνα 5.9: Μενού Συσκευής Android

2. Πατήστε στο «**sHome**» για να ανοίξει η εφαρμογή. Θα εμφανιστεί ένα μήνυμα που σας ζητάει την άδεια να ενεργοποιήσετε το Bluetooth πατήστε «**Αποδοχή**» για να εμφανιστεί το interface της εισόδου:

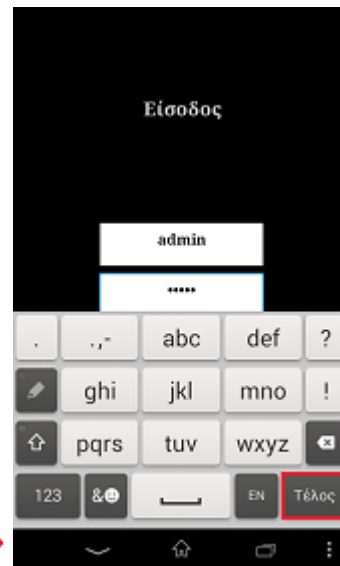


Εικόνα 5.10: Ενεργοποίηση Bluetooth κατά την έναρξη της εφαρμογής

3. Τοποθετήστε το Όνομα χρήστη και τον κωδικό πρόσβασης (τα default είναι τα username: admin και password: admin):



Εικόνα 5.11: Είσοδος στην εφαρμογή



Εικόνα 5.12: Εισαγωγή Username και Password

4. Πατήστε «Τέλος» μετά την εισαγωγή των στοιχείων και στην συνέχεια «Είσοδο»:



Εικόνα 5.13: Είσοδος στην εφαρμογή

Και θα εμφανιστεί η κεντρική οθόνη της εφαρμογής:



Εικόνα 5.14 και 5.15: Κυρίως οθόνη-Πίνακας Ελέγχου

### 5.3 Ρύθμιση Σαλονιού

1. ΣΤΟΝ «Πίνακα Ελέγχου» ΕΠΙΛΕΞΤΕ ΤΟ «Σαλόνι»



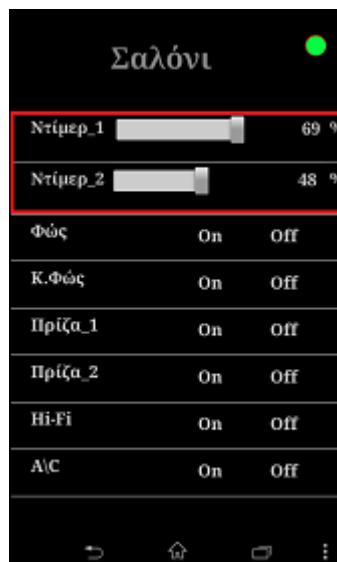
Εικόνα 5.16: Επιλογή Σαλονιού

2. Στο interface του «**Σαλονιού**» θα δείτε τις παρακάτω επιλογές:



Εικόνα 5.17: Μενού του Σαλονιού

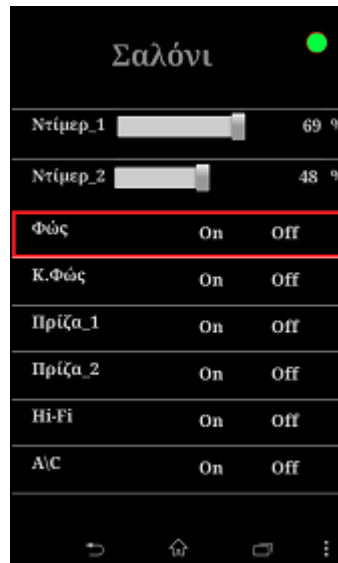
- Στις επιλογές «Ντίμερ\_1» και «Ντίμερ\_2» μπορείτε να ρυθμίσετε την ένταση στα 2 από τα φώτα του σαλονιού.



Εικόνα 5.18: Χρήση Dimmer

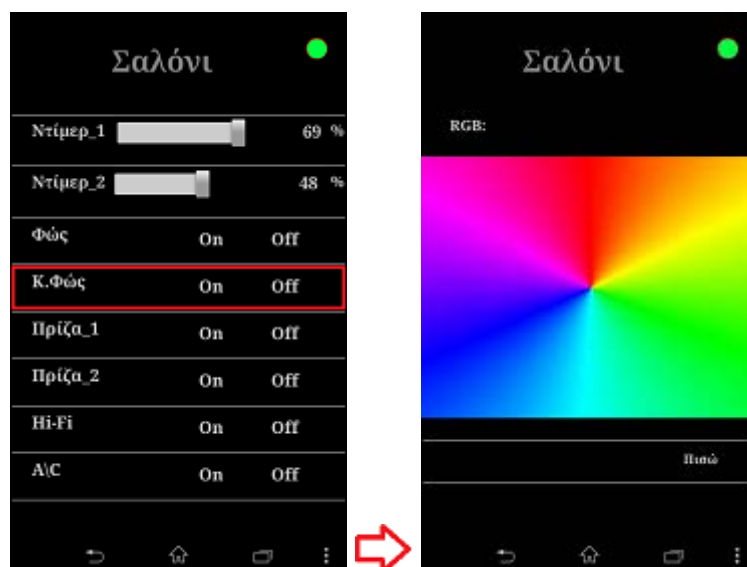


- Στην επιλογή «Φως» μπορείτε να ανάψετε το φως του σαλονιού πατώντας «on» ή να το απενεργοποιήσουμε πατώντας το «off».



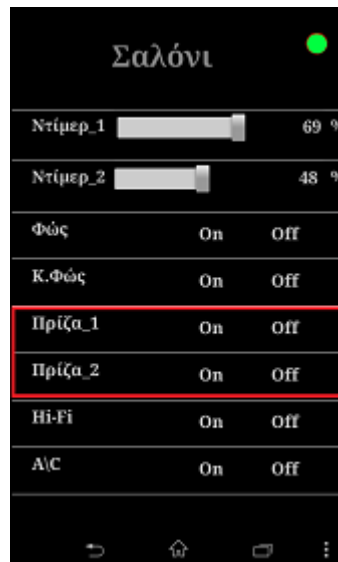
Εικόνα 5.19: On/Off Λάμπας

- Πατώντας «Κ.Φως» (Κύριο Φως) ανοίγει το παρακάτω interface όπου σας επιτρέπει να διαλέξετε την απόχρωση του φωτισμού στο σαλόνι σας. Καθώς επίσης και την ενεργοποίηση και την απενεργοποίηση αυτού του φωτός με τα «On» και «Off»



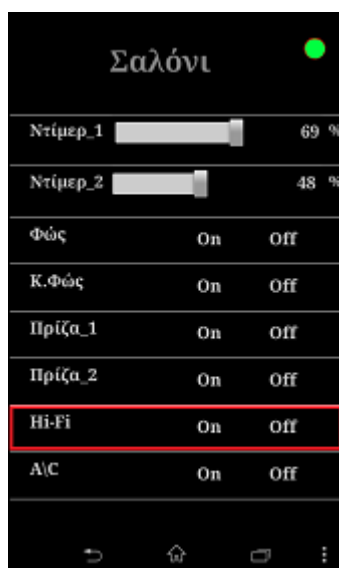
Εικόνα 5.20: On/Off Κρυφού φωτισμού Εικόνα 5.21: Επιλογή RGB Κρυφού φωτισμού

- Στις επιλογές «Πρίζα\_1» και «Πρίζα\_2» μπορείτε να ενεργοποιήσετε και να απενεργοποιήσετε τις συσκευές που είναι στις αντίστοιχες πρίζες πατώντας «On» ή «Off».

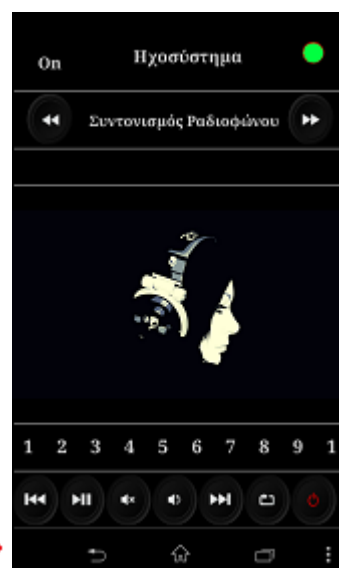


Εικόνα 5.22: On/Off Πριζών Σαλονιού

- Στην επιλογή «Hi-Fi» πατώντας «On» ή «Off» μπορείτε να ενεργοποιήσουμε και να απενεργοποιήσουμε την συσκευή του Hi-Fi. Πατώντας στο Hi-Fi να δίνει την δυνατότητα να επιλέξουμε ένα από τους 10 σταθμούς που είναι αποθηκευμένοι. \*Σημείωση: Το ραδιόφωνο θα ανοίγει στον σταθμό που ήταν την τελευταία φορά που χρησιμοποιήθηκε.

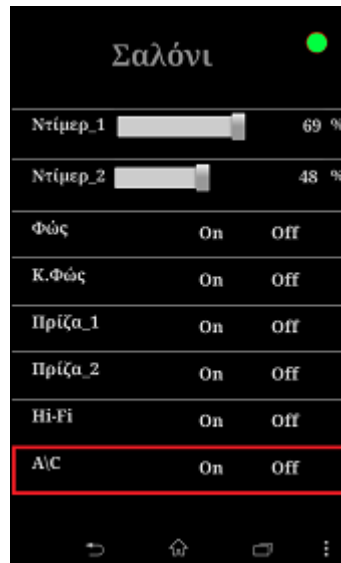


Εικόνα 5.23: On/Off Hi-Fi



Εικόνα 5.24: Διαχείριση Hi-Fi

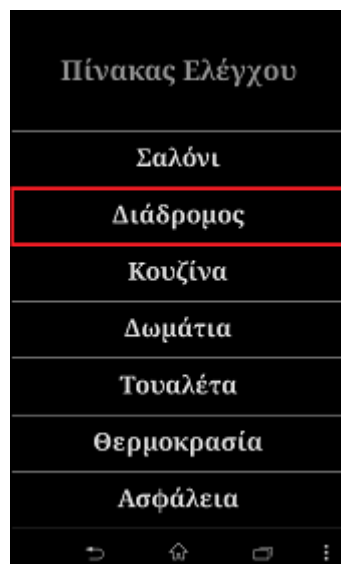
- Τέλος στο interface του σαλονιού μπορείτε να ανοίξετε και να κλείσετε το «aircondition» με τις επιλογές «On» και «Off»



Εικόνα 5.25: On/Off Air condition

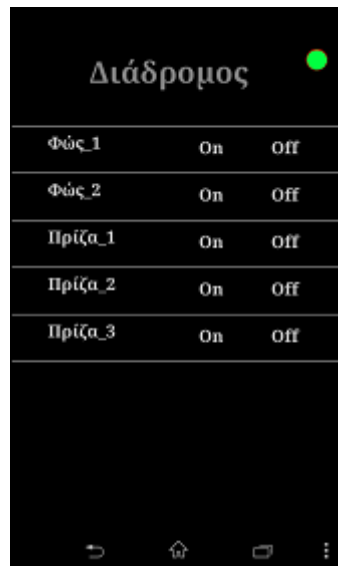
## 5.4 Ρύθμιση Διαδρόμου

1. ΣΤΟΝ «Πίνακα Ελέγχου» ΕΠΙΛΕΞΤΕ ΤΟ «Διάδρομος»



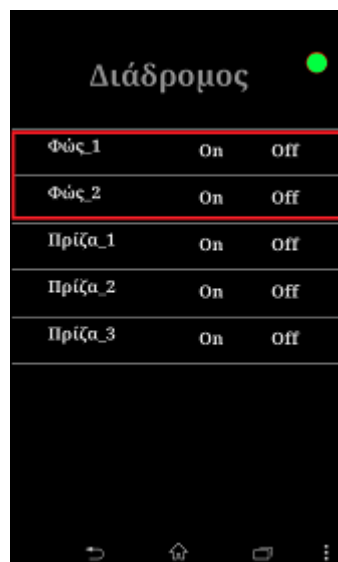
Εικόνα 5.26: Επιλογή διαδρόμου

2. Στο interface του «**Διαδρόμου**» θα δείτε τις παρακάτω επιλογές:



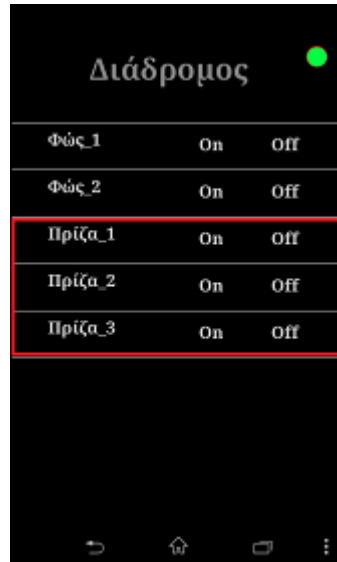
Εικόνα 5.27: Μενού Διαδρόμου

- Στις επιλογές «**Φως\_1**» και «**Φως\_2**» μπορείτε να ανάψετε τα φώτα του διαδρόμου πατώντας «**On**» ή να τα απενεργοποιήσετε πατώντας το «**Off**».



Εικόνα 5.28: On/Off στα φώτα του Διαδρόμου

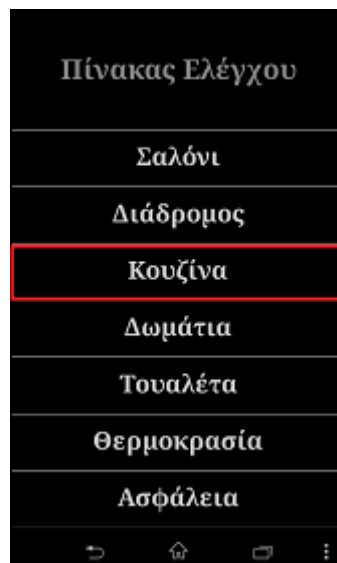
- Στις επιλογές «Πρίζα\_1», «Πρίζα\_2» και «Πρίζα\_3» μπορείτε να ενεργοποιήσετε και να απενεργοποιήσετε τις συσκευές που είναι στις αντίστοιχες πρίζες πατώντας «On» ή «Off» αντίστοιχα.



Εικόνα 5.29: On/Off Πριζών του Διαδρόμου

## 5.5 Ρύθμιση Κουζίνας

1. Στον «Πίνακα Ελέγχου» επιλέξτε το «Κουζίνα»



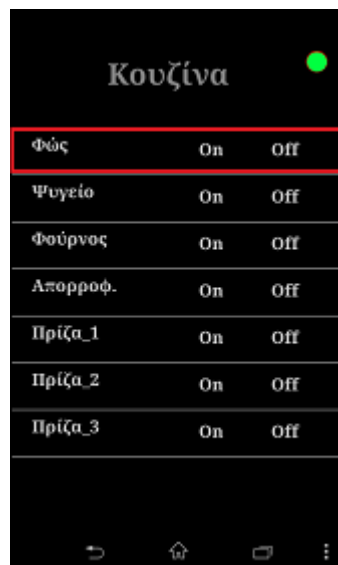
Εικόνα 5.30: Επιλογή Κουζίνας

2. Στο interface της «Κουζίνας» θα δείτε τις παρακάτω επιλογές:



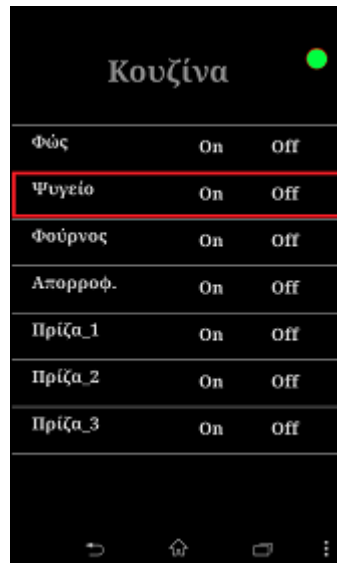
Εικόνα 5.31: Μενού Κουζίνας

- Στην επιλογή «Φως» μπορείτε να ανάψετε το φως της κουζίνας πατώντας «On» ή να τα απενεργοποιήσετε πατώντας το «Off».



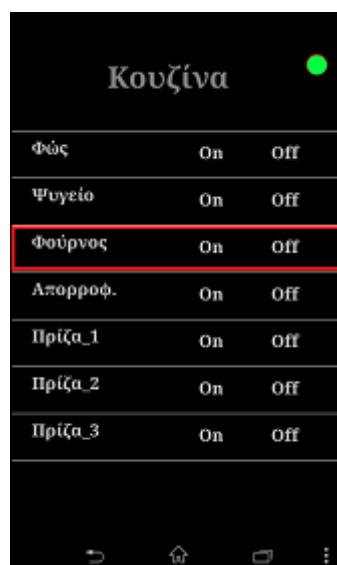
Εικόνα 5.32: On/Off στα φώτα της Κουζίνας

- Στην επιλογή «**Ψυγείο**» μπορείτε να ανοίξετε και να κλείσετε το ψυγείο από το ρεύμα με τα κουμπιά «**On**» και «**Off**» αντίστοιχα.



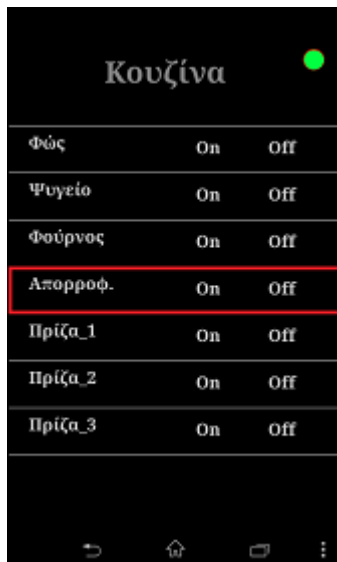
Εικόνα 5.33: On/Off στο ψυγείου της Κουζίνας

- Στην επιλογή «**Φούρνος**» μπορείτε με πολύ απλό τρόπο να ενεργοποιήσετε τον φούρνο σας πατώντας τα πλήκτρα «**On**» και «**Off**». \*Σημείωση: Όταν ανοίγουμε τον φούρνο στην θερμοκρασία και τις ρυθμίσεις ψησίματος όπου έχουμε αφήσει τον φούρνο.



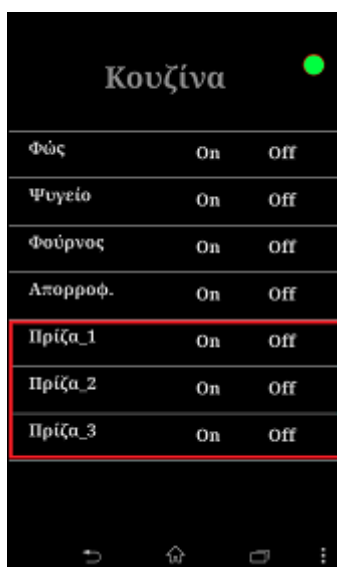
Εικόνα 5.34: On/Off Φούρνου της Κουζίνας

- Στην επιλογή «**Απορροφ.**» (Απορροφητήρας) πατώντας «**On**» ή «**Off**» μπορείτε να ενεργοποιήσετε και να απενεργοποιήσετε τον απορροφητήρα σας.



Εικόνα 5.35: On/Off Απορροφητήρα της Κουζίνας

- Στις επιλογές «**Πρίζα\_1**», «**Πρίζα\_2**» και «**Πρίζα\_3**» μπορείτε να ενεργοποιήσετε και να απενεργοποιήσετε τις συσκευές που είναι στις αντίστοιχες πρίζες πατώντας «**On**» ή «**Off**» αντίστοιχα.

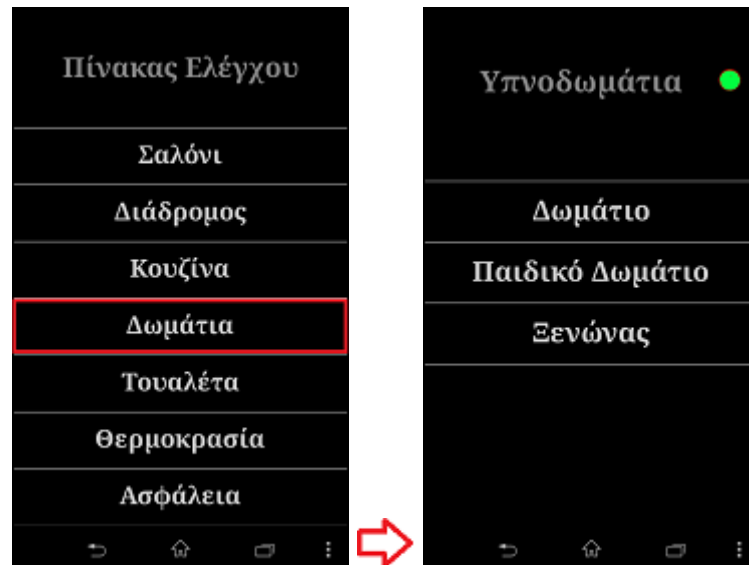


Εικόνα 5.36: On/Off Απορροφητήρα της Κουζίνας



## 5.6 Ρύθμιση Δωματίων

1. ΣΤΟΝ «Πίνακα Ελέγχου» ΕΠΙΛΕΞΤΕ ΤΟ «Δωμάτια»



Εικόνα 5.37: Επιλογή Δωματίων

Εικόνα 5.38: Επιλογή Υπνοδωματίου

2. Επιλέξτε ποιο Υπνοδωμάτιο θέλετε να διαχειριστείτε:

✓ Δωμάτιο

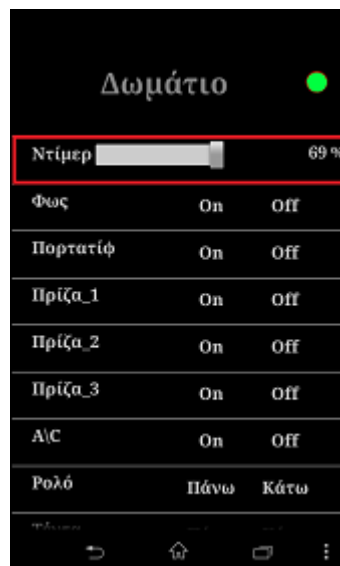


Εικόνα 5.39: Επιλογή Υπνοδωματίου

Εικόνα 5.40: Μενού Δωματίου (1)

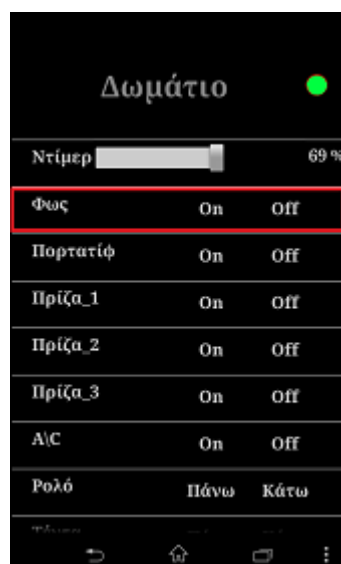
Εικόνα 5.41: Μενού Δωματίου (2)

- Στην επιλογή «**Ντίμερ**» μπορούμε να ρυθμίσουμε την ένταση στο φως του δωματίου.



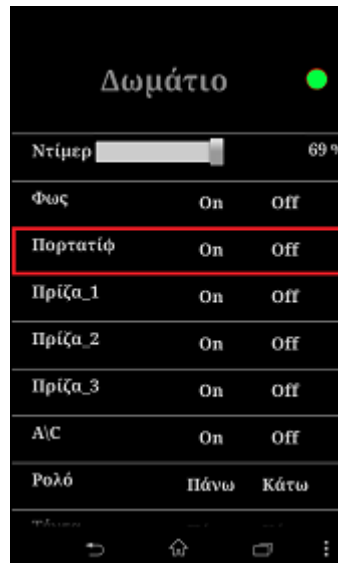
Εικόνα 5.42: Dimmer Δωματίου

- Στην επιλογή «**Φως**» μπορούμε να ενεργοποιήσουμε το φως πατώντας «**On**» και «**Off**».



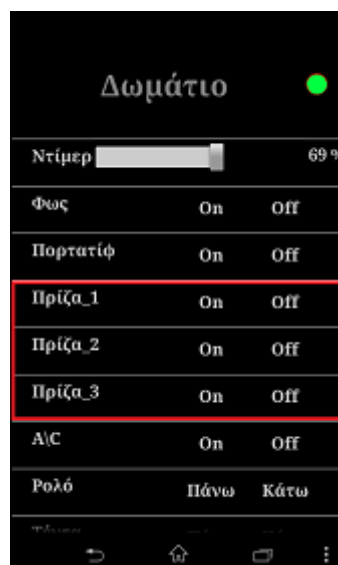
Εικόνα 5.43: On/Off στο Φως του Δωματίου

- Στην επιλογή **«Πορτατίφ»** μπορείτε να ενεργοποιήσετε και να απενεργοποιήσετε το πορτατίφ του δωματίου.



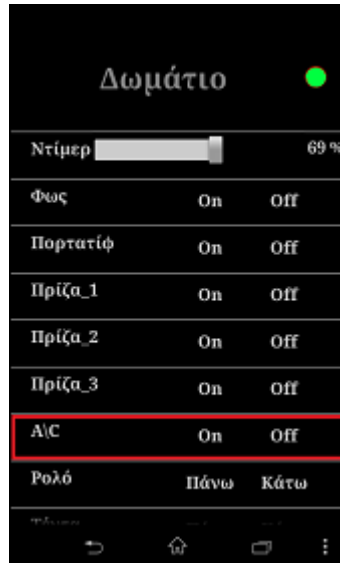
Εικόνα 5.44: On/Off Πορτατίφ του Δωματίου

- Στις επιλογές **«Πρίζα\_1»**, **«Πρίζα\_2»** και **«Πρίζα\_3»** μπορείτε να ενεργοποιήσετε και να απενεργοποιήσετε τις συσκευές που είναι στις αντίστοιχες πρίζες πατώντας **«On»** ή **«Off»** αντίστοιχα.



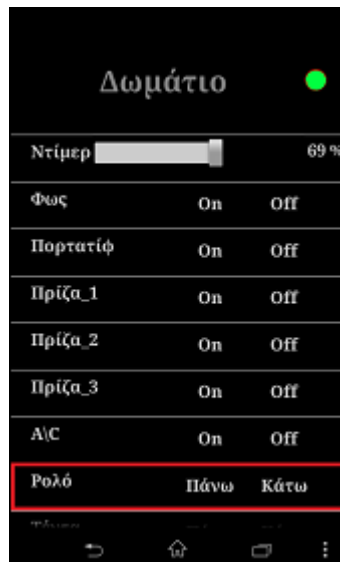
Εικόνα 5.45: On/Off στις Πρίζες του Δωματίου

- Στην επιλογή «**A/C**» ενεργοποιείτε και απενεργοποιείτε χειροκίνητα το aircondition του δωματίου.



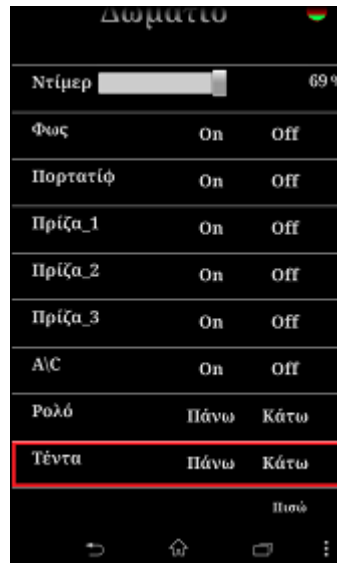
Εικόνα 5.46: On/Off στο Air του Δωματίου

- Στην επιλογή «**Ρολό**» μπορείτε να ανεβοκατεβάσετε τα ρολά από τις μπαλκονόπορτες.



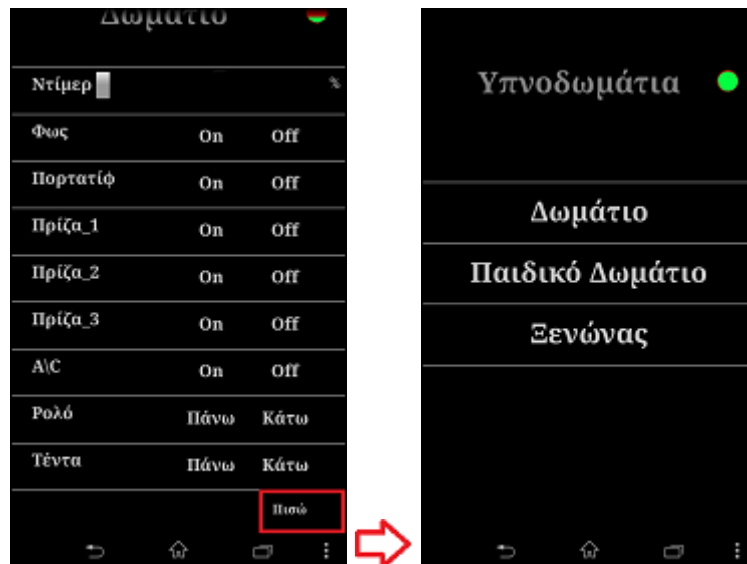
Εικόνα 5.47: Διαχείριση Ρολό του Δωματίου

- Στην επιλογή «**Τέντα**» μπορείτε να ανεβοκατεβάσετε την τέντα από το μπαλκόνι του υπνοδωματίου.



Εικόνα 5.48: Διαχείριση Τέντας του Δωματίου

- Πατήστε «**Πίσω**» για να επιστρέψετε στο μενού «**Υπνοδωμάτια**»



Εικόνα 5.49: Πίσω στο Μενού Υπνοδωματίων

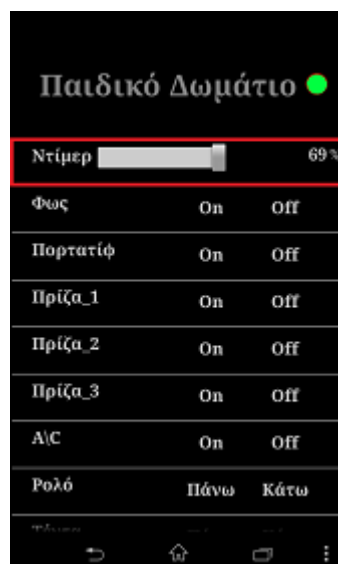
Εικόνα 5.50: Επιλογή Υπνοδωματίου

✓ Παιδικό Δωμάτιο



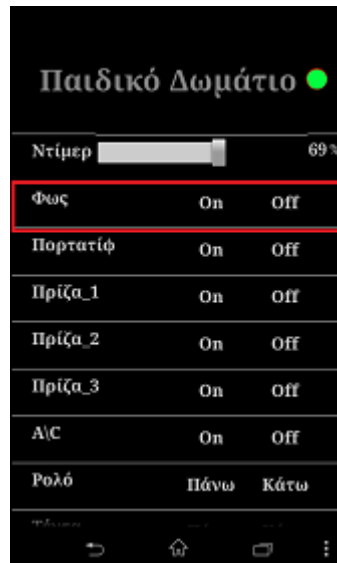
Εικόνα 5.51: Επιλογή Υπνοδωματίου Εικόνα 5.52/Εικόνα 5.53: Μενού Παιδικού Δωματίου (1) και (2)

- Στην επιλογή «**Ντίμερ**» μπορείτε να ρυθμίσετε την ένταση στο φως του παιδικού δωματίου.



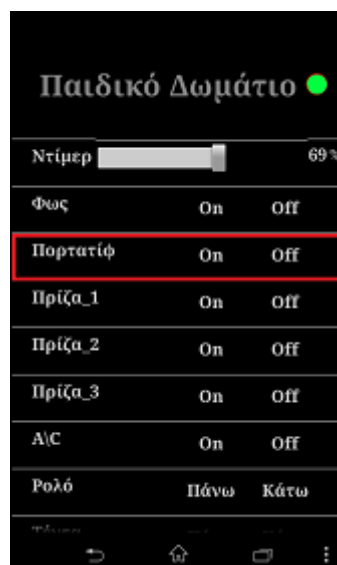
Εικόνα 5.54: Dimmer Παιδικού Δωματίου

- Στην επιλογή «**Φως**» μπορούμε να ενεργοποιήσουμε το φως πατώντας «**On**» και «**Off**».



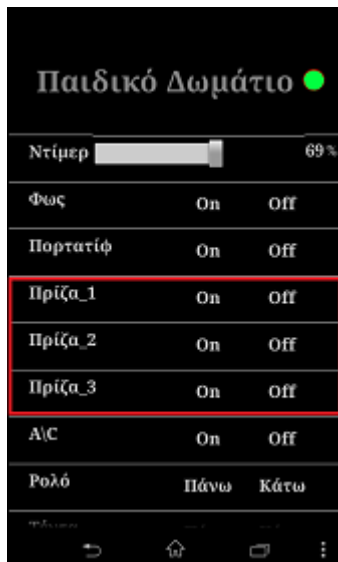
Εικόνα 5.55: On/Off στο Φως του Παιδικού Δωματίου

- Στην επιλογή «**Πορτατίφ**» μπορείτε να ενεργοποιήσετε και να απενεργοποιήσετε το πορτατίφ του παιδικού δωματίου με τα κουμπιά «**On**» και «**Off**».



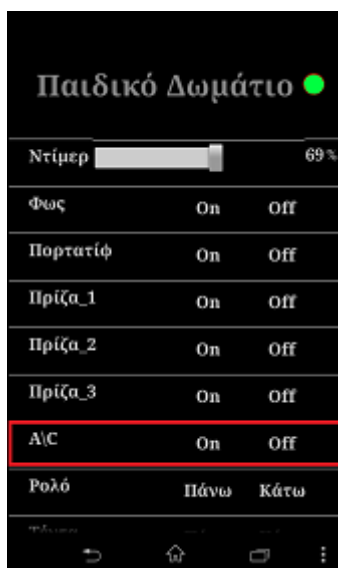
Εικόνα 5.56: On/Off στο Πορτατίφ του Παιδικού Δωματίου

- Στις επιλογές «Πρίζα\_1», «Πρίζα\_2» και «Πρίζα\_3» μπορείτε να ενεργοποιήσετε και να απενεργοποιήσετε τις συσκευές που είναι στις αντίστοιχες πρίζες πατώντας «On» ή «Off» αντίστοιχα.



Εικόνα 5.57: On/Off στις Πρίζες του Παιδικού Δωματίου

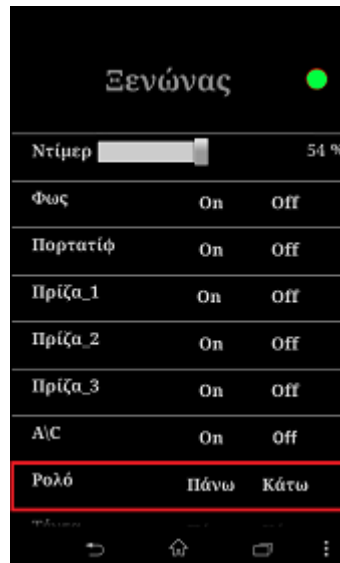
- Στην επιλογή «A\C» ενεργοποιούμε και απενεργοποιούμε χειροκίνητα το aircondition του παιδικού δωματίου πατώντας «On» ή «Off» αντίστοιχα.



Εικόνα 5.58: On/Off στο condition του Παιδικού Δωματίου

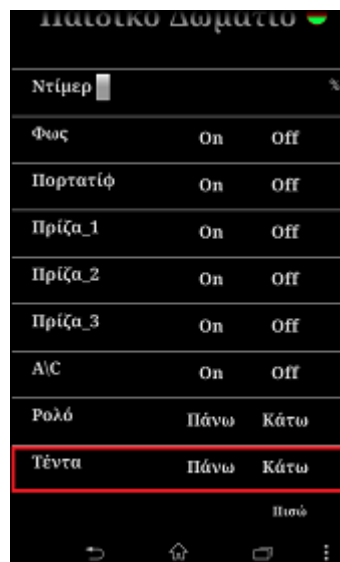


- Στην επιλογή **«Ρολό»** μπορούμε να ανεβοκατεβάσουμε τα ρολά από τις μπαλκονόπορτες πατώντας **«On»** ή **«Off»** αντίστοιχα.



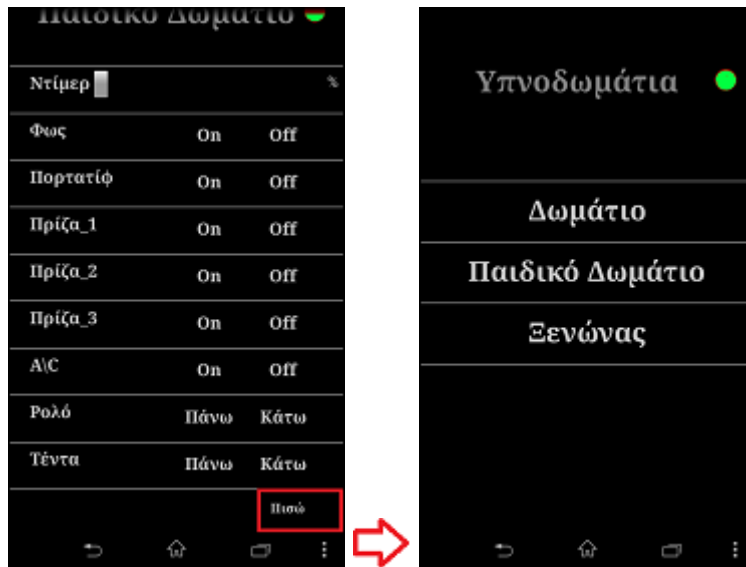
Εικόνα 5.59: Διαχείριση Ρολό του Παιδικού Δωματίου

- Στην επιλογή **«Τέντα»** μπορείτε να ανεβοκατεβάσετε τις τέντες του παιδικού δωματίου πατώντας **«On»** ή **«Off»** αντίστοιχα.



Εικόνα 5.60: Διαχείριση Τέντας του Παιδικού Δωματίου

- Πατήστε «Πίσω» για α επιστρέψετε στο μενού «Υπνοδωμάτια»



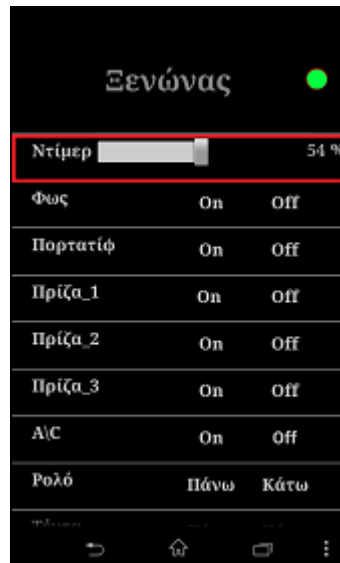
Εικόνα 5.61: Πίσω στο Μενού Παιδικού Δωματίου Εικόνα 5.62: Επιλογή Υπνοδωματίου

✓ Ξενώνας



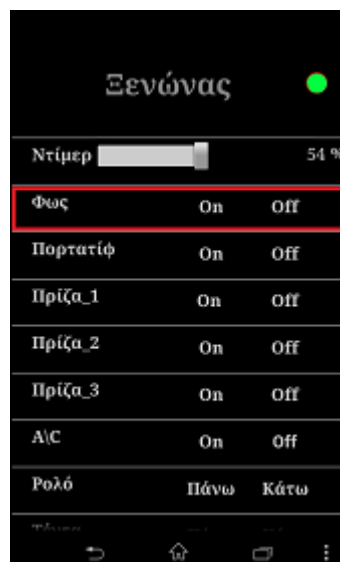
Εικόνα 5.63: Επιλογή Υπνοδωματίου Εικόνα 5.64: Μενού Ξενώνα (1) Εικόνα 5.65: Μενού Ξενώνα (2)

- Στην επιλογή «**Ντίμερ**» μπορούμε να ρυθμίσουμε την ένταση στο φως του ξενώνα.



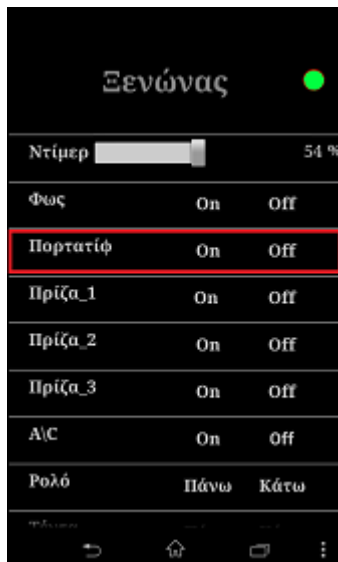
Εικόνα 5.66: Dimmer Ξενώνα

- Στην επιλογή «**Φως**» μπορούμε να ενεργοποιήσουμε το φως πατώντας «On» και «Off».



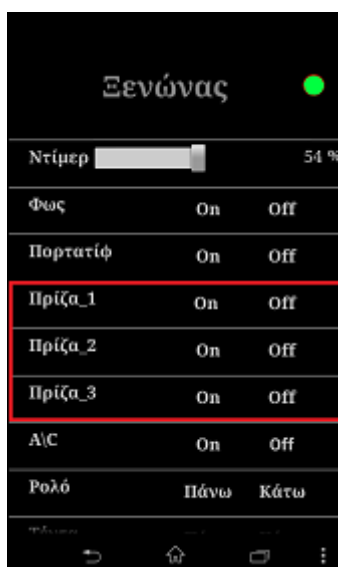
Εικόνα 5.67: On/Off στο Φως του Ξενώνα

- Στην επιλογή **«Πορτατίφ»** μπορούμε να ενεργοποιήσουμε και να απενεργοποιήσουμε και να απενεργοποιήσουμε το πορτατίφ του ξενώνα, πατώντας **«On»** ή **«Off»** αντίστοιχα.



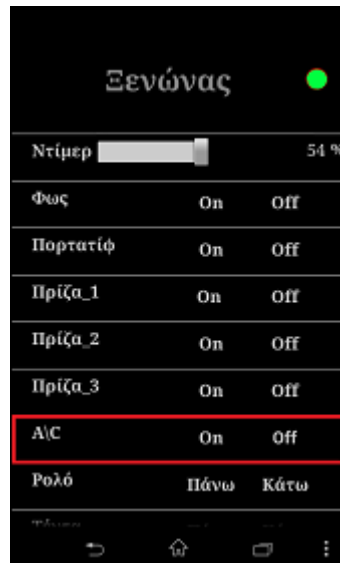
Εικόνα 5.68: On/Off στο Πορτατίφ του Ξενώνα

- Στις επιλογές **«Πρίζα\_1»**, **«Πρίζα\_2»** και **«Πρίζα\_3»** μπορείτε να ενεργοποιήσετε και να απενεργοποιήσετε τις συσκευές που είναι στις αντίστοιχες πρίζες πατώντας **«On»** ή **«Off»** αντίστοιχα.



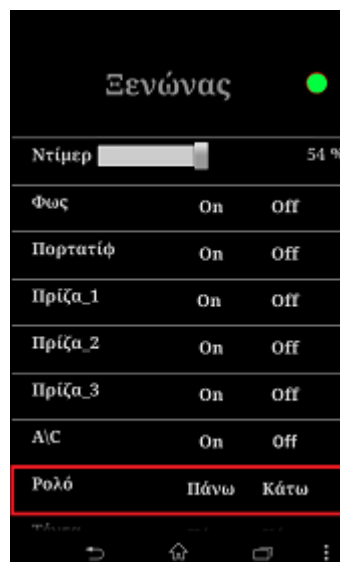
Εικόνα 5.69: On/Off στις του Ξενώνα

- Στην επιλογή «Α\C» ενεργοποιούμε και απενεργοποιούμε χειροκίνητα το aircondition του ξενώνα.



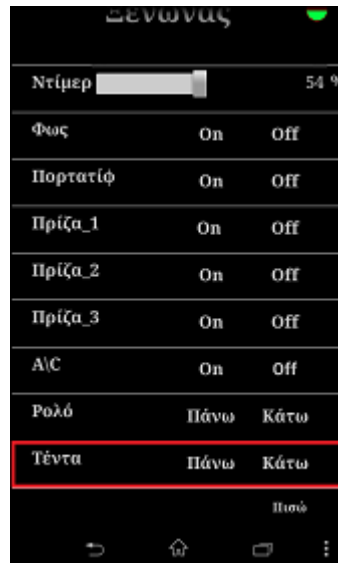
Εικόνα 5.70: On/Off στο Φως του Ξενώνα

- Στην επιλογή «**Ρολό**» μπορούμε να ανεβοκατεβάσουμε τα ρολά από τις μπαλκονόπορτες πατώντας «Πάνω» ή «Κάτω» αντίστοιχα.



Εικόνα 5.71: Διαχείριση Ρολό του Ξενώνα

- Στην επιλογή «**Τέντα**» μπορείτε να ανεβοκατεβάσετε τις τέντες του παιδικού δωματίου πατώντας «**On**» ή «**Off**» αντίστοιχα.

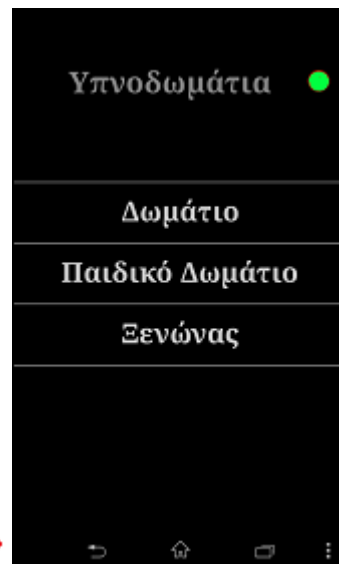


Εικόνα 5.72: Διαχείριση Τέντας του Ξενώνα

- Πατήστε «**Πίσω**» για α επιστρέψετε στο μενού «**Υπνοδωμάτια**»



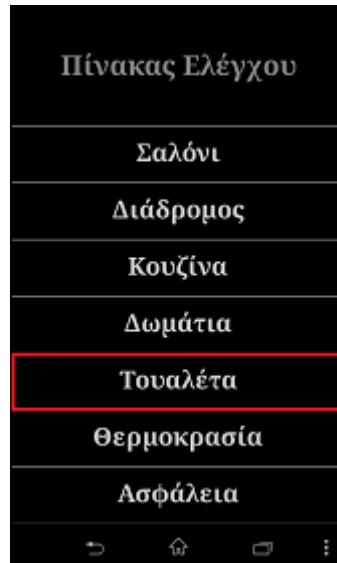
Εικόνα 5.73: Πίσω στο Μενού Ξενώνα



Εικόνα 5.74: Επιλογή Υπνοδωματίου

## 5.7 Ρύθμιση Τουαλέτας

1. ΣΤΟΝ «Πίνακα Ελέγχου» ΕΠΙΛΕΞΤΕ ΤΟ «Τουαλέτα»



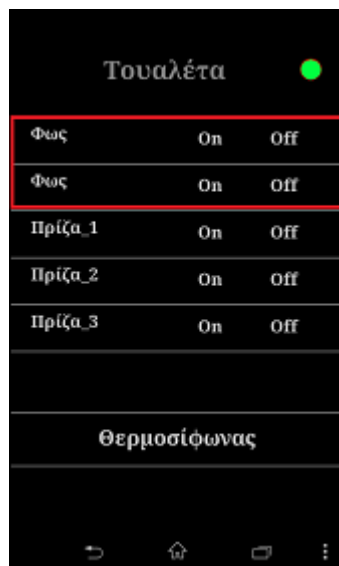
Εικόνα 5.75: Επιλογή Τουαλέτας

2. Στο interface του «Τουαλέτα» θα δείτε τις παρακάτω επιλογές:



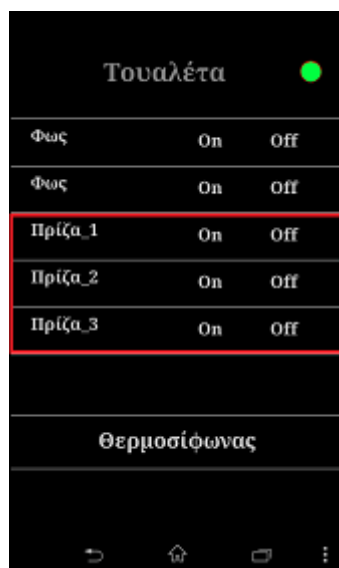
Εικόνα 5.76: Μενού Τουαλέτας

- Στις επιλογές «**Φως**» μπορείτε να ανάψετε τα φώτα της τουαλέτας πατώντας «**On**» ή να τα απενεργοποιήσετε πατώντας το «**Off**».



Εικόνα 5.77: On/Off στο Φως της Τουαλέτας

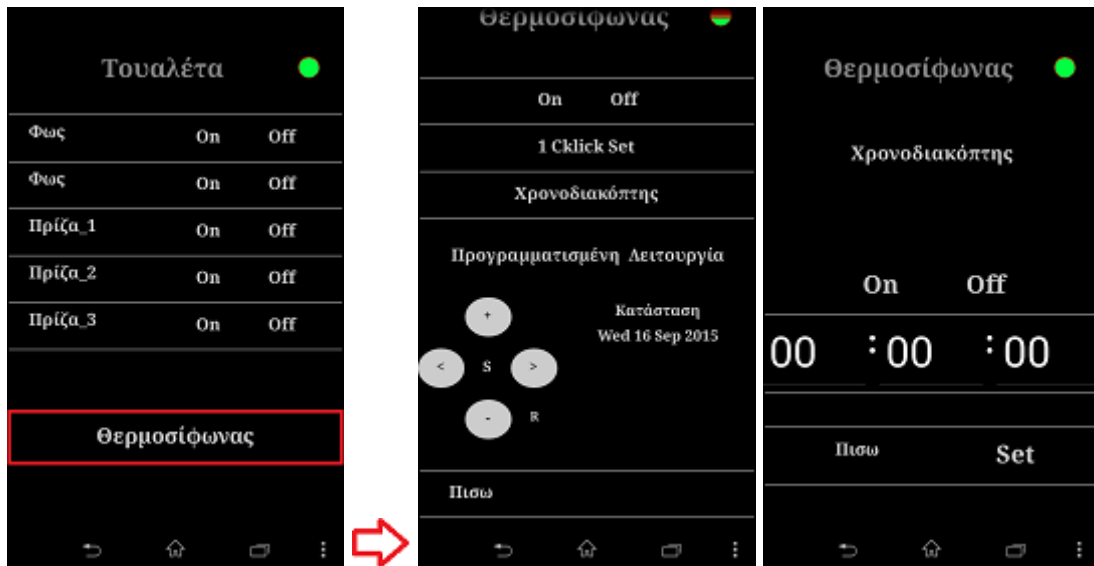
- Στις επιλογές «**Πρίζα\_1**», «**Πρίζα\_2**» και «**Πρίζα\_3**» μπορείτε να ενεργοποιήσετε και να απενεργοποιήσετε τις συσκευές που είναι στις αντίστοιχες πρίζες πατώντας «**On**» ή «**Off**» αντίστοιχα.



Εικόνα 5.78: On/Off στις Πρίζες της Τουαλέτας



- ο Στην επιλογή «Θερμοσίφωνα» μπορείτε να διαχειριστείτε τον θερμοσίφωνα



Εικόνα 5.79: Επιλογή Θερμοσίφωνα

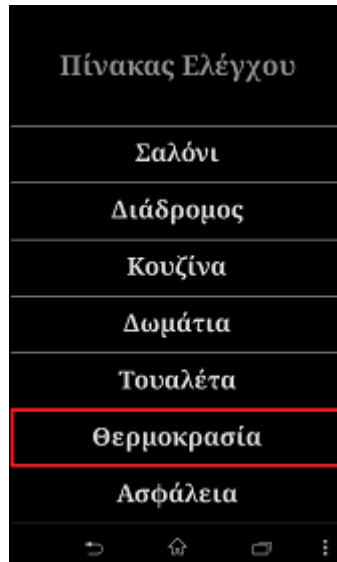
Εικόνα 5.80: Μενού Θερμοσίφωνα

Εικόνα 5.81: Ρύθμιση ώρας θερμοσίφωνα

- Στα «**On**» και «**Off**» που βρίσκονται πάνω μπορείτε να ενεργοποιήσετε και να απενεργοποιήσετε χειροκίνητα τον θερμοσίφωνα. Επίσης μπορείτε να ρυθμίσετε πόση ώρα θέλετε να παραμείνει ανοιχτός ο θερμοσίφωνα. Στην επιλογή «Προγραμματισμένη Λειτουργία» μπορείτε να ρυθμίσετε τη επιθυμητή ώρα, ημέρα και ημερομηνία για το πότε θέλετε να ανοίξει ο θερμοσίφωνα.

## 5.8 Ρύθμιση Θερμοκρασίας

1. ΣΤΟΝ «Πίνακα Ελέγχου» ΕΠΙΛΕΞΤΕ ΤΟ «Θερμοκρασία»



Εικόνα 5.82: Επιλογή Θερμοκρασίας

2. Στο interface της «Θερμοκρασίας» μπορείτε να πάρετε μετρήσεις για την παρούσα θερμοκρασία και υγρασία του χώρου, επίσης να θέσουμε τις επιθυμητές θερμοκρασίες ψύξης και θέρμανσης.



Εικόνα 5.83: Interface Θερμοκρασίας

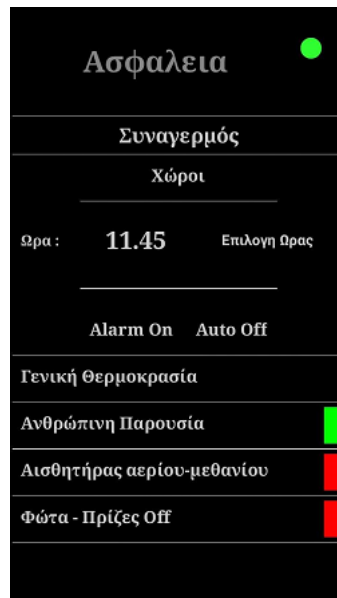
## 5.9 Ρύθμιση Ασφάλειας

1. ΣΤΟΝ «Πίνακα Ελέγχου» ΕΠΙΛΕΞΤΕ ΤΟ «Ασφάλεια»



Εικόνα 5.84: Επιλογή Ασφάλειας

2. Στο interface της «Ασφάλειας» θα δείτε τις παρακάτω επιλογές:



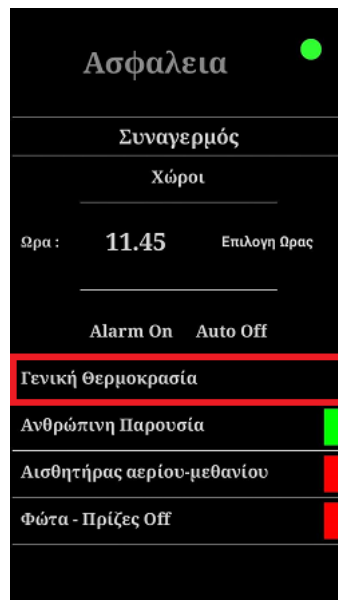
Εικόνα 5.85: Interface Ασφάλειας

- Μπορείτε να διαχειριστείτε τον συναγερμό του χώρου και την ώρα που θέλετε να ενεργοποιηθεί ο συναγερμός.



Εικόνα 5.86: Διαχείριση Συναγερμού

- Μπορείτε να έχετε μία γενική εικόνα για την θερμοκρασία του σπιτιού



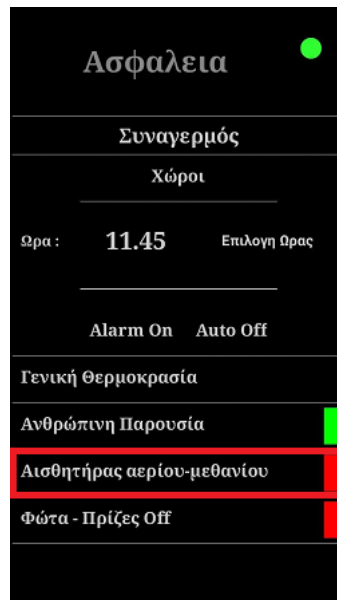
Εικόνα 5.87: Έλεγχος γενικής θερμοκρασίας

- Μπορείτε να ελέγξετε αν υπάρχει άνθρωπος στον χώρο



Εικόνα 5.88: Έλεγχος Ανθρώπινης παρουσίας

- Μπορείτε να ελέγξετε αν υπάρχει κάποια διαρροή αερίου/ μεθανίου



Εικόνα 5.89: Έλεγχος διαρροής αερίου

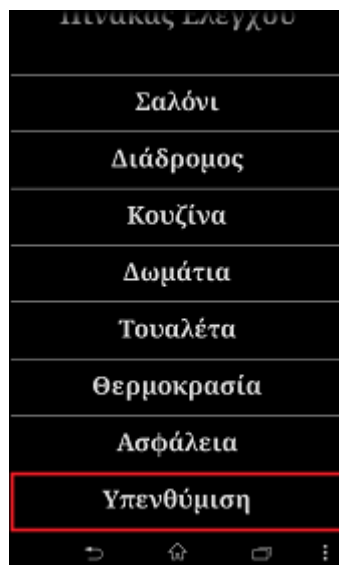
- Τέλος μπορείτε να ελέγξετε αν υπάρχουν ανοιχτά φώτα στους χώρους του σπιτιού



Εικόνα 5.90: Έλεγχος Πριζών του χώρου

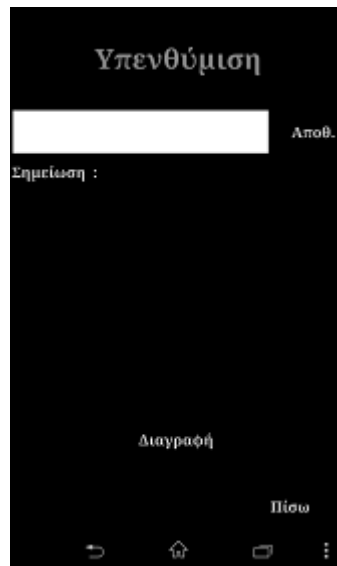
## 5.10 Ρύθμιση Υπενθύμισης

1. Στον «Πίνακα Ελέγχου» επιλέξτε το «Υπενθύμιση»



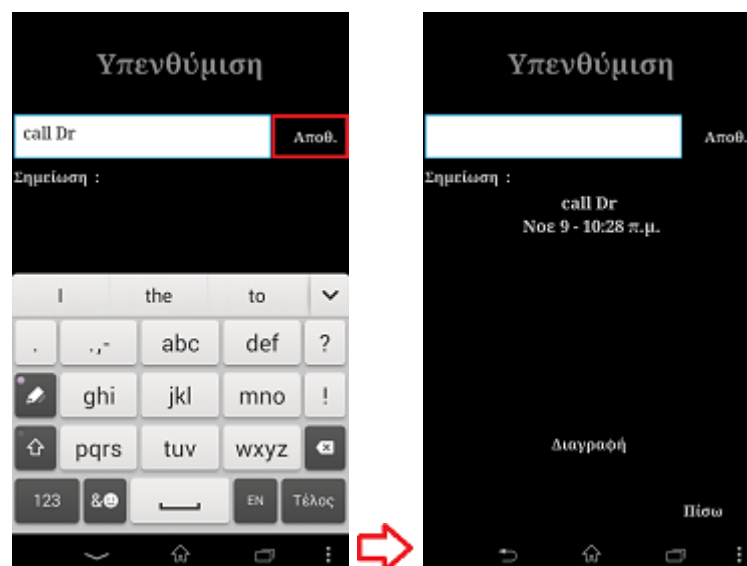
Εικόνα 5.91: Επιλογή Υπενθύμισης

2. Ανοίγει το παρακάτω interface:



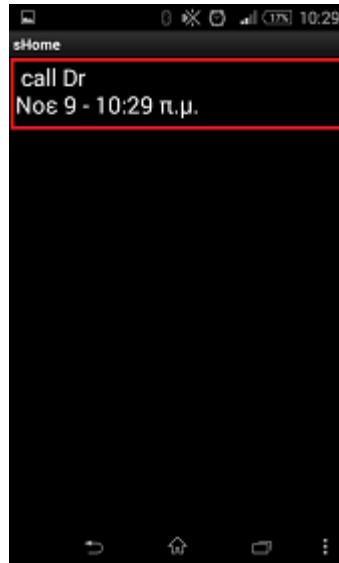
Εικόνα 5.92: Μενού Υπενθύμισης

3. Πατώντας πάνω το πεδίο του κειμένου (text field) εμφανίζεται το πληκτρολόγιο και γράφετε οποιαδήποτε σημείωση/υπενθύμιση επιθυμείτε. Πατήστε αποθήκευση για να καταχωρηθεί η υπενθύμιση.



Εικόνα 5.93: Πληκτρολόγηση Υπενθυμίσεις Εικόνα 5.94: Καταχωρημένη Υπενθύμισης

4. Πατώντας «Διαγραφή» εμφανίζεται η λίστα για να επιλέξετε ποιά από τις υπενθυμίσεις επιθυμείτε να διαγράψετε.



Εικόνα 5.85: Διαγραφή Υπενθυμίσεις



**ΠΑΡΑΡΤΗΜΑ Α: ΚΩΔΙΚΑΣ ARDUINO**

```

#include <Wire.h> // I2C βιβλιοθήκη
#include "RTCLib.h" // RTC βιβλιοθήκη
RTC_DS1307 RTC; // RTC Module
#include <OneWire.h> // OneWire βιβλιοθήκη
#include <DallasTemperature.h> // DS18B20 library
#include "SoftwareSerial.h"
#include <SPI.h>
#include <TEA5767.h>
#include <Button.h>
#include "Adafruit_MCP23017.h"

//*****DS18b20*****
#define DS18B20_PIN 3 // αρχικοποίηση pin DS18B20
OneWire oneWire(DS18B20_PIN); // OneWire αναφορά
DallasTemperature sensors(&oneWire); // αρχικοποίηση DS18B20
//*****Ldr***
int sensorPin = A4; // επιλογή εισόδου για ο LDR
int ledStatus= 22;
//*****Sound_Sensor*****_
int SoundSensorPin = A4; //αναλογικό pin 4 μιροφώνου
int YellowLED = 23;
int RedLED = 24;
int SoundSensorValue = 0;
//*****mcp*****
char junk;
String inputString="";
char rx_byte = 0;
Adafruit_MCP23017 mcp;
//*****Radio_Mp3*****
TEA5767 Radio; //Pinout SLC and SDA - Arduino pins A5 and A4
//=====Setup_Loop=====
void setup(void) {
  Wire.begin(); //αρχικοποίηση I2C
  RTC.begin(); // αρχικοποίηση RTC
  Serial.begin(38400);
  Serial.println("Date - Time and Temperature ");
  if (! RTC.isrunning()) { // έλεγχος αν το RTC τρέχει
    RTC.adjust(DateTime(__DATE__, __TIME__)); // Set current date and
time, if the clock is not running
    Serial.println("RTC has been started and is set to system time.");
  }
  else Serial.println("RTC is already running");
  sensors.begin(); // έναρξη DS18B20
  //-----Radio-Mp3-Analog Switch-----
  Radio.init();
  Radio.set_frequency(95.2); //συntonισμός στον σταθμό 95.2
  pinMode(12, OUTPUT);
  pinMode(12, HIGH);
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT); //pinMode(13, OUTPUT);
  //-----mcp230-----

```

## Έξυπνο Σπίτι - Smarthome

```

    mcp.begin();          // use default address 0
    mcp_Function_pinMode();
}
//=====Void_Loop=====
void loop(){

    DateTime now=RTC.now();          //τρέχουσα ώρα
    show_time_and_date(now);          //αποστολή ημερομηνίας και ώρας
    sensors.requestTemperatures();    //διάβασμα θερμοκρασίας
    show_temperature(sensors.getTempCByIndex(0)); //εμφάνιση θερμοκρασίας
    delay(1000);                      //αναμονή 1s για το επόμενο γεγονός
if (Serial.available() > 0) {        //αν υπάρχει διαθέσιμος χαρακτήρας
    rx_byte = Serial.read();
    Serial.print(rx_byte);
    delay(50);
    switch (rx_byte) {
        case '0':                    // έλεγχος Radio-Mp3 player
            Mp3_Radio_Function();
            break;
        case '1':                    // έλεγχος θερμοσίφωνα
            Heater_State();
            break;
        case '2':                    // σαλόνι
            Saloni_State_Function();
            break;
        case '3':                    //διάδρομος
            Diadromos_State_Function();
            break;
        case '4':                    // κουζίνα
            Kouzina_State_Function();
            break;
        case '5':                    //δωμάτιο
            Dwmatio_1_State_Function();
            break;
        case '6':                    //δωμάτιο 2
            Dwmatio_2_State_Function();
            break;
        case '7':                    //δωμάτιο 3
            Dwmatio_3_State_Function();
            break;
        case '8':                    //τουαλέτα
            Toualeta_State_Function();
            break;
        case '9':
            Asfaleia_State_Function();
            Thermokrasia_State_Function() ;
            break;
        default:
            // Serial.println("Enter option");
            Serial.println(" ");
            break;

        LdrInternalFunction();        // LDR φως
        SoundSensorFunction();        // αισθητήρας ήχου
    }                                  // τέλος της switch mode
}                                      // τέλος της switch case select
}                                      //τέλος τηςloop
//-----RTC Determine the week -----
String get_day_of_week(uint8_t dow){   // Determine the week
    String dows=" ";

```

```

    switch(dow){
    case 0: dows="Κυριακκη"; break;
    case 1: dows="Δευτερα"; break;
    case 2: dows="Τριτη"; break;
    case 3: dows="Τεταρτη"; break;
    case 4: dows="Πεμπτη"; break;
    case 5: dows="Παρασκευη"; break;
    case 6: dows="Σαββατο"; break;
    }
    return dows;
}
//----- RTC date-time -----
void show_time_and_date(DateTime datetime){ // αποστολή ημερομηνία
    Serial.print(get_day_of_week(datetime.dayOfWeek())); //ημερομηνία
    Serial.print(": ");
    if(datetime.day()<10)Serial.print(0);
    Serial.print(datetime.day(),DEC);
    Serial.print(".");
    if(datetime.month()<10)Serial.print(0);
    Serial.print(datetime.month(),DEC);
    Serial.print(".");
    Serial.println(datetime.year(),DEC);

    Serial.print("Ωρα: ");
    if(datetime.hour()<10)Serial.print(0); // ωρα/λεπιά/δευτερόλεπτα
    Serial.print(datetime.hour(),DEC);
    Serial.print(":");
    if(datetime.minute()<10)Serial.print(0);
    Serial.print(datetime.minute(),DEC);
    Serial.print(":");
    if(datetime.second()<10)Serial.print(0);
    Serial.println(datetime.second(),DEC);
}
//-----DS18B20 temp-----
void show_temperature(float temp){ // εξωτερική θερμοκρασία
    Serial.print("Θερμ/σια: ");
    Serial.print(temp);
    Serial.print(" "); //Here we have a little trick
    // Serial.write(176); //To display the character correctly °
    Serial.println("*C");
}
//-----LdrFunc-----
void LdrInternalFunction(){
    int sensorValue = 0; // μεταβλητή αισθητήρα
    sensorValue = analogRead(sensorPin);
    if (sensorValue < 40){
        digitalWrite(ledStatus,HIGH);
        delay(500);
    } else
        {digitalWrite(ledStatus,LOW);
        }
}
//-----Sound Sensor -----
void SoundSensorFunction(){
    SoundSensorValue = analogRead(SoundSensorPin); //μεταβλητή αισθητήρα
    Serial.println(SoundSensorValue);
    Serial.println(SoundSensorValue); //- with:- if (SoundSensorValue <
509){ Serial.println(SoundSensorValue); }
    digitalWrite(YellowLED, HIGH); //
Yellow is always on.

```

## Έξυπνο Σπίτι - Smarthome

```
    if (SoundSensorValue>50){                                     // The
'silence' sensor value is 509-511
    digitalWrite(RedLED,HIGH);
    delay(15 );                                                // The
red LEDs stay on for 2 seconds
    } else {
    digitalWrite(RedLED,LOW);
    }
}
//-----mcp-----
void mcp_Function_pinMode(){

    mcp.pinMode(0, OUTPUT);
    mcp.pinMode(1, OUTPUT);
    mcp.pinMode(2, OUTPUT);
}
//-----Σαλόνη-----
void Saloni_State_Function() {
if (Serial.available() > 0) { // έλεγχος ύπαρξης χαρακτήρα
    rx_byte = Serial.read();
    switch (rx_byte) {
        case 'A':
            mcp.readOutput(0);
            if (mcp.readOutput(0) == HIGH) {
                mcp.digitalWrite(0,LOW);
                Serial.println("OFF");
            } else {
                mcp.digitalWrite(0, HIGH);
                Serial.println("ON");
            }
            break;

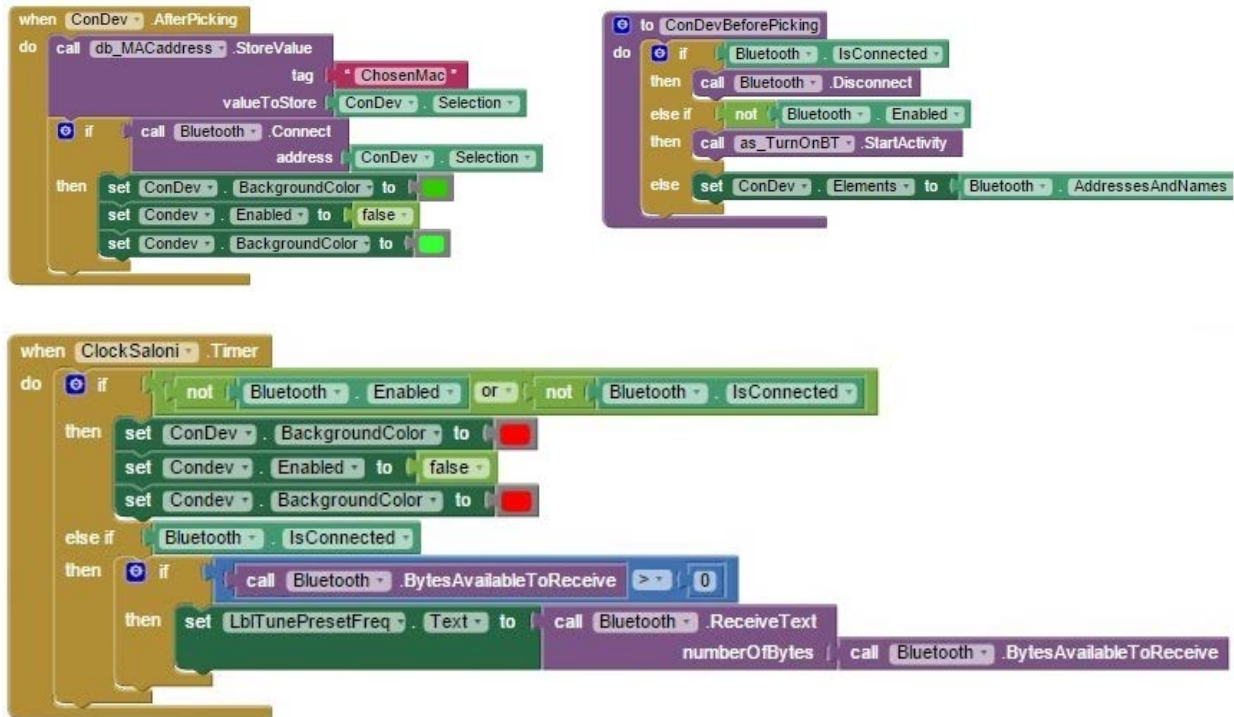
            case 'B':
            mcp.readOutput(1);
            if (mcp.readOutput(1) == HIGH) {
                mcp.digitalWrite(1,LOW);
                Serial.println("OFF");
            } else {
                mcp.digitalWrite(1, HIGH);
                Serial.println("LED is ON");
            }
            break;

            case 'C':
            mcp.readOutput(2);
            if (mcp.readOutput(2) == HIGH) {
                mcp.digitalWrite(2,LOW);
                Serial.println("OFF");
            } else {
                mcp.digitalWrite(2, HIGH);
                Serial.println("ON");
            }
            break;

            default: Serial.println(" "); break;
        } // end: switch (rx_byte)
    } // end: if (Serial.available() > 0)
}
```

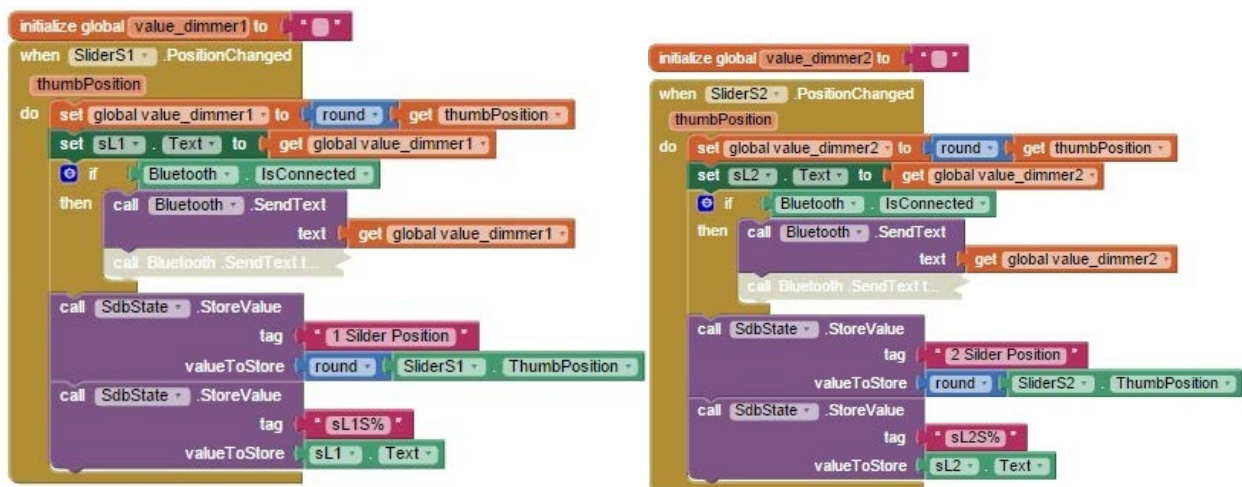
## ΠΑΡΑΡΤΗΜΑ Β: BLOCKS ΤΟΥ ANDROID

### Έλεγχος ενεργοποίησης Bluetooth και ενημέρωση κατάστασης του Bluetooth



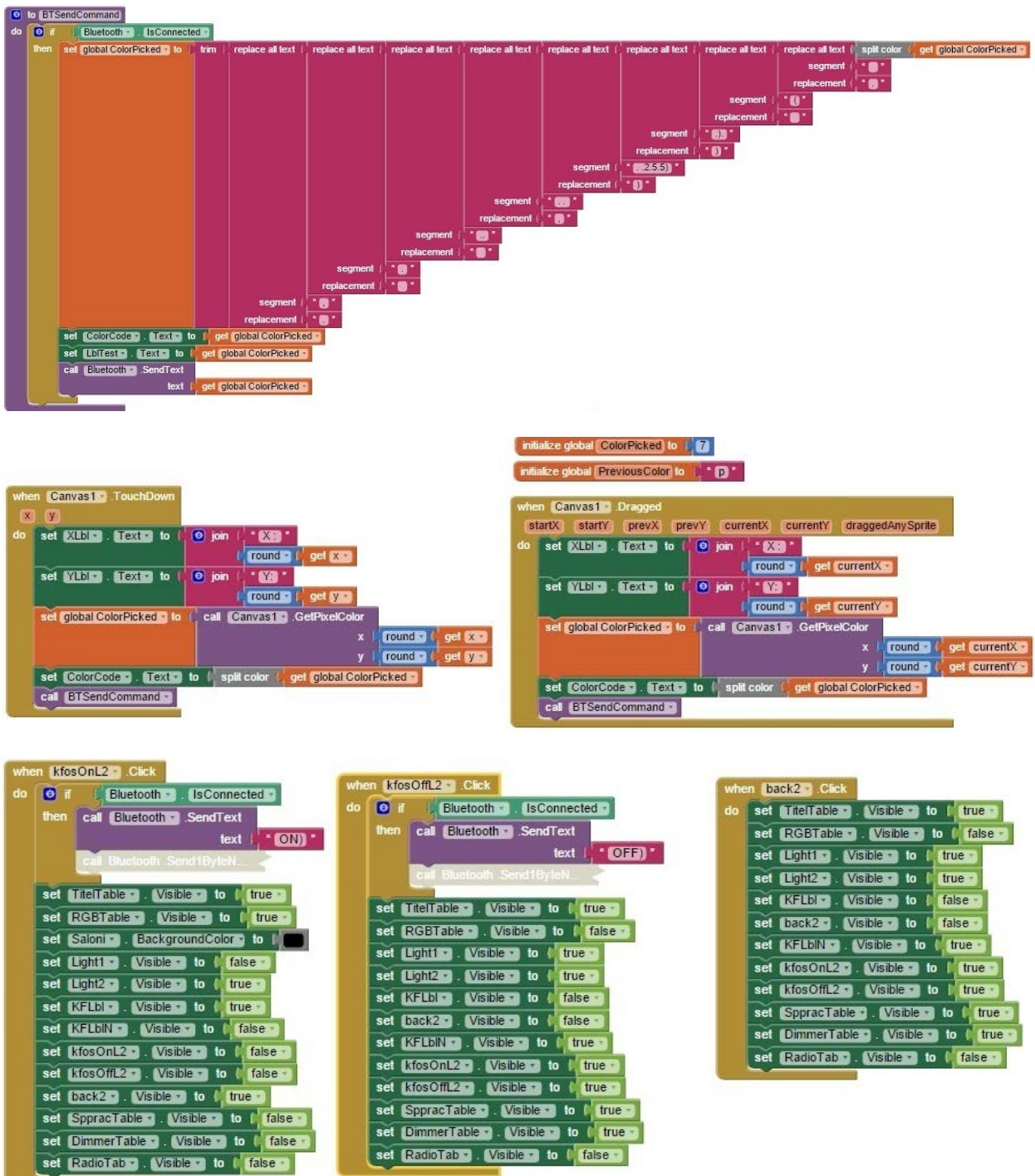
Εικόνα Β.1: Έλεγχος Bluetooth

### Έλεγχος των dimmer και αποθήκευση της κατάστασης σε Βάση Δεδομένων



Εικόνα Β.2: Έλεγχος Dimmer

## Κρυφός φωτισμός: On/Off και αλλαγή χρώματος



Εικόνα Β.3: Έλεγχος Κρυφού φωτισμού



## Ενεργοποίηση Ραδιοφώνου

```
when HiFi_On .Click
do
  call FRadio1 .ReadFrom
  fileName "Station.txt"
  set LblTunePresetFreq .Text to call DBRadio .GetValue
  tag "Station"
  valueIfTagNotThere ""
  call FRadio .ReadFrom
  fileName "tempStation.txt"
  set LblTunePresetFreq .Text to call DBRadio .GetValue
  tag "tempStation"
  valueIfTagNotThere ""
  set HiFi_On .TextColor to green
  set HiFi_Off .TextColor to grey
  set TitelTable .Visible to false
  set RGBTable .Visible to false
  set RadioTab .Visible to true
  set Saloni .BackgroundColor to black
  set Light1 .Visible to false
  set Light2 .Visible to false
  set SpracTable .Visible to false
  set DimmerTable .Visible to false

when FRadio .GotText
text
do set LblTunePresetFreq .Text to get text

when FRadio1 .GotText
text
do set LblTunePresetFreq .Text to get text
```

Εικόνα Β.4: Ενεργοποίηση ραδιοφώνου

## Επιστροφή στο σαλόνι από το Ραδιόφωνο και αποθήκευση ρυθμίσεων

```
when back1 .Click
do
  set TitelTable .Visible to true
  set RGBTable .Visible to false
  set Light1 .Visible to true
  set Light2 .Visible to true
  set SpracTable .Visible to true
  set DimmerTable .Visible to true
  set RadioTab .Visible to false
  call DBRadio .StoreValue
  tag "tempStation"
  valueToStore LblTunePresetFreq .Text
  call FRadio .SaveFile
  text LblTunePresetFreq .Text
  fileName "tempStation.txt"
  call DBRadio .StoreValue
  tag "Station"
  valueToStore LblTunePresetFreq .Text
  call FRadio1 .SaveFile
  text LblTunePresetFreq .Text
  fileName "Station.txt"
```

Εικόνα Β5: Επιστροφή στο Σαλόνι από το Ραδιόφωνο

## Επιλογή των 10 σταθμών του Ραδιοφώνου



Εικόνα Β6: 10 σταθμοί του Ραδιοφώνου

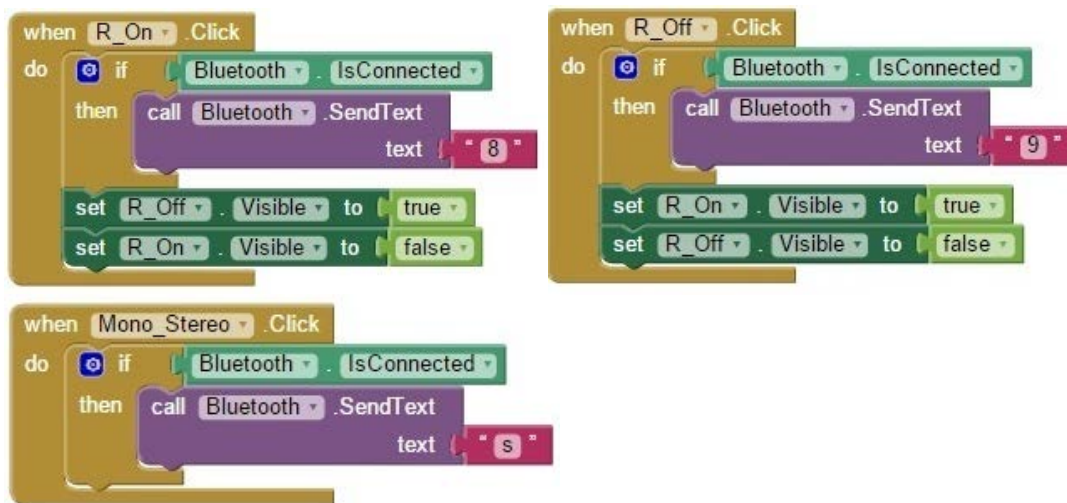


### Λειτουργία κουμπιών MP3



Εικόνα B7: Λειτουργίες MP3

## On/Off Ραδιοφώνου για να μην λειτουργεί μαζί με το MP3



Εικόνα Β6: On/Off Ραδιοφώνου χωρίς MP3

## ΒΙΒΛΙΟΓΡΑΦΙΑ

### ΒΙΒΛΙΑ

Ανάπτυξη εφαρμογών με το Android εκδόσεις Γκιούρδας

Μάθετε την ανάπτυξη εφαρμογών για το Android σε 24 ώρες

Practical Arduino

Exploring Arduino

### ΙΣΤΟΣΕΛΙΔΕΣ

<https://el.wikipedia.org/wiki/Arduino>

<http://deltahacker.gr/arduino-intro/>

<https://th.wikispaces.com/file/view/%CE%95%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE%B3%CE%AE%CF%83%CF%84%CE%BFArduino.pdf>

<https://www.arduino.cc/>

<https://el.wikipedia.org/wiki/Android>

<http://ti-einai.gr/android/>

<https://developer.android.com/training/index.html>

<http://appinventor.mit.edu/explore/>

[http://sepchiou.gr/docs/appinventor\\_code\\_club/App%20Inventor%20Programming%20\(Part%20A\).pdf](http://sepchiou.gr/docs/appinventor_code_club/App%20Inventor%20Programming%20(Part%20A).pdf)

<http://www.appinventor.org/>

<http://tronixstuff.com/2012/02/29/tutorial-analog-input-for-multiple-buttons-part-two/>

[http://www.pjrc.com/teensy/td\\_libs\\_CapacitiveSensor.html](http://www.pjrc.com/teensy/td_libs_CapacitiveSensor.html)

<https://arduino-info.wikispaces.com/Brick-Temperature-DS18B20>

<https://startingelectronics.org/software/arduino/learn-to-program-course/13-switch-break/>

<https://codebender.cc/>