

**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Πλήρης Διαδικτυακός Έλεγχος Οικίας**

**Γεώργιος Γωνιωτάκης  
Νικόλαος Παπαζιάν**

**Εισηγητής: Δρ. Παύλος Κούρος**

**ΑΘΗΝΑ  
ΦΕΒΡΟΥΑΡΙΟΣ 2016**

**(Κενό φύλλο)**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Πλήρης Διαδικτυακός Έλεγχος Οικίας**

**Γεώργιος Γωνιωτάκης  
Α.Μ. 42045**

**Νικόλαος Παπαζιάν  
Α.Μ. 41785**

**Εισηγητής:**

**Δρ. Παύλος Κούρος**

**Εξεταστική Επιτροπή:**

1. ....
2. ....
3. ....

**Ημερομηνία εξέτασης .....**

**(Κενό φύλλο)**



## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος **ΓΕΩΡΓΙΟΣ ΓΩΝΙΩΤΑΚΗΣ**, του **ΓΕΩΡΓΙΟΥ**, με αριθμό μητρώου **42045** φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασής της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού δμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»



---

**(Κενό φύλλο)**


## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος **ΝΙΚΟΛΑΟΣ ΠΑΠΑΖΙΑΝ**, του **ΜΗΝΑ**, με αριθμό μητρώου **41785** φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού δμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»



---

**(Κενό φύλλο)**

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θέλουμε να ευχαριστήσουμε τις οικογένειές μας που μας στάθηκαν τόσο ηθικά όσο και οικονομικά κατά την δημιουργία αυτής της πτυχιακής εργασίας. Επίσης θέλουμε να ευχαριστήσουμε για την βοήθεια, υποστήριξη και καθοδήγηση που μας προσέφερε τον εισηγητή της πτυχιακής άσκησής μας Δρ. Παύλο Κούρο.

**(Κενό φύλλο)**

## ΠΕΡΙΛΗΨΗ

Στην σημερινή εποχή όπου οι καθημερινές ανθρώπινες ανάγκες πολλαπλασιάζονται δημιουργείται η ανάγκη η κατοικία μας να αποκτάει συνεχώς μεγαλύτερη λειτουργικότητα έτσι ώστε να μας καλύπτει. Γι' αυτό τον λόγο διάφορες εταιρίες έχουν δημιουργήσει έξυπνες εφαρμογές οι οποίες εγκαθίστανται σε σπίτια. Κάποια από τα προβλήματα που προκύπτουν από την χρήση αυτών των τεχνολογιών είναι η σχεδόν απαγορευτική τιμή την οποία έχει να αντιμετωπίσει ο καταναλωτής καθώς και διάφοροι περιορισμοί σχετικά με τις λειτουργίες που προσφέρει κάθε τέτοια λύση.

Σκοπός αυτής της πτυχιακής εργασίας είναι η δημιουργία μιας φθηνότερης, πιο εύκολης στην χρήση, πιο διαχειρίσιμης λύσης η οποία θα βελτιώσει την καθημερινότητα του ανθρώπου μετατρέποντας το σπίτι του σε ένα πλήρως διαδικτυακά ελεγχόμενο σπίτι. Ο χρήστης θα μπορεί οποιαδήποτε στιγμή, από οποιοδήποτε μέρος του κόσμου, μέσω οποιασδήποτε συσκευής είναι εφοδιασμένη με πρόσβαση στο διαδίκτυο και πρόγραμμα περιηγητή (browser) να αποκτήσει πρόσβαση στον έλεγχο των συσκευών του σπιτιού του, να δει ζωντανά τιμές διάφορων αισθητηρίων, να παρακολουθήσει μέσω κυκλώματος κλειστής τηλεόρασης (CCTV) την οικία του και διάφορα άλλα. Αυτό θα επιτευχθεί με την χρήση της μητρικής πλακέτας ανοικτού κώδικα Arduino.

**ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ:** Μικροελεγκτές και προγραμματισμός ηλεκτρονικών υπολογιστών

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** μικροελεγκτής, Arduino, αισθητήρια, διαδίκτυο, διαδικτυακός έλεγχος, οικία

## **ABSTRACT**

Nowadays that the everyday human needs are being multiplied, also the need for our house constantly becoming more functional to cover our needs is rising. This is why several companies have created smart applications that are installed into homes. Some of the problems arising from the use of these technologies is the almost prohibitive price which the consumer has to deal with as well as various restrictions on the services offered by each such solution.

The present thesis concern the creation of a cheaper, easier to use, more manageable solution that will improve the daily life of human turning his house into a fully controlled online home. The user may at any time, from anywhere in the world, on any device equipped with internet access and a browser program take control of his home appliances, see live values of various sensors, watch video footage of his home through closed circuit television system (CCTV) and several other functions. This project will be done using the open source Arduino microcontroller board.



## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ</b> .....	15
<b>ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ</b> .....	16
<b>ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ</b> .....	17
<b>1. ΕΙΣΑΓΩΓΗ</b> .....	23
1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας.....	23
1.2 Τι είναι ένας μικροελεγκτής.....	24
1.3 Ιστορική αναδρομή στους μικροελεγκτές.....	24
1.4 Τι κάνει ιδανική την χρήση ενός μικροελεγκτή .....	25
<b>2. ΠΡΟΓΡΑΜΜΑΤΙΖΟΜΕΝΕΣ ΠΛΑΚΕΤΕΣ ΕΦΑΡΜΟΓΩΝ</b> .....	27
2.1 Arduino.....	27
2.2 PINGUINO PIC32.....	32
2.3 Microchip chipKIT Max32 .....	34
2.4 Intel Galileo Gen 2.....	35
2.5 Raspberry Pi 2 – Zero .....	36
2.6 Field Programmable Gate Array (FPGA).....	38
<b>3. ΛΕΙΤΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ</b> .....	39
3.1 Λειτουργία Εφαρμογής .....	39
3.2 Πρόσθετες λειτουργίες εντός του σπιτιού .....	40
3.3 Πρόσθετες λειτουργίες της ιστοσελίδας .....	42
<b>4. ΕΠΙΛΟΓΗ ΠΛΑΚΕΤΑΣ ΕΦΑΡΜΟΓΗΣ</b> .....	43
4.1 Πλακέτα εξοπλισμένη με μικροελεγκτή ή μικροεπεξεργαστή .....	43
4.2 Γιατί Arduino.....	44
4.3 Μοντέλο Arduino που θα χρησιμοποιηθεί .....	44
4.4 Τροφοδοσία του Arduino MEGA .....	46
4.5 Μνήμη Arduino MEGA.....	46
4.6 Ακίδες εισόδου – εξόδου Arduino MEGA .....	46
4.7 Επικοινωνία με το Arduino MEGA .....	47
4.8 Προστασία θυρών USB .....	47
4.9 Προγραμματισμός του Arduino MEGA 2560 .....	47
<b>5. ΑΙΣΘΗΤΗΡΙΑ ΚΑΙ ΠΕΡΙΦΕΡΕΙΑΚΑ</b> .....	49
5.1 Αισθητήριο θερμοκρασίας – υγρασίας DHT11 .....	49
5.2 MQ2 - Αισθητήριο μέτρησης επικίνδυνων αερίων .....	51

5.3	Αισθητήριο εντοπισμού υπέρυθρης ακτινοβολίας (Passive Infrared Sensor) ...	54
5.4	Αισθητήριο βροχής – νερού.....	56
5.5	Αισθητήριο Μέτρησης Υγρασίας Εδάφους .....	57
5.6	Αισθητήριο Μέτρησης Φωτεινότητας.....	58
5.7	Αισθητήριο Αφής .....	59
5.8	Οθόνη TFT 3.2” .....	60
5.9	Οθόνη OLED 0.9” .....	62
5.10	Οθόνη LCD 16x2.....	63
5.11	Πληκτρολόγιο 12 πλήκτρων .....	64
5.12	Κινητήρας DC.....	65
5.13	Βηματικός κινητήρας .....	66
5.14	Ρελέ σε μορφή πλακέτας (Relay Module).....	67
<b>6.</b>	<b>ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ARDUINO</b> .....	<b>69</b>
6.1	Επιλογή αριθμού πλακετών Arduino .....	69
6.2	Εισαγωγικά σχόλια ως προς τον πηγαίο κώδικα.....	69
6.3	Πρωτόκολλο επικοινωνίας Arduino – προγράμματος C# .....	72
<b>7.</b>	<b>ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ</b> .....	<b>95</b>
7.1	Ρόλος βάσεως δεδομένων .....	95
7.2	Επιλογή είδους βάσεως δεδομένων .....	95
7.3	Λειτουργία βάσεως δεδομένων .....	95
7.4	Ρύθμιση βάσεως δεδομένων .....	96
7.5	Δομή βάσεως δεδομένων.....	97
7.6	Ασφάλεια βάσεως δεδομένων .....	100
7.7	Κώδικας δημιουργίας βάσεως δεδομένων.....	101
<b>8.</b>	<b>ΚΕΝΤΡΙΚΟ ΠΡΟΓΡΑΜΜΑ ΔΙΑΜΕΣΟΛΑΒΗΣΗΣ</b> .....	<b>103</b>
8.1	Γλώσσα προγραμματισμού C# και πρόγραμμα ανάπτυξης εφαρμογής.....	103
8.2	Εισαγωγικά για το πρόγραμμα .....	103
8.3	Δημιουργία παραθυρικής εφαρμογής με το Microsoft Visual Studio.....	104
8.4	Κώδικας πρωτοκόλλου επικοινωνίας .....	106
8.5	Κεντρικό παράθυρο εφαρμογής .....	110
8.6	Παράθυρο οδηγιών εφαρμογής.....	122
8.7	Παράθυρο ρυθμίσεων εφαρμογής.....	123
8.8	Παράθυρο πληροφοριών πτυχιακής εργασίας .....	125
<b>9.</b>	<b>ΙΣΤΟΣΕΛΙΔΑ ΕΦΑΡΜΟΓΗΣ</b> .....	<b>127</b>
9.1	Σκοπός κατασκευής ιστοσελίδας .....	127

9.2 Τεχνολογίες που χρησιμοποιήθηκαν .....	127
9.3 Συλλογή εργαλείων ανοιχτού κώδικα – Bootstrap .....	128
9.4 Αρχική σελίδα εφαρμογής .....	133
9.5 Σελίδα εισόδου στο σύστημα .....	134
9.6 Κεντρική σελίδα συστήματος .....	134
9.7 Κεντρικό Πάνελ Αισθητηρίων .....	136
9.8 Συνεχής Προβολή Αισθητηρίων .....	136
9.9 Σελίδα παρακολούθησης κλειστού κυκλώματος παρακολούθησης (CCTV)...	137
9.10 Σελίδα προβολής συμβάντων.....	138
9.11 Κεντρική σελίδα προβολής στατιστικών στοιχείων .....	139
9.12 Σελίδα επεξεργασίας συσκευών οικίας.....	139
9.13 Σελίδα ρυθμίσεων.....	140
9.14 Σελίδα ρυθμίσεων λογαριασμού .....	141
9.15 Σελίδα εξόδου από το σύστημα .....	142
9.16 Είσοδος επαλήθευσης δύο σταδίων (Two Step Verification).....	142
9.17 Ειδοποιήσεις συστήματος και ηλεκτρονικού ταχυδρομείου .....	144
9.18 Έλεγχος μέσω φωνητικών εντολών .....	147
9.19 Ασφάλεια διαδικτυακής εφαρμογής .....	149
9.20 Συμβατότητα με φορητές συσκευές.....	149
<b>10. ΜΑΚΕΤΑ ΕΦΑΡΜΟΓΗΣ</b> .....	<b>155</b>
10.1 Σκοπός κατασκευής μακέτας.....	155
10.2 Φυσικά χαρακτηριστικά μακέτας.....	155
10.3 Υλικά κατασκευής.....	155
10.4 Συσκευές και αισθητήρια μακέτας.....	159
10.5 Παρουσίαση χώρων .....	160
<b>11. ΕΦΑΡΜΟΓΗ ΣΕ ΠΡΑΓΜΑΤΙΚΕΣ ΣΥΝΘΗΚΕΣ</b> .....	<b>167</b>
11.1 Υλικό πραγματικής εφαρμογής.....	167
11.2 Πηγαίος κώδικας Arduino .....	167
11.3 Οδηγίες χρήσης πραγματικής εφαρμογής .....	169
<b>ΠΑΡΑΡΤΗΜΑ Α΄</b> .....	<b>170</b>
<b>ΠΑΡΑΡΤΗΜΑ Β΄</b> .....	<b>175</b>
<b>ΠΑΡΑΡΤΗΜΑ Γ΄</b> .....	<b>191</b>
<b>ΠΑΡΑΡΤΗΜΑ Δ΄</b> .....	<b>193</b>
<b>ΠΑΡΑΡΤΗΜΑ Ε΄</b> .....	<b>208</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b> .....	<b>210</b>

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

RAM – Random Access Memory

EPROM – Erasable Programmable Read-Only Memory

EEPROM – Electrically Erasable Programmable Read-Only Memory

I2C – I squared C

SPI – Serial Peripheral Interface

USB – Universal Serial Bus

USART – Universal Synchronous/Asynchronous Receiver/Transmitter

UART - Universal Asynchronous Receiver/Transmitter

ROM – Read Only Memory

PWM – Pulse-Width Modulation

HDMI – High-Definition Multimedia Interface

FPGA – Field Programmable Gate Array

PCB – Printed Circuit Board

TTL – Transistor Transistor Logic

PIR – Passive Infrared

LDR – Light Dependent Resistors

GUI – Graphical User Interface

XAML – Extensible Application Markup Language

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Συγκριτικός πίνακας τεχνικών χαρακτηριστικών πλακετών Arduino .....	31
Πίνακας 2: Τεχνικά χαρακτηριστικά οθόνης TFT 3.2” .....	61
Πίνακας 3: Συνδεσμολογία οθόνης TFT 3.2” με Arduino MEGA .....	61
Πίνακας 4: Τεχνικά χαρακτηριστικά οθόνης OLED .....	62
Πίνακας 5: Τεχνικά χαρακτηριστικά οθόνης LCD 16x2 .....	63
Πίνακας 6: Λειτουργίες Arduino συσκευών και επεξήγηση λειτουργίας .....	82
Πίνακας 7: Λειτουργίες Arduino αισθητηρίων και επεξήγηση λειτουργίας τους .....	84
Πίνακας 8: Επεξήγηση πεδίων πίνακα Users .....	97
Πίνακας 9: Επεξήγηση πεδίων πίνακα Preferences .....	98
Πίνακας 10: Επεξήγηση πεδίων πίνακα Logs .....	98
Πίνακας 11: Επεξήγηση πεδίων πίνακα Appliances .....	99
Πίνακας 12: Επεξήγηση πεδίων πίνακα Sensors .....	99
Πίνακας 13: Επεξήγηση πεδίων πίνακα Surveillance .....	100
Πίνακας 14: Διαστάσεις Ξύλινων Κομματιών Μακέτας .....	156
Πίνακας 15: Υλικά κατασκευής .....	156
Πίνακας 16: Ηλεκτρικό – ηλεκτρονικό υλικό μακέτας .....	160

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Gary Boone .....	25
Εικόνα 2: Michael Cochran .....	25
Εικόνα 3: Ένα από τα πρώτα Arduino .....	27
Εικόνα 4: Arduino Mini .....	28
Εικόνα 5: Arduino Nano .....	28
Εικόνα 6: Arduino Uno .....	28
Εικόνα 7: Arduino Diecimila .....	28
Εικόνα 8: Arduino Extreme .....	28
Εικόνα 9: Arduino Duemilanove .....	28
Εικόνα 10: Arduino Yun .....	29
Εικόνα 11: Arduino Mega 2560 .....	29
Εικόνα 12: Arduino Lilypad .....	29
Εικόνα 13: Arduino Gemma .....	29
Εικόνα 14: GSM Shield .....	31
Εικόνα 15: Wi-Fi Shield .....	31
Εικόνα 16: Ethernet Shield .....	32
Εικόνα 17: Motor Shield .....	32
Εικόνα 18: Pinguino PIC32 .....	33
Εικόνα 19: Pinguino PIC32 Micro .....	33
Εικόνα 20: Pinguino IDE .....	33
Εικόνα 21: chipKIT Max32 .....	35
Εικόνα 22: MPIDE .....	35
Εικόνα 23: Μπροστινή όψη Intel Galileo Gen 2 .....	36
Εικόνα 24: Πίσω όψη Intel Galileo Gen 2 .....	36
Εικόνα 25: Raspberry Pi 2 .....	37
Εικόνα 26: Raspberry Pi Zero .....	37
Εικόνα 27: Xilinx Spartan 6 SP605 .....	38
Εικόνα 28: Altera Cyclone V GX .....	38
Εικόνα 29: Κεντρική λειτουργία εφαρμογής .....	40
Εικόνα 30: Arduino MEGA 2560 και σχολιασμός των περιφερειακών του .....	45
Εικόνα 31: Λειτουργία DHT11 .....	50
Εικόνα 32: Τάση λειτουργίας και κατανάλωση του DHT11 .....	50

Εικόνα 33: DHT11 .....	50
Εικόνα 34: Συνδεσμολογία DHT11 με το Arduino MEGA.....	51
Εικόνα 35: Χαρακτηριστικά λειτουργίας MQ2.....	52
Εικόνα 36: Συνδεσμολογία αισθητηρίου MQ2.....	52
Εικόνα 37: Μετρήσεις διαφορετικών αερίων .....	53
Εικόνα 38: MQ2 Module.....	53
Εικόνα 39: Τρόπος συνδεσμολογίας MQ2 Module .....	53
Εικόνα 40: Φυσικά χαρακτηριστικά, χαρακτηριστικά λειτουργίας και δομή PIR ....	55
Εικόνα 41: Μεταλλικό κάλυμμα αισθητηρίου .....	55
Εικόνα 42: Πλαστική επικάλυψη .....	55
Εικόνα 43: Εύρος κάλυψης με την χρήση της πλαστικής επικάλυψης .....	56
Εικόνα 44: PIR Sensor Module .....	56
Εικόνα 45: Συνδεσμολογία PIR με Arduino .....	56
Εικόνα 46: Αισθητήριο νερού (module) .....	57
Εικόνα 47: Συνδεσμολογία αισθητηρίου με Arduino.....	57
Εικόνα 48: Αισθητήριο μέτρησης υγρασίας χώματος .....	58
Εικόνα 49: Συνδεσμολογία αισθητηρίου.....	58
Εικόνα 50: Φωτοαντίσταση (Module) .....	59
Εικόνα 51: Σύνδεση φωτοαντίστασης με Arduino .....	59
Εικόνα 52: Τρόπος λειτουργίας αισθητηρίου.....	60
Εικόνα 53: Αισθητήριο Αφής .....	60
Εικόνα 54: Μπροστά Όψη Οθόνης .....	61
Εικόνα 55: Πίσω Όψη Οθόνης .....	61
Εικόνα 56: Οθόνη OLED 0.96” .....	63
Εικόνα 57: Σύνδεση οθόνης με Arduino MEGA .....	63
Εικόνα 58: Μπροστινή όψη οθόνης .....	64
Εικόνα 59: Πίσω όψη οθόνης.....	64
Εικόνα 60: Πληκτρολόγιο 4x3 .....	64
Εικόνα 61: Σύνδεση πληκτρολογίου με Arduino.....	64
Εικόνα 62: Κινητήρας συνεχούς.....	65
Εικόνα 63: Σύνδεση του κινητήρα με Arduino .....	65
Εικόνα 64: Βηματικός κινητήρας .....	66
Εικόνα 65: Σύνδεση κινητήρα με Arduino Mega.....	66
Εικόνα 66: Relay Module .....	67
Εικόνα 67: Relay Module 8 καναλιών .....	67

Εικόνα 68: Παράδειγμα σύνδεσης ρελέ με Arduino και φωτιστικό.....	68
Εικόνα 69: Πρωτόκολλο σύζευξης επικοινωνίας .....	73
Εικόνα 70: Παράδειγμα αίτησης κοινοποίησης τιμής αισθητηρίου .....	76
Εικόνα 71: Παράδειγμα αίτησης ενεργοποίησης συσκευής.....	79
Εικόνα 72: Δομή βάσεως δεδομένων.....	100
Εικόνα 73: Διαδικασία δημιουργίας νέου προγράμματος στο Microsoft Visual Studio.....	104
Εικόνα 74: Διαδικασία δημιουργίας νέας παραθυρικής εφαρμογής στο Microsoft Visual Studio .....	105
Εικόνα 75: Παράθυρο σχεδίασης (Designer) .....	106
Εικόνα 76: Επιτυχία σύνδεσης πλακετών .....	122
Εικόνα 77: Αποτυχία σύνδεσης πλακετών .....	122
Εικόνα 78: Παράθυρο οδηγιών χρήσης .....	123
Εικόνα 79: Παράθυρο ρυθμίσεων εφαρμογής.....	125
Εικόνα 80: Παράθυρο πληροφοριών πρότζεκτ .....	126
Εικόνα 81: EBay.com σε ανάλυση οθόνης 1920 x 900 .....	128
Εικόνα 82: EBay.com σε ανάλυση οθόνης 980 x 1280 .....	129
Εικόνα 83: EBay.com σε ανάλυση οθόνης 768 x 1024 .....	129
Εικόνα 84: EBay.com σε ανάλυση 360 x 640 .....	130
Εικόνα 85: EBay.com σε ανάλυση 320 x 480 .....	130
Εικόνα 86: Λειτουργία συστήματος πλέγματος .....	131
Εικόνα 87: Δομή συστήματος πλέγματος.....	132
Εικόνα 88: Σχεδίαση ιστοσελίδας για συσκευές διαφορετικών αναλύσεων οθόνης .....	132
Εικόνες 89 – 95: Αρχική σελίδα εφαρμογής .....	133
Εικόνα 96: Σελίδα εισόδου στο σύστημα.....	134
Εικόνα 97: Παράδειγμα ελέγχου συσκευών σαλονιού.....	135
Εικόνα 98: Παράδειγμα ελέγχου συσκευών τουαλέτας .....	135
Εικόνα 99: Κεντρικό Πάνελ Αισθητηρίων .....	136
Εικόνα 100: Σελίδα συνεχούς προβολής τιμών αισθητηρίων .....	137
Εικόνα 101: Σελίδα παρακολούθησης κλειστού κυκλώματος τηλεόρασης (CCTV) .....	138
Εικόνα 102: Σελίδα προβολής συμβάντων.....	138
Εικόνα 103: Κεντρική σελίδα προβολής στατιστικών στοιχείων .....	139
Εικόνα 104: Σελίδα επεξεργασίας συσκευών οικίας.....	140
Εικόνα 105: Σελίδα προσωπικών ρυθμίσεων .....	141



Εικόνα 106: Σελίδα ρυθμίσεων λογαριασμού.....	141
Εικόνα 107: Σελίδα εξόδου από το σύστημα.....	142
Εικόνα 108: Ενεργοποίηση επαλήθευσης δύο σταδίων.....	143
Εικόνα 109: Ρύθμιση εφαρμογής.....	144
Εικόνα 110: Επαλήθευση δύο σταδίων.....	144
Εικόνα 111: Μήνυμα ηλεκτρονικού ταχυδρομείου σε περίπτωση ενεργοποίησης της λειτουργίας διακοπών.....	145
Εικόνα 112: Μήνυμα ηλεκτρονικού ταχυδρομείου σε περίπτωση ενεργοποίησης του συστήματος πυρασφάλειας.....	145
Εικόνα 113: Μήνυμα ηλεκτρονικού ταχυδρομείου σε περίπτωση ενεργοποίησης του συστήματος συναγερμού.....	146
Εικόνα 114: Μήνυμα ηλεκτρονικού ταχυδρομείου σε περίπτωση ενεργοποίησης του διακόπτη έκτακτης ανάγκης.....	146
Εικόνα 115: Ειδοποίηση (Popup Notification) σε περίπτωση έκτακτου συμβάντος.....	147
Εικόνα 116: Παράθυρο φωνητικών εντολών.....	148
Εικόνα 117: Λειτουργία ηχογράφησης αντί για πληκτρολόγηση.....	148
Εικόνες 118 – 149: Εμφάνιση μηνυμάτων ηλεκτρονικού ταχυδρομείου και ιστοσελίδας σε κινητή συσκευή.....	150
Εικόνα 150: Χρωματολόγιο οικίας.....	157
Εικόνα 151: Δάπεδο κουζίνας.....	157
Εικόνα 152: Δάπεδο μπάνιου.....	157
Εικόνα 153: Εσωτερική όψη παραθύρου.....	157
Εικόνα 154: Εξωτερική όψη παραθύρου.....	157
Εικόνα 155: Κεντρική πόρτα οικίας.....	158
Εικόνα 156: Πόρτα δωματίου οικίας.....	158
Εικόνα 157: Πόρτα χώρου στάθμευσης οικίας.....	158
Εικόνα 158: Ηλεκτρολογικός Πίνακας Οικίας.....	160
Εικόνα 159: Τηλεόραση Οικίας.....	161
Εικόνα 160: Ηλεκτρολογικός Πίνακας.....	161
Εικόνα 161: Σύστημα συναγερμού.....	161
Εικόνα 162: Μηχανισμός τέντας.....	161
Εικόνα 163: Ανεμόμετρο.....	161
Εικόνα 164: Ηλιακός θερμοσίφωνας.....	161
Εικόνα 165: Μπροστά όψη πινακίδας logo.....	162
Εικόνα 166: Πίσω όψη πινακίδας logo.....	162

Εικόνα 167: Πάνελ ρύθμισης τέντας και ελέγχου διακόπτη έκτακτης ανάγκης ....	162
Εικόνα 168: Χώρος στάθμευσης .....	163
Εικόνα 169: Χώρος καθιστικού .....	163
Εικόνα 170: Χώρος μπάνιου .....	163
Εικόνα 171: Υπνοδωμάτιο .....	163
Εικόνα 172: Κουζίνα Εικόνα .....	164
Εικόνα 173: Είσοδος οικίας .....	164
Εικόνα 174: Μπροστά όψη οικίας .....	164
Εικόνα 175: Πίσω όψη οικίας .....	164
Εικόνα 176: Πλαϊνή όψη οικίας .....	165

**(Κενό φύλλο)**

## ΚΕΦΑΛΑΙΟ 1

### ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο θα αναλυθεί το αντικείμενο αυτής της πτυχιακής εργασίας καθώς θα γίνει και μια γρήγορη ιστορική αναδρομή όσον αφορά τις τεχνολογίες που ήδη υπάρχουν στον τομέα των έξυπνων οικιακών εφαρμογών.

#### 1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας

Σκοπός αυτής της πτυχιακής εργασίας είναι η δημιουργία ενός συστήματος το οποίο θα επιτρέπει στον χρήστη να ελέγχει οποιαδήποτε συσκευή της οικίας του θέλει αυτός, οποιαδήποτε στιγμή, από οποιοδήποτε σημείο του κόσμου υπάρχει κάλυψη ίντερνετ καθώς και να ελέγχει διάφορες συνθήκες όπως είναι τα επίπεδα θερμοκρασίας και υγρασίας, το επίπεδο ποτίσματος των φυτών του κ.α. με την βοήθεια ειδικών αισθητηρίων. Επιπρόσθετα θα μπορεί να παρακολουθήσει την κατάσταση του χώρου του μέσω συστήματος βιντεοσκόπησης, να δει στατιστικά υγρασίας και θερμοκρασίας του χώρου του, να ειδοποιείται για τις σημαντικές εξελίξεις στο σπίτι του μέσω μηνυμάτων ηλεκτρονικού ταχυδρομείου. Αυτές οι λειτουργίες είναι μόνο κάποιες από αυτές που θα αποδώσει στον χρήστη η τελική εφαρμογή μέσω ενός περιβάλλοντος το οποίο θα παρέχει ασφάλεια πληροφοριών, οικονομία και πλήρη διαχείριση.

Για να επιτευχθούν όλα αυτά χρειαζόμαστε μια συσκευή η οποία θα είναι ικανή να καταγράφει από τα επιμέρους αισθητήρια μετρήσεις και να τις δημοσιεύει στο διαδίκτυο. Παρομοίως θα χρειαστούμε αυτή η συσκευή να δέχεται από το διαδίκτυο εντολές όπως για παράδειγμα το να ενεργοποιηθεί ή να απενεργοποιηθεί μια συσκευή. Για αυτόν τον λόγο θα χρησιμοποιήσουμε την μητρική πλακέτα Arduino. Έπειτα οι μετρήσεις των αισθητηρίων θα αποστέλλονται μέσω του πρωτόκολλου USB σε ένα παραθυρικό πρόγραμμα κατασκευασμένο στην γλώσσα προγραμματισμού C# το οποίο θα είναι αρμόδιο για την καταχώρηση των πληροφοριών σε μια βάση δεδομένων. Τέλος ο χρήστης θα έχει πρόσβαση σε όλο αυτό το υλικό μέσω μιας ιστοσελίδας.

## 1.2 Τι είναι ένας μικροελεγκτής

Ένας μικροελεγκτής είναι για την ακρίβεια ένας μικρός υπολογιστής ο οποίος έρχεται στα χέρια μας στην μορφή τσιπ και μπορεί να εμπεριέχει πολλά διαφορετικά εξαρτήματα όπως είναι ένας μονοπύρηνος επεξεργαστής, μνήμη μόνιμης αποθήκευσης (flash memory), μνήμη προσωρινής αποθήκευσης (RAM memory), μνήμη EEPROM, προγραμματιζόμενες αναλογικές/ψηφιακές εισόδους/εξόδους, εσωτερικό κρύσταλλο χρονισμού, πρωτόκολλα επικοινωνίας μεταξύ άλλων συσκευών (I<sup>2</sup>C, SPI, USB, USART, UART). [1]

## 1.3 Ιστορική αναδρομή στους μικροελεγκτές

Όλα ξεκίνησαν το 1971 όπου οι δύο μηχανικοί της εταιρίας Texas Instruments, Gary Boone και Michael Cochran κατασκεύασαν με επιτυχία τον πρώτο μικροελεγκτή με ονομασία TMS 1000. Ο TMS 1000 ήταν εξοπλισμένος με δύο μνήμες, μία μόνο ανάγνωσης και μία ανάγνωσης και εγγραφής. Δεν άργησε να γίνει διαθέσιμος στο ευρύ κοινό μόλις λίγα χρόνια αργότερα το 1974. Αργότερα το 1977 η εταιρία Intel έδωσε την απάντησή της με την δημιουργία του Intel 8048 ο οποίος συνδυάζε μνήμη RAM και ROM και χρησιμοποιήθηκε ευρέως σε ηλεκτρολόγια υπολογιστών της εποχής.

Μέχρι εκείνη την στιγμή οι μικροελεγκτές αντιμετώπιζαν ένα πρόβλημα και αυτό είχε να κάνει με τον επαναπρογραμματισμό τους. Υπήρχαν δύο είδη μικροελεγκτών εκείνη την εποχή αυτοί οι οποίοι προγραμματιζόντουσαν μόνο μια φορά (One Time Programmable – OTP) και αυτοί οι οποίοι είχαν την δυνατότητα επανεγγραφής δεδομένων με την μέθοδο καθαρισμού της μνήμης μέσω της υπεριώδους ακτινοβολίας (UV). Για να συμβεί αυτό οι μικροελεγκτές ήταν εφοδιασμένοι με ένα μικρό παράθυρο κατασκευασμένο από κρύσταλλο και με την έκθεση αυτού του παραθύρου σε υπεριώδη ακτινοβολία για μερικές ώρες η μνήμη άδειαζε.

Όλα αυτά μέχρι το 1993 που η εταιρία Microchip δημιουργώντας την σειρά μικροελεγκτών της PIC16X84 εισήγαγε την τεχνολογία μνήμης EEPROM. Τώρα πια οι χρήστες μπορούσαν να αδειάσουν την μνήμη του μικροελεγκτή μέσω του ρεύματος.

Τον ίδιο χρόνο η εταιρία Atmel κατασκεύασε τον πρώτο μικροελεγκτή ο οποίος χρησιμοποιούσε Flash μνήμη. [1]



Εικόνα 1: Gary Boone



Εικόνα 2: Michael Cochran

#### 1.4 Τι κάνει ιδανική την χρήση ενός μικροελεγκτή

Ο μικροελεγκτής είναι ένα τσιπ όπως αναφέραμε και παραπάνω το οποίο περικλείει διάφορα άλλα συστήματα στο εσωτερικό του όπως μνήμη, για την αποθήκευση προσωρινών και μόνιμων δεδομένων, και πρωτόκολλα επικοινωνίας, για την επικοινωνία με διάφορες άλλες συσκευές. Ανάμεσα σε αυτά αναφέραμε ότι εμπεριέχει και επεξεργαστή ο οποίος δεν είναι μεγάλης ισχύος και ταχύτητας πράγμα που δεν τον κάνει ιδανικό για χρήση σε εφαρμογές μεγάλου φόρτου επεξεργασίας δεδομένων. Στην περίπτωση που θα θέλαμε να δώσουμε μία λύση σε ένα τέτοιο πρόβλημα θα χρησιμοποιούσαμε έναν μικροεπεξεργαστή.

Ο μικροεπεξεργαστής αποτελείται από πολλαπλάσιες φορές δυνατότερο επεξεργαστή ο οποίος μπορεί να επιλύσει δύσκολους και μεγάλους υπολογισμούς αλλά εξειδικεύεται σε αυτό. Αυτό σημαίνει ότι ένας μικροεπεξεργαστής στερείται τεχνολογίες που έχει ένας μικροελεγκτής. Για αυτό στην περίπτωση που θα θέλαμε να χρησιμοποιήσουμε έναν μικροεπεξεργαστή θα χρειαζόμασταν να τον εφοδιάσουμε με διάφορα ακόμα περιφερειακά όπως μνήμη και άλλα. Με αυτόν τον τρόπο ανεβάζουμε το κόστος κατά πολύ συνθήκη η οποία σε μία μεγάλη γραμμή παραγωγής μπορεί να αποβεί καταστροφική.

Το πλεονέκτημα της χρήσης του μικροελεγκτή είναι ότι ο αριθμός περιφερειακών συσκευών που χρειάζεται για να λειτουργήσει είναι ελάχιστος πράγμα που μειώνει κατά πολύ το κόστος υλοποίησης μιας συσκευής. Σε αυτό έρχεται να προστεθεί φυσικά και η τιμή του μικροελεγκτή σαν συσκευή η οποία ανέρχεται σε κάποια λεπτά (cents) του ευρώ.

Στην σημερινή εποχή ένας μικροελεγκτής βρίσκεται σχεδόν σε κάθε συσκευή που χρησιμοποιούμε καθώς είναι κατάλληλος για γρήγορους υπολογισμούς. Επίσης

Πλήρης Διαδικτυακός Έλεγχος Οικίας

μικροελεγκτές χρησιμοποιούνται και στους χώρους της ρομποτικής και της αυτοκινητοβιομηχανίας.

## ΚΕΦΑΛΑΙΟ 2

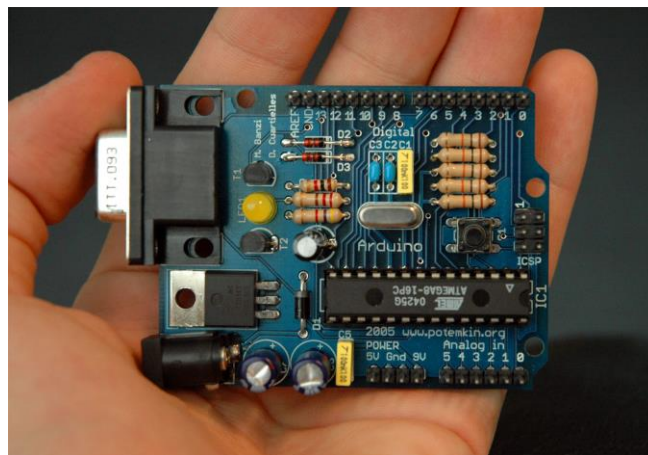
### ΠΡΟΓΡΑΜΜΑΤΙΖΟΜΕΝΕΣ ΠΛΑΚΕΤΕΣ ΕΦΑΡΜΟΓΩΝ

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τις πιο γνωστές προγραμματιζόμενες πλακέτες εφαρμογών που χρησιμοποιούνται αυτήν την στιγμή.

#### 2.1 Arduino

Ας ξεκινήσουμε από το Arduino το οποίο θα αναλύσουμε και λίγο περισσότερο από τα υπόλοιπα μιας και θα είναι η πλακέτα εφαρμογών που θα χρησιμοποιήσουμε.

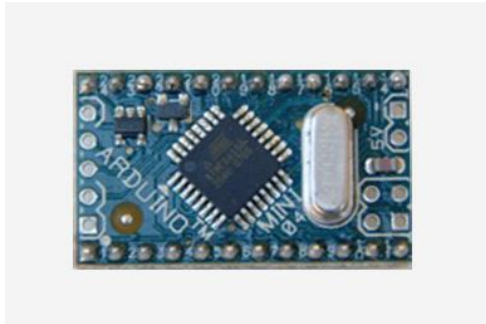
Η ιδέα για την κατασκευή του Arduino προήλθε το 2005 από τους Ιταλούς Massimo Banzi και David Cueartielles σε μια προσπάθειά τους να φτιάξουν μια πλακέτα εφαρμογών χαμηλού κόστους για μαθησιακούς λόγους. Ένα από τα πρώτα Arduino που κατασκευάστηκαν όπως φαίνεται και στην εικόνα 3 χρησιμοποιούσε μη επιφανειακά εξαρτήματα (through hole) και επίσης ο προγραμματισμός του γινόταν μέσω σειριακής θύρας 9 ακίδων (RS232). [2]



Εικόνα 3: Ένα από τα πρώτα Arduino

Το Arduino κυκλοφορεί σε πολλές διαφορετικές εκδόσεις αλλά όλες οι εκδόσεις του έχουν ένα κοινό μεταξύ τους και αυτό είναι ότι χρησιμοποιούν μικροελεγκτή της εταιρίας Atmel. Στις παρακάτω εικόνες θα δούμε κάποιες από τις διαφορετικές εκδόσεις Arduino που υπάρχουν.





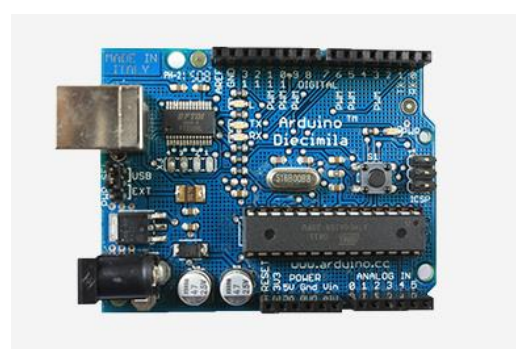
Εικόνα 4: Arduino Mini



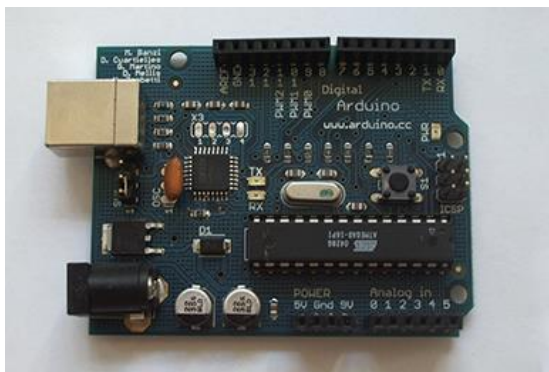
Εικόνα 5: Arduino Nano



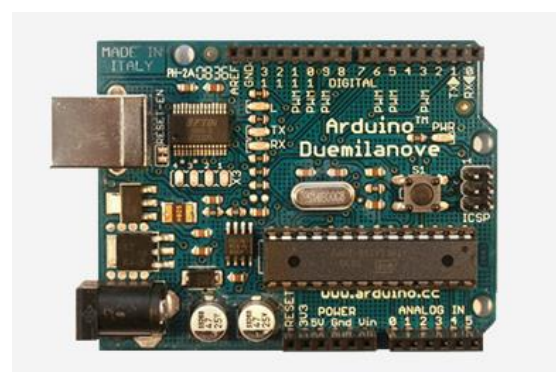
Εικόνα 6: Arduino Uno



Εικόνα 7: Arduino Diecimila



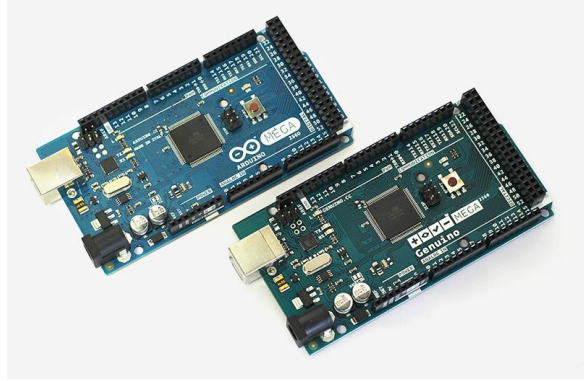
Εικόνα 8: Arduino Extreme



Εικόνα 9: Arduino Duemilanove

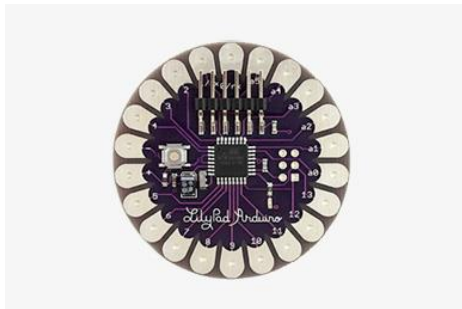


Εικόνα 10: Arduino Yun



Εικόνα 11: Arduino Mega 2560

Από ότι βλέπουμε η γκάμα των πλακετών Arduino είναι μεγάλη. Παρακάτω θα δούμε κάποιες εκδόσεις Arduino οι οποίες είναι τόσο μικρές έτσι ώστε να μπορούν να φορευθούν για παράδειγμα στον ανθρώπινο καρπό.



Εικόνα 12: Arduino Lilypad



Εικόνα 13: Arduino Gemma

Φυσικά όλες αυτές οι εκδόσεις διαφέρουν η μία από την άλλη ως προς τις δυνατότητες που προσφέρουν. Ο παρακάτω πίνακας έχει τα πιο βασικά χαρακτηριστικά κάθε μοντέλου.

Name	Processor	Operating /Input Voltage	CPU Speed	Analog In/Output	Digital IO/ PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
<b>Due</b>	ATSAM3X8E	3.3 V / 7-12 V	84 MHz	12/2	54/12	-	96	512	2 Micro	4
<b>Gemma</b>	ATtiny85	3.3 V / 4-16 V	8 MHz	1/0	3/2	0.5	0.5	8	Micro	0
<b>LilyPad</b>	ATmega168V ATmega328P	2.7-5.5 V / 2.7-5.5 V	8MHz	6/0	14/6	0.512	1	16	-	-
<b>LilyPad Simple Snap</b>	ATmega328P	2.7-5.5 V / 2.7-5.5 V	8 MHz	4/0	9/4	1	2	32	-	-
<b>LilyPad USB</b>	ATmega32U4	3.3 V / 3.8-5 V	8 MHz	4/0	9/4	1	2.5	32	Micro	-
<b>Mega 2560</b>	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
<b>Micro</b>	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
<b>Nano</b>	ATmega168 ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	0.512 1	1 2	16 32	Mini	1

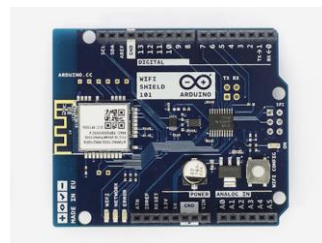
<b>Pro</b>	ATmeg a168 ATmeg a328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/ 6	0.512 1	1 2	16 32	-	1
<b>Pro Mini</b>	ATmeg a328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/ 6	1	1	32	-	1
<b>Uno</b>	ATmeg a328P	5 V / 7-12 V	16 MHz	6/0	14/ 6	1	2	32	Reg ular	1
<b>Yùn</b>	ATmeg a32U4 AR933 1 Linux	5 V	16 MHz 400MH z	12/0	20/ 7	1	2.5 16M B	32 64 MB	Micr o	1
<b>Mini</b>	ATmeg a328P	5 V / 7-9 V	16 MHz	8/0	14/ 6	1	2	32	-	-

Πίνακας 1: Συγκριτικός πίνακας τεχνικών χαρακτηριστικών πλακετών Arduino

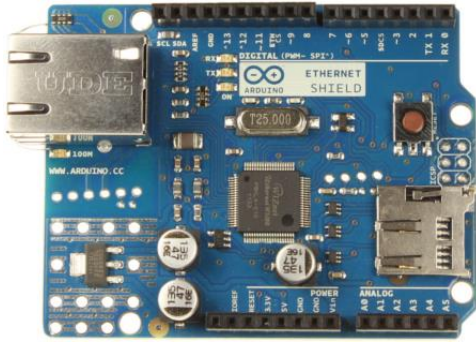
Η αρχιτεκτονική του Arduino είναι ιεραρχημένη έτσι ώστε η πλακέτα να δέχεται άλλες πλακέτες οι οποίες προσαρμόζονται από πάνω τις (shields) που προσδίδουν στο Arduino περαιτέρω λειτουργικότητα χωρίς να κάνουν κατάληψη των εισόδων του (headers) αφήνοντας έτσι τον χρήστη να τις χρησιμοποιήσει.



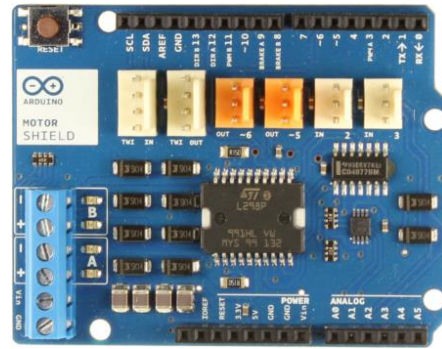
Εικόνα 14: GSM Shield



Εικόνα 15: Wi-Fi Shield



Εικόνα 16: Ethernet Shield



Εικόνα 17: Motor Shield

Το Arduino αποτελεί ανοικτή κοινότητα κώδικα (software) και υλικού (hardware) που αυτό πρακτικά σημαίνει ότι όλα τα σχέδια των πλακετών Arduino είναι διαθέσιμα προς λήψη στον ιστότοπο της Arduino στο <http://www.arduino.cc>. Γι' αυτόν τον λόγο πολλές εταιρίες σε όλο τον κόσμο αναλαμβάνουν την κατασκευή απομιμήσεων του Arduino σε χαμηλότερες τιμές. Παράλληλα οι πλακέτες αυτές συνοδεύονται από το ολοκληρωμένο περιβάλλον ανάπτυξης του (IDE) το οποίο είναι δωρεάν διαθέσιμο προς λήψη στην ιστοσελίδα τους. Μέσα σε αυτό το πρόγραμμα ο χρήστης μπορεί να χρησιμοποιήσει τις ήδη υλοποιημένες βιβλιοθήκες που υπάρχουν για να δημιουργήσει τις δικές του ιδέες γράφοντας στην γλώσσα προγραμματισμού Wiring η οποία ουσιαστικά βασίζεται στην γλώσσα προγραμματισμού C++. [2]

Το Arduino χαρακτηρίζεται από το μέγεθος της κοινότητάς του το οποίο απαρτίζεται από χιλιάδες χρήστες, είτε επαγγελματίες είτε χομπίστες, οι οποίοι σπεύδουν να δημιουργήσουν καινούργιες βιβλιοθήκες ή να δημοσιεύσουν κώδικά τους με σκοπό να βοηθήσουν την κοινότητα.

## 2.2 PINGUINO PIC32

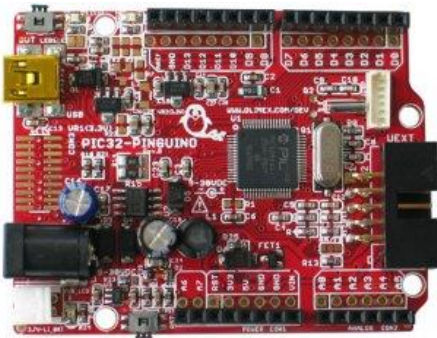
Το Pinguino όπως λέει και το αστείο όνομά τους είναι μία σειρά από πλακέτες εφαρμογών οι οποίες αποτελούνται από ανοιχτό κώδικα (software) και ανοιχτό υλικό (hardware) και χρησιμοποιούν μικροελεγκτές της εταιρίας Microchip. Οι πλακέτες αυτές παράγονται από την εταιρία Olimex και προγραμματίζονται μέσω του προγράμματός τους.

Η ονομασία των πλακετών αυτών προέρχεται από την συνένωση των λέξεων Arduino και PIC. Οι πλακέτες αυτές προγραμματίζονται μέσω του ολοκληρωμένου



περιβάλλοντος ανάπτυξης τους (IDE) το οποίο έχει ήδη προεγκατεστημένες και προσαρμοσμένες τις περισσότερες βιβλιοθήκες που έχει και το πρόγραμμα του Arduino και είναι διαθέσιμο δωρεάν για λήψη στην ιστοσελίδα <http://www.pinguino.cc> Εμφανισιακά το IDE της Pinguino είναι ίδιο με το αντίστοιχο της Arduino και αυτό συμβαίνει για να νιώσει ο χρήστης άνετα με το πρόγραμμα. Παράλληλα οι εντολές που χρησιμοποιούνται είναι ακριβώς ίδιες και στο Pinguino, Επίσης το Pinguino έχει συμβατότητα ως προς τα Shields του Arduino.

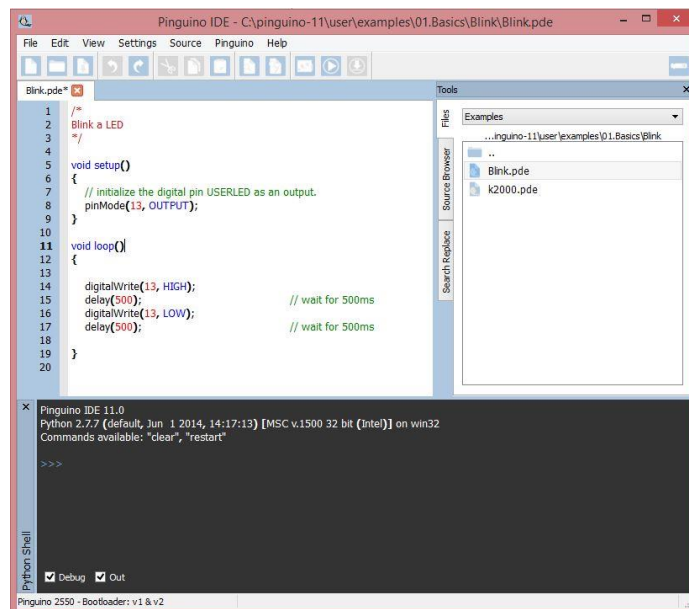
Παρακάτω θα δούμε τρεις φωτογραφίες, μία του προγράμματος και δύο από δύο διαφορετικές εκδόσεις της πλακέτας.



Εικόνα 18: Pinguino PIC32



Εικόνα 19: Pinguino PIC32 Micro



Εικόνα 20: Pinguino IDE

Κάποια από τα τεχνικά χαρακτηριστικά της πλακέτας της εικόνας 18 είναι τα παρακάτω:

- Συχνότητα 80 Mhz
- 256KB Flash Μνήμη
- 32KB RAM Μνήμη
- Τάση λειτουργίας από 9 μέχρι 30V DC
- Ρυθμιστές τάσεως χαμηλής ισχύος
- RTC - Real Time Clock
- 5 PWM Εξόδους
- 2 I<sup>2</sup>C, 2 UART Εξόδους
- USB On The Go (OTG)
- 10 Αναλογικές Εισόδους/Εξόδους [3]

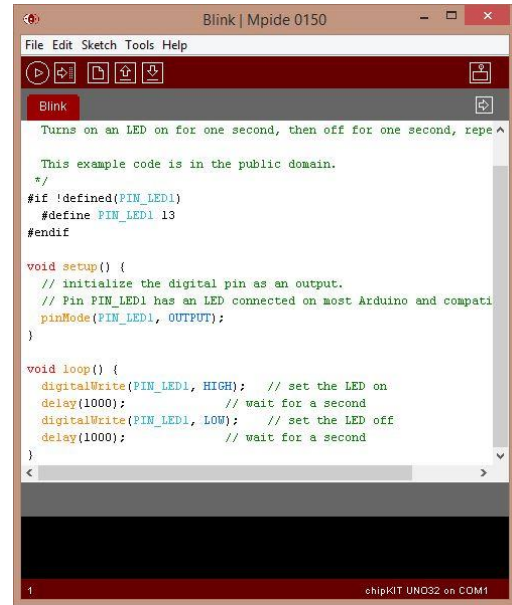
### 2.3 Microchip chipKIT Max32

Η πλακέτα chipKIT Max32 αποτελεί μια συνεργασία της εταιρίας Microchip και της εταιρίας Digilent. Η πλακέτα χρησιμοποιεί μικροελεγκτή της εταιρίας Microchip και έχει σχεδιασθεί για να έχει πλήρη συμβατότητα με τις πλακέτες Shields της εταιρίας Arduino. Όσον αφορά τα φυσικά της χαρακτηριστικά μοιάζει με την πλακέτα Arduino Mega. Η ιδιομορφία αυτής της πλακέτας είναι το περιβάλλον προγραμματισμού της. Χρησιμοποιεί το πρόγραμμα MPIDE της εταιρίας chipKIT το οποίο αποτελεί πρακτικά πιστό αντίγραφο του προγράμματος του Arduino τόσο σε περιβάλλον εντολών όσο και σε εμφάνιση. Στο πρόγραμμα αυτό βρίσκονται υλοποιημένες από πριν οι γνωστές βιβλιοθήκες που βρίσκονται και στο πρόγραμμα του Arduino. Παράλληλα ο μικροελεγκτής μπορεί να προγραμματιστεί μέσω οποιασδήποτε άλλης γλώσσας (Assembly, c) μέσα από την χρήση του επίσημου προγράμματος της Microchip με την ονομασία MPLAB IDE. Η πλακέτα υπάρχει σε πολλές διαφορετικές εκδόσεις με ποικιλία μικροελεγκτών και μεγεθών. Άλλες εκδόσεις αυτής της πλακέτας αποτελούν είναι οι:

- chipKIT Uno32
- chipKIT Pro



Εικόνα 21: chipKIT Max32



Εικόνα 22: MPIDE

Κάποια από τα τεχνικά χαρακτηριστικά της πλακέτας της εικόνας 21 είναι τα παρακάτω:

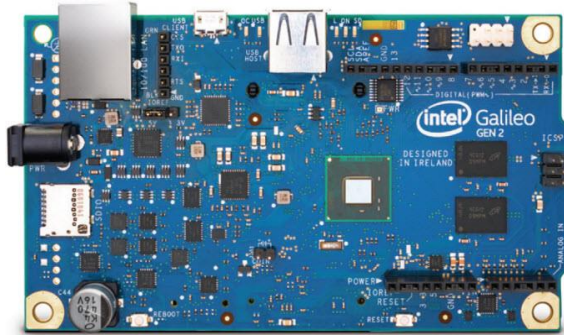
- 512KB Μνήμη Flash
- 128KB Μνήμη RAM
- Ταχύτητα λειτουργίας 80 MHz
- 4 SPI και 5 I<sup>2</sup>C εξόδους
- 16 channel 10-bit ADC, 5 PWM εξόδους
- 83 ακίδες εισόδου/εξόδου
- 3.3V τάση λειτουργίας
- 16 αναλογικές εισόδους
- USB 2.0 OTG
- 10/100 Ethernet MAC [4]

## 2.4 Intel Galileo Gen 2

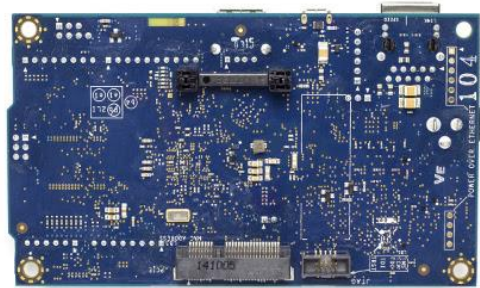
Η εταιρία Intel κατασκεύασε το 2014 την δεύτερη έκδοση της προγραμματιζόμενης πλακέτας εφαρμογών της. Όπως και όλες οι παραπάνω πλακέτες εφαρμογών προσφέρει πλήρη συμβατότητα με τις πρόσθετες πλακέτες (Shields) καθώς και το περιβάλλον προγραμματισμού του Arduino. Μία διαφορά από τις άλλες προγραμματιζόμενες πλακέτες είναι ότι αυτή η πλακέτα αποτελεί επίσης και έναν μονοπύρηνου υπολογιστή ο οποίος λειτουργεί σε περιβάλλον Linux. Έτσι ο χρήστης



στήνοντας πολύ εύκολα ένα περιβάλλον ασφαλούς δικτυακού πρωτοκόλλου (SSH) μπορεί σε μικρό χρόνο να έχει πρόσβαση στο λειτουργικό περιβάλλον της συσκευής.



Εικόνα 23: Μπροστινή όψη Intel Galileo Gen 2



Εικόνα 24: Πίσω όψη Intel Galileo Gen 2

Κάποια από τα τεχνικά χαρακτηριστικά της συσκευής βρίσκονται παρακάτω:

- SoC: Quark X1000 @ 400MHz
- RAM: 256MByte DDR3; 512KByte embedded SRAM
- Flash Storage: 8MByte NOR Flash; 8Kbit EEPROM
- LAN: 10/100 Ethernet (RJ-45 connector)
- PCI-E: Mini-PCI Express 1x Slot
- USB 2.0
- 20 Digital I/O
- 6 Analog
- UART ,I<sup>2</sup>C
- SD Card: MicroSD
- JTAG
- 1 PWM [5]

## 2.5 Raspberry Pi 2 – Zero

Το Raspberry Pi είναι ένας υπολογιστής σε μέγεθος τσέπης ο οποίος δημιουργήθηκε από την ομώνυμη εταιρία με σκοπό να βοηθήσει στην διδασκαλία της τεχνολογίας των υπολογιστών σε μαθητές. [6]

Πρακτικά το Raspberry Pi διαφέρει πάρα πολύ από τις προηγούμενες πλακέτες εφαρμογών που αναφέραμε και αυτό συμβαίνει επειδή βάση της αρχιτεκτονικής του μοιάζει περισσότερο σε ένα πλήρες υπολογιστικό σύστημα. Το Raspberry Pi είναι

εξοπλισμένο με θύρα HDMI έτσι ώστε να επιτρέπει την σύνδεση του με οθόνη. Επίσης είναι εφοδιασμένο με τέσσερις θύρες USB, υποδοχή βύσματος 3.5 χιλιοστών για την αναπαραγωγή ήχου, εξωτερικές υποδοχές πολλαπλών χρήσεων όπως είναι βιντεοκάμερες, οθόνες και άλλες συσκευές. Τέλος είναι εφοδιασμένο με υποδοχή καλωδίου τύπου Ethernet έτσι ώστε να επιτρέπει την σύνδεσή του στο διαδίκτυο.

Προς το παρόν υπάρχουν δεκάδες ελεύθερα λειτουργικά συστήματα τα οποία είναι διαθέσιμα για εγκατάσταση στην συσκευή αυτή σε μεγάλη ποικιλία. Ο χρήστης της πλακέτας αυτής μπορεί να επιλέξει ανάμεσα σε λειτουργικά συστήματα που είναι βασισμένα στα Linux καθώς και σε άλλα όπως για παράδειγμα είναι τα Windows 10.

Φυσικά οι δυνατότητες δεν σταματάνε εδώ αφού το Raspberry Pi είναι εφοδιασμένο με 40 προγραμματιζόμενες ακίδες οι οποίες λειτουργούν ως είσοδοι και έξοδοι. Η επιλογή της γλώσσας προγραμματισμού της πλακέτας αυτής είναι ξεκάθαρα επιλογή του χρήστη αφού αυτή η πλακέτα μέσω της χρήσης βιβλιοθηκών μπορεί να προγραμματιστεί με τις περισσότερες γνωστές γλώσσες προγραμματισμού όπως είναι για παράδειγμα η Java, C, C++, Ruby, Python και άλλες.

Πρόσφατα η εταιρία έκανε την παραγωγή του μικρότερου υπολογιστή που υπάρχει στον κόσμο και κοστίζει μόλις 5 ευρώ καθώς δεν δίστασε να το δώσει και δωρεάν μέσω του περιοδικού της. Φυσικά το αποτέλεσμα ήταν η πλακέτα να εξαντληθεί από όλα τα διαθέσιμα σημεία αγοράς της μόλις σε μερικές ώρες.



Εικόνα 25: Raspberry Pi 2



Εικόνα 26: Raspberry Pi Zero

## 2.6 Field Programmable Gate Array (FPGA)

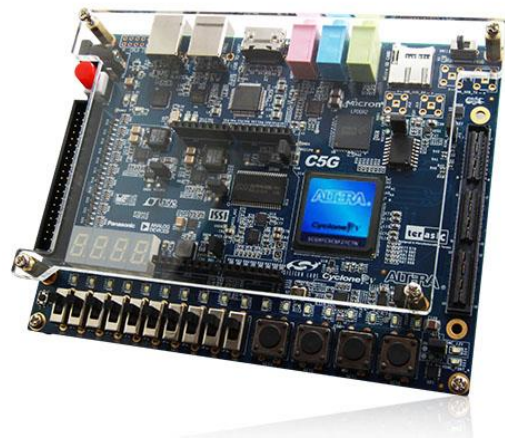
Το FPGA ή Field Programmable Gate Array είναι ένα είδος προγραμματιζόμενου ολοκληρωμένου κυκλώματος γενικής χρήσης το οποίο εμπεριέχει μεγάλο αριθμό πυλών, απαριθμητών, καταχωρητών και άλλων.

Ένα FPGA μπορεί να προγραμματιστεί με γλώσσες περιγραφής υλικού (hardware) όπως είναι η VHDL, η AHDL και η Verilog. Το αρνητικό του FPGA είναι ότι κάθε φορά που σταματάει να τροφοδοτείται από ηλεκτρική ενέργεια χάνει τον προγραμματισμό του πράγμα που κάνει απαραίτητο την συνεργασία του με κάποια εξωτερική μνήμη μόνιμης αποθήκευσης.[7]

Οι δύο γνωστότερες εταιρίες στον χώρο των FPGA είναι η Altera και η Xilinx οι οποίες παράγουν πλακέτες εφαρμογών για εκπαιδευτικούς σκοπούς που χρησιμοποιούν FPGAs. Αυτές οι πλακέτες συνήθως είναι εξοπλισμένες με το κατάλληλο εκπαιδευτικό υλικό όπως είναι led, διακόπτες, οθόνες LCD, οθόνες 7-Segment και άλλα έτσι ώστε να βοηθήσουν τον εκπαιδευόμενο να πειραματιστεί με τον κώδικα και το υλικό φτιάχνοντας απλές εφαρμογές όπως για παράδειγμα μετρητές,



Εικόνα 27: Xilinx Spartan 6 SP605



Εικόνα 28: Altera Cyclone V GX

## ΚΕΦΑΛΑΙΟ 3

### ΛΕΙΤΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ

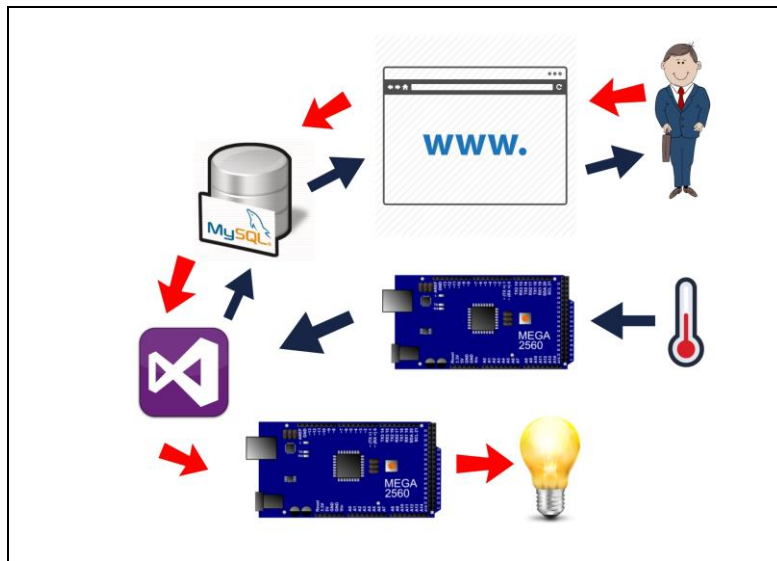
#### 3.1 Λειτουργία Εφαρμογής

Η εφαρμογή μας αποτελείται από 4 βασικά σημεία λειτουργίας:

- Την πλακέτα εφαρμογών Arduino
- Μία βάση δεδομένων
- Ένα πρόγραμμα διαμεσολαβητή κατασκευασμένο στην γλώσσα προγραμματισμού C#
- Μια διαδικτυακή ιστοσελίδα

Το πλάνο λειτουργίας της εφαρμογής μας είναι το ακόλουθο:

- Όλες οι οικιακές συσκευές που θέλουμε να ελέγξουμε βρίσκονται τοποθετημένες πάνω στην πλακέτα Arduino
- Όλα τα αισθητήρια που θέλουμε να ενημερωνόμαστε για τις τιμές τους βρίσκονται συνδεδεμένα πάνω στην δεύτερη πλακέτα Arduino
- Σε κάθε μία από τις πλακέτες παραπάνω βρίσκεται εγκατεστημένο το ειδικά κατασκευασμένο πρόγραμμα το οποίο μέσω σειριακής USB επικοινωνίας θα επικοινωνεί με το πρόγραμμά μας φτιαγμένο στην γλώσσα προγραμματισμού C#
- Το πρόγραμμά μας θα στέλνει στην πλακέτα συσκευών οποιαδήποτε πληροφορία σε σχέση με την κατάσταση κάθε συσκευής αφού ενημερωθεί από την βάση
- Το πρόγραμμά μας θα δέχεται από την πλακέτα αισθητηρίων τις μετρήσεις τους και θα ενημερώνει την βάση δεδομένων
- Η διαδικτυακή μας ιστοσελίδα θα λειτουργήσει ως χειριστήριο (controller) επιτρέποντας στον χρήστη τον έλεγχο των συσκευών του καθώς και την ενημέρωσή του για τις τιμές των αισθητηρίων και άλλα.



Εικόνα 29: Κεντρική λειτουργία εφαρμογής

### 3.2 Πρόσθετες λειτουργίες εντός του σπιτιού

Η εφαρμογή μας προσφέρει διάφορες άλλες παροχές οι οποίες προέρχονται από την χρήση αισθητηρίων εντός και εκτός του σπιτιού. Για παράδειγμα η εφαρμογή προσφέρει αυτόματη ρύθμιση της εξωτερικής τέντας σε περίπτωση έντονου ανέμου, βροχής καθώς και σε διαφορετικές περιπτώσεις ηλιοφάνειας. Έτσι σε περίπτωση έντονου ανέμου η τέντα θα σηκωθεί έτσι ώστε να αποτραπεί η καταστροφή της. Σε περίπτωση βροχής η τέντα θα κατέβει για να προστατέψει τον κήπο ή το μπαλκόνι από το νερό. Το απόγευμα η τέντα θα ανέβει έτσι ώστε να προσφέρει όσο το δυνατόν περισσότερο φως μέσα στο σπίτι έτσι ώστε να μην χρειάζεται η παροχή φωτός μέσω ηλεκτρικών μέσων π.χ. φωτιστικά. Φυσικά ο ιδιοκτήτης της οικίας θα μπορεί οποιαδήποτε στιγμή να απενεργοποιήσει αυτό το σύστημα απλά με το πάτημα ενός κουμπιού.

Μια επιπρόσθετη λειτουργία της οικίας είναι και το αυτόματο σύστημα ποτίσματος. Το πότισμα των φυτών είναι ένα μεγάλο θέμα προς συζήτηση λόγω του ότι μερικές φορές δεν είμαστε σίγουροι για το πότε χρειάζεται ή δεν χρειάζεται πότισμα ένα φυτό, συνήθεια που μας οδηγεί να καταστρέφουμε τα φυτά μας χωρίς να το θέλουμε. Ένα ακόμα πρόβλημα είναι ότι τα αυτοματοποιημένα συστήματα ποτίσματος της αγοράς λειτουργούν με χρονοδιακόπτες πράγμα που σημαίνει ότι σε περίπτωση ενεργοποίησης του συστήματος ποτίσματος σε μια συγκεκριμένη χρονική στιγμή

υπάρχει η περίπτωση να μην χρειάζονται πότισμα όλα τα φυτά που είναι συνδεδεμένα πάνω στο σύστημα πράγμα που οδηγεί στην καταστροφή τους. Πρόβλημα επίσης αποτελεί και το πότισμα κατά την μακρόχρονη απουσία π.χ. καλοκαιρινές διακοπές, των κατοίκων της οικίας. Για να αποφευχθούν όλα αυτά κάθε φυτό της οικίας είναι εφοδιασμένο με αισθητήριο μέτρησης υγρασίας χώματος το οποίο βρίσκεται συνδεδεμένο σε ξεχωριστή αντλία ποτίσματος. Ο αισθητήρας παίρνει συνεχώς μετρήσεις για να αποφανθεί ποια είναι η καλύτερη στιγμή για να ανοίξει την αντλία και ποια η καλύτερη στιγμή για να την κλείσει.

Η οικία είναι εξοπλισμένη με συστήματα αυτόματης θέρμανσης και κλιματισμού έτσι ώστε ο χώρος να βρίσκεται πάντα στην επιθυμητή θερμοκρασία. Παράλληλα τα συστήματα αυτά συντελούν στην εξοικονόμηση ηλεκτρικής ενέργειας καθώς και την προστασία του περιβάλλοντος.

Οθόνες με τεχνολογία LED βρίσκονται εγκατεστημένες στα δωμάτια του σπιτιού προσφέροντας εύκολη ενημέρωση των κατοίκων για τις συνθήκες θερμοκρασίας που επικρατούν μέσα σε κάθε δωμάτιο του σπιτιού.

Μέσω των αισθητηρίων καπνού και επικίνδυνων αερίων που βρίσκονται στην οικία κάτοικος ενημερώνεται οποιαδήποτε στιγμή μέσω μηνύματος ηλεκτρονικού ταχυδρομείου για τυχόν κινδύνους όπως για παράδειγμα φωτιά, διαρροή υγραερίου από κάποιο σύστημα θέρμανσης, αν η ηλεκτρική κουζίνα έχει μείνει ανοικτή κ.α. έτσι ώστε να μπορεί να προλαβαίνει τέτοιες καταστάσεις.

Στο σπίτι βρίσκεται εγκατεστημένο πλήκτρο έκτακτης ανάγκης (panic button) το οποίο κατά την ενεργοποίησή του αναλαμβάνει να ενημερώσει τον διακομιστή μας ότι ο κάποιος κάτοικος της οικίας βρίσκεται σε έκτακτη ανάγκη. Ο διακομιστής στην συνέχεια παρέχει την δυνατότητα αποστολής μηνύματος ηλεκτρονικού ταχυδρομείου σε μία διεύθυνση που έχει ορίσει ο χρήστης. Έτσι αυτό το σύστημα γίνεται ιδανικό για οποιαδήποτε ανάγκη αντιμετωπίσει ένας κάτοικος, για χρήση σε σπίτια που κατοικούν άτομα με ειδικές ανάγκες ή ηλικιωμένοι κ.α.

Το σύστημα συναγερμού το οποίο είναι εγκατεστημένο στην οικία ενημερώνει συνεχώς τον διακομιστή για οποιαδήποτε παραβίαση του χώρου συμβεί. Σε περίπτωση παραβίασης του χώρου ο ιδιοκτήτης της οικίας ενημερώνεται μέσω μηνύματος ηλεκτρονικού ταχυδρομείου.

Η ασφάλεια του χώρου είναι μεγάλο ζήτημα για αυτό η οικία έχει εγκατεστημένη διαδικτυακή κάμερα η οποία εφόσον είναι συνδεδεμένη σε κάποιον υπολογιστή μπορεί σε πραγματικό χρόνο να εκπέμψει το υλικό της στην ιστοσελίδα της εφαρμογής μας.

### 3.3 Πρόσθετες λειτουργίες της ιστοσελίδας

Φυσικά μία ιστοσελίδα η οποία περιέχει τον έλεγχο οποιασδήποτε συσκευής, τις μετρήσεις όλων των αισθητηρίων, ζωντανή παρακολούθηση μέσω βίντεο και προσωπικές πληροφορίες για τους κατοίκους μιας οικίας δεν θα μπορούσε να έχει χαμηλή ασφάλεια. Για αυτόν τον λόγο η ιστοσελίδα παρακολουθείται από γνωστή εταιρία αντιικών προγραμμάτων για οποιαδήποτε παραβίαση του συστήματος ασφαλείας και είναι έτοιμη ακόμη και να την θέσει εκτός λειτουργίας σε περίπτωση που αυτό κριθεί αναγκαίο.

Οποιοδήποτε αίτημα που προσπαθήσει να εισέλθει ή εξέλθει στην βάση δεδομένων ελέγχεται και καθορίζεται αν είναι κατάλληλο για να επεξεργαστεί ή να απορριφθεί από αυτήν.

Ως επιπρόσθετα μέτρα ασφάλειας είναι η καταγραφή αναλυτικών πληροφοριών για την είσοδο και έξοδο κάποιου χρήστη στο σύστημα καθώς και η διαπίστευση δύο βημάτων (two step verification). Μέσω αυτού του τύπου διαπίστευσης ο χρήστης χρησιμοποιεί την κινητή συσκευή του ως γεννήτρια τυχαίων κωδικών και βάσει ενός μοναδικού κλειδιού που του παρέχουμε μπορούμε μέσω ενός αλγορίθμου να κρίνουμε αν όντως πρόκειται για πραγματικό χρήστη ή για χρήστη που προσπαθεί να αποκτήσει πρόσβαση χωρίς να έχει το δικαίωμα. Έτσι μαζί με τα βασικά διαπιστευτήριά του, όνομα χρήστη και κωδικό ασφαλείας, ο χρήστης θα πρέπει να πληκτρολογήσει και τον κωδικό που θα του επιδείξει η γεννήτρια εκείνη την στιγμή.

Πέρα από την ασφάλεια ο χρήστης μπορεί να κάνει εξατομικευμένες ρυθμίσεις ως προς το ποιοι χώροι ελέγχονται μέσω του συστήματος, ποια είδη μηνυμάτων ηλεκτρονικού ταχυδρομείου θέλει να λαμβάνει ακόμη και να προσθέσει καινούργιες συσκευές που έχει συνδέσει πάνω στην πλακέτα και θέλει να είναι διαθέσιμες από τον διαδικτυακό έλεγχο.

Επίσης σε συσκευές οι οποίες έχουν διαθέσιμη την δυνατότητα ηχογράφησης φωνής αντί για πληκτρολόγηση ο χρήστης μπορεί βάσει ενός μοτίβου που έχει οριστεί, να δώσει φωνητικές εντολές έτσι ώστε να ενεργοποιήσει ή να απενεργοποιήσει μια συσκευή της οικίας του.

Τέλος ο χρήστης έχει την δυνατότητα μέσω ζωντανών γραφημάτων να εξετάσει τις συνθήκες θερμοκρασίας και υγρασίας του χώρου του οποιαδήποτε στιγμή.



## ΚΕΦΑΛΑΙΟ 4

### ΕΠΙΛΟΓΗ ΠΛΑΚΕΤΑΣ ΕΦΑΡΜΟΓΗΣ

#### 4.1 Πλακέτα εξοπλισμένη με μικροελεγκτή ή μικροεπεξεργαστή

Κατά την έρευνά μας για την κατασκευή αυτής της εφαρμογής έπρεπε να επιλέξουμε ανάμεσα στη χρήση μιας πλακέτας η οποία θα είχε μεγαλύτερη επεξεργαστική ισχύ μέσα από την χρήση μικροεπεξεργαστή αλλά θα είχε ανάγκη διάφορα περιφερειακά όπως είναι ξεχωριστά τσιπ μνήμης έτσι ώστε να αποθηκεύει το πρόγραμμα που θα δημιουργούσαμε, ξεχωριστά τσιπ με τεχνολογίες σειριακής επικοινωνίας όπως διασύνδεση με θύρα USB, ξεχωριστά τσιπ παροχής τεχνολογιών γραφικών όπως είναι η συμβατότητα με οθόνες VGA, HDMI. Μετά από δοκιμές και στα δύο είδη πλακετών καταλήξαμε ότι ένα σύστημα εξοπλισμένο με μικροεπεξεργαστή δεν θα ήταν ιδανικό για χρήση γιατί από την στιγμή που θέλουμε να ανταλλάζουμε πληροφορίες με μια βάση δεδομένων πολλές από τις τεχνολογίες, όπως για παράδειγμα η σύνδεση με οθόνη HDMI, η σύνδεση ήχου κ.α. θα παραμέναν αναξιοποίητες. Επίσης οι πλακέτες μικροεπεξεργαστών λειτουργούν και ως ηλεκτρονικοί υπολογιστές που είναι εξοπλισμένοι με λειτουργικά συστήματα, πράγμα που δεν θα αξιοποιούσαμε από την στιγμή που θέλουμε να χρησιμοποιήσουμε μόνο μια παραθυρική εφαρμογή η οποία θα καταναλώσει ελάχιστη μνήμη και σχεδόν μηδενική υπολογιστική ισχύ.

Έπειτα ένας σημαντικός παράγοντας ήταν το κόστος υλοποίησης. Λόγω του ότι οι πλακέτες μικροεπεξεργαστών έχουν ως στόχο την επεξεργαστική ισχύ δεν χρησιμοποιούν τις τεχνολογίες System On Chip (SoC) πράγμα που σημαίνει ότι χρειάζονται και άλλα περιφερειακά για να λειτουργήσουν. Αυτό σημαίνει άνοδος στην τιμή πώλησής τους. Από την άλλη πλευρά οι πλακέτες μικροελεγκτών λόγω της αυξημένης ζήτησής τους από το κοινό έχουν πιο μειωμένες τιμές. Επιπρόσθετα οι πλακέτες μικροεπεξεργαστών στην συντριπτική τους πλειοψηφία δεν αποτελούν συσκευές ανοικτού υλικού πράγμα που σημαίνει ότι δεν δημιουργείται ανταγωνισμός ως προς την κατασκευή τους αφού η τιμή καθορίζεται από έναν κατασκευαστή. Η διαφορά όμως σε αυτόν τον τομέα είναι μεγάλη διότι ήδη πολλές πλακέτες μικροελεγκτών αποτελούν συσκευές ανοικτού υλικού επιτρέποντας σε διαφορετικά εργοστάσια να τις κατασκευάσουν και να τις προμηθεύσουν στο κοινό. Αυτό



δημιουργεί ανταγωνισμό στις τιμές πώλησης με αποτέλεσμα την μείωση των τιμών τους.

Εκτός από ανοικτό υλικό οι περισσότερες πλακέτες μικροελεγκτών αποτελούν και συσκευές ανοικτού κώδικα. Αυτό πρακτικά σημαίνει ότι όλες οι λεπτομέρειες που αφορούν την δομή και την λειτουργία μιας πλακέτας από πλευράς κώδικα είναι ελεύθερες για το κοινό. Μέσα από αυτήν την ανοιχτή κοινότητα κώδικα οι χρήστες μπορούν να κατανοήσουν ευκολότερα το υλικό και να κατασκευάσουν ευκολότερα εφαρμογές. Αυτό έχει ως συνέπεια την συσπείρωση μιας μεγάλης κοινότητας τέτοιων χρηστών οι οποίοι μοιράζονται τις ιδέες τους, δημιουργούν νέες βιβλιοθήκες κώδικα για την οδήγηση διαφορετικών συσκευών όπως βηματικούς κινητήρες, οθόνες LCD, πληκτρολογίων και άλλων περιφερειακών τα οποία είναι απαραίτητα κομμάτια ενός προϊόντος.

## **4.2 Γιατί Arduino**

Θέλαμε να κάνουμε αυτήν την εφαρμογή διαθέσιμη σε μια μεγάλη γκάμα διαφορετικών μικροελεγκτών. Αυτό σημαίνει ότι με τον ίδιο κώδικα όλες οι πλακέτες Arduino που είναι εφοδιασμένες με υποστήριξη σειριακής επικοινωνίας USB αυτομάτως συγχρονίζονται με το κεντρικό μας πρόγραμμα χωρίς κάποια τροποποίηση. Στην περίπτωση που κάποιος χρήστης θα ήθελε να χειριστεί το μέγιστο έως 12 συσκευές της οικίας του θα μπορούσε να χρησιμοποιήσει με ελάχιστο κόστος ένα Arduino Uno. Αν ο χρήστης ήθελε να ελαττώσει ακόμα πιο πολύ το κόστος θα μπορούσε να χρησιμοποιήσει ένα Arduino Nano χωρίς να θυσιάσει ούτε μία από τις 12 συσκευές που θα ήθελε να ελέγξει.

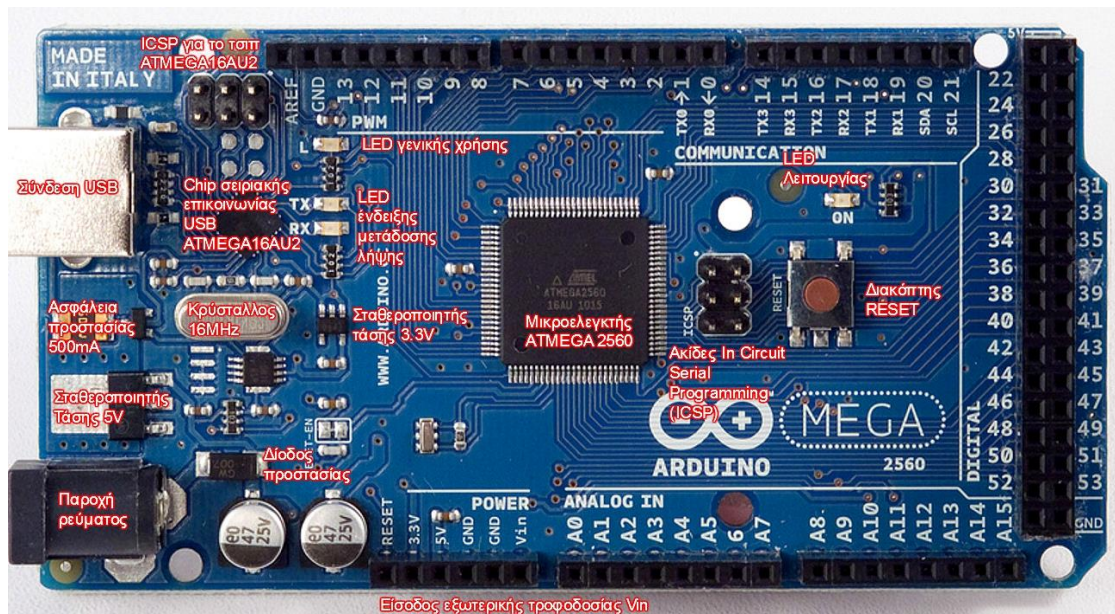
Πολύ σημαντικό είναι ότι το Arduino βγαίνει σε διαφορετικά μεγέθη πράγμα που το κάνει ιδανικό για την τοποθέτηση του σε πολλά διαφορετικά σημεία. Για παράδειγμα τα φυσικά χαρακτηριστικά ενός Arduino Uno είναι 68,6mm X 53,4mm και ενός Arduino Nano είναι 45mm X 18mm παρέχοντας τις ίδιες δυνατότητες στον χρήστη.

## **4.3 Μοντέλο Arduino που θα χρησιμοποιηθεί**

Σε αυτήν την εφαρμογή θα θέλαμε να παρουσιάσουμε όσο το δυνατόν περισσότερες συσκευές και περισσότερους αισθητήρες μπορούμε για αυτό θα χρησιμοποιήσουμε δύο Arduino MEGA 2560 τα οποία παρέχουν 54 ψηφιακές εισόδους/εξόδους, 16 αναλογικές εισόδους/εξόδους, υποστήριξη σειριακής επικοινωνίας μέσω USB κ.α. Περισσότερες λεπτομέρειες παρουσιάζονται παρακάτω.

### Τεχνικά χαρακτηριστικά πλακέτας Arduino MEGA 2560:

- Τάση λειτουργίας 5V
- Μικροελεγκτής ATMEGA 2560 8-Bit
- Όριο τάσεως λειτουργίας 6-20V
- 54 ψηφιακοί είσοδοι / έξοδοι 15 εκ των οποίων παρέχουν έξοδο Pulse With Modulation (PWM)
- 16 αναλογικοί είσοδοι/έξοδοι
- 256KB Flash μνήμη
- 8KB SRAM μνήμη
- 4KB EEPROM μνήμη
- Συχνότητα λειτουργίας 16MHz
- 101,52 χιλιοστά πλάτος
- 53,3 χιλιοστά μήκος
- Βάρος 37 γραμμάρια [8]



Εικόνα 30: Arduino MEGA 2560 και σχολιασμός των περιφερειακών του

Τα αρχεία σχεδίασης, λίστας εξαρτημάτων και κυκλώματος πλακέτας PCB βρίσκονται στην ιστοσελίδα του κατασκευαστή και είναι διαθέσιμα για λήψη στον σύνδεσμο <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

#### 4.4 Τροφοδοσία του Arduino MEGA

Το Arduino MEGA μπορεί να τροφοδοτηθεί είτε από τη θύρα USB είτε από εξωτερική πηγή τροφοδοσίας. Η εξωτερική τροφοδοσία μπορεί να προέρχεται είτε από τροφοδοτικό πρίζας είτε από μπαταρία. Η πλακέτα είναι εφοδιασμένη με κυκλικό βύσμα σύνδεσης διαμέτρου 2,1 χιλιοστών. Επίσης για την παροχή εξωτερικής τροφοδοσίας μπορεί να χρησιμοποιηθεί η ακίδα Vin. Σε περίπτωση που η τάση τροφοδοσίας δεν είναι αρκετή, μικρότερη από 5V, η πλακέτα μπορεί να αρχίσει να είναι ασταθής. Από την άλλη αν η πλακέτα τροφοδοτηθεί με περισσότερα από 12V το ρεύμα θα υπερθερμάνει τους σταθεροποιητές της με συνέπεια να τους καταστρέψει. [8]

#### 4.5 Μνήμη Arduino MEGA

Από πλευράς μνήμης ο μικροελεγκτής ATmega2560 εμπεριέχει όπως αναφέραμε και παραπάνω 256KB flash μνήμης για την αποθήκευση του κώδικα 8 εκ των οποίων εμπεριέχουν το bootloader, 8KB μνήμης SRAM και 4KB μνήμης EEPROM.

#### 4.6 Ακίδες εισόδου – εξόδου Arduino MEGA

Η πλακέτα Arduino MEGA αποτελείται από 54 ψηφιακές εισόδους-εξόδους οι οποίες μπορούν να δώσουν ή να δεχτούν μέχρι 40mA φορτίου ρεύματος. Κάποιες από αυτές τις ακίδες παρέχουν και επιπρόσθετες λειτουργίες όπως βλέπουμε παρακάτω

- 4 κανάλια σειριακής ανταλλαγής δεδομένων RX-TX στις ακίδες 0-1,19-18,17-16,15-14
- 6 ακίδες εξωτερικών διακοπών στους αριθμούς ακίδων 2,3,18,19,20,21
- 4 εξόδους σημάτων Pulse With Modulation (PWM) στις ακίδες 2,13,44,46
- Επικοινωνία μέσω SPI στις ακίδες 50,51,52,53
- Επικοινωνία μέσω I<sup>2</sup>C στις ακίδες 20,21

Η πλακέτα επίσης χρησιμοποιεί και 16 αναλογικές εισόδους-εξόδους των 10-Bit με ανώτατη τιμή το 1024 έχοντας ως κατώτατο όριο σύγκρισης τη γη και ως ανώτατο τα

5V το οποίο μπορεί να αλλάξει με την χρήση της ακίδας Analog Reference (AREF). [8]

#### **4.7 Επικοινωνία με το Arduino MEGA**

Το Arduino MEGA έχει διάφορες τεχνολογίες έτσι ώστε να μπορεί να επικοινωνήσει με άλλες πλακέτες Arduino, άλλους μικροελεγκτές ή κάποιον υπολογιστή. Από μόνος του ο μικροελεγκτής ATmega 2560 διαθέτει 4 κανάλια σειριακής TTL επικοινωνίας. Το USB τσιπ ATmega16AU2 που βρίσκεται πάνω στην πλακέτα αναλαμβάνει να πάρει αυτά τα σειριακά μηνύματα από τις ακίδες 0 και 1 του Arduino και να τα μεταδώσει στον υπολογιστή δημιουργώντας παράλληλα ένα κανάλι USB επικοινωνίας που μπορεί να χρησιμοποιήσει ο χρήστης μέσω του Arduino IDE για να ανταλλάξει μηνύματα, μέσω του σειριακού παράθυρου (Serial Monitor), με τον μικροελεγκτή. [8]

#### **4.8 Προστασία θυρών USB**

Το Arduino MEGA είναι εφοδιασμένο με ασφάλεια έτσι ώστε να προστατέψει τις θύρες USB του υπολογιστή από βραχυκύκλωμα ή υπερβολική τάση. Παρότι όλοι οι υπολογιστές έχουν υλοποιημένα τέτοια κυκλώματα προστασίας, η ασφάλεια προσφέρει άλλο ένα επιπλέον επίπεδο ασφάλειας. Έτσι αν περάσουν πάνω από 500mA ηλεκτρικού φορτίου από την πλακέτα και επιχειρήσουν να πάνε στην θύρα USB η ασφάλεια θα ενεργοποιηθεί προστατεύοντάς την. [8]

#### **4.9 Προγραμματισμός του Arduino MEGA 2560**

Το Arduino MEGA μπορεί να προγραμματιστεί όπως όλες οι πλακέτες Arduino μέσα από το πρόγραμμα ανάπτυξης εφαρμογών Arduino IDE. Ο μικροελεγκτής ATMEGA 2560 περιέχει ένα πρόγραμμα το οποίο επιτρέπει τον προγραμματισμό του χωρίς τη χρήση εξωτερικού προγραμματιστή. Αυτό το πρόγραμμα ονομάζεται Bootloader και αναλαμβάνει να επαναπρογραμματίσει τον μικροελεγκτή χρησιμοποιώντας το πρωτόκολλο επικοινωνίας STK500. Υπό κανονικές συνθήκες για να προγραμματιστεί ένας μικροελεγκτής πρέπει να μπει σε κατάσταση προγραμματισμού όμως στην περίπτωση του Arduino αυτό γίνεται αυτόματα από το πρόγραμμα. Αντί να χρειάζεται το πάτημα του πλήκτρου Reset από τον χρήστη, το Arduino χρησιμοποιεί έναν πυκνωτή 100nF ο οποίος οδηγεί την γραμμή του Reset να ενεργοποιηθεί και όλα αυτά συμβαίνουν με το πάτημα του πλήκτρου "Upload Sketch" μέσα στο Arduino IDE.

Αυτό το αυτόματο RESET μπορεί να απενεργοποιηθεί από τον χρήστη αφαιρώντας το ίχνος που τιτλοφορείται ως RESET-EN πάνω στην πλακέτα. [8]

## ΚΕΦΑΛΑΙΟ 5

### ΑΙΣΘΗΤΗΡΙΑ ΚΑΙ ΠΕΡΙΦΕΡΕΙΑΚΑ

#### 5.1 Αισθητήριο θερμοκρασίας – υγρασίας DHT11

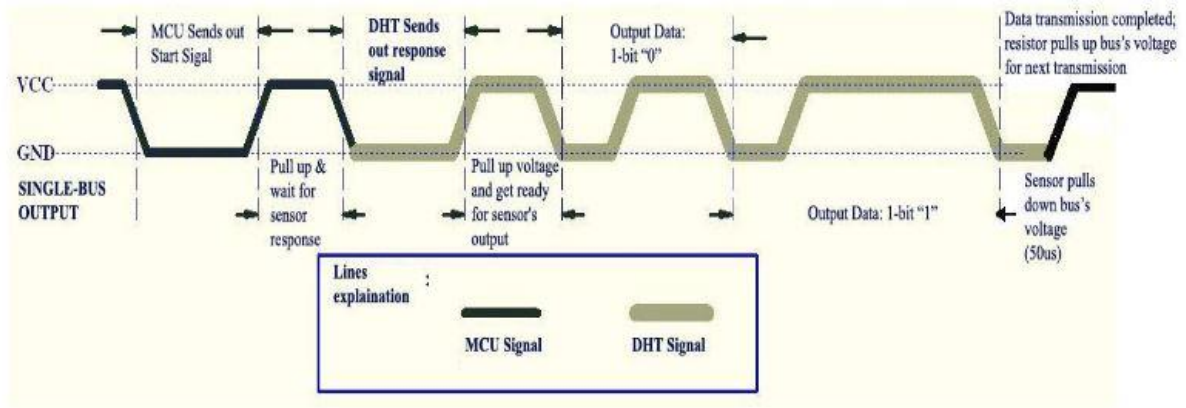
Ο DHT11 αποτελεί ένα ψηφιακό αισθητήριο μέτρησης θερμοκρασίας και υγρασίας το οποίο χρησιμοποιεί υλικό αντίστασης στην υγρασία, θερμίστορ για την μέτρηση της θερμοκρασίας και συνδέεται σε 8-bit μικροελεγκτή για τη σειριακή μεταβίβαση των μετρήσεων.

Η τάση λειτουργίας του είναι από 3-5.5V και πρέπει να δουλεύει ανάμεσα σε αυτήν γιατί αλλιώς μπορεί να προκληθεί αλλοίωση των μετρήσεων του αισθητηρίου μέχρι και καταστροφή του. Σε περίπτωση αλλοίωσης των μετρήσεων το αισθητήριο πρέπει να λειτουργήσει μόνο τους σε δύο διαφορετικές συνθήκες τις οποίες ορίζει ο κατασκευαστής του για να επανέλθει στην κανονική του λειτουργία.

Εξαιρετική προσοχή πρέπει να δοθεί και στην έκθεση του αισθητηρίου σε υψηλές θερμοκρασίες, σε εκτεταμένα επίπεδα υπεριώδους ακτινοβολίας, σε θερμοκρασίες άνω των 40 βαθμών κελσίου και σε επίπεδα υγρασίας μεγαλύτερα του 60% καθώς αυτό μπορεί να μειώσει την ζωή του αισθητηρίου ή να αλλοιώσει την ποιότητα των μετρήσεών του.

Όσον αφορά τον τρόπο επικοινωνίας των εσωτερικών αισθητηρίων με τον μικροελεγκτή αυτή επιτυγχάνεται μέσω πακέτων των 40 bit.

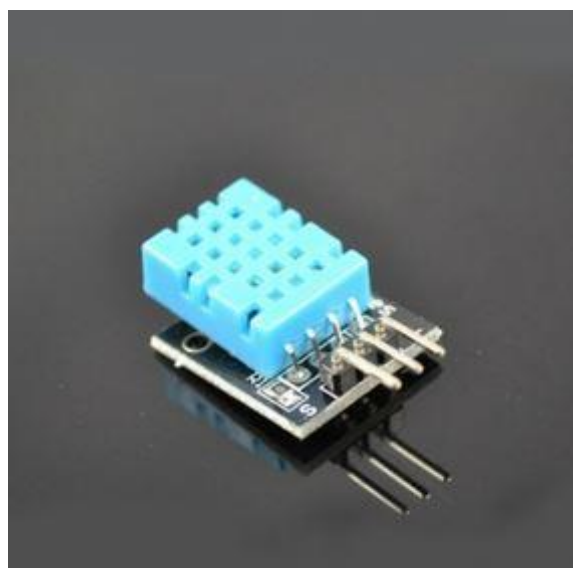
Αρχικά ο μικροελεγκτής στέλνει μια ακολουθία από υψηλά και χαμηλά σήματα συγκεκριμένου χρόνου έτσι ώστε να ενεργοποιήσει τα αισθητήρια τα οποία μέχρι εκείνη την στιγμή βρίσκονται σε λειτουργία εξοικονόμησης ενέργειας. Μόλις αυτό συμβεί ο αισθητήρας απαντάει πίσω στον μικροελεγκτή με ένα μήνυμα το οποίο του λέει να ετοιμαστεί να αποδεχτεί τις πληροφορίες που θα του αποσταλούν. Έπειτα ο αισθητήρας αποστέλλει με την σειρά, 8 bit που περιέχουν την ακέραια τιμή της υγρασίας, 8 bit με τη δεκαδική τιμή της υγρασίας, 8 bit με την ακέραια τιμή της θερμοκρασίας, 8 bit με τη δεκαδική τιμή της θερμοκρασίας και 8 bit τα οποία έχουν ένα μήνυμα που χρησιμοποιείται για τον έλεγχο από τυχόν αλλοίωση των δεδομένων που λαμβάνονται.



Εικόνα 31: Λειτουργία DHT11

	Conditions	Minimum	Typical	Maximum
Power Supply	DC	3V	5V	5.5V
Current Supply	Measuring	0.5mA		2.5mA
	Average	0.2mA		1mA
	Standby	100uA		150uA
Sampling period	Second	1		

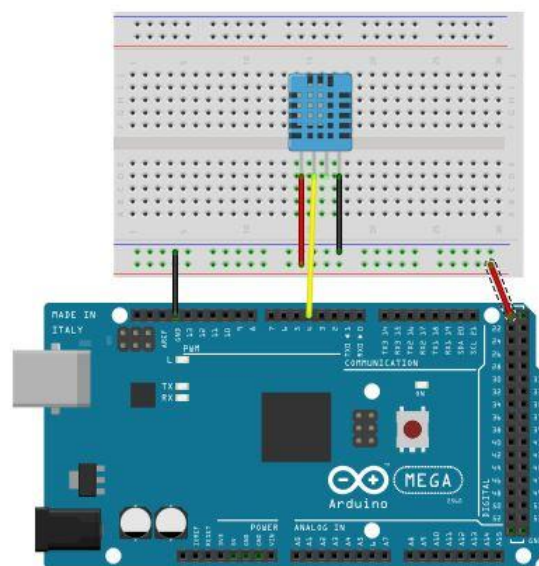
Εικόνα 32: Τάση λειτουργίας και κατανάλωση του DHT11



Εικόνα 33: DHT11

Για να καταγράψουμε τις τιμές του αισθητηρίου θα χρησιμοποιήσουμε σε συνδυασμό με το Arduino IDE την βιβλιοθήκη με τίτλο “dht.h” την οποία μπορούμε να κατεβάσουμε εδώ <https://github.com/RobTillaart/Arduino/tree/master/libraries/DHTlib>

Όσον αφορά την σύνδεση του αισθητηρίου αυτό δεν θα είναι δύσκολο λόγω του ότι το αισθητήριο το έχουμε σε μορφή έτοιμης πλακέτας (module) και θα ακολουθήσουμε τις οδηγίες που είναι τυπωμένες πάνω του τοποθετώντας την ακίδα 1 στα 5V, την ακίδα 2 σε μία από τις ψηφιακές εισόδους του Arduino και την ακίδα 3 στη γείωση όπως φαίνεται και στην Εικόνα 34.



Εικόνα 34: Συνδεσμολογία DHT11 με το Arduino MEGA

## 5.2 MQ2 - Αισθητήριο μέτρησης επικίνδυνων αερίων

Το MQ2 αποτελεί ένα αναλογικό αισθητήριο μέτρησης πολλών επικίνδυνων αερίων όπως είναι το υγραέριο, το βουτάνιο, το μεθάνιο, το οινόπνευμα, το υδρογόνο και ο καπνός. [10] Το MQ2 χρησιμοποιείται τόσο σε συστήματα πυρασφάλειας όσο και σε συστήματα θέρμανσης όπου η διαρροή για παράδειγμα κάποιου εύφλεκτου υλικού όπως είναι το φυσικό αέριο μπορεί να δημιουργήσει τεράστιο κίνδυνο αν δεν προβλεφθεί εγκαίρως. Ο αισθητήρας αποτελείται από δύο μέρη, το ασάλινο δίχτυ και το δαχτυλίδι. Το δαχτυλίδι ενώνεται απευθείας με το ηλεκτρικό φορτίο με σκοπό να ζεσταθεί. Αφού ζεσταθεί αναλαμβάνει να ιονίσει το εισερχόμενο αέριο.[11] Έτσι το



αισθητήριο κοινοποιεί την αντίσταση η οποία δημιουργείται λόγω της διέλευσης του αερίου, τιμή την οποία μπορούμε πολύ απλά μέσω αναλογικής ανάγνωσης να διαβάσουμε με την βοήθεια του μικροελεγκτή και να την εκτιμήσουμε.

Το module που θα χρησιμοποιήσουμε είναι εφοδιασμένο με ένα ολοκληρωμένο που λειτουργεί ως συγκριτής και μία μεταβλητή αντίσταση (ποτενσιόμετρο) έτσι ώστε να μπορεί με την κατάλληλη ρύθμιση να ενεργοποιήσει μία ψηφιακή συσκευή έως 5V χωρίς την χρήση μικροελεγκτή, λειτουργία την οποία δεν θα χρησιμοποιήσουμε.

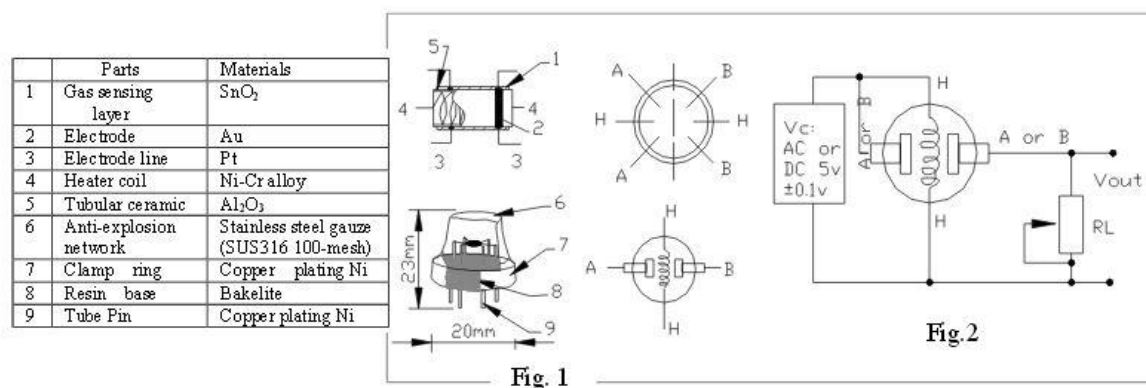
Στις παρακάτω εικόνες θα μπορέσουμε να δούμε κάποια από τα χαρακτηριστικά λειτουργίας του αισθητηρίου καθώς και ένα γράφημα μετρήσεων διαφόρων αερίων.

A. Standard work condition			
Symbol	Parameter name	Technical condition	Remarks
$V_c$	Circuit voltage	$5V \pm 0.1$	AC OR DC
$V_H$	Heating voltage	$5V \pm 0.1$	AC OR DC
$R_L$	Load resistance	can adjust	
$R_H$	Heater resistance	$33 \Omega \pm 5\%$	Room Tem
$P_H$	Heating consumption	less than 800mw	

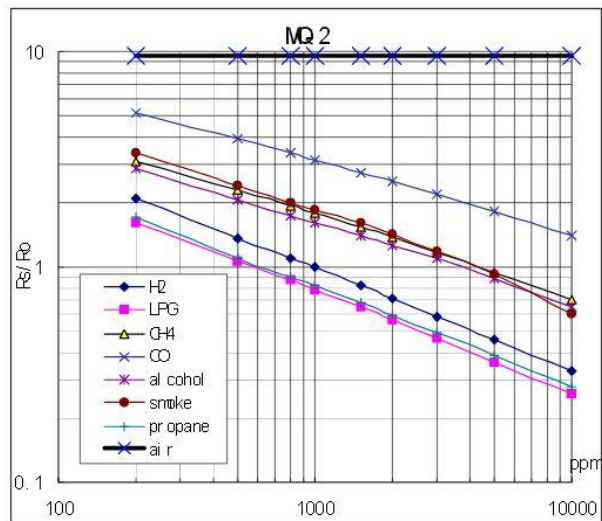
  

B. Environment condition			
Symbol	Parameter name	Technical condition	Remarks
$T_{ao}$	Using Tem	$-20^\circ C - 50^\circ C$	
$T_{as}$	Storage Tem	$-20^\circ C - 70^\circ C$	
$R_H$	Related humidity	less than 95%Rh	
$O_2$	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

Εικόνα 35: Χαρακτηριστικά λειτουργίας MQ2



Εικόνα 36: Συνδεσμολογία αισθητηρίου MQ2

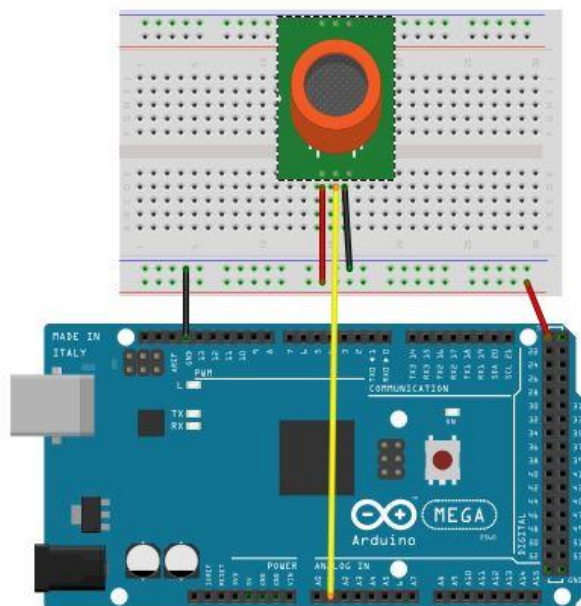


Εικόνα 37: Μετρήσεις διαφορετικών αερίων



Εικόνα 38: MQ2 Module

Στην εικόνα 39 μπορούμε να δούμε τον τρόπο συνδεσμολογίας του αισθητηρίου MQ2 με την πλακέτα Arduino MEGA.



Εικόνα 39: Τρόπος συνδεσμολογίας MQ2 Module

### 5.3 Αισθητήριο εντοπισμού υπέρυθρης ακτινοβολίας (Passive Infrared Sensor)

Ο PIR λειτουργεί με διαφορετικό τρόπο από τα υπόλοιπα αισθητήρια. Στηρίζεται στο φαινόμενο ότι κάθε αντικείμενο μεταδίδει ένα ποσοστό θερμότητας με την μορφή υπέρυθρης ακτινοβολίας.

Ο PIR αποτελείται από δύο μέρη τα οποία είναι κατασκευασμένα από πυροηλεκτρικά υλικά. Το αισθητήριο συνήθως περνάει μία περίοδο σταθεροποίησης κατά την οποία τα κομμάτια του αισθητηρίου που βρίσκονται συνδεδεμένα σε έναν διαφορικό ενισχυτή καταφέρνουν να απομονώσουν οποιοδήποτε αντικείμενο μπορεί να προκαλέσει λάθος συναγερμό. Με αυτόν τον τρόπο μετά από ένα σύντομο χρονικό διάστημα το αισθητήριο μπορεί να κρίνει την διαφορά μεταξύ των αντικειμένων ενός δωματίου με την ύπαρξη κάποιου ανθρώπου ή ζώου στο δωμάτιο.

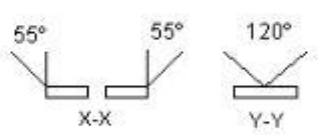
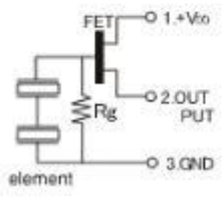
Τα εξαρτήματα του αισθητηρίου καλύπτονται από ένα μεταλλικό και θερμικά σφραγισμένο αντικείμενο το οποίο εξασφαλίζει την προστασία των αισθητηρίων από λάθος μετρήσεις που μπορούν να προκληθούν από θόρυβο, θερμοκρασία ή υγρασία. Το αντικείμενο αυτό στο κέντρο του είναι εξοπλισμένο με ένα παράθυρο κατασκευασμένο από υλικό το οποίο επιτρέπει την πρόσβαση της υπέρυθρης ακτινοβολίας. Για να επιτραπεί στο αισθητήριο να μετρήσει σε μεγαλύτερες περιοχές το εφοδιάζουν με ένα πλαστικό κάλυμμα το οποίο είναι χωρισμένο σε ζώνες επιτρέποντάς του την ανίχνευση ακτινοβολίας σε σχήμα κώνου και όχι ευθύγραμμο.

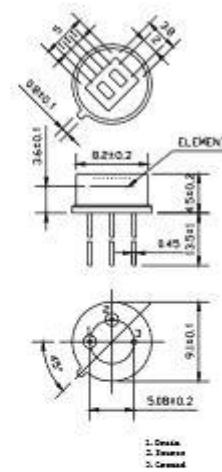
Το αισθητήριο κυκλοφορεί σε μορφή έτοιμης πλακέτας (module) και είναι εφοδιασμένο με τον αποδέκτη των σημάτων που προέρχονται από τα κομμάτια του αισθητηρίου, το ολοκληρωμένο BISS0001. Το BISS0001 είναι αρμόδιο για την αποδοχή των σημάτων και την απόφαση για το αν θα σημάνει ότι εντόπισε ένα αντικείμενο. Συνήθως τα modules είναι εφοδιασμένα με δύο ποτενσιόμετρα επιτρέποντας την ρύθμιση της ευαισθησίας και τον χρόνο επανεκπομπής σήματος συναγερμού. Στα νεότερα modules υπάρχει επιλογή αν το τσιπ θα σημάνει συναγερμό μόλις εντοπίσει κάποιο αντικείμενο και καθ' όλη την διάρκεια που το εντοπίζει ή αν θα σημάνει συναγερμό μόνο την πρώτη φορά που το εντοπίζει. [13]

Ο PIR χρησιμοποιείται σε συστήματα ασφαλείας καθώς και σε συστήματα αυτόματου φωτισμού κ.α.

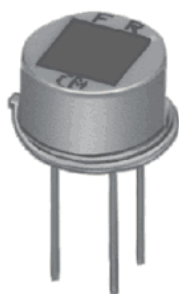
Η χρήση του αισθητηρίου με μικροελεγκτή είναι εύκολη καθώς λόγω του ότι το αισθητήριο είναι αναλογικό δεν χρειάζεται κάποια εξειδικευμένη μέθοδος για να καταγράψουμε την μέτρησή του. Για να πάρουμε την μέτρησή του θα χρησιμοποιήσουμε αναλογική ανάγνωση της τιμής του.

Παρακάτω παρουσιάζονται τα χαρακτηριστικά λειτουργίας του καθώς και διάφορες εικόνες συνδεσμολογίας του.

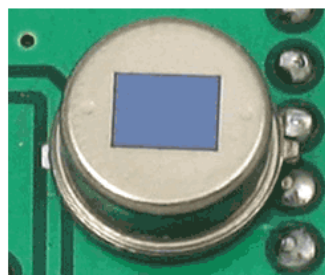
Recommended Model	D203B
Encapsulation Type	TO-5
IR Receiving Electrode	2×1mm, 2 elements
Window Size	5×3.8mm
Spectral Response	5-14μm
Transmittance	≥75%
Signal Output [Vp-p]	≥3500mV
Sensitivity	≥3300V/W
Detectivity (D*)	≥1.4 ×10 <sup>8</sup> cmHz <sup>1/2</sup> /W
Noise[Vp-p]	<70mV
Output Balance	<10%
Offset Voltage	0.3-1.2V
Supply Voltage	3-15V
Operating Temp.	-30-70°C
Storage Temp.	-40-80°C
Field of View Equivalent Circuit	
Equivalent Circuit	



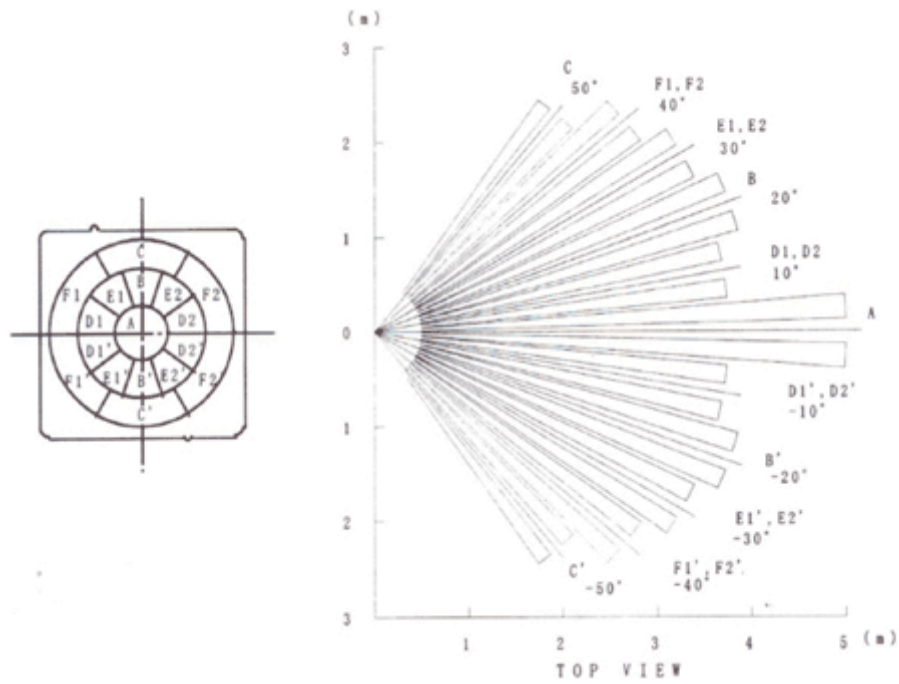
Εικόνα 40: Φυσικά χαρακτηριστικά, χαρακτηριστικά λειτουργίας και δομή PIR



Εικόνα 41: Μεταλλικό κάλυμμα αισθητήριου



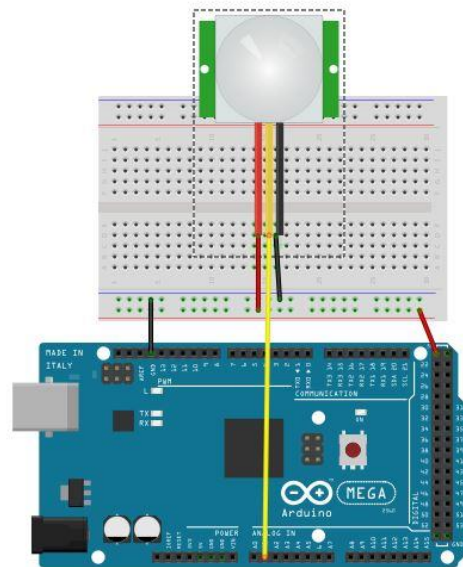
Εικόνα 42: Πλαστική επικάλυψη



Εικόνα 43: Εύρος κάλυψης με την χρήση της πλαστικής επικάλυψης



Εικόνα 44: PIR Sensor Module



Εικόνα 45: Συνδεσμολογία PIR με Arduino

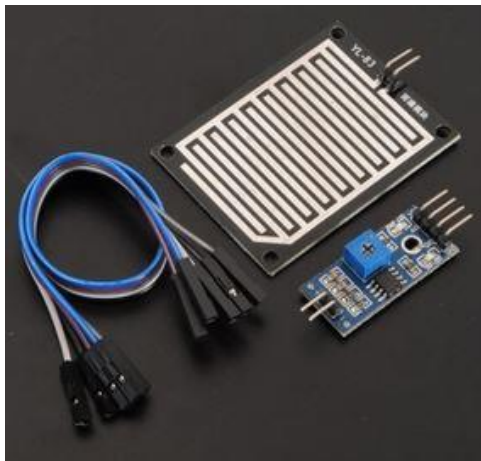
#### 5.4 Αισθητήριο βροχής – νερού

Το αισθητήριο νερού αποτελείται από μία μεταλλική πλάκα η οποία περιέχει αγώγιμους μεταλλικούς διαδρόμους. Σε περίπτωση επαφής νερού με αυτήν την επιφάνεια λόγω της αγωγιμότητας του ενώνει τους διαδρόμους της επιφάνειας μεταξύ τους μειώνοντας την αντίσταση του ρεύματος που διαρρέει αυτήν την επιφάνεια. Η

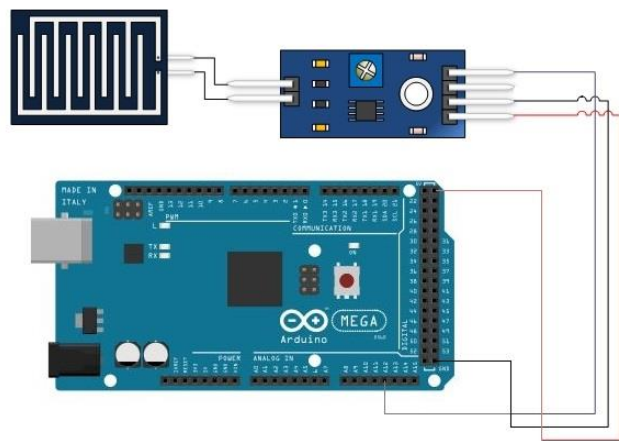
επιφάνεια αυτή συνδέεται με ένα απλό κύκλωμα συγκριτή και ένα ποτενσιόμετρο σε μορφή έτοιμης πλακέτας (module) επιτρέποντας την ενεργοποίηση κάποιας συσκευής σε περίπτωση που η τιμή του αισθητηρίου ξεπεράσει την τιμή που έχουμε ορίσει, χωρίς να είναι απαραίτητη η ύπαρξη μικροελεγκτή. Η τιμή αυτή ορίζεται μέσω του ποτενσιόμετρου.

Το αισθητήριο είναι αναλογικό, πράγμα που σημαίνει ότι η τιμή του θα καταγραφεί από τον μικροελεγκτή χωρίς την ανάγκη ύπαρξης κάποιας βιβλιοθήκης αλλά με αναλογική ανάγνωση.

Παρακάτω θα δούμε μια εικόνα του αισθητηρίου καθώς και τον τρόπο συνδεσμολογίας του με την πλακέτα εφαρμογών Arduino Mega.



Εικόνα 46: Αισθητήριο νερού (module)



Εικόνα 47: Συνδεσμολογία αισθητηρίου με Arduino

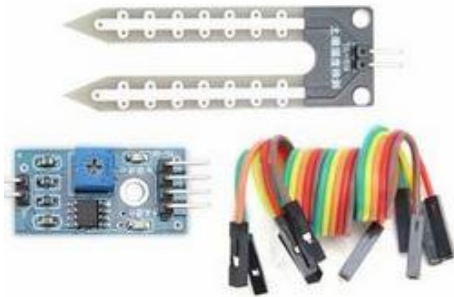
## 5.5 Αισθητήριο Μέτρησης Υγρασίας Εδάφους

Παρόμοιο με το παραπάνω αισθητήριο είναι και το αισθητήριο μέτρησης υγρασίας του χώματος. Το αισθητήριο αποτελείται από ένα πλαστικό αντικείμενο, σε σχήμα πιρουνιού, το οποίο είναι εξοπλισμένο με αγωγούς επιτρέποντας με τον ίδιο τρόπο που περιγράψαμε και πρωτύτερα στο νερό να δημιουργήσει αγωγιμότητα μεταξύ τους. Το σχήμα του διευκολύνει την τοποθέτησή του στο έδαφος. Το αισθητήριο συνοδεύεται με ένα ολοκληρωμένο κύκλωμα συγκριτή και ένα ποτενσιόμετρο διευκολύνοντας την ενεργοποίηση κάποιας συσκευής όπως ένα σύστημα ποτίσματος χωρίς την χρήση κάποιας μικροελεγκτή.

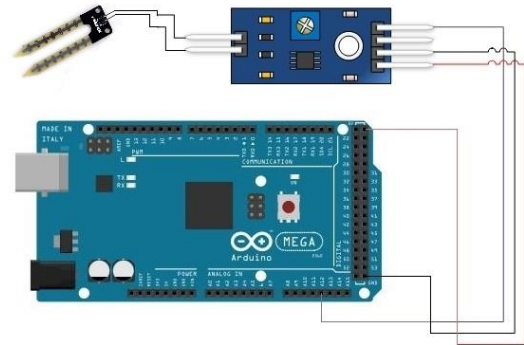
Η ανάγνωση της τιμής του γίνεται με αναλογική ανάγνωση μέσω μικροελεγκτή από την στιγμή που αποτελεί αναλογικό αισθητήριο.



Στις παρακάτω εικόνες βλέπουμε το αισθητήριο καθώς και τον τρόπο σύνδεσής του με την πλακέτα Arduino Mega.



Εικόνα 48: Αισθητήριο μέτρησης υγρασίας χώματος



Εικόνα 49: Συνδεσμολογία αισθητηρίου

## 5.6 Αισθητήριο Μέτρησης Φωτεινότητας

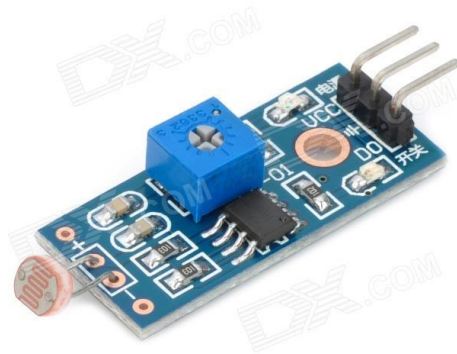
Το αισθητήριο μέτρησης φωτεινότητας είναι πρακτικά μία φωτοαντίσταση (LDR) η οποία αποτελείται από ημιαγώγιμο υλικό υψηλής αντίστασης. Καθώς το φως την διαπέρνει αυτή μειώνει την αντίστασή της. Η φωτοαντίσταση πρακτικά αποτελεί ηλεκτρονικό εξάρτημα με πολύ μικρότερη ακρίβεια από άλλα εξαρτήματα όπως είναι η φωτοδίοδος ή το φωτοτρανζίστορ. [14]

Η φωτοαντίσταση βρίσκεται σε πολλές εφαρμογές όπου πρέπει να γίνεται αυτόματη προσαρμογή κάποιας παραμέτρου σε περίπτωση εντοπισμού χαμηλού ή υψηλού ποσοστού φωτεινότητας όπως είναι για παράδειγμα η αυτόματη ρύθμιση φωτεινότητας μιας οθόνης ανάλογα με την ύπαρξη φωτός σε ένα δωμάτιο κλπ.

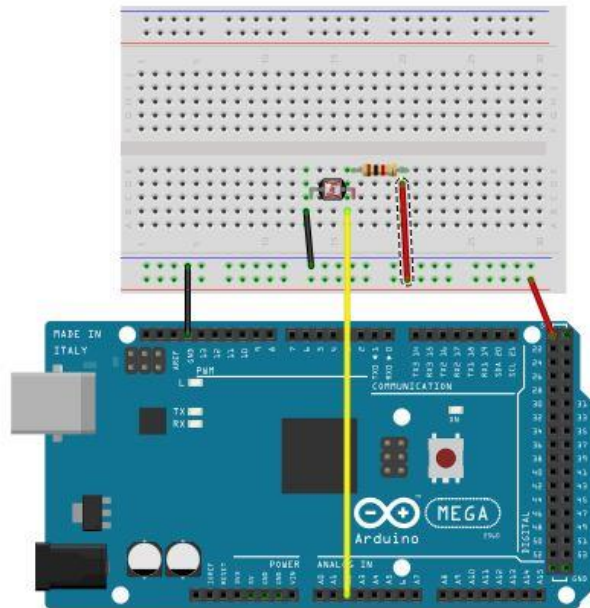
Το αισθητήριο μέτρησης φωτεινότητας έχει την μορφή έτοιμης πλακέτας (module) και εμπεριέχει όπως και τα παραπάνω αισθητήρια ένα ποτενσιόμετρο και ένα ολοκληρωμένο που λειτουργεί ως συγκριτής σε περίπτωση που θέλουμε να ελέγξουμε μία συσκευή χωρίς την ύπαρξη μικροελεγκτή.

Η τιμή αυτού του αισθητηρίου είναι αναλογική και θα την καταγράψουμε χρησιμοποιώντας αναλογική ανάγνωση αφότου την συνδέσουμε με την πλακέτα εφαρμογών Arduino.

Παρακάτω θα δούμε μία φωτογραφία του αισθητηρίου καθώς και τον τρόπο συνδεσμολογίας του με το Arduino MEGA.



Εικόνα 50: Φωτοαντίσταση (Module)



Εικόνα 51: Σύνδεση φωτοαντίστασης με Arduino

## 5.7 Αισθητήριο Αφής

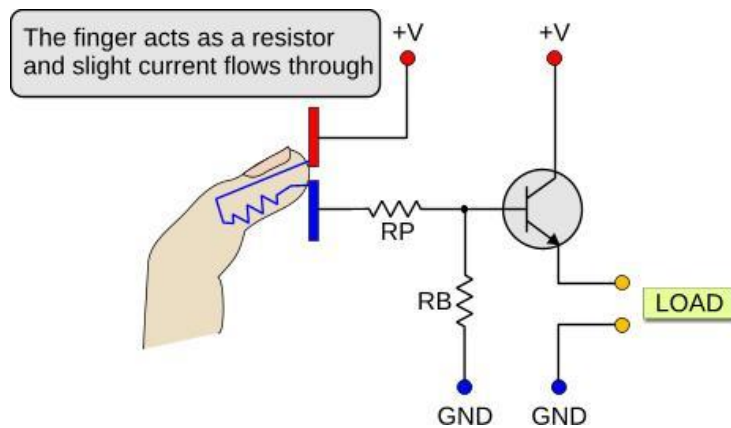
Το αισθητήριο αφής είναι ένας διακόπτης ο οποίος ενεργοποιείται όταν έρθει σε επαφή με το ανθρώπινο δάκτυλο. Το αισθητήριο εκμεταλλεύεται το γεγονός ότι η ανθρώπινη επιδερμίδα περιέχει ένα μεγάλο ποσοστό νερού και άλατος πράγμα που την κάνει αγώγιμη. Έτσι με το πάτημα το διακόπτη το δάκτυλό μας λειτουργεί ως αγωγός μεταξύ της τάσης και της γης κλείνοντας το κύκλωμα. Το αισθητήριο είναι εξοπλισμένο με ένα τρανζίστορ το οποίο αναλαμβάνει μετά να δημιουργήσει το φαινόμενο ύπαρξης του διακόπτη. [15]

Το αισθητήριο αφής χρησιμοποιείται στις περισσότερες νέες εφαρμογές αντικαθιστώντας τον παλιό κλασσικό διακόπτη.

Το αισθητήριο στέλνει στον μικροελεγκτή, μέσω ψηφιακής ανάγνωσης τιμής, την τιμή '0' όταν δεν είναι πατημένο και την τιμή '1' όταν είναι.

Στην εικόνα 52 φαίνεται ο τρόπος λειτουργίας του.





Εικόνα 52: Τρόπος λειτουργίας αισθητηρίου



Εικόνα 53: Αισθητήριο Αφής

## 5.8 Οθόνη TFT 3.2"

Παρακάτω έχουμε την οθόνη 3.2" που χρησιμοποιούμε στην κατασκευή μας ως την τηλεόραση του σπιτιού. Η οθόνη έρχεται σε μορφή έτοιμης πλακέτας (module) η οποία είναι εξοπλισμένη με υποδοχή κάρτας μνήμης SD για την αποθήκευση περιεχομένου και την προβολή του μέσα από αυτήν.

Για την οδήγησή της χρησιμοποιήσαμε την βιβλιοθήκη "UTFT.h" η οποία είναι διαθέσιμη για λήψη στο διαδίκτυο και με την χρήση της μπορεί να οδηγηθεί μία μεγάλη γκάμα από οθόνες TFT. Η βιβλιοθήκη έχει τις παρακάτω επιλογές:

- Αλλαγή χρώματος παρασκηνίου (background color)
- Αλλαγή χρώματος γραμματοσειράς (font color)
- Εισαγωγή εικόνων bitmap
- Επιλογή μεγέθους κειμένου (font size)
- Επιλογή εύκολης ενεργοποίησης / απενεργοποίησης
- Τοποθέτηση αντικειμένων μέσω συντεταγμένων X και Y
- Λειτουργία μεγέθυνσης κ.α.

Εδώ πρέπει να τονίσουμε ότι στην περίπτωση που θέλουμε να εμφανίσουμε κάποιο γραφικό στοιχείο όπως είναι για παράδειγμα μια εικόνα θα πρέπει πρώτα να μετατρέψουμε την εικόνα σε μορφή bitmap και έπειτα χρησιμοποιώντας κάποιο πρόγραμμα να αναλύσουμε αυτήν την εικόνα σε χρωματικούς κωδικούς της κάθε ψηφίδας της με σκοπό να εισάγουμε στο πρόγραμμα έναν πίνακα με τους κωδικούς αυτούς. Επίσης σε περίπτωση που ενεργοποιήσουμε την οθόνη και έπειτα προσπαθήσουμε να την απενεργοποιήσουμε ο οπίσθιος φωτισμός της δεν

απενεργοποιείται πράγμα που σημαίνει ότι πρέπει να συνδεθεί ξεχωριστά σε ψηφιακή ακίδα του μικροελεγκτή και να ελεγχθεί χειροκίνητα.

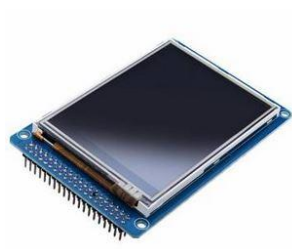
Τάση τροφοδοσίας	5V
Τάση τροφοδοσίας οπίσθιου φωτισμού	3.3V
Ανάλυση	240p X 320p
Μέγεθος	3.2 inch
Διαστάσεις	9.4 x 6.1cm(L x W)

Πίνακας 2: Τεχνικά χαρακτηριστικά οθόνης TFT 3.2"

Ο τρόπος συνδεσμολογίας της είναι απλός καθώς δίπλα σε κάθε ακίδα της υπάρχει η λειτουργία της. Ο τρόπος συνδεσμολογίας που ακολουθήσαμε φαίνεται στον πίνακα 3.

<u>Αριθμός Ακίδας Οθόνης</u>	<u>Αριθμός Ακίδας Arduino</u>
DB0 - DB7	D37 - D30
DB8 - DB15	D22 - D29
RESET	D41
CS	D40
RD	D42
WR	D39
RS	D38
VCC	5V
GND	GND

Πίνακας 3: Συνδεσμολογία οθόνης TFT 3.2" με Arduino MEGA



Εικόνα 54: Μπροστά Όψη Οθόνης



Εικόνα 55: Πίσω Όψη Οθόνης

## 5.9 Οθόνη OLED 0.9”

Παρακάτω θα δούμε την οθόνη την οποία θα χρησιμοποιήσουμε ως ενημερωτική πηγή για τις συνθήκες θερμοκρασίας που επικρατούν στο χώρο του σπιτιού και θα τοποθετηθεί στα περισσότερα δωμάτια.

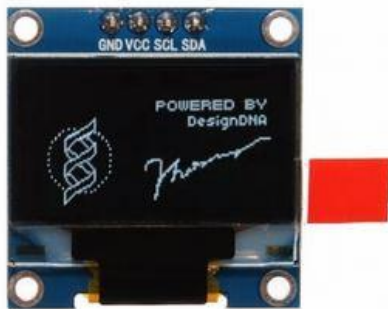
Η οθόνη αυτή αποτελεί μια πάρα πολύ μικρή, μόλις μίας ίντσας, συσκευή η οποία χρησιμοποιεί το πρωτόκολλο επικοινωνίας I<sup>2</sup>C για να συνομιλήσει με τον μικροελεγκτή. Για την οδήγησή της θα χρησιμοποιήσουμε τις βιβλιοθήκες της εταιρίας Adafruit με ονομασίες “Adafruit\_GFX” για να δημιουργήσουμε γραφικά και “Adafruit\_SSD1306” για να οδηγήσουμε την συσκευή. Επίσης για να ενεργοποιήσουμε την δυνατότητα χρήσης του πρωτοκόλλου I<sup>2</sup>C θα χρησιμοποιήσουμε και την βιβλιοθήκη “Wire.h” η οποία βρίσκεται μέσα στο περιβάλλον του Arduino IDE.

Ο τρόπος συνδεσμολογίας της είναι πολύ απλός μιας και αποτελείται μόνο από τέσσερις ακίδες εκ των οποίων οι δύο αποτελούν την τάση τροφοδοσίας και την γείωση. Τις υπόλοιπες ακίδες θα τις συνδέσουμε με τις ακίδες 20 και 21 του Arduino MEGA αντίστοιχα.

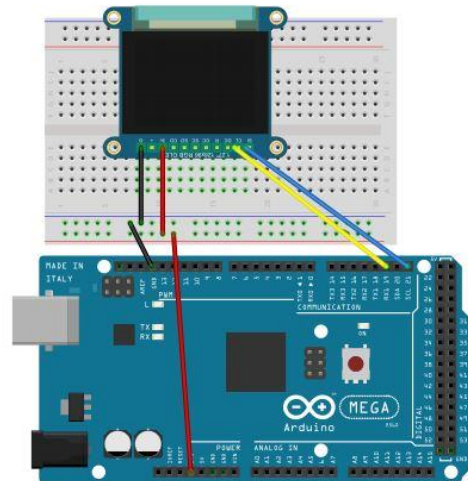
Κάποια από τα τεχνικά χαρακτηριστικά της οθόνης φαίνονται στον πίνακα παρακάτω.

Οδηγός οθόνης	SSD1306
Ανάλυση	128 X 64
Τάση τροφοδοσίας	3.3V – 6V
Διαστάσεις	2.7 x 2.8cm
Μέγεθος	0.96”

Πίνακας 4: Τεχνικά χαρακτηριστικά οθόνης OLED



Εικόνα 56: Οθόνη OLED 0.96”



Εικόνα 57: Σύνδεση οθόνης με Arduino MEGA

### 5.10 Οθόνη LCD 16x2

Η παρακάτω συσκευή αποτελεί μία οθόνη 2 γραμμών και 16 χαρακτήρων την οποία χρησιμοποιούμε στο σύστημα συναγερμού της μακέτας μας και την οδηγούμε με την βοήθεια της βιβλιοθήκης “Liquidcrystal” η οποία εμπεριέχεται στο Arduino IDE. Η οθόνη έχει οπίσθιο μπλε φωτισμό και λειτουργεί με τάση τροφοδοσίας 5V. Υποστηρίζει λειτουργία με ή χωρίς φωτισμό καθώς δίνει την δυνατότητα στον χρήστη για ρύθμιση της αντίθεσής της με την χρήση ενός ποτενσιόμετρου 10KΩ στην ακίδα 3.

Η τιμή της είναι χαμηλή και σε συνδυασμό με τον αριθμό χαρακτήρων που δείχνει μπορεί να χρησιμοποιηθεί σε πάρα πολλές εφαρμογές όπως είναι εκτυπωτές, συστήματα συναγερμών, φούρνοι μικροκυμάτων κλπ.

Κάποια από τα τεχνικά χαρακτηριστικά της φαίνονται στον πίνακα που βρίσκεται παρακάτω.

Τάση τροφοδοσίας	5V
Οδηγός Οθόνης	HD44780
Διαστάσεις	80mm x 36mm x 12mm
Χρώμα οπίσθιου φωτισμού	Μπλε
Περιοχή θέασης	64.5mm x 16mm

Πίνακας 5: Τεχνικά χαρακτηριστικά οθόνης LCD 16x2

Παρακάτω φαίνεται η φωτογραφία του προϊόντος καθώς και ο τρόπος συνδεσμολογίας του με την πλακέτα Arduino Mega.



Εικόνα 58: Μπροστινή όψη οθόνης



Εικόνα 59: Πίσω όψη οθόνης

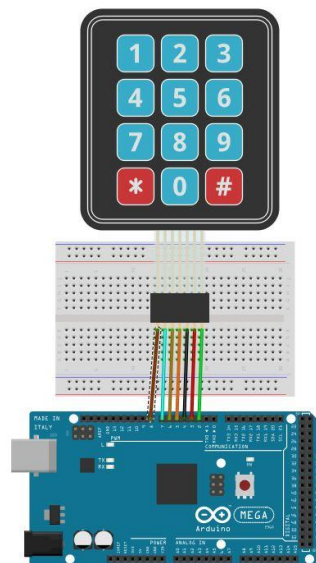
### 5.11 Πληκτρολόγιο 12 πλήκτρων

Το πληκτρολόγιο που θα παρουσιάσουμε παρακάτω αποτελείται από 4 γραμμές και 3 στήλες οι οποίες έχουν τους αριθμούς 0-9 και τα σύμβολα \* και #. Χρησιμοποιεί το φαινόμενο της πολυπλεξίας των πλήκτρων, δηλαδή την ένωση των πλήκτρων ανά γραμμή και στήλη με σκοπό να ελαχιστοποιήσει τον αριθμό των ακίδων που χρειάζεται για να συνδεθεί. Ο τρόπος σύνδεσής του στην πλακέτα εφαρμογών είναι ένα εύκαμπτο βύσμα.

Για την χρήση του σε συνεργασία με την πλακέτα εφαρμογών θα χρησιμοποιήσουμε την βιβλιοθήκη "Keypad.h" που μας παρέχεται μέσα από το περιβάλλον Arduino IDE.



Εικόνα 60: Πληκτρολόγιο 4x3



Εικόνα 61: Σύνδεση πληκτρολογίου με Arduino

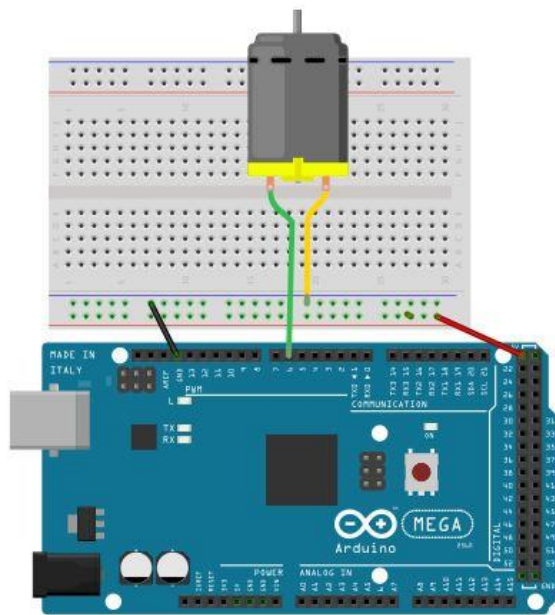
## 5.12 Κινητήρας DC

Χρησιμοποιούμε επίσης στην μακέτα μας τρεις κινητήρες συνεχούς ρεύματος. Τους δύο τους χρησιμοποιούμε για να προσομοιώσουμε το σύστημα κλιματισμού. Ο τρίτος αποτελεί ένα πολύ μικρό κινητήρα συνεχούς με τάση τροφοδοσίας μόλις 2V τον οποίο θα τον χρησιμοποιήσουμε ως ανεμόμετρο. Τους πρώτους δύο με την βοήθεια του μικροελεγκτή μας θα τους χρησιμοποιήσουμε μέσω ψηφιακής ενεργοποίησης / απενεργοποίησης. Στον τρίτο κινητήρα θα χρησιμοποιήσουμε το φαινόμενο της παραγωγής ρεύματος μέσω της κίνησής του από τον άνεμο μετρώντας ως αναλογικό μέγεθος την παραγόμενη λόγω αυτού του φαινομένου τιμή.

Ο τρόπος σύνδεσης των δύο ακίδων ενός κινητήρα συνεχούς καθορίζει και τη φορά κατά την οποία θα κινηθεί. Έτσι μπορούμε να τον συνδέσουμε με όποιο τρόπο θέλουμε είτε συνδέοντας την μία ακίδα του στην τάση και την άλλη στην γείωση είτε το αντίθετο.



Εικόνα 62: Κινητήρας συνεχούς



Εικόνα 63: Σύνδεση του κινητήρα με Arduino



### 5.13 Βηματικός κινητήρας

Ο βηματικός κινητήρας έχει διαφορετικό τρόπο χρήσης και ελέγχου από τον κινητήρα συνεχούς. Ο βηματικός κινητήρας εξασφαλίζει ακρίβεια στην κίνησή του και έτσι είναι ιδανικός προς χρήση μέσα σε εκτυπωτές κ.α. Στη μακέτα μας υπάρχουν δύο βηματικοί κινητήρες, ένας ο οποίος αποτελεί το μοτέρ της τέντας και ένας ο οποίος αποτελεί το μοτέρ της πόρτας του γκαράζ.

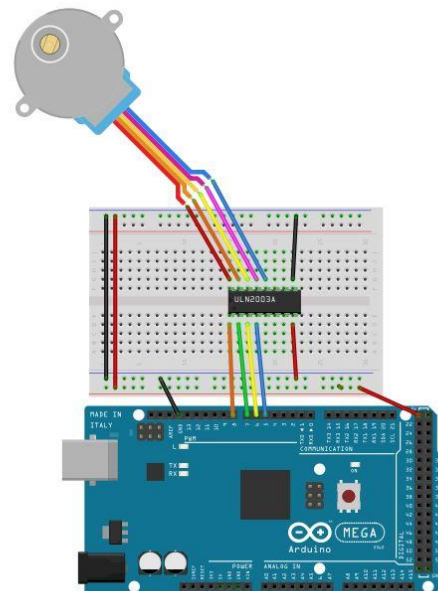
Ο βηματικός κινητήρας που χρησιμοποιούμε είναι μονοπολικός και η οδήγησή του επιτυγχάνεται μέσω του ειδικού ολοκληρωμένου για αυτήν την δουλειά ULN2003 της εταιρίας Fairchild. Ο κινητήρας έχει τάση τροφοδοσίας 5V και το ολοκληρωμένο ULN2003 έχει την δυνατότητα οδήγησης βηματικών κινητήρων από 3-30V. Για την οδήγησή του θα χρησιμοποιήσουμε την βιβλιοθήκη “Stepper.h” που βρίσκεται μέσα στο πρόγραμμα Arduino IDE. Η βιβλιοθήκη, μας επιτρέπει με ευκολία να ελέγξουμε την ταχύτητα, το βήμα και την φορά περιστροφής του κινητήρα.

Ο οδηγός του κινητήρα βρίσκεται στα χέρια μας σε μορφή έτοιμης πλακέτας (module) και είναι εξοπλισμένος με 4 led προβολής κατάστασης τα οποία μας επιτρέπουν να ελέγξουμε την σωστή λειτουργία του.

Παρακάτω φαίνεται ο τρόπος σύνδεσής του στην πλακέτα εφαρμογών Arduino Mega.



Εικόνα 64: Βηματικός κινητήρας



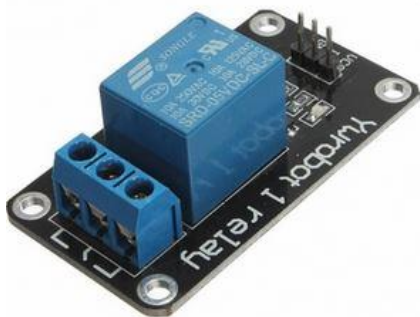
Εικόνα 65: Σύνδεση κινητήρα με Arduino Mega

### 5.14 Ρελές σε μορφή πλακέτας (Relay Module)

Ο ρελές είναι ένα είδος ηλεκτρικού-μηχανικού διακόπτη ο οποίος αποτελείται από έναν ηλεκτρομαγνήτη και επαφές. Ο ρελές είναι ένα πολύ χρήσιμο εξάρτημα το οποίο χρησιμοποιείται μέσα σε μια πληθώρα σημερινών εφαρμογών όπου υπάρχει ανάγκη για ενεργοποίηση/απενεργοποίηση ενός κυκλώματος.

Ο ρελές που χρησιμοποιούμε εμείς είναι σε μορφή έτοιμης πλακέτας (module) και είναι εξοπλισμένος με δίοδο προστασίας, οπτικό απομονωτή (optocoupler) για τον διαχωρισμό της τροφοδοσίας του module από την τροφοδοσία της συσκευής και led ένδειξης λειτουργίας και κατάστασης. Από τη μία πλευρά τροφοδοτείται το module και τοποθετείται το ψηφιακό σήμα ενεργοποίησης/απενεργοποίησης που έρχεται από το Arduino και από την άλλη πλευρά έχει τρεις εισόδους οι οποίες κλειδώνουν με βίδες (terminals). Η μεσαία είσοδος αποτελεί το κοινό σήμα, η δεξιά το σημείο που συνδέεται με την κοινή είσοδο όταν ο ρελές είναι ανενεργός και η αριστερή είσοδος το σημείο που συνδέεται με την μεσαία είσοδο όταν ο ρελές είναι ενεργός.

Το module θα το χρησιμοποιήσουμε τόσο στην κατασκευή μας όσο και στον έλεγχο των πραγματικών συσκευών μιας οικίας μέσω της πλακέτας εφαρμογών Arduino.

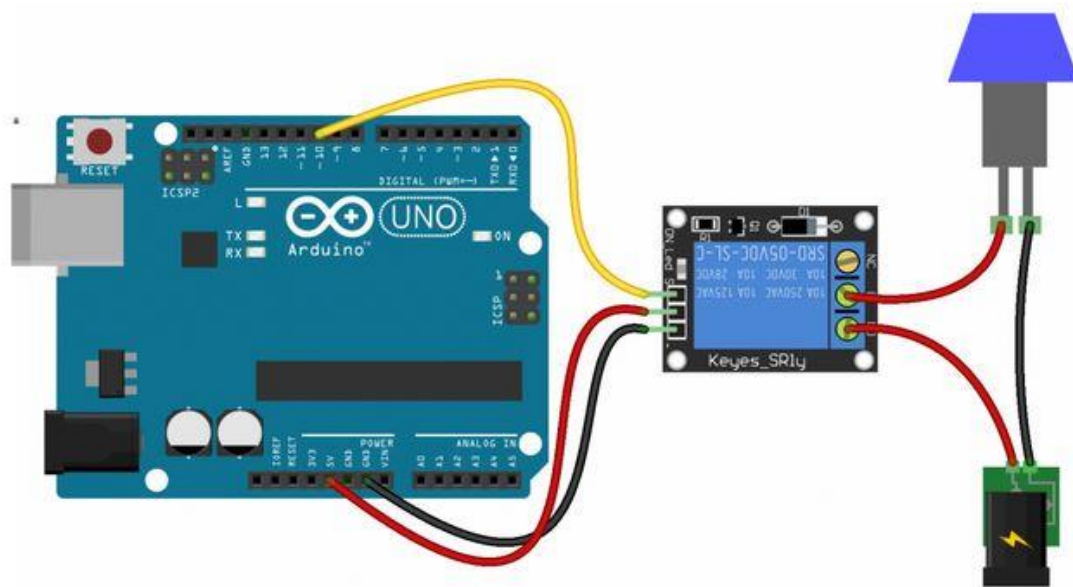


Εικόνα 66: Relay Module



Εικόνα 67: Relay Module 8 καναλιών





Εικόνα 68: Παράδειγμα σύνδεσης ρελέ με Arduino και φωτιστικό

## ΚΕΦΑΛΑΙΟ 6

### ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ARDUINO

#### 6.1 Επιλογή αριθμού πλακετών Arduino

Λόγω της συνεχής αποστολής δεδομένων των αισθητηρίων στην βάση δεδομένων δημιουργείται το φαινόμενο αύξησης εκθετικά του χρόνου του οποίου χρειάζεται η πλακέτα για να κάνει μια πλήρη επανάληψη του προγράμματός της όσο αυξάνεται ο αριθμός των αισθητηρίων. Έτσι αν αποφασίσουμε να έχουμε στην ίδια πλακέτα και τις συσκευές της οικίας αυτό σημαίνει ότι θα έχουμε μεγάλη καθυστέρηση από την στιγμή που θα πραγματοποιήσουμε μια αλλαγή από την ιστοσελίδα μέχρι αυτή η αλλαγή πραγματικά να συμβεί. Για αυτό τον λόγο αποφασίσαμε να χρησιμοποιήσουμε δύο διαφορετικές πλακέτες Arduino, μία για τις συσκευές και μία για τα αισθητήρια, έτσι ώστε ο χρήστης να μπορεί να πραγματοποιεί ταχύτατες αλλαγές στις συσκευές χωρίς να τον ενδιαφέρει ο αριθμός των αισθητηρίων που έχει στο σπίτι του. Παρόλα αυτά ο χρήστης εφόσον θέλει να έχει έναν μικρό αριθμό αισθητηρίων σπίτι του, κάτω από 5, και παράλληλα όσες συσκευές θέλει θα μπορούσε να χρησιμοποιήσει μία πλακέτα Arduino.

#### 6.2 Εισαγωγικά σχόλια ως προς τον πηγαίο κώδικα

Κάθε εφαρμογή Arduino αποτελείται από τρεις χώρους:

- Την λειτουργία void setup() όπου βρίσκεται ο κώδικας που θα εκτελεστεί μόνο μία φορά κατά την έναρξη
- Την λειτουργία void loop() όπου βρίσκεται ο κώδικας που θα εκτελείται ξανά και ξανά καθ' όλη την διάρκεια
- Ο εξωτερικός χώρος που πρακτικά είναι ότι βρίσκεται έξω από τις δύο προαναφερθείσες λειτουργίες όπου μπορεί να βρίσκεται η δήλωση μεταβλητών, βιβλιοθηκών, ορισμών ακόμα και άλλες λειτουργίες (functions) που θα χρησιμοποιηθούν

Ξεκινώντας λοιπόν με την βοήθεια της εντολή include θα ορίσουμε τις βιβλιοθήκες που θα χρησιμοποιηθούν αντίστοιχα στο Arduino συσκευών και στο Arduino αισθητηρίων.

## Arduino No1

```
#include <UTFT.h> //Βιβλιοθήκη οθόνης TFT
#include <avr/pgmspace.h> //Βιβλιοθήκη γενικής χρήσης
#include <Stepper.h> //Βιβλιοθήκη βηματικού κινητήρα
```

## Arduino No2

```
#include <SPI.h> //Βιβλιοθήκη Τεχνολογίας SPI
#include <Wire.h> //Βιβλιοθήκη Τεχνολογίας I2C
#include <Adafruit_GFX.h> //Βιβλιοθήκη οθονών OLED
#include <Adafruit_SSD1306.h> //Βιβλιοθήκη οθονών OLED
#include <dht.h> //Βιβλιοθήκη αισθητηρίων DHT11
#include <LiquidCrystal.h> //Βιβλιοθήκη οθόνης LCD 2X16
#include <Keypad.h> //Βιβλιοθήκη Πληκτρολογίου 4X3
#include <Stepper.h> //Βιβλιοθήκη Βηματικού Μοτέρ
```

Έπειτα με την βοήθεια της εντολής define θα θέσουμε κάποια ονόματα ορισμών να αντιστοιχούν σε μία τιμή. Το κομμάτι κώδικα που ακολουθεί χρησιμοποιείται στο Arduino αισθητηρίων για την δήλωση των ακίδων που βρίσκονται τα αισθητήρια θερμοκρασίας.

```
#define DHTPINLIV 22 //Ακίδα σύνδεσης αισθητηρίου θερμοκρασίας-υγρασίας σαλονιού
#define DHTPINBAT 23 //Ακίδα σύνδεσης αισθητηρίου θερμοκρασίας-υγρασίας τουαλέτας
#define DHTPINBED 24 //Ακίδα σύνδεσης αισθητηρίου θερμοκρασίας-υγρασίας υπνοδωματίου
#define DHTPINKIT 25 //Ακίδα σύνδεσης αισθητηρίου θερμοκρασίας-υγρασίας κουζίνας
#define DHTPINOUT 26 //Ακίδα σύνδεσης αισθητηρίου θερμοκρασίας-υγρασίας εξωτερικού χώρου
#define DHTPINROO 27 //Ακίδα σύνδεσης αισθητηρίου θερμοκρασίας Θερμοσίφωνα
```

Στη συνέχεια θα δηλώσουμε κάποιες μεταβλητές τις οποίες θα χρησιμοποιούμε καθ' όλη την διάρκεια του προγράμματός μας στον εξωτερικό χώρο αντίστοιχα στο Arduino με τις συσκευές και στο Arduino με τα αισθητήρια.

## Arduino No1

```
bool garagestatus = false; //Μεταβλητή αποθήκευσης κατάστασης γκαραζόπορτας
boolean tvstatus = false; //Μεταβλητή αποθήκευσης κατάστασης τηλεόρασης
int FirstTime = 0; //Μεταβλητή γκαραζόπορτας
int LostConnection = 0; //Μεταβλητή αποθήκευσης κατάστασης σύνδεσης της πλακέτας με το πρόγραμμα
extern uint8_t SmallFont[]; //Δήλωση μικρού μεγέθους γραμματοσειράς οθόνης TFT
extern uint8_t BigFont[]; //Δήλωση μεγάλου μεγέθους γραμματοσειράς οθόνης TFT
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
extern unsigned int alt[0xD16]; //Δήλωση πίνακα λογότυπου προς εμφάνιση
στην οθόνη TFT
byte inputByte[] = {0,0,0}; //Μεταβλητή αποθήκευσης μηνυμάτων σειριακής
επικοινωνίας
```

### Arduino No2

```
//Μεταβλητές Συστήματος
Ασφαλείας////////////////////////////////////
bool AlarmStatus=false; //Συναγερμός απενεργοποιημένος
int intruder=0; //Μηδενική Παραβίαση
int datacount=0; //Αριθμός ψηφίων που έχει πληκτρολογήσει ο χρήστης
char Code[5]={'7','7','7','7'}; //Σωστός κωδικός συναγερμού 7777
char tempCode[5]; //Μεταβλητή προσωρινής αποθήκευσης πληκτρολόγησης
const byte ROWS = 4; //4 γραμμές πληκτρολόγιο
const byte COLS = 3; //3 στήλες πληκτρολόγιο
byte rowPins[ROWS] = {8,9,10,11}; //Ακίδες σύνδεσης των τεσσάρων γραμμών
του πληκτρολόγιου
byte colPins[COLS] = {12,14,15}; //Ακίδες σύνδεσης των τριών στηλών του
πληκτρολόγιου

//Σχεδίαση διαθέσιμων ψηφίων του πληκτρολόγιου
char hexaKeys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

//Γενικές
Μεταβλητές////////////////////////////////////
int Thermostat1 = 18; //Μεταβλητή θερμοστάτη πρώτου καλοριφέρ
int Thermostat2 = 18; //Μεταβλητή θερμοστάτη δεύτερου καλοριφέρ
int Thermostat3 = 30; //Μεταβλητή θερμοστάτη πρώτου air condition
int Thermostat4 = 30; //Μεταβλητή θερμοστάτη δεύτερου air condition
int i; //Μεταβλητή εκάστοτε αναλογικού αισθητηρίου προς ανάγνωση
byte PinA[] = {A0,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10}; //Ακίδες που έχουν
συνδεδεμένες αναλογικά αισθητήρια
int TempHum[12]; //Πίνακας αποθήκευσης αποτελεσμάτων θερμοκρασίας και
υγρασίας
int flag = 0; //Μεταβλητή μέτρησης σελίδας για τις οθόνες OLED
unsigned long previousMillis = 0; //Μεταβλητή αποθήκευσης χρόνου
const long interval = 2000; //Χρονοκαθυστέρηση πριν την αλλαγή σελίδας στις
οθόνες OLED
byte inputByte[]={0,0,0}; //Πίνακας αποθήκευσης μηνυμάτων σειριακής
επικοινωνίας

//Μεταβλητές Συστήματος Ρύθμισης
Τέντας////////////////////////////////////
int angle=0; //Προσωρινή Μεταβλητή Υπολογισμού
int lastpos=-6000; //Μεταβλητή αποθήκευσης τελευταίας θέσης
int pos1=0; //Θέση τέντας σε περίπτωση ανέμου
int pos2=-6000; //Θέση τέντας σε περίπτωση βροχής
int pos3=-5000; //Θέση τέντας σε περίπτωση υψηλής φωτεινότητας
int pos4=-3000; //Θέση τέντας σε περίπτωση μέτριας φωτεινότητας
int pos5=-1000; //Θέση τέντας σε περίπτωση χαμηλής φωτεινότητας
int DueToWind=0; //Μεταβλητή αποθήκευσης μετρήσεων ανέμου
int DueToRain=0; //Μεταβλητή αποθήκευσης μετρήσεων βροχής
int DueToLight1=0; //Μεταβλητή αποθήκευσης μετρήσεων υψηλής φωτεινότητας
int DueToLight2=0; //Μεταβλητή αποθήκευσης μετρήσεων μεσαίας φωτεινότητας
```

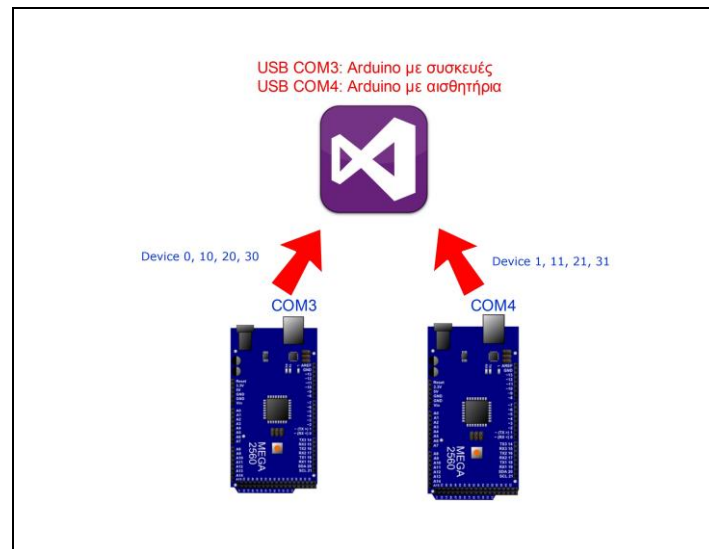
```
int DueToLight3=0; //Μεταβλητή αποθήκευσης μετρήσεων χαμηλής φωτεινότητας
int DueNotToWind=0; //Μεταβλητή αποθήκευσης μετρήσεων μη ύπαρξης ανέμου
int DueNotToRain=0; //Μεταβλητή αποθήκευσης μετρήσεων μη ύπαρξης βροχής
int TentIdle=0; //Τένια απασχολημένη λόγω ανέμου
int TentIdle2=0; //Τένια απασχολημένη λόγω βροχής
unsigned long previousMillis2 = 0; //Μεταβλητή αποθήκευσης χρονικής στιγμής
const long interval2 = 1000; //Έλεγχος καιρικών συνθηκών κάθε 1
δευτερόλεπτο
////////////////////////////////////
////////////////////////////////////
```

### 6.3 Πρωτόκολλο επικοινωνίας Arduino – προγράμματος C#

Από τη στιγμή που χρησιμοποιούμε πλακέτες οι οποίες ανταλλάσσουν με τον υπολογιστή μας πολύ σημαντικές πληροφορίες, μία απλή ανάγνωση των δεδομένων που κινούνται σε μία θύρα USB θα ήταν μεγάλο ρίσκο. Το ρίσκο αυτό προκύπτει από το γεγονός ότι το πρόγραμμα μπορεί να μπερδέψει τις πλακέτες με κάποιο άλλο περιφερειακό το οποίο χρησιμοποιεί την ίδια δομή, μικροελεγκτή και USB τσιπ, και να δεχτεί πληροφορίες από αυτό. Επιπρόσθετα αν κάποιο μήνυμα από ή προς το πρόγραμμα αλλοιωθεί, αυτό θα δημιουργήσει πρόβλημα στις συσκευές-αισθητήρια οι οποίες βρίσκονται τοποθετημένες πάνω στις πλακέτες.

Για αυτόν τον λόγο μετά από έρευνα, χρησιμοποιήσαμε ανοιχτό κώδικα από τις πηγές [16], [17] οι οποίες είναι η ανοιχτή μαθησιακή κοινότητα του Arduino και μία παρουσίαση, στον οποίο αρχικά κάναμε αποσφαλμάτωση και τροποποίηση έτσι ώστε να υποστηρίζει παραπάνω από μία συσκευές Arduino και να τις ξεχωρίζει με βάση την λειτουργία τους π.χ. αν ελέγχουν συσκευές, αν κοινοποιούν τιμές αισθητηρίων και να ανταλλάσσει δεδομένα με αυτές γνωρίζοντας πάντα αν αυτές έχουν χάσει την σύνδεση με το κύριο πρόγραμμα ή όχι.

Το πρωτόκολλο σύζευξης επικοινωνίας λειτουργεί με την εκπομπή ενός μηνύματος αποτελούμενου από μία συμβολοσειρά, Device X και τρεις αριθμούς με βάση το μοτίβο 1X, 2X, 3X. Το κύριο πρόγραμμα περιμένει να εντοπίσει τουλάχιστον μία συσκευή που εκπέμπει αυτό το μήνυμα και εφόσον τη βρει ταυτίζει την USB θύρα επικοινωνίας με τη λειτουργία που έχει π.χ. αν η πλακέτα χειρίζεται συσκευές ή αισθητήρια.



Εικόνα 69: Πρωτόκολλο σύζευξης επικοινωνίας

Για να ενεργοποιήσουμε την σειριακή επικοινωνία μεταξύ του μικροελεγκτή και της θύρας χρησιμοποιούμε την εντολή `Serial.begin(9600)` και για να ελέγξουμε αν υπάρχουν πληροφορίες στην γραμμή χρησιμοποιούμε την εντολή `Serial.available()`. Για να διαβάσουμε μια σειριακή πληροφορία χρησιμοποιούμε την εντολή `Serial.read()` ενώ για να γράψουμε μια σειριακή πληροφορία χρησιμοποιούμε την εντολή `Serial.print()`. Παρακάτω φαίνεται ο πηγαίος κώδικας σε Arduino της αρχικής ρουτίνας `void setup()` και του πρωτοκόλλου σύζευξης επικοινωνίας:

### Arduino 1

```
void setup() {  
  
  myGLCD.InitLCD(); //Σύνδεση οθόνης TFT  
  myStepper.setSpeed(60); //Ορισμός ταχύτητας βηματικού κινητήρα  
  
  Serial.begin(9600); //Έναρξη σειριακής επικοινωνίας  
  
  //Ορισμός λειτουργίας ψηφιακών ακίδων ως εξόδους  
  for (int i=2; i<14; i++)  
  {  
  
    pinMode(i,OUTPUT);  
  
  }  
  
  for(int i=42; i<49; i++)  
  {  
  
    pinMode(i,OUTPUT);  
  
  }  
}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
//Ορισμός των ψηφιακών ακίδων που βρίσκονται οι διακόπτες αφής ως
εισόδους
pinMode(16,INPUT);
pinMode(17,INPUT);
pinMode(18,INPUT);
pinMode(19,INPUT);
pinMode(20,INPUT);
pinMode(21,INPUT);

}

//Λειτουργία σύνδεσης πλακέτας με το πρόγραμμα C#
bool sindesi(){

    boolean sin = false;

    if (Serial.available())
    {
        inputByte[0] = Serial.read();
        delay(10);
        inputByte[1] = Serial.read();
        delay(10);
        inputByte[2] = Serial.read();
        delay(10);
    }

    if(inputByte[0] == 10 && inputByte[1] == 20 && inputByte[2] == 30)
    {
        Serial.print("Device 0");
        return(true);
    }
}
```

## Arduino 2

```
void setup() {

Serial.begin(9600); //Εναρξη σειριακής επικοινωνίας
Measure(); //Κλήση λειτουργίας μέτρησης θερμοκρασίας και υγρασίας
display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //Εναρξη οθόνης OLED
display.clearDisplay(); //Καθάρισμα περιεχομένου της οθόνης OLED
UpdateScreen(); //Κλήση λειτουργίας ανανέωσης οθονών
//Διαχείριση ακίδων σύνδεσης του συστήματος ποτίσματος
pinMode(34,OUTPUT);
pinMode(35,OUTPUT);
digitalWrite(34,LOW);
digitalWrite(35,LOW);
////////////////////////////////////

//Διαχείριση ακίδων σύνδεσης του συστήματος Aircondition
pinMode(30,OUTPUT);
pinMode(32,OUTPUT);
digitalWrite(30,LOW);
digitalWrite(32,LOW);
////////////////////////////////////
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
//Διαχείριση ακίδων σύνδεσης του συστήματος Θέρμανσης
pinMode(31,OUTPUT);
pinMode(33,OUTPUT);
digitalWrite(31,LOW);
digitalWrite(33,LOW);
////////////////////////////////////

//Διαχείρισης ακίδων σύνδεσης του συστήματος Συναγερμού
pinMode(40,OUTPUT);
pinMode(13,OUTPUT);
digitalWrite(40,LOW);
digitalWrite(13,LOW);
lcd.begin(16, 2);
lcd.setCursor(0, 0);
lcd.print("Alarm System");
lcd.setCursor(0, 1);
lcd.print("Press * to enable");
////////////////////////////////////

//Διαχείριση συστήματος κουτιού του πίνακα ελέγχου
pinMode(42,INPUT);
pinMode(45,OUTPUT);
pinMode(43,OUTPUT);
pinMode(53,OUTPUT);
AwningBox(); //Κλήση λειτουργίας σεταρίσματος κουτιού του πίνακα ελέγχου
πριν την έναρξη εκτέλεσης του προγράμματος
////////////////////////////////////

}

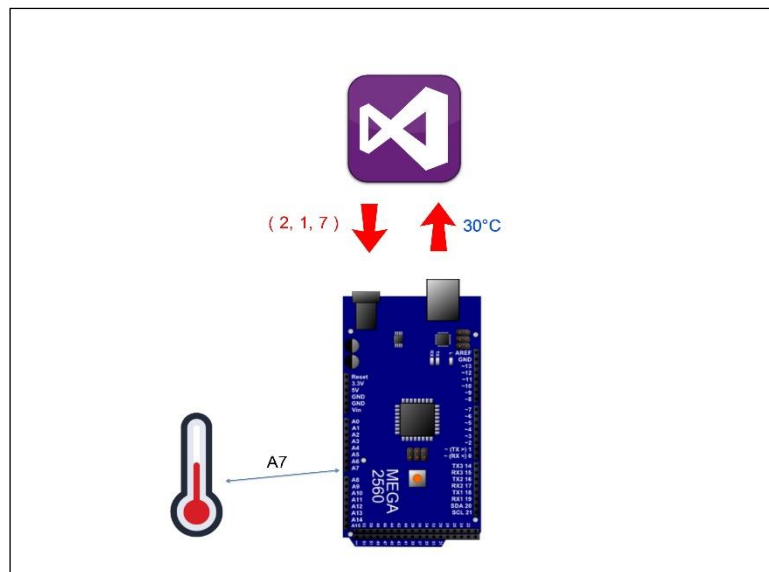
//Λειτουργία ανταλλαγής σειριακών μηνυμάτων με το πρόγραμμα C#
bool sindesi(){
    boolean sin = false; //Αρχική κατάσταση ΜΗ ΣΥΝΔΕΔΕΜΕΝΟ

//Εφόσον υπάρχουν διαθέσιμες πληροφορίες στο κανάλι επικοινωνίας διάβασέ
τις
    if (Serial.available() > 2)
    {
        inputByte[0] = Serial.read();
        delay(10);
        inputByte[1] = Serial.read();
        delay(10);
        inputByte[2] = Serial.read();
        delay(10);
    }

//Μήνυμα επίτευξης σύνδεσης του Arduino αισθητηρίων
if(inputByte[0] == 11 && inputByte[1] == 21 && inputByte[2] == 31)
    {
        Serial.print("Device 1");
        return(true);
    }
}
```



Μόλις ολοκληρωθεί η διαδικασία σύζευξης επικοινωνίας ακολουθεί το μήνυμα ανταλλαγής πληροφοριών. Το μήνυμα αυτό αποτελείται από τρεις αριθμούς όπου για το Arduino που ελέγχει τις συσκευές ακολουθεί το εξής μοτίβο 5, αριθμός ακίδας που βρίσκεται η συσκευή, 1 ή 0 ανάλογα αν θέλουμε ενεργοποίηση ή απενεργοποίηση της συσκευής. Για το Arduino που χειρίζεται τα αισθητήρια τα πράγματα είναι διαφορετικά λόγω του ότι στέλνει πληροφορίες και δεν λαμβάνει. Το μοτίβο που ακολουθείται για επικοινωνία με αυτή την συσκευή είναι 2, 1, ακίδα αναλογικού αισθητηρίου που είναι συνδεδεμένος ο αισθητήρας του οποίου θέλουμε την μέτρηση.



Εικόνα 70: Παράδειγμα αίτησης κοινοποίησης τιμής αισθητηρίου

Για να ελέγξουμε αν η σύνδεση μεταξύ της πλακέτας και του προγράμματος συνεχίζει να υπάρχει χρησιμοποιούμε την ρουτίνα `sindesi()`. Παρακάτω ακολουθεί ο κώδικας αυτού του τρόπου επικοινωνίας καθώς και της κεντρικής λειτουργίας `void loop()`:

```
//Κεντρική λειτουργία προγράμματος
void loop() {

    //Αν η πλακέτα έχει συνδεθεί με το πρόγραμμα
    if(sindesi() == true)
    {
        Measure();

        //Αν είναι ενεργοποιημένος ο έλεγχος της τέντας από το κουτί ελέγχου
        if (digitalRead(44) == LOW){
            unsigned long currentMillis2 = millis();
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
if(currentMillis2 - previousMillis2 >= interval2 && digitalRead(43)==LOW)
{
    previousMillis2 = currentMillis2;
    Awning();
}

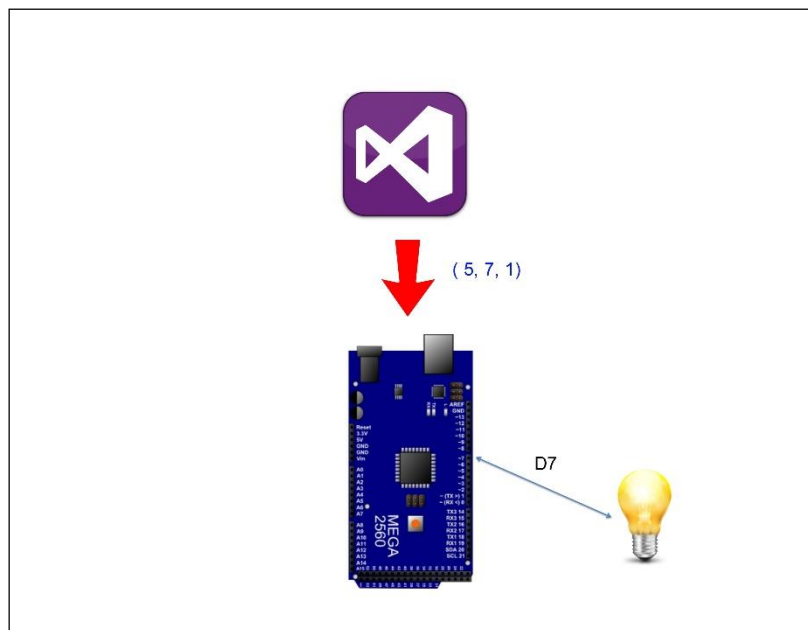
if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 11 )    {
    Serial.print(TempHum[1]);
}
else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 12 )
{
    Serial.print(TempHum[2]);
}
else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 13 )
{
    Serial.print(TempHum[3]);
}
else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 14 )
{
    Serial.print(TempHum[4]);
}
else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 15 )
{
    Serial.print(TempHum[5]);
}
else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 16 )
{
    Serial.print(TempHum[6]);
}
else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 17 )
{
    Serial.print(TempHum[7]);
}
else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 18 )
{
    Serial.print(TempHum[8]);
}
else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 19 )
{
    Serial.print(TempHum[9]);
}
}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 20 )
{
    Serial.print(TempHum[10]);
}
else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 21 )
{
    Serial.print(TempHum[11]);
}
//Σε περίπτωση φωτιάς στείλε την απάντηση 1 στην αίτηση 7,1,96
else if (inputByte[0]==7 && inputByte[1]==1 && inputByte[2]==96){
    if (analogRead(A1) > 600 || analogRead(A9) >600 ||
analogRead(A10) >240){
        Serial.print("1");
    }
    else{
        Serial.print("0");
    }
}
//Σε περίπτωση ενεργοποίησης του κουμπιού έκτακτης ανάγκης στείλε την
απάντηση 1 στην αίτηση 7,1,97
else if (inputByte[0]==7 && inputByte[1]==1 && inputByte[2]==97){
    if (digitalRead(53)==HIGH){
        Serial.print("1");
    }
    else if(digitalRead(53) == LOW){
        Serial.print("0");
    }
}
//Σε περίπτωση παραβίασης του χώρου την ώρα που ήταν ενεργοποιημένος ο
συναγερμός στείλε την απάντηση 1 στην αίτηση 7,1,98
else if (inputByte[0]==7 && inputByte[1] == 1 && inputByte[2]==98){
    if(intruder==1){
        Serial.print("1");
    }
    else if(intruder==0){
        Serial.print("0");
    }
}
//Αλλιώς αν η αίτηση έχει την μορφή 2,1,X όπου X = 0-10 στείλε την
αναλογική τιμή αισθητηρίου
else if(inputByte[0] == 2 && inputByte[1] == 1 ){
    i=inputByte[2];
    Serial.print(analogRead(PinA[i]));
}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    }  
    //Αν έχει άλλη μορφή το μήνυμα μην κάνεις τίποτα  
    else{  
        }  
    }  
//Αν δεν έχει επιτευχθεί σύνδεση μην κάνεις τίποτα  
else{  
    }  
  
UpdateScreen(); //Κλήση λειτουργίας ανανέωσης οθονών  
NeedWatering(); //Κλήση λειτουργίας ελέγχου ποτίσματος  
Aircondition(); //Κλήση λειτουργίας ελέγχου κλιματισμού  
Radiator(); //Κλήση λειτουργίας ελέγχου θέρμανσης  
AlarmSystem(); //Κλήση λειτουργίας ελέγχου συστήματος συναγερμού  
  
inputByte[0] = 0;  
inputByte[1] = 0;  
inputByte[2] = 0;  
  
}
```



Εικόνα 71: Παράδειγμα αίτησης ενεργοποίησης συσκευής

Η μεταβλητή `LostConnection` θα μας βοηθήσει να παρέχουμε στον χρήστη τον πλήρη έλεγχο των συσκευών του σε περίπτωση απώλειας σύνδεσης στο δίκτυο απελευθερώνοντας εφόσον συμβεί αυτό τον έλεγχο των συσκευών του από τον

διαδικτυακό έλεγχο. Παρακάτω ακολουθεί ο κώδικας του παραδείγματος αίτησης ενεργοποίησης ή απενεργοποίησης συσκευής καθώς και της κεντρικής λειτουργίας void loop():

```
void loop() {

    //Αν επιτευχθεί σύνδεση με το πρόγραμμα
    if(sindesi() == true)
    {

        if (LostConnection == 0){
            LostConnection=1;
        }

        if(inputByte[0] == 5)
        {

            if (inputByte[2] == 1){

                //Αν η αίτηση είναι η 5,30,1 ενεργοποίησε την τηλεόραση
                if (inputByte[1] == 30){
                    televisionon();
                }

                //Αν η αίτηση είναι η 5,49,1 άνοιξε την πόρτα του γκαράζ
                else if(inputByte[1] == 49 && garagestatus==false){
                    if (FirstTime==0){
                        garagestatus=true;
                        FirstTime++;
                    }
                    else{
                        garageon();
                    }
                }

                //Ενεργοποίηση φωτισμού δωματίων και απενεργοποίηση πινακίδων
                Exit

                else if(inputByte[1] == 2){
                    digitalWrite(2, HIGH);
                    digitalWrite(46,LOW);
                }
                else if(inputByte[1] == 9){
                    digitalWrite(9, HIGH);
                    digitalWrite(47,LOW);
                }
                else if(inputByte[1] == 12){
                    digitalWrite(12, HIGH);
                    digitalWrite(48,LOW);
                }

                //Ενεργοποίηση της συσκευής που βρίσκεται στην ακίδα X όταν η
                αίτηση έχει την μορφή 5,X,1
                else{
                    digitalWrite(inputByte[1], HIGH);
                }
            }

            else if (inputByte[2]==0){
```

```
    if (inputByte[1] == 30){
        televisionoff();
    }

    else if(inputByte[1] == 49 && garagestatus==true){

        if (FirstTime==0){
            garagestatus=false;
            FirstTime++;
        }

        else{
            garageoff();
        }
    }

    else if(inputByte[1] == 2){
        digitalWrite(2, LOW);
        digitalWrite(46,HIGH);
    }
    else if(inputByte[1] == 9){
        digitalWrite(9, LOW);
        digitalWrite(47,HIGH);
    }
    else if(inputByte[1] == 12){
        digitalWrite(12, LOW);
        digitalWrite(48,HIGH);
    }
    else{
        digitalWrite(inputByte[1], LOW);
    }
}

}

sendButtonState(); //Κλήση λειτουργίας ελέγχου κατάστασης διακοπών
αφής

inputByte[0] = 0;
inputByte[1] = 0;
inputByte[2] = 0;

}

//Σε περίπτωση διακοπής Internet ή κάποιου άλλου προβλήματος ενεργοποίησε
όλες τις συσκευές έτσι ώστε να μπορεί να τις χειριστεί ο χρήστης
else if(LostConnection==0){

    televisionon();

    for (int i=2; i<14; i++)
    {
        digitalWrite(i,HIGH);
    }

    for(int i=42; i<46; i++)
    {
        digitalWrite(i,HIGH);
    }
}
```

```

    }
  }
}

```

Στον παρακάτω πίνακα θα δούμε τις βοηθητικές λειτουργίες με την λειτουργία τους όσον αφορά την πλακέτα Arduino που χειρίζεται τις συσκευές.

<u>Όνομα ρουτίνας</u>	<u>Λειτουργία Ρουτίνας</u>
garageon()	Ρουτίνα ανοίγματος πόρτας γκαράζ
garageoff()	Ρουτίνα κλεισίματος πόρτας γκαράζ
televisionon()	Ρουτίνα ενεργοποίησης τηλεόρασης
televisionoff()	Ρουτίνα απενεργοποίησης τηλεόρασης
sendButtonState()	Έλεγχος πατήματος διακοπών αφής και αποστολή της κατάστασής τους στο κεντρικό πρόγραμμα ελέγχου

Πίνακας 6: Λειτουργίες Arduino συσκευών και επεξήγηση λειτουργίας

Παρακάτω ακολουθεί ο πηγαίος κώδικας των λειτουργιών που εξηγήσαμε στον πίνακα 6.

```

//Λειτουργία ανοίγματος πόρτας γκαράζ
void garageon() {
myStepper.step(3500);
garagestatus =true;
}

//Λειτουργία κλεισίματος πόρτας γκαράζ
void garageoff() {
myStepper.step(-2000);
garagestatus=false;
}

//Λειτουργία ενεργοποίησης τηλεόρασης
void televisionon()
{

unsigned long currentMillis = millis();
unsigned long timeOfLastLedEvent = 0;
int intervalOFF = 0;

if (currentMillis - timeOfLastLedEvent > intervalOFF && tvstatus == false) {

timeOfLastLedEvent = currentMillis;
tvstatus = true;
digitalWrite(42,HIGH); //Ενεργοποίηση οπίσθιου φωτισμού τηλεόρασης

```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
myGLCD.clrScr(); //Καθαρισμός περιεχομένου οθόνης
myGLCD.setColor(VGA_WHITE); //Ορισμός λευκού χρώματος γραμμών
myGLCD.setBackgroundColor(0, 0, 0); //Ορισμός μαύρου χρώματος background
myGLCD.drawBitmap (90, 100, 67, 50, alt, 2); //Σχεδιασμός λογότυπου στην
οθόνη
myGLCD.setFont(BigFont); //Ορισμός μεγάλης γραμματοσειράς
myGLCD.print("House Web Control", CENTER, 40);
myGLCD.print("AEI Piraeus TT MHYS", CENTER, 65);
myGLCD.setFont(SmallFont); //Ορισμός μικρής γραμματοσειράς
myGLCD.print("By:",RIGHT , 200);
myGLCD.print("Giorgos Goniotakis & Nikos Papazian",RIGHT , 220);

}
}

//Λειτουργία απενεργοποίησης τηλεόρασης
void televisionoff(){
  myGLCD.clrScr(); //Καθαρισμός περιεχομένου οθόνης
  digitalWrite(42,LOW); //Απενεργοποίηση οπίσθιου φωτισμού
  tvstatus = false;
}

//Λειτουργία αποστολής κατάστασης πλήκτρων αφής
void sendButtonState(){

  //Αν έχει πατηθεί το πλήκτρο αφής στείλε αίτηση με αριθμοί 5,2,2
  if (digitalRead(16)==HIGH){
    if (inputByte[0]== 5 && inputByte[1]==2 && inputByte[2]==2 ){
      Serial.print("1");
    }
  }
  else if (digitalRead(17)==HIGH){
    if (inputByte[0]== 5 && inputByte[1]==2 && inputByte[2]==3 ){
      Serial.print("1");
    }
  }
  else if (digitalRead(18)==HIGH){
    if (inputByte[0]== 5 && inputByte[1]==2 && inputByte[2]==6 ){
      Serial.print("1");
    }
  }
  else if (digitalRead(19)==HIGH){
    if (inputByte[0]== 5 && inputByte[1]==2 && inputByte[2]==9 ){
      Serial.print("1");
    }
  }
  else if (digitalRead(20)==HIGH){
    if (inputByte[0]== 5 && inputByte[1]==2 && inputByte[2]==12 ){
      Serial.print("1");
    }
  }
  else if (digitalRead(21)==HIGH){
    if (inputByte[0]== 5 && inputByte[1]==2 && inputByte[2]==44 ){
      Serial.print("1");
    }
  }
}
}
```



Στον παρακάτω πίνακα θα δούμε τις βοηθητικές λειτουργίες με την λειτουργία τους όσον αφορά την πλακέτα Arduino που χειρίζεται τα αισθητήρια.

Όνομα ρουτίνας	Λειτουργία Ρουτίνας
Measure()	Μέτρηση τιμών αισθητηρίου υγρασίας - θερμοκρασίας
UpdateScreen()	Ανανέωση περιεχομένου ενημερωτικών οθονών OLED
NeedWatering()	Έλεγχος κατάστασης ποτίσματος φυτών
Aircondition()	Αυτόματη ενεργοποίηση/απενεργοποίηση συστήματος κλιματισμού
Radiator()	Αυτόματη ενεργοποίηση/απενεργοποίηση συστήματος θέρμανσης
AlarmSystem()	Έλεγχος συστήματος συναγερμού
checkAlarm()	Έλεγχος παραβίασης χώρου
checkPassword()	Έλεγχος πληκτρολόγησης κωδικού συστήματος συναγερμού
enableAlarm()	Ενεργοποίηση συστήματος συναγερμού
checkEntry()	Έλεγχος πληκτρολόγησης
AwningBox()	Χειροκίνητη ρύθμιση τέντας στο κατώτερο σημείο της πριν την έναρξη
Awning()	Αυτόματη ρύθμιση τέντας βάση του ποσοστού ανέμου, βροχής και φωτεινότητας

Πίνακας 7: Λειτουργίες Arduino αισθητηρίων και επεξήγηση λειτουργίας τους

Παρακάτω ακολουθεί ο πηγαίος κώδικας των λειτουργιών που εξηγήσαμε στον πίνακα 7.

```
//Λειτουργία καταγραφής τιμών αισθητηρίων θερμοκρασίας - υγρασίας
void Measure () {

    TempHum[1] = DHT.read11 (DHTPINLIV) ; //Σύνδεση με το αισθητήριο τύπου
DHT11
    TempHum[1] = (int)DHT.temperature; //Διάβασμα τιμής θερμοκρασίας
αισθητηρίου
    TempHum[2] = (int)DHT.humidity; //Διάβασμα τιμής υγρασίας αισθητηρίου
    TempHum[3] = DHT.read11 (DHTPINBAT) ;
    TempHum[3] = (int)DHT.temperature;
    TempHum[4] = (int)DHT.humidity;
    TempHum[5] = DHT.read11 (DHTPINBED) ;
    TempHum[5] = (int)DHT.temperature;
    TempHum[6] = (int)DHT.humidity;
    TempHum[7] = DHT.read11 (DHTPINKIT) ;
    TempHum[7] = (int)DHT.temperature;
    TempHum[8] = (int)DHT.humidity;
    TempHum[9] = DHT.read11 (DHTPINOUT) ;
    TempHum[9] = (int)DHT.temperature;
    TempHum[10] = (int)DHT.humidity;
    TempHum[11] = DHT.read11 (DHTPINROO) ;
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
TempHum[11] = (int)DHT.temperature;
TempHum[12] = (int)DHT.humidity;

}

//Λειτουργία ενημέρωσης οθονών OLED
void UpdateScreen() {

//Αν έχουν περάσει 2 δευτερόλεπτα
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {

    previousMillis = currentMillis;

    //Αν βρίσκεσαι στην σελίδα 1 ενημέρωσε για τις τιμές των αισθητηρίων
    θερμοκρασίας του σαλονιού, του υπνοδωματίου και της τουαλέτας
    if (flag == 0)
    {
        display.clearDisplay();
        display.setTextSize(2);
        display.setTextColor(WHITE);
        display.setCursor(0,0);
        display.print("LIVIN: ");
        display.print(TempHum[1]);
        display.print("C");
        display.setCursor(0,25);
        display.print("BEDR: ");
        display.print(TempHum[5]);
        display.print("C");
        display.setCursor(0,50);
        display.print("BATH: ");
        display.print(TempHum[3]);
        display.print("C");
        display.display();

        flag=1;
    }

    //Αν βρίσκεσαι στην σελίδα 2 ενημέρωσε για τις τιμές των αισθητηρίων
    θερμοκρασίας της κουζίνας και του εξωτερικού χώρου
    else if (flag== 1){
        display.clearDisplay();
        display.setTextSize(2);
        display.setTextColor(WHITE);
        display.setCursor(0,0);
        display.print("KITCH: ");
        display.print(TempHum[7]);
        display.print("C");
        display.setCursor(0,25);
        display.print("EXTER: ");
        display.print(TempHum[9]);
        display.print("C");
        display.setCursor(0,50);
        display.display();

        flag=0;
    }

}

}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
//Λειτουργία ελέγχου ποτίσματος
void NeedWatering() {

    //Ενεργοποίησε το σύστημα ποτίσματος ανάλογα με την τιμή του αισθητηρίου
του φυτού νούμερο 1
    if (analogRead(A3) > 500) {

        digitalWrite(35,HIGH);

    }
    else{

        digitalWrite(35,LOW);

    }

    //Ενεργοποίησε το σύστημα ποτίσματος ανάλογα με την τιμή του αισθητηρίου
του φυτού νούμερο 2
    if (analogRead(A4) > 500) {

        digitalWrite(34,HIGH);

    }
    else{

        digitalWrite(34,LOW);

    }
}

//Λειτουργία ελέγχου συστήματος κλιματισμού
void Aircondition() {

    //Αν η θερμοκρασία έχει φτάσει τους 30 βαθμούς ενεργοποίησε τον
κλιματισμού μέχρι η θερμοκρασία να φτάσει τους 28 βαθμούς
    if (TempHum[1] > Thermostat3) {

        digitalWrite(30,HIGH);
        Thermostat3 = 28;

    }
    else{

        Thermostat3 = 30;
        digitalWrite(30,LOW);

    }

    //Αν η θερμοκρασία έχει φτάσει τους 30 βαθμούς ενεργοποίησε τον
κλιματισμού μέχρι η θερμοκρασία να φτάσει τους 28 βαθμούς
    if (TempHum[5] > Thermostat4) {

        digitalWrite(32,HIGH);
        Thermostat4 = 28;

    }
    else{

        Thermostat4 = 30;
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
        digitalWrite(32,LOW);
    }
}

//Λειτουργία ελέγχου συστήματος θέρμανσης
void Radiator(){

    //Αν η θερμοκρασία έχει φτάσει τους 18 βαθμούς ενεργοποίησε την θέρμανση
    μέχρι η θερμοκρασία να φτάσει τους 20 βαθμούς
    if (TempHum[1] < Thermostat1){

        digitalWrite(31,HIGH);
        Thermostat1 = 20;

    }
    else{

        Thermostat1 = 18;
        digitalWrite(31,LOW);

    }

    //Αν η θερμοκρασία έχει φτάσει τους 18 βαθμούς ενεργοποίησε την θέρμανση
    μέχρι η θερμοκρασία να φτάσει τους 20 βαθμούς
    if (TempHum[5] < Thermostat2){

        digitalWrite(33,HIGH);
        Thermostat2=20;

    }
    else{

        Thermostat2=20;
        digitalWrite(33,LOW);

    }
}

//Λειτουργία συστήματος συναγερμού
void AlarmSystem(){

    char customKey = customKeypad.getKey(); //Σε περίπτωση που πατηθεί
    πλήκτρο αποθήκευσε το

    //Αν πατηθεί το * ενεργοποίησε τον συναγερμό
    if (customKey == '*' && AlarmStatus == false){

        enableAlarm();

    }
    //Αν είναι ενεργοποιημένος κάνε έλεγχο σωστού κωδικού ή παραβίασης
    χώρου
    else if (AlarmStatus==true){

        checkAlarm();
        checkEntry();

    }

}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    }
    //Αλλιώς ίμην κάνεις κάτι
    else{

    }
}

//Λειτουργία ελέγχου παραβίασης χώρου σε περίπτωση που το σύστημα
//συναγερμού είναι ενεργοποιημένο
void checkAlarm() {

    //Αν γίνει παραβίαση του χώρου ενεργοποίησε τον συναγερμό
    if (analogRead(A6) > 500 || analogRead(A8) > 500 && intruder==0) {

        digitalWrite(40,HIGH);
        intruder=1;

    }

    //Αν είναι ήδη ενεργοποιημένος άφησέ τον ενεργοποιημένο
    else if (intruder==1){
        digitalWrite(40,HIGH);
    }

    //Αν δεν υπάρχει κίνηση άφησε τον συναγερμό απενεργοποιημένο
    else{
        digitalWrite(40,LOW);
    }
}

//Λειτουργία ελέγχου κωδικού
void checkPassword() {

    //Αν ο κωδικός είναι σωστός
    if(!strcmp(tempCode,Code)){

        AlarmStatus=false;
        intruder=0;
        datacount=0;
        digitalWrite(40,LOW);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Disarmed");
        delay(1000);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Alarm System");
        lcd.setCursor(0, 1);
        lcd.print("Press * to enable");

        int rep;

        for(rep=0;rep<4;rep++){

            tempCode[rep]='0';
        }
    }
}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
//Αν ο κωδικός είναι λαθός, προσπάθεια ξανά
else{

    datacount=0;
    int rep;

    for(rep=0;rep<4;rep++){

        tempCode[rep]='0';

    }

    lcd.setCursor(0,1);
    lcd.print("Wrong Password");
    delay(1000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Enter Password...");
    lcd.setCursor(0, 1);
    lcd.blink();

}
}

//Λειτουργία ενεργοποίησης συστήματος συναγερμού
void enableAlarm(){

AlarmStatus=true;

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Enabling Alarm...");
delay(10000); //Περίμενε 10 δευτερόλεπτα
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Enter Password...");
lcd.setCursor(0, 1);
lcd.blink();

}

void checkEntry(){

char customKey = customKeypad.getKey();

//Αν πατηθεί πλήκτρο αποθήκευσε το
if(customKey){

    if (customKey!= '*' && customKey!='#'){

        lcd.print("*");
        tempCode[datacount] =customKey;
        datacount++;

    }

    //Αν ο χρήστης εισάγει 4 χαρακτήρες έλεγξε τον κωδικό
    if (datacount ==4){

        checkPassword();

    }

}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
}

//Αν πατήσει # καθαρισμός των ψηφίων που έχει εισάγει
else if (customKey == '#' && datacount<4){

    datacount=0;
    int rep;

    for(rep=0;rep<4;rep++){

        tempCode[rep]='0';

    }

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Enter Password...");
    lcd.setCursor(0, 1);
    lcd.blink();
}

}

}

//Λειτουργία ρύθμισης της τέντας από το κουτί που υπάρχει στον πίνακα
ελέγχου
void AwningBox(){

    const int stepsPerRevolution = 200; //Βήματα ανά επανάληψη
    Stepper myStepper(200,36,37,38,39); //Δήλωση βηματικού κινητήρα
    myStepper.setSpeed(60); //Ορισμός ταχύτητας
    int SetUp = 0; //Μεταβλητή αποθήκευσης κατάστασης διακόπτη Push Button
    digitalWrite(45,HIGH); //Ενεργοποίηση LED ένδειξης κατάστασης

    //Όσο δεν έχει πατηθεί το Push Button ρύθμιση της τέντας μέσω του
    ποτενσιόμετρου
    while (SetUp != 1){

        digitalWrite(52,HIGH);
        int Potentiometer = analogRead(A11);
        int PushButton = digitalRead(42);

        if (SetUp == 0){
            if (PushButton == HIGH){

                SetUp++;
                digitalWrite(45,LOW);
                delay(500);
                digitalWrite(45,HIGH);
                delay(1000);
            }
            else if (Potentiometer > 820 && Potentiometer < 1023){

                myStepper.step(200);

            }

        }

    }

}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
        else if (Potentiometer > 0 && Potentiometer <268){  
            myStepper.step(-200);  
        }  
        else{  
            //POTENTIOMETER MEDIUM  
        }  
    }  
}  
digitalWrite(45,LOW);  
}
```

```
void Awning() {
```

```
    Stepper myStepper(200,36,37,38,39); //Δήλωση οντότητας βηματικού κινητήρα  
    myStepper.setSpeed(60);  
    int WindSensor = analogRead(A5); //Αναλογική μέτρηση από τον αισθητήρα  
    ανέμου  
    int RainSensor = analogRead(A7); //Αναλογική μέτρηση από τον αισθητήρα  
    βροχής  
    int LightSensor = analogRead(A2); //Αναλογική μέτρηση από τον αισθητήρα  
    φωτεινότητας
```

```
    //Αν έχει άνεμο
```

```
    if (WindSensor >= 70 && TentIdle!=1){
```

```
        DueToLight1=0;  
        DueToLight2=0;  
        DueToLight3=0;  
        DueToWind++;  
        DueNotToWind=0;
```

```
    }
```

```
    //Αν δεν έχει άνεμο
```

```
    else if (WindSensor<70 && TentIdle==1){
```

```
        DueNotToWind++;  
        DueToWind=0;
```

```
    }
```

```
    //Αν πάρουμε 10 συνεχόμενες μετρήσεις ότι είχαμε άνεμο μετακινούμε την  
    τέντα
```

```
    if (DueToWind >=10 && TentIdle==0){
```

```
        DueToLight1=0;  
        DueToLight2=0;  
        DueToLight3=0;  
        TentIdle=1;  
        angle=pos1-lastpos;  
        lastpos=pos1;  
        myStepper.step(angle);
```

```
    }
```



## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
//Αν πάρουμε 10 συνεχόμενες μετρήσεις ότι δεν έχει άνεμο αφήνουμε τον
έλεγχο της τέντας στα υπόλοιπα αισθητήρια
else if (DueNotToWind ==10 && TentIdle==1){

    DueToLight1=0;
    DueToLight2=0;
    DueToLight3=0;
    TentIdle=0;

}

//Αν έχει βροχή και όχι άνεμο
if (RainSensor < 800 && TentIdle==0){

    DueToLight1=0;
    DueToLight2=0;
    DueToLight3=0;
    DueToRain++;
    DueNotToRain=0;

}

//Αν δεν έχει βροχή
else{

    DueToRain=0;
    DueNotToRain++;

}

//Αν πάρουμε 10 συνεχόμενες μετρήσεις ότι είχαμε βροχή και δεν είχαμε
άνεμο μετακινούμε την τέντα
if (DueToRain >= 10 && TentIdle!=1){

    DueToLight1=0;
    DueToLight2=0;
    DueToLight3=0;
    angle=pos2-lastpos;
    lastpos=pos2;
    TentIdle2=1;
    myStepper.step(angle);

}

//Αν πάρουμε 10 συνεχόμενες μετρήσεις ότι δεν έχει βροχή αφήνουμε τον
έλεγχο της τέντας στα υπόλοιπα αισθητήρια
else if (DueNotToRain == 10 && TentIdle2==1){

    DueToLight1=0;
    DueToLight2=0;
    DueToLight3=0;
    TentIdle2=0;

}

//Αν έχουμε υψηλή φωτεινότητα και δεν έχουμε άνεμο ή βροχή
if (LightSensor > 0 && LightSensor < 300 && TentIdle!=1 && TentIdle2!=1){

    DueToLight1++;
    DueToLight2=0;
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    DueToLight3=0;

}
//Αν έχουμε μεσαία φωτεινότητα και δεν έχουμε άνεμο ή βροχή
else if (LightSensor >= 300 && LightSensor < 600 && TentIdle!=1 &&
TentIdle2 !=1){

    DueToLight1=0;
    DueToLight2++;
    DueToLight3=0;

}

//Αν έχουμε χαμηλή φωτεινότητα και δεν έχουμε άνεμο ή βροχή
else if (LightSensor >= 600 && LightSensor <= 1024 && TentIdle!=1 &&
TentIdle2!=1){

    DueToLight1=0;
    DueToLight2=0;
    DueToLight3++;

}
//Αν δεν συμβαίνει τίποτα από όλα αυτά μην κάνεις τίποτα
else{
}

//Αν έχουμε πάρει 10 μετρήσεις υψηλής φωτεινότητας και δεν υπάρχει άνεμος
και βροχή μετακίνησε την τέντα
if (DueToLight1>=10){

    angle=pos3-lastpos;
    lastpos=pos3;
    myStepper.step(angle);

}

//Αν έχουμε πάρει 10 μετρήσεις μεσαίας φωτεινότητας και δεν υπάρχει
άνεμος και βροχή μετακίνησε την τέντα
else if (DueToLight2>=10){

    angle=pos4-lastpos;
    lastpos=pos4;
    myStepper.step(angle);

}

//Αν έχουμε πάρει 10 μετρήσεις χαμηλής φωτεινότητας και δεν υπάρχει
άνεμος και βροχή μετακίνησε την τέντα
else if (DueToLight3>=10){

    angle=pos5-lastpos;
    lastpos=pos5;
    myStepper.step(angle);

}

//Αλλιώς συνέχισε να μετράς
else{
}
}
```

**(Κενό φύλλο)**

## ΚΕΦΑΛΑΙΟ 7

### ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

#### 7.1 Ρόλος βάσεως δεδομένων

Η βάση δεδομένων παίζει σημαντικό ρόλο στην εφαρμογή μας από την στιγμή που έχουμε να κάνουμε με αλλαγές που γίνονται μέσα στην οικία οι οποίες πρέπει να καταγράφονται. Οι τιμές των αισθητηρίων πρέπει να αποθηκεύονται σε μία βάση δεδομένων έτσι ώστε ο χρήστης να είναι σε θέση να τις παρακολουθεί από το διαδίκτυο. Οι αλλαγές που θα πραγματοποιήσει ο χρήστης στις συσκευές του, ενεργοποίηση ή απενεργοποίηση, προσθήκη ή κατάργηση πρέπει επίσης να καταγράφονται με ακρίβεια.

#### 7.2 Επιλογή είδους βάσεως δεδομένων

Για την εφαρμογή μας επιλέξαμε ως βάση δεδομένων την MySQL για τον λόγο της ευκολίας πρόσβασης και χειρισμού της από τις πιο πολλές γλώσσες προγραμματισμού.

Έτσι θα μας είναι πολύ εύκολο να έχουμε πρόσβαση στο περιεχόμενό της από την ιστοσελίδα χρησιμοποιώντας εντολές που ήδη εμπεριέχονται στην σουίτα εντολών της γλώσσας Php. Επιπρόσθετα θα μπορούμε πολύ εύκολα μέσω της χρήσης μίας μόνο βιβλιοθήκης να την ενημερώνουμε μέσω του κεντρικού προγράμματός μας που είναι φτιαγμένο στην γλώσσα προγραμματισμού C#.

#### 7.3 Λειτουργία βάσεως δεδομένων

Η ύπαρξη της βάσεως, όπως αναφέραμε και πρωτίτερα, παίζει καθοριστικό ρόλο στην εφαρμογή μας καθώς το κεντρικό μας πρόγραμμα και η ιστοσελίδα την χρησιμοποιούν ως σημείο αναφοράς.

Πρακτικά όταν ο χρήστης ενεργοποιεί μία συσκευή μέσω της ιστοσελίδας αλλάζει μία τιμή για αυτό το αντικείμενο στην βάση δεδομένων και προσθέτει την ημερομηνία και ώρα τελευταίας τροποποίησης αυτής της τιμής. Το κεντρικό μας πρόγραμμα ελέγχει για αλλαγές που έχουν συμβεί στην βάση τα τελευταία 10 δευτερόλεπτα και τις κοινοποιεί μέσω του πρωτοκόλλου επικοινωνίας στην πλακέτα Arduino που χειρίζεται τις συσκευές. Το Arduino με την σειρά του ενεργοποιεί την συσκευή αυτή.

Όσον αφορά τις τιμές των αισθητηρίων, το Arduino στο οποίο έχουν τοποθετηθεί τα αισθητήρια ενημερώνει συνεχώς το κεντρικό μας πρόγραμμα για τις τιμές των αισθητηρίων. Το πρόγραμμά μας στην συνέχεια καταχωρεί τις πληροφορίες αυτές στην βάση δεδομένων. Ο χρήστης έπειτα μπορεί να δει τις τιμές των αισθητηρίων του μέσα από την ιστοσελίδα.

#### 7.4 Ρύθμιση βάσεως δεδομένων

Η βάση δεδομένων μας όπως αναφέραμε και παραπάνω είναι MySQL. Το είδος των πινάκων είναι InnoDB και η σύνθεση της βάσης είναι utf8\_general\_ci πράγμα που σημαίνει ότι είναι δυνατή η αποθήκευση UTF8 χαρακτήρων, όπως για παράδειγμα ελληνικά και αγγλικά, χωρίς να δημιουργείται καμία αλλοίωση πληροφορίας. Ο λόγος που χρησιμοποιήσαμε είδος πινάκων InnoDB είναι λόγω του ότι επιτρέπει την χρήση συναλλαγών και την ευκολότερη σταθεροποίηση της σε περίπτωση δημιουργίας σφάλματος.

Συναλλαγές (transactions) σε μία βάση δεδομένων πρακτικά σημαίνει ότι η βάση εξυπηρετεί έναν χρήστη την φορά σε κάθε πίνακα. Έτσι όταν ένας χρήστης επιθυμεί να πραγματοποιήσει μια αλλαγή σε έναν πίνακα, ο πίνακας κλειδώνει αποτρέποντας να πραγματοποιηθούν αλλαγές κατά την διάρκεια τροποποίησης του από τον χρήστη. Οι υπόλοιποι χρήστες που θέλουν να αλληλεπιδράσουν με τον πίνακα μπαίνουν σε σειρά προτεραιότητας.

Ως σταθεροποίηση σε περίπτωση δημιουργίας σφάλματος (crash recovery) ορίζεται η διαδικασία που ακολουθεί η βάση δεδομένων έτσι ώστε να ξεμπλοκαρισθεί σε περίπτωση που κάτι δεν πάει έτσι όπως θα έπρεπε. Μερικές φορές κατά την ταυτόχρονη αλληλεπίδραση των χρηστών με την βάση δημιουργούνται κάποια σφάλματα τα οποία οδηγούν την βάση να μην αποκρίνεται πια στα αιτήματα των χρηστών. Έτσι η βάση δεδομένων αναλαμβάνει να επαναφερθεί μόνη της στο τελευταίο σημείο σταθερής λειτουργίας, όσον το δυνατόν πιο γρήγορα γίνεται, επιτρέποντας ξανά στους χρήστες να πραγματοποιήσουν ενέργειες. Το θετικό από την ύπαρξη αυτής της λειτουργίας είναι ότι ο υπεύθυνος της βάσης σε μία τέτοια περίπτωση δεν θα χρειαστεί να κάνει επανεκκίνηση στην βάση δεδομένων και η βάση θα είναι πάντα διαθέσιμη για τους χρήστες.

## 7.5 Δομή βάσεως δεδομένων

Η βάση δεδομένων μας αποτελείται από έξι πίνακες οι οποίοι αναλαμβάνουν να αποθηκεύσουν πληροφορίες όσον αφορά τις συσκευές, τα αισθητήρια και τους χρήστες του συστήματος.

Ο πίνακας Users αναλαμβάνει να αποθηκεύσει σημαντικές πληροφορίες όσον αφορά τους χρήστες του συστήματος και τις οικίες τους. Στον πίνακα που ακολουθεί θα δούμε αναλυτικά τα πεδία του πίνακα.

<u>Όνομασία πεδίου</u>	<u>Σκοπός πεδίου</u>
Username	Όνομα χρήστη για πρόσβαση στο σύστημα
Password	Κωδικός ασφαλείας χρήστη για πρόσβαση στο σύστημα
Name	Όνοματεπώνυμο χρήστη
Image	Σύνδεσμος εικόνας χρήστη
Email	Διεύθυνση ηλεκτρονικού ταχυδρομείου χρήστη
CustomerFrom	Ημερομηνία έναρξης χρήσης
HomeAdr	Διεύθυνση οικίας χρήστη
LastCon	Ημερομηνία και ώρα τελευταίας σύνδεσης χρήστη στο σύστημα
HomeImg	Σύνδεσμος εικόνας οικίας
Dimensions	Διαστάσεις οικίας χρήστη
Rooms	Αριθμός δωματίων οικίας
SmartSystem	Έλεγχος ενεργοποίησης λειτουργίας SmartSystem
CustomerID	Μοναδικός αριθμός χρήστη συστήματος
AuthenticatorKey	Μοναδικό κλειδί λειτουργίας γεννήτριας κωδικών χρήστη
EmergencyKey	Ενναλακτικός κωδικός ασφαλείας
EmergencyPerson	Διεύθυνση ηλεκτρονικού ταχυδρομείου ατόμου που θα ειδοποιηθεί σε περίπτωση ανάγκης

Πίνακας 8: Επεξήγηση πεδίων πίνακα Users

Ο μοναδικός αριθμός χρήστη συστήματος (CustomerID) αποτελεί το πεδίο συσχέτισης όλων των πινάκων. Στην συνέχεια θα δούμε την δομή του πίνακα Preferences ο οποίος αποθηκεύει προτιμήσεις του χρήστη.

<u>Ονομασία πεδίου</u>	<u>Σκοπός πεδίου</u>
CustomerID	Μοναδικός αριθμός χρήστη
TypeOfPreference	Ονομασία ρύθμισης
RequiresAction	Τιμή ενεργοποίησης/απενεργοποίησης λειτουργίας
Title	Περιγραφή ρύθμισης

Πίνακας 9: Επεξήγηση πεδίων πίνακα Preferences

Οι βάσεις δεδομένων χρησιμοποιούνται από τις πιο πολλές εφαρμογές και για την καταγραφή περιστατικών (logs) που συμβαίνουν στο σύστημα και έχουν να κάνουν με τους χρήστες. Έτσι δημιουργήσαμε τον πίνακα Logs του οποίου ο σκοπός είναι να αποθηκεύσει σημαντικά γεγονότα, την ημερομηνία και ώρα που συνέβησαν και το αν ο χρήστης ενημερώθηκε για αυτά σε περίπτωση που δημιουργηθεί κάποιο σφάλμα με τον διακομιστή ηλεκτρονικής αλληλογραφίας μας (email server). Η δομή του πίνακα φαίνεται παρακάτω.

<u>Ονομασία πεδίου</u>	<u>Σκοπός πεδίου</u>
CustomerID	Μοναδικός αριθμός χρήστη
IncidentName	Ονομασία συμβάντος
Timestamp	Ημερομηνία και ώρα ύπαρξης συμβάντος
InformedClient	Αποθήκευση κατάσταση ενημέρωσης χρήστη για συμβάν
ID	Μοναδικός αριθμός συμβάντος

Πίνακας 10: Επεξήγηση πεδίων πίνακα Logs

Έπειτα χρειαστήκαμε έναν πίνακα για να αποθηκεύουμε τις συσκευές που υπάρχουν συνδεδεμένες σε κάθε οικία καθώς και διάφορα στοιχεία σχετικά με αυτές. Δημιουργήσαμε λοιπόν για αυτόν τον σκοπό τον πίνακα Appliances του οποίου η δομή φαίνεται παρακάτω.

<u>Ονομασία πεδίου</u>	<u>Σκοπός πεδίου</u>
CustomerID	Μοναδικός αριθμός χρήστη
Gate	Ακίδα σύνδεσης συσκευής
Id	Αριθμός συσκευής
Name	Ονομασία συσκευής
Area	Δωμάτιο που βρίσκεται η συσκευή
State	Κατάσταση συσκευής
ImageOn	Σύνδεσμος εικόνας ενεργοποιημένης συσκευής
ImageOff	Σύνδεσμος εικόνας απενεργοποιημένης συσκευής
update_time	Ημερομηνία και ώρα τελευταίας αλλαγής συσκευής

Πίνακας 11: Επεξήγηση πεδίων πίνακα Appliances

Έπειτα θα θέλαμε έναν πίνακα στον οποίο θα αποθηκεύονται οι τιμές που καταγράφουν τα αισθητήρια της οικίας, ανάγκη η οποία μας οδήγησε στην κατασκευή του πίνακα Sensors. Η δομή του πίνακα παρουσιάζεται παρακάτω.

<u>Ονομασία πεδίου</u>	<u>Σκοπός πεδίου</u>
CustomerID	Μοναδικός αριθμός χρήστη
Gate	Ακίδα σύνδεσης αισθητηρίου
Id	Αριθμός αισθητηρίου
Name	Όνομα αισθητηρίου
Area	Δωμάτιο που βρίσκεται το αισθητήριο
Value	Τιμή αισθητηρίου
Image	Σύνδεσμος εικόνας αισθητηρίου
update_time	Ημερομηνία και ώρα τελευταίας ενημέρωσης αισθητηρίου

Πίνακας 12: Επεξήγηση πεδίων πίνακα Sensors

Τέλος χρειαστήκαμε έναν πίνακα για να αποθηκεύουμε τους συνδέσμους από τις διάφορες υπηρεσίες ζωντανής διαδικτυακής βίντεο παρακολούθησης χώρων τους οποίους θα έχουν οι χρήστες στην διάθεσή τους σε περίπτωση που διαθέτουν κάμερες καταγραφής μέσω διαδικτύου (IP Cameras). Έτσι εφόσον μας τον δώσουν μπορούμε

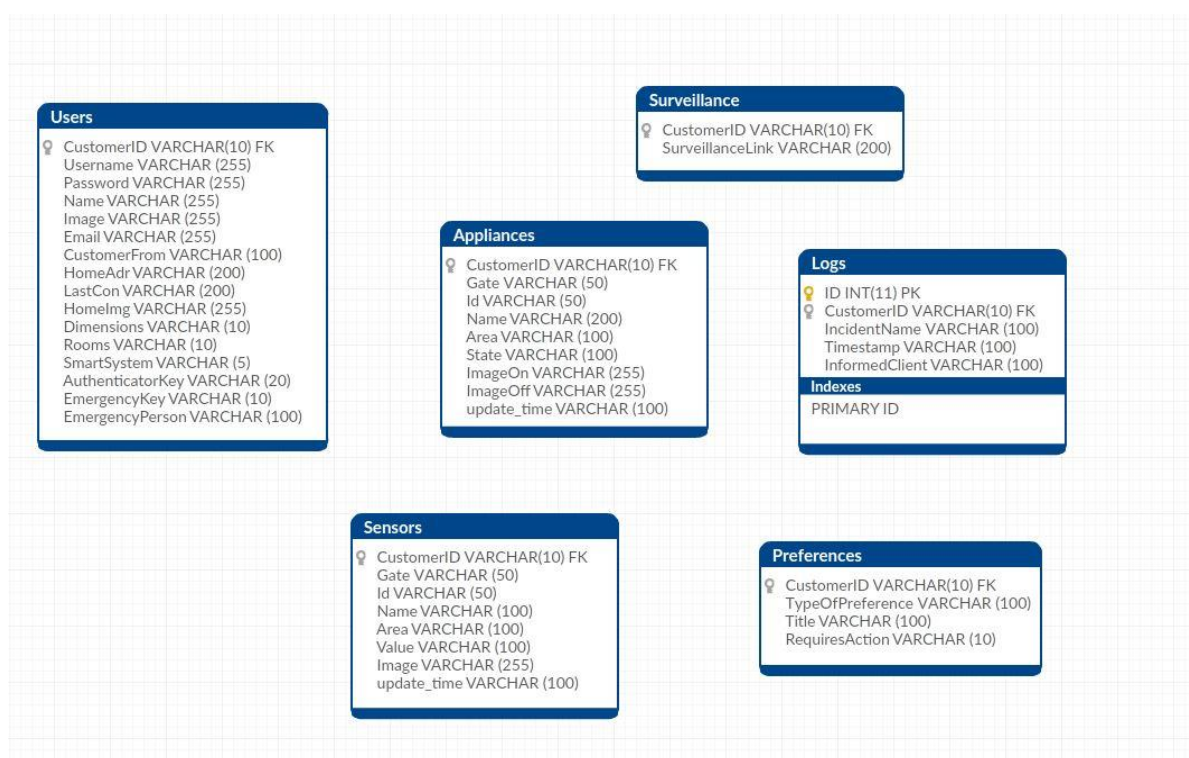


να τους παρέχουμε παρακολούθηση του χώρου τους και μέσα από το δικό μας σύστημα. Η δομή του πίνακα που δημιουργήσαμε για αυτόν τον σκοπό βρίσκεται από κάτω.

<u>Όνομασία πεδίου</u>	<u>Σκοπός πεδίου</u>
CustomerID	Μοναδικός αριθμός χρήστη
SurveillanceLink	Σύνδεσμος παροχής υλικού βιντεοσκόπησης από διαδικτυακή κάμερα (IP Camera)

Πίνακας 13: Επεξήγηση πεδίων πίνακα Surveillance

Στην εικόνα που βρίσκεται παρακάτω φαίνεται η δομή της βάσεως δεδομένων μας (schema).



Εικόνα 72: Δομή βάσεως δεδομένων

## 7.6 Ασφάλεια βάσεως δεδομένων

Για να αυξήσουμε την ασφάλεια της βάσεως δεδομένων μας μετά την εγκατάστασή της στο σύστημά μας δημιουργούμε ένα αντίγραφο του διαχειριστή (root) που μας παρέχει η MySQL και ανακαλούμε όλα τα δικαιώματα από τον αρχικό διαχειριστή

(root). Αυτήν την ενέργεια την πράττουμε για τον λόγο του ότι επειδή είναι γνωστό ότι ο αρχικός διαχειριστής των MySQL βάσεων έχει το όνομα χρήστη root, σε περίπτωση επίθεσης brute force στον χρήστη αυτόν, ο επιτιθέμενος το μόνο που θα καταφέρει να κάνει είναι να αποκτήσει πρόσβαση σε έναν χρήστη ο οποίος πρακτικά δεν έχει κανένα δικαίωμα στο σύστημα. Επιπρόσθετα, λόγω του ότι τα στοιχεία αυτού του χρήστη δεν θα χρησιμοποιούνται θα μπορέσουμε να καταλάβουμε ότι κάποιος άγνωστος κατάφερε να αποκτήσει πρόσβαση στο σύστημά μας και έτσι θα λάβουμε τα αντίστοιχα μέτρα.

Αφού έχουμε δημιουργήσει το αντίγραφο του διαχειριστή πρέπει να δημιουργήσουμε έναν διαφορετικό χρήστη ο οποίος θα έχει μόνο συγκεκριμένα δικαιώματα στην βάση δεδομένων μας. Έτσι θα καταφέρουμε να προστατεύσουμε τις υπόλοιπες βάσεις δεδομένων μας από ανεπιθύμητους και θα περιορίσουμε την πρόσβαση των χρηστών μόνο σε στοιχεία που αφορούν αυτούς.

## 7.7 Κώδικας δημιουργίας βάσεως δεδομένων

Παρακάτω παρουσιάζεται ο κώδικας που θα πληκτρολογήσαμε στην γραμμή εντολών του περιβάλλοντος της MySQL σε περίπτωση που δεν χρησιμοποιούσαμε κάποιο παραθυρικό πρόγραμμα διαχείρισης βάσης όπως είναι για παράδειγμα το MySQL Workbench, PhpmyAdmin, Navicat κλπ.

```
CREATE TABLE `Users` (
  `Username` varchar(255) DEFAULT NULL,
  `Password` varchar(255) DEFAULT NULL,
  `Name` varchar(255) DEFAULT NULL,
  `Image` varchar(255) DEFAULT NULL,
  `Email` varchar(255) DEFAULT NULL,
  `CustomerFrom` varchar(100) DEFAULT NULL,
  `HomeAdr` varchar(200) DEFAULT NULL,
  `LastCon` varchar(200) DEFAULT NULL,
  `HomeImg` varchar(255) DEFAULT NULL,
  `Dimensions` varchar(10) DEFAULT NULL,
  `Rooms` varchar(10) DEFAULT NULL,
  `SmartSystem` varchar(5) DEFAULT NULL,
  `CustomerID` varchar(10) DEFAULT NULL,
  `AuthenticatorKey` varchar(20) DEFAULT NULL,
  `EmergencyKey` varchar(10) DEFAULT NULL,
  `EmergencyPerson` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `Appliances` (
  `CustomerID` varchar(50) DEFAULT NULL,
  `Gate` varchar(50) DEFAULT NULL,
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
`Id` varchar(50) DEFAULT NULL,  
`Name` varchar(200) DEFAULT NULL,  
`Area` varchar(100) DEFAULT NULL,  
`State` varchar(100) DEFAULT NULL,  
`ImageOn` varchar(255) DEFAULT NULL,  
`ImageOff` varchar(255) DEFAULT NULL,  
`update_time` varchar(100) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `Logs` (  
`CustomerID` varchar(10) DEFAULT NULL,  
`IncidentName` varchar(100) DEFAULT NULL,  
`Timestamp` varchar(100) DEFAULT NULL,  
`InformedClient` varchar(100) DEFAULT NULL,  
`ID` int(11) NOT NULL AUTO_INCREMENT,  
PRIMARY KEY (`ID`)  
) ENGINE=InnoDB AUTO_INCREMENT=366 DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `Preferences` (  
`CustomerID` varchar(10) DEFAULT NULL,  
`TypeOfPreference` varchar(100) DEFAULT NULL,  
`RequiresAction` varchar(10) DEFAULT NULL,  
`Title` varchar(100) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `Sensors` (  
`CustomerID` varchar(100) DEFAULT NULL,  
`Gate` varchar(50) DEFAULT NULL,  
`Id` varchar(50) DEFAULT NULL,  
`Name` varchar(100) DEFAULT NULL,  
`Area` varchar(100) DEFAULT NULL,  
`Value` varchar(100) DEFAULT NULL,  
`Image` varchar(255) DEFAULT NULL,  
`update_time` varchar(100) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `Surveillance` (  
`CustomerID` varchar(100) DEFAULT NULL,  
`SurveillanceLink` varchar(200) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

## ΚΕΦΑΛΑΙΟ 8

### ΚΕΝΤΡΙΚΟ ΠΡΟΓΡΑΜΜΑ ΔΙΑΜΕΣΟΛΑΒΗΣΗΣ

#### 8.1 Γλώσσα προγραμματισμού C# και πρόγραμμα ανάπτυξης εφαρμογής

Για να δημιουργήσουμε την εφαρμογή που θα επικοινωνεί με τις πλακέτες Arduino και τη βάση δεδομένων επιλέξαμε την γλώσσα προγραμματισμού C# (C Sharp) η οποία μπορεί να χρησιμοποιηθεί για να κατασκευαστούν τόσο παραθυρικές όσο και μη παραθυρικές εφαρμογές μέσω του προγράμματος Microsoft Visual Studio.

Η C# είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού πράγμα που σημαίνει ότι χρησιμοποιούνται οικογένειες αντικειμένων που ονομάζονται κλάσεις. Με αυτόν τον τρόπο επιτυγχάνεται ελαχιστοποίηση του κώδικα και ομαδοποίηση των ιδιοτήτων που περιβάλλουν ένα αντικείμενο.

Το πρόγραμμα Microsoft Visual Studio υποστηρίζει την ανάπτυξη εφαρμογών σε αρκετές γλώσσες προγραμματισμού αλλά ο προγραμματιστής μπορεί πραγματοποιώντας λήψη πρόσθετων πακέτων από την διαδικτυακή κοινότητα της Microsoft να επεκτείνει το φάσμα γλωσσών που υποστηρίζονται από το πρόγραμμα.

#### 8.2 Εισαγωγικά για το πρόγραμμα

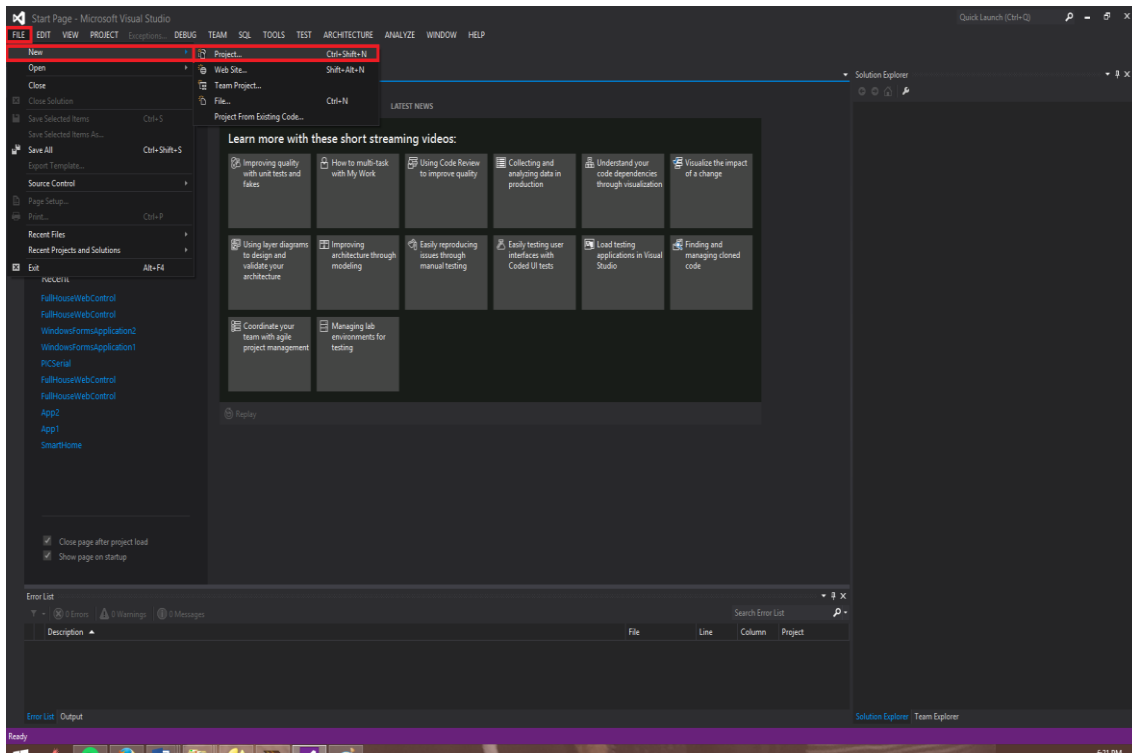
Το πρόγραμμα διαμεσολάβησης που κατασκευάσαμε αποφασίσαμε να είναι παραθυρικό (GUI) επιτρέποντας έτσι στο χρήστη μέσω του ποντικιού και του πληκτρολογίου του να πραγματοποιήσει ρυθμίσεις πράγμα που θα ήταν σχεδόν αδύνατο να κάνει σε ένα πρόγραμμα που θα εκτελούταν στην γραμμή εντολών. Τα γραφικά αυτά στοιχεία θα δημιουργηθούν αυτόματα από το Microsoft Visual Studio με την γλώσσα εμφάνισης XAML και θα αποτελέσουν το frontend της εφαρμογής.

Το πρόγραμμα είναι κατασκευασμένο στην τεχνολογία πλαισίου .NET 4.5 της Microsoft και είναι συμβατό για χρήση με τα λειτουργικά συστήματα Windows Vista, Windows 7, Windows 8, Windows 8.1 και Windows 10.

Οι απαιτήσεις του προγράμματος είναι η ύπαρξη σύνδεσης με το διαδίκτυο (ενσύρματη ή ασύρματη), η ύπαρξη προγράμματος οδήγησης (drivers) για τις πλακέτες Arduino και η ύπαρξη του πλαισίου .NET 4.5 που εφόσον δεν υπάρχει εγκαθίσταται μετά από ενημέρωση αυτόματα από το λειτουργικό σύστημα μετά την έναρξη της εφαρμογής.

### 8.3 Δημιουργία παραθυρικής εφαρμογής με το Microsoft Visual Studio

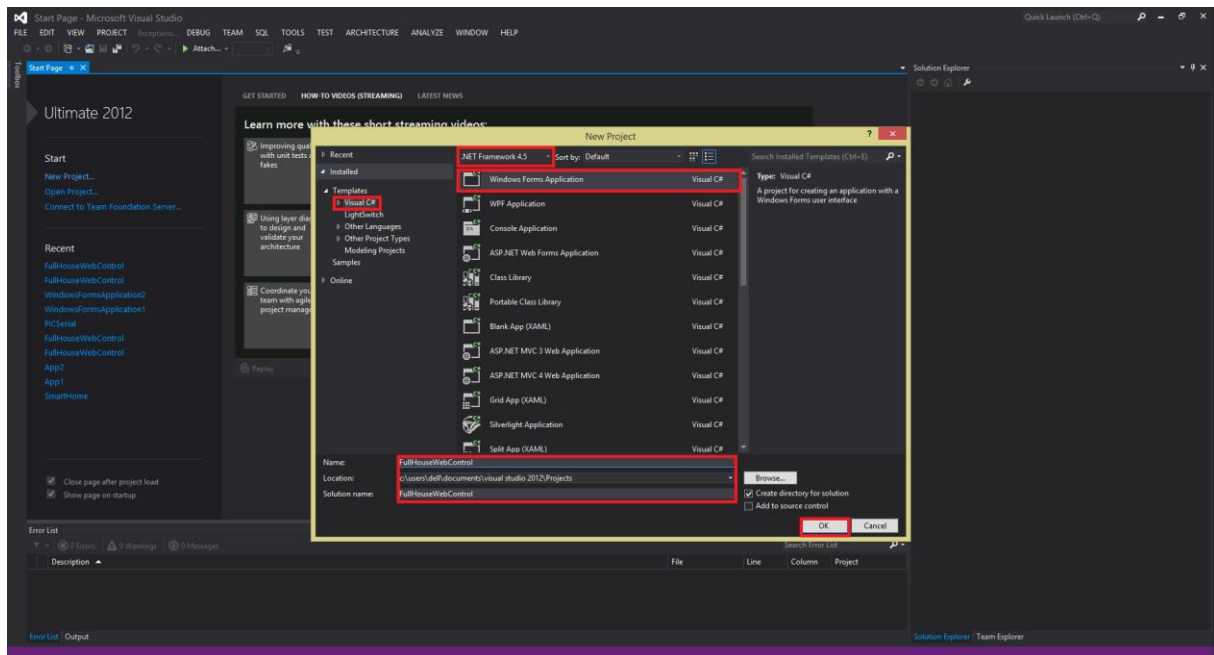
Εισερχόμενοι στο πρόγραμμα μπορούμε πολύ εύκολα να δημιουργήσουμε ένα νέο πρόγραμμα ακολουθώντας το κεντρικό μενού όπως φαίνεται και παρακάτω.



Εικόνα 73: Διαδικασία δημιουργίας νέου προγράμματος στο Microsoft Visual Studio

Έπειτα θα μας εμφανιστεί ένα μενού με επιλογές στο οποίο μεταξύ άλλων πρέπει με μεγάλη προσοχή να διαλέξουμε την γλώσσα προγραμματισμού που θέλουμε, την μορφή της εφαρμογής μας (παραθυρική ή όχι), την έκδοση .NET στην οποία θέλουμε να την κατασκευάσουμε, τον τίτλο του προγράμματός μας και την τοποθεσία στον σκληρό δίσκο στην οποία θα αποθηκευτεί.

Η σωστή επιλογή του ονόματος είναι πολύ σημαντική επειδή μετά την δημιουργία του project θα καταχωρηθεί σε πολλά αρχεία και θα είναι αδύνατη η αλλαγή του. Επίσης μεγάλη σημασία πρέπει να δοθεί και στην επιλογή του πλαισίου .NET (Framework) στο οποίο θα κατασκευαστεί το πρόγραμμα. Πάντα επιλέγουμε την νεότερη και πιο σταθερή έκδοση του πλαισίου που είναι διαθέσιμη έτσι ώστε η εφαρμογή μας να χρησιμοποιεί τις πιο καινούργιες βιβλιοθήκες κλάσεων και ιδιοτήτων. Για να κατασκευάσουμε παραθυρική εφαρμογή διαλέγουμε την επιλογή Windows Forms Application.



Εικόνα 74: Διαδικασία δημιουργίας νέας παραθυρικής εφαρμογής στο Microsoft Visual Studio

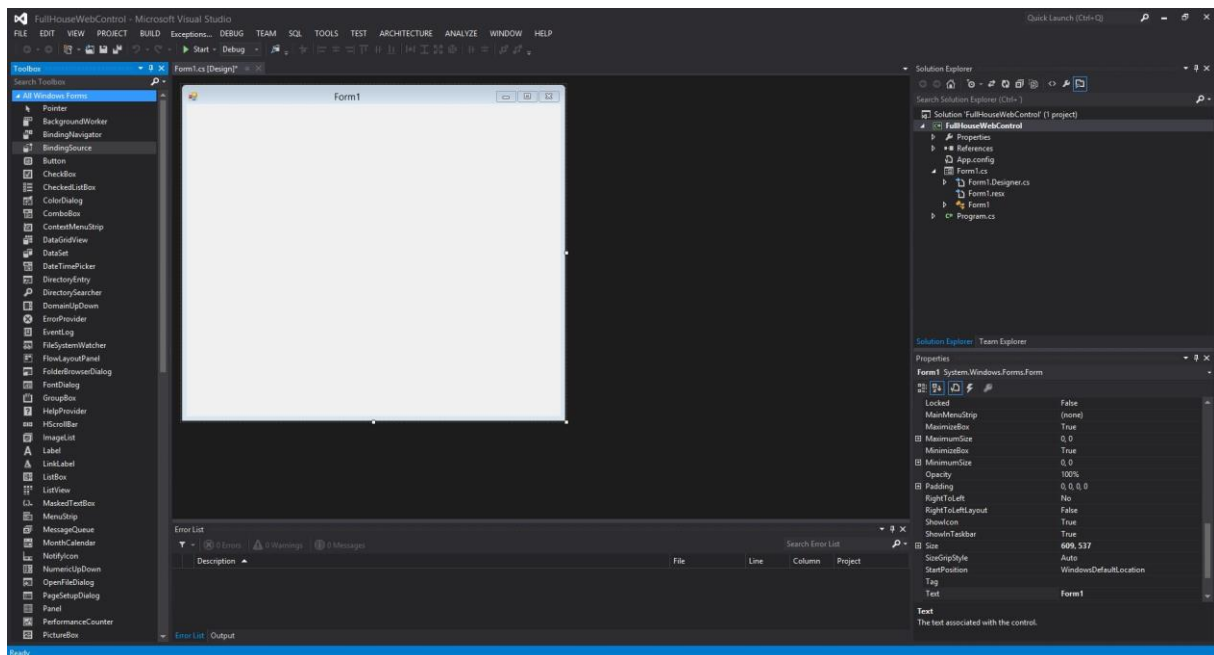
Μόλις πραγματοποιήσουμε τις παραπάνω ενέργειες μετά από σύντομο χρονικό διάστημα το Microsoft Visual Studio έχει δημιουργήσει όλα τα απαραίτητα αρχεία που θα χρειαστούν και μας μεταφέρει στην καρτέλα σχεδιασμού του πρώτου παραθύρου της εφαρμογής μας.

Στο αριστερό μέρος της οθόνης μας βλέπουμε το κουτί εργαλείων (Toolbox) το οποίο εμπεριέχει όλα τα απαραίτητα οπτικά αντικείμενα, για παράδειγμα εισαγωγή εικόνας, εισαγωγή κειμένου, εισαγωγή περιοχής πληκτρολόγησης κειμένου (Textbox), εισαγωγής κουμπιών αλληλεπίδρασης κ.α., τα οποία θα χρειαστούμε κατά την κατασκευή της εφαρμογής μας. Για όποιο αντικείμενο διαλέξουμε είτε αυτό είναι το κύριο παράθυρο είτε κάποιο άλλο οπτικό αντικείμενο, το Microsoft Visual Studio μας δίνει την δυνατότητα στο κάτω δεξιό κομμάτι της οθόνης μας να δούμε και να τροποποιήσουμε τις οπτικές και λειτουργικές του ιδιότητες μέσω του μενού Properties. Έτσι μπορούμε πολύ εύκολα να αλλάξουμε το όνομα, το μέγεθος, το χρώμα, το εικονίδιο του παραθύρου και άλλα πολλά.

Όσον αφορά την τοποθέτηση των αντικειμένων στο παράθυρο το Microsoft Visual Studio μας επιτρέπει να χρησιμοποιούμε το ποντίκι μας και να τα τοποθετούμε όπου εμείς επιθυμούμε κάνοντας τα πράγματα πολύ εύκολα.

Η ύπαρξη και τοποθέτηση όλων αυτών των αντικειμένων στο κεντρικό παράθυρο μεταφράζεται αυτόματα από το Microsoft Visual Studio σε κώδικα εμφάνισης XAML

επιτρέποντας έτσι σε χρήστες που δεν είναι εξοικειωμένοι με την γλώσσα XAML να δημιουργήσουν τα δικά τους παράθυρα.



Εικόνα 75: Παράθυρο σχεδίασης (Designer)

Μετά την τοποθέτηση των αντικειμένων υπάρχει η περίπτωση να θέλουμε κάποια αντικείμενα όπως ένα κουμπί να αποκτήσουν και λειτουργικό χαρακτήρα. Για αυτό τον λόγο χρησιμοποιούμε τα συμβάντα (events) στα οποία μπορούμε να έχουμε πρόσβαση από το μενού ιδιότητες (properties) που αναφέραμε πρωτύτερα. Έτσι μπορούμε να ανιχνεύσουμε διάφορες ενέργειες που πραγματοποιεί ο χρήστης σε σχέση με τα αντικείμενά μας όπως είναι για παράδειγμα το πάτημα του αντικειμένου μέσω του αριστερού κλικ του ποντικιού, η διέλευση του κέρσορα πάνω από το αντικείμενο και άλλων. Αν θέλουμε κάποιο από αυτά να ανιχνευθεί απλά το διαλέγουμε με διπλό κλικ. Το πρόγραμμα έπειτα δημιουργεί έναν χειριστή για αυτό το συμβάν (event handler) ο οποίος είναι αρμόδιος εφόσον ανιχνεύσει ότι συμβαίνει το συγκεκριμένο συμβάν για το οποίο έχει οριστεί να εκτελέσει τον πηγαίο κώδικα που του έχει ορίσει ο προγραμματιστής.

## 8.4 Κώδικας πρωτοκόλλου επικοινωνίας

Πριν ξεκινήσουμε την ενασχόλησή μας με τα διαφορετικά παράθυρα που θα έχει η εφαρμογή μας, θα κατασκευάσουμε το πρωτόκολλο επικοινωνίας που αναφέραμε στο κεφάλαιο 6.3 αλλά αυτήν την φορά από την πλευρά αυτού του προγράμματος. Η



υλοποίησή του θα γίνει σύμφωνα με το υλικό που χρησιμοποιήσαμε από τους συνδέσμους [16], [17] αφότου το τροποποιήσουμε για να υποστηρίξει την ταυτόχρονη σύνδεση – ανίχνευση παραπάνω από μίας συσκευής Arduino.

Ξεκινάμε λοιπόν δημιουργώντας μέσα στην λύση μας (project) ένα νέο αρχείο. Όλα τα αρχεία που θα δούμε σε αυτό το πρόγραμμα αποτελούνται από τις εξής περιοχές:

- Περιοχή namespace και public partial class()
- Περιοχή public Form1()
- Εξωτερικός χώρος εκτός του namespace και public partial class()

Η περιοχή namespace παίρνει το όνομά της αυτόματα από το όνομα της λύσης (project) και εμπεριέχει οτιδήποτε έχει σχέση με το αρχείο στο οποίο βρίσκεται. Η κλάση public partial class βρίσκεται μέσα στην περιοχή namespace και περιβάλλει και αυτή όλα τα στοιχεία του αρχείου. Μέσα σε αυτήν εμπεριέχονται όλες οι κλάσεις, λειτουργίες, δηλώσεις μεταβλητών καθώς και η public Form1().

Η περιοχή public Form1() παίρνει την ονομασία της από το όνομα που δίνουμε στο παράθυρο στο οποίο αντιστοιχίζεται. Η λειτουργία αυτή εμφανίζεται μόνο στις σελίδες παραθύρων και εμπεριέχει τις ρουτίνες που θα εκτελεστούν μόλις ξεκινήσει το πρόγραμμα. Πάντα περιέχει την ρουτίνα InitializeComponent() η οποία όπως λέει και το όνομά της κάνει προετοιμασία όλων των γραφικών στοιχείων που θα απεικονιστούν στο παράθυρο.

Στον εξωτερικό χώρο μπορούμε να τοποθετήσουμε τις βιβλιοθήκες τις οποίες θα χρησιμοποιήσουμε πληκτρολογώντας την εντολή using και την βιβλιοθήκη που θέλουμε.

Ξεκινώντας λοιπόν θα δηλώσουμε με την βοήθεια της εντολής using τις βιβλιοθήκες τις οποίες θα χρησιμοποιήσουμε σε αυτήν την σελίδα.

```
using System;  
using System.Collections.Generic;  
using System.Text;  
using System.IO.Ports;  
using System.Threading;  
using System.Data;
```

Στον χώρο εξωτερικά από τις δύο μας λειτουργίες θα δηλώσουμε κάποιες μεταβλητές στις οποίες θέλουμε να έχουν πρόσβαση και οι δύο λειτουργίες.



```

public string port = ""; //Όνομασία πόρτας επικοινωνίας USB
public SerialPort currentPort; //Αριθμός πόρτας επικοινωνίας USB
public bool portConnection = false; //Κατάσταση σύνδεσης πόρτας USB

```

εξ ορισμού

Έπειτα ακολουθεί η λειτουργία επίτευξης σύνδεσης της συσκευής Arduino με το πρόγραμμά μας.

```

/*
    Λειτουργία επίτευξης σύνδεσης με τις πλακέτες Arduino. Αυτή η
    λειτουργία δέχεται ως όρισμα την ταχύτητα επικοινωνίας, μία συμβολοσειρά και
    τρεις αριθμούς
    και διατρέπει όλες τις διαθέσιμες πόρτες επικοινωνίας USB
    προσπαθώντας να δει αν κάποια συσκευή αναπαράγει αυτό το μήνυμα. Εφόσον την
    βρει επιστρέφει
    το όνομα και τον αριθμό της πόρτας επικοινωνίας της συσκευής.
*/
public Boolean connect(int baud, string recognizeText, byte
paramone, byte paramtwo, byte paramthree)
{
    try
    {
        byte[] buffer = new byte[3]; //Πίνακας αποθήκευσης των
τριών αριθμών
        buffer[0] = Convert.ToByte(paramone); //Μετατροπή του
πρώτου αριθμού σε byte
        buffer[1] = Convert.ToByte(paramtwo); //Μετατροπή του
δεύτερου αριθμού σε byte
        buffer[2] = Convert.ToByte(paramthree); //Μετατροπή του
 τρίτου αριθμού σε byte

        int intReturnASCII = 0; //Μεταβλητή μέτρησης αριθμού ψηφίων
που θα διαβαστούν
        string[] ports = SerialPort.GetPortNames(); //Αποθήκευση
διαθέσιμων ονομάτων συσκευών USB σε έναν πίνακα
        foreach (string newport in ports) //Για κάθε μία πόρτα στην
λίστα
        {
            currentPort = new SerialPort(newport, baud); //Επίτευξη
σύνδεσης στην πόρτα αυτή με ταχύτητα 9600
            currentPort.Open(); //Άνοιγμα θύρας
            currentPort.Write(buffer, 0, 3); //Εγγραφή μηνύματος
            Thread.Sleep(200); //Αναμονή 200ms
            int count = currentPort.BytesToRead; //Ανάγνωση
            μηνύματος που προέρχεται από την συσκευή USB
            string returnMessage = ""; //Αποθήκευση σε πίνακα
            while (count > 0) //Αν υπάρχει περιεχόμενο στο μήνυμα
            {
                intReturnASCII = currentPort.ReadByte();
                //Αποθήκευση σε μεταβλητή του περιεχομένου του μηνύματος
                returnMessage = returnMessage +
                Convert.ToChar(intReturnASCII); //Μετατροπή σε ASCII χαρακτήρες και προσθήκη
                στο τέλος της μεταβλητής
                count--;
            }

            currentPort.Close(); //Κλείσιμο θύρας επικοινωνίας

```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
port = newport; //Συνέχεια με την επόμενη διαθέσιμη
πόρτα της λίστας
    if (returnMessage.Contains(recognizeText)) //Αν το
περιεχόμενο που δώσαμε και το μήνυμα της θύρας ήταν ίδια επέστρεψε αλήθεια
    {
        return true;
    }
}
return false; //Άλλιώς επέστρεψε λάθος
}
finally { }
}
```

Τέλος ακολουθεί η λειτουργία επικοινωνίας μεταξύ των πλακετών Arduino και του προγράμματος.

```
//Λειτουργία επικοινωνίας μεταξύ θυρών στις οποίες έχει διασφαλισθεί σύνδεση
σύμφωνα με την μέθοδο που βρίσκεται παραπάνω
public string message(byte paramone, byte paramtwo, byte
paramthree)
{
    byte[] buffer = new byte[3]; //Πίνακας αποθήκευσης των τριών
αριθμών
    buffer[0] = Convert.ToByte(paramone); //Μετατροπή του πρώτου
αριθμού σε byte
    buffer[1] = Convert.ToByte(paramtwo); //Μετατροπή του δεύτερου
αριθμού σε byte
    buffer[2] = Convert.ToByte(paramthree); //Μετατροπή του τρίτου
αριθμού σε byte
    currentPort.Open(); //Άνοιγμα θύρας
    currentPort.Write(buffer, 0, 3); //Εγγραφή μηνύματος
    int intReturnASCII = 0; //Μεταβλητή μέτρησης αριθμού ψηφίων που
θα διαβαστούν
    Thread.Sleep(200); //Αναμονή 200ms
    int count = currentPort.BytesToRead; //Ανάγνωση μηνύματος που
προέρχεται από την συσκευή USB
    string returnMessage = ""; //Αποθήκευση σε πίνακα
    while (count > 0) //Αν υπάρχει περιεχόμενο στο μήνυμα
    {
        intReturnASCII = currentPort.ReadByte(); //Αποθήκευση σε
μεταβλητή του περιεχομένου του μηνύματος
        returnMessage = returnMessage +
Convert.ToChar(intReturnASCII); //Μετατροπή σε ASCII χαρακτήρες και προσθήκη
στο τέλος της μεταβλητής
        count--;
    }
    currentPort.Close(); //Κλείσιμο θύρας επικοινωνίας
    return returnMessage; //Επιστροφή μηνύματος από συσκευή USB
}
```

## 8.5 Κεντρικό παράθυρο εφαρμογής

Αφού κατασκευάσαμε το αρχείο που περιέχει τον τρόπο επίτευξης επικοινωνίας και ανταλλαγής πληροφοριών αποθηκεύουμε το αρχείο και πάμε να ξεκινήσουμε την κατασκευή της πρώτης σελίδας του προγράμματός μας.

Οι βιβλιοθήκες που χρησιμοποιήσαμε οι οποίες δεν ήταν τοποθετημένες στην λύση μας από την αρχή είναι η βιβλιοθήκη για διεπαφή με την βάση MySQL, η βιβλιοθήκη επικοινωνίας με τις συσκευές USB και η γενικότερη βιβλιοθήκη παραλληλοποίησης.

Από κάτω ακολουθεί η δήλωση των βιβλιοθηκών της πρώτης σελίδας.

```
using System;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Collections.Generic;
using System.Windows.Forms;
using System.IO.Ports; //Βιβλιοθήκη για τις συσκευές USB
using MySql.Data.MySqlClient; //Βιβλιοθήκη της βάσεως δεδομένων
using System.Threading; //Βιβλιοθήκη παραλληλοποίησης
using System.Threading.Tasks; //Βιβλιοθήκη παραλληλοποίησης
using System.Linq;
using System.Diagnostics;
```

Έπειτα θα ακολουθήσει η δήλωση μεταβλητών στον εξωτερικό χώρο στις οποίες θέλουμε να έχουν πρόσβαση όλες οι λειτουργίες κατά την εκτέλεση του προγράμματος.

```
public static string CustomerID = "1"; //Κωδικός πελάτη 1 εξ ορισμού
public static Boolean ButtonPref = false; //Κατάσταση λειτουργίας ελέγχου
διακοπιών αφής
int ArduinoID = 0; //Μεταβλητή μέτρησης του αριθμού των πλακετών που θα
συνδεθούν
int SurveillanceButtonStatus = 0; //Μεταβλητή κατάστασης κουμπιού
παρακολούθησης χώρου
Process myProcess;
List<communicator> Communicators = new List<communicator>();
```

Αρχικά θα δημιουργήσουμε την λειτουργία η οποία θα καλείται στην έναρξη του προγράμματος και θα κάνει αυτόματη αναζήτηση στον υπολογιστή του χρήστη για διαθέσιμες συσκευές Arduino που ακολουθούν τους κανόνες του πρωτοκόλλου.

```
private void ConnectArduino ()
{
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
communicator comport = new communicator(); //Αίτηση για νέα
σύζευξη επικοινωνίας του προγράμματος με πλακέτα Arduino

//Μήνυμα σύνδεσης πλακέτας συσκευών: Device 0, 10, 20,30
//Μήνυμα σύνδεσης πλακέτας αισθητηρίων: Device 1, 11, 21, 31

if (comport.connect(9600, "Device " + ArduinoID,
(byte) (ArduinoID + 10), (byte) (ArduinoID + 20), (byte) (ArduinoID + 30)))
{
    if (ArduinoID == 0)
    {
        label1.Text = "Connection of Arduino 1 Successful -
Connected to " + comport.port;
    }
    else if (ArduinoID == 1)
    {
        label2.Text = "Connection of Arduino 2 Successful -
Connected to " + comport.port;
    }
    comport.portConnection = true;

    Communicators.Add(comport);
    ArduinoID++;
}

//Σε περίπτωση μη σύνδεσης μιας από τις δύο πλακέτες ενημέρωση
του χρήστη
else
{
    if (ArduinoID == 0)
    {
        label1.Text = "Arduino Number : 1 is not connected . .
. ";
        comport.portConnection = false;
        button1.Enabled = false;
        ArduinoID++;
    }
    else if (ArduinoID == 1)
    {
        label2.Text = "Arduino Number : 2 is not connected . .
. ";
        comport.portConnection = false;
        button1.Enabled = false;
    }
}
}
```

Στην συνέχεια αφού έχουν συνδεθεί με επιτυχία δύο πλακέτες Arduino το πρόγραμμα θα ξεκλειδώσει το πλήκτρο έναρξης λειτουργίας που κανονικά είναι απενεργοποιημένο. Με το πάτημα του κουμπιού αυτού ξεκινάει μια σειρά ενεργειών.

Η πρώτη ενέργεια που πραγματοποιείται είναι το Arduino των συσκευών να συγχρονιστεί με την βάση δεδομένων.

```

public void SetUp()
{
    ρυθμίσειςToolStripMenuItem.Enabled = false; //Αν έχει ξεκινήσει
    η σύνδεση το ID πελάτη δεν μπορεί να αλλάξει πια

    //Αν επιτευχθεί η σύνδεση με την πλακέτα συσκευών ενημέρωση
    όλων των συσκευών σύμφωνα με την βάση δεδομένων
    if (Communicators[0].portConnection == true)
    {
        string ConnectionString =
        "datasource=83.212.120.81;port=3306;username=Arduino;password=ard1234";
        MySqlConnection myConn = new
        MySqlConnection(ConnectionString);
        MySqlDataReader reader = null;
        try
        {
            myConn.Open();
            using (MySqlCommand cmd = new MySqlCommand("Select
            Id,State from Full_House_Web_Control.Appliances WHERE
            CustomerID="+CustomerID, myConn))
            {

                reader = cmd.ExecuteReader();
                while (reader.Read())
                {

                    int Value = int.Parse(reader.GetString(1));
                    byte ID = (byte)int.Parse(reader.GetString(0));

                    //Αποστολή αίτησης 5,αριθμός συσκευής,1 για
                    ενεργοποίηση συσκευής και 5,αριθμός συσκευής,0 για απενεργοποίηση συσκευής
                    if (Value == 1)
                    {

                        Communicators[0].message(5, ID, 1);

                    }
                    else
                    {

                        Communicators[0].message(5, ID, 0);

                    }

                }

            }
            myConn.Close();
        }
        catch { }
    }
}

```

```

        finally
        {

        }

    }

}

```

Στην συνέχεια πραγματοποιείται ασύγχρονα η επικοινωνία με την κάθε πλακέτα Arduino και το πρόγραμμα. Έτσι επιτυγχάνουμε με μόνο δύο νήματα του επεξεργαστή να μειώσουμε τον χρόνο εκτέλεσης του προγράμματος σημαντικά.

Ξεκινώντας από το Arduino που χειρίζεται τις συσκευές δημιουργήσαμε μία λειτουργία η οποία ελέγχει για τις αλλαγές που έχουν πραγματοποιηθεί στην βάση δεδομένων σε σχέση με τις συσκευές της οικίας τα τελευταία 10 δευτερόλεπτα. Αυτό συμβαίνει αφενός για να είμαστε σίγουροι ότι αν υπάρξει κάποιο πρόβλημα με την ταχύτητα σύνδεσης του χρήστη στο διαδίκτυο η πλακέτα θα ενημερωθεί για κάποια αλλαγή και αφετέρου γιατί θα ήταν ανούσιο η πλακέτα να ενημερώνεται για αλλαγές που έγιναν για παράδειγμα πριν εξήντα δευτερόλεπτα οι οποίες θα είχαν επίπτωση στην συνολική της απόδοση. Ο μέσος χρόνος ενημέρωσης της πλακέτας από την στιγμή που θα γίνει μία αλλαγή είναι περίπου ένα δευτερόλεπτο.

```

//Λειτουργία ενημέρωσης Arduino συσκευών για τυχόν αλλαγές που προέκυψαν τα
τελευταία 10 δευτερόλεπτα στην βάση δεδομένων
public async void CheckStatus()
{
    //Παραλληλοποιημένη λειτουργία
    await Task.Run(() =>
    {
        if (Communicators[0].portConnection == true)
        {
            string connectionString =
"datasource=83.212.120.81;port=3306;username=Arduino;password=ard1234";
            MySqlConnection myConn = new
MySqlConnection(connectionString);
            MySqlDataReader reader = null;
            try
            {

                myConn.Open();
                using (MySqlCommand cmd = new MySqlCommand("Select
Id,State from Full_House_Web_Control.Appliances WHERE update_time > (NOW()
- INTERVAL 10 SECOND) AND CustomerID="+CustomerID, myConn))
                {

                    reader = cmd.ExecuteReader();
                    while (reader.Read())

```

```
        {  
  
            int Value = int.Parse(reader.GetString(1));  
            byte ID =  
(byte)int.Parse(reader.GetString(0));  
  
            if (Value == 1)  
            {  
  
                Communicators[0].message(5, ID, 1);  
  
            }  
            else  
            {  
  
                Communicators[0].message(5, ID, 0);  
  
            }  
  
        }  
        myConn.Close();  
    }  
  
    }  
  
    catch {  
  
    }  
  
    finally  
    {  
  
    }  
    if (ButtonPref == true)  
    {  
        CheckButtons(); //Κλήση λειτουργίας ελέγχου για  
        τυχόν πάτημα κάποιο αισθητήρα αφής  
    }  
    }  
});  
  
    CheckStatus(); //Κλήση της ίδιας λειτουργίας ξανά μετά το τέλος  
}
```

Από ότι παρατηρούμε κάποιες γραμμές πριν το τέλος καλείται μία λειτουργία με την ονομασία CheckButtons(). Η λειτουργία αυτή είναι αρμόδια για την ενημέρωση της βάσεως δεδομένων σε περίπτωση που πατηθεί κάποιος διακόπτης αφής της οικίας.

Προς το παρόν η λειτουργία όπως είναι υλοποιημένη ελέγχει για τους διακόπτες που βρίσκονται στις ακίδες 2, 3, 6, 9, 12, 44 του Arduino. Η λειτουργία αυτή αποτελεί προαιρετική λειτουργία αφενός λόγω του ότι σε κάποια σπίτια μπορεί να μην υπάρχουν διακόπτες αφής οι οποίοι είναι συμβατοί με το σύστημα και αφετέρου λόγω της καθυστέρησης που προκαλείται στο Arduino συσκευών. Το Arduino συσκευών όπως είχαμε αναφέρει σε προηγούμενο κεφάλαιο μόνο δέχεται πληροφορίες. Η χρήση όμως αυτής της λειτουργίας σημαίνει ότι ενώ η συσκευή απλά περίμενε να δεχτεί και να εκτελέσει εντολές, τώρα πρέπει να ελέγχει σε κάθε επανάληψή της και την κατάσταση των έξι διακοπών. Έτσι δέχεται έξι αιτήματα από την εφαρμογή μας κάθε φορά όποια και αν είναι η κατάσταση των διακοπών τα οποία πρέπει να επεξεργασθεί. Για αυτόν τον λόγο έχει προστεθεί επιλογή ενεργοποίησης/απενεργοποίησης αυτής της λειτουργίας στο παράθυρο 4 της εφαρμογής μας που θα δούμε αργότερα.

```
//Λειτουργία ελέγχου πατήματος κάποιου από τα πλήκτρα αφής
public void CheckButtons ()
{

    string connectionString =
"datasource=83.212.120.81;port=3306;username=Arduino;password=ard1234";
    MySqlConnection myConn = new MySqlConnection(connectionString);

    myConn.Open();
    string[] buttons = new string[7];
    int c = 0;
    int j = 0;
    for (int i = 0; i < 6; i++)
    {
        if (i == 0) { j = 2; }
        if (i == 1) { j = 3; }
        if (i == 2) { j = 6; }
        if (i == 3) { j = 9; }
        if (i == 4) { j = 12; }
        if (i == 5) { j = 44; }

        byte k = (byte)j;

        /*Κάνει αίτηση για την ανάγνωση της κατάστασης όλων των
κουμπιών αφής και αν κάποιο είναι πατημένο ενημερώνει το αντίστοιχο φως του
αντίστοιχου χώρου
στην βάση δεδομένων με αντίστροφη λογική. Πρακτικά αν ένας
διακόπτης είναι πατημένος και το αντίστοιχο φως είναι απενεργοποιημένο η
βάση θα ενημερωθεί
ότι το φως αυτού του χώρου είναι πια ενεργοποιημένο.

Η μορφή της αίτησης είναι η εξής: 5,2,αριθμός διακόπτη
*/
        buttons[c] = Communicators[0].message(5, 2, k);
    }
}
```



```

        if (buttons[c] == "0")
        {

        }
        else if (buttons[c] == "1")
        {
            using (MySqlCommand cmd = new MySqlCommand("UPDATE
Full_House_Web_Control.Appliances SET State=NOT State,update_time = NOW()
WHERE Id=@ID AND CustomerID="+CustomerID, myConn))
            {

                cmd.Parameters.AddWithValue("@ID", j);
                cmd.ExecuteNonQuery();

            }
        }
        c++;
    }

    myConn.Close();
}

```

Από την άλλη πλευρά έχουμε τις λειτουργίες του Arduino που χειρίζεται τα αισθητήρια της οικίας. Η λειτουργία CheckSensors() ζητάει με μορφή αίτησης τις τιμές όλων των αισθητηρίων της οικίας και αφού τις λάβει τις δημοσιεύει στην βάση δεδομένων όπου από εκεί μέσω της ιστοσελίδας μπορεί να τις δει ο χρήστης.

```

//Λειτουργία αίτησης των τιμών των αισθητηρίων από το Arduino με τα
αισθητήρια και ενημέρωση της βάσεως δεδομένων με τις τιμές αυτές
public async void CheckSensors()
{
    //Παραλληλοποιημένη λειτουργία
    await Task.Run(() =>
    {

        if (Communicators[1].portConnection == true)
        {
            try
            {
                CheckAlerts(); //Κλήση λειτουργίας ελέγχου
                συστήματος συναγερμού, πυρασφάλειας και πλήκτρου έκτακτης ανάγκης
                string[] sensors = new string[22];

                string ConnectionString =
                "datasource=83.212.120.81;port=3306;username=Arduino;password=ard1234";
            }
            catch { }
        }
    });
}

```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
        MySqlConnection myConn = new
MySqlConnection(ConnectionString);

        myConn.Open();
        //Ζητάει από το Arduino με τα αισθητήρια τις τιμές
τους με την μορφή αίτησης: 2,1,αριθμός αισθητηρίου και ενημερώνει την βάση
δεδομένων
        for (int i = 1; i < 22; i++)
        {
            byte k = (byte)i;

            sensors[i] = Communicators[1].message(2, 1, k);
            using (MySqlCommand cmd = new
MySqlCommand("UPDATE Full_House_Web_Control.Sensors SET
Value=@Value,update_time = NOW() WHERE Id=@ID AND CustomerID="+CustomerID,
myConn))
            {

                cmd.Parameters.AddWithValue("@ID", i);
                cmd.Parameters.AddWithValue("@Value",

sensors[i]);

                cmd.ExecuteNonQuery();

            }

        }

        myConn.Close();

    }
    catch
    {
    }
    finally
    {

    }

}
});

    CheckSensors(); //Κλήση της ίδιας λειτουργίας ξανά μετά το
τέλος εκτέλεσής της
}
```

Στην αρχή της εκτέλεσης αυτής της λειτουργίας παρατηρούμε ότι εμπεριέχεται και η λειτουργία CheckAlerts(). Η λειτουργία αυτή αναλαμβάνει μέσω τριών αιτημάτων να ενημερωθεί για τα σημαντικά συμβάντα της οικίας όπως είναι ο πιθανός κίνδυνος

πυρκαγιάς, η παραβίαση της οικίας την ώρα που ήταν ενεργοποιημένο το σύστημα συναγερμού και η κλήση σε βοήθεια μέσω της ενεργοποίησης του πλήκτρου έκτακτης ανάγκης της οικίας. Εφόσον συμβεί κάτι από τα παραπάνω πρώτα ελέγχει αν έχει γίνει καταχώρηση για την ίδια κατηγορία συμβάντος στην βάση δεδομένων τα τελευταία δέκα λεπτά και αν δεν έχει γίνει την πραγματοποιεί. Από εκεί και πέρα αναλαμβάνει δουλειά ο διακομιστής ηλεκτρονικής αλληλογραφίας μας ο οποίος αναλαμβάνει να ενημερώσει τον χρήστη εντός περίπου ενός λεπτού.

//Λειτουργία ελέγχου συστήματος πυρασφάλειας, κουμπιού έκτακτης ανάγκης και συστήματος συναγερμού

```

public void CheckAlerts ()
{

    if (Communicators[1].portConnection == true)
    {
        string ConnectionString =
"datasource=83.212.120.81;port=3306;username=Arduino;password=ard1234";
        MySqlConnection myConn = new
MySqlConnection(ConnectionString);
        MySqlConnection myConn2 = new
MySqlConnection(ConnectionString);

        /*
        Το πρόγραμμα ζητάει από το Arduino με τα αισθητήρια να του
        κοινοποιήσει αν στο σπίτι έχει ενεργοποιηθεί το σύστημα πυρασφάλειας,
        σύστημα συναγερμού ή
        ο διακόπτης έκτακτης ανάγκης. Αν κάποιο από αυτά έχει
        ενεργοποιηθεί κοιτάει την βάση δεδομένων για να δει αν έχει αναφέρει κάποια
        ίδια ενέργεια στον
        χρήστη τα τελευταία 10 λεπτά. Αν έχει αναφέρει δεν τον
        ειδοποιεί ξανά. Αν όμως δεν έχει αναφέρει ενημερώνει την βάση δεδομένων με
        αυτή την πληροφορία.

        Τρόπος αίτησης για σύστημα πυρασφάλειας: 7,1,96
        Τρόπος αίτησης για ενεργοποίηση πλήκτρου έκτακτης ανάγκης:
7,1,97

        Τρόπος αίτησης για σύστημα συναγερμού: 7,1,98

        */
        string[] AlertMessages = { "Empty Alert", "Fire Alert",
"Emergency Alert", "Burglar Alert" };
        string[] alarm = new string[4];
        int c = 1;

        MySqlDataReader reader = null;

        for (int i = 96; i < 99; i++)
        {
            byte k = (byte)i;

            alarm[c] = Communicators[1].message(7, 1, k);
            if (alarm[c] == "1")
            {
                string AlertName = AlertMessages[c];
    
```

```

        using (MySqlCommand cmd = new MySqlCommand("Select
Timestamp from Full_House_Web_Control.Logs WHERE IncidentName='" +
AlertName + "' AND CustomerID='" + CustomerID + "' ORDER BY ID DESC LIMIT 1",
myConn))
    {
        //SELECT
        myConn.Open();
        myConn2.Open();
        reader = cmd.ExecuteReader();
        if (reader.Read())
        {
            DateTime Timestamp = reader.GetDateTime(0);
            DateTime dtStartString = DateTime.Now;
            dtStartString = dtStartString.AddMinutes(-
10);

            if (Timestamp < dtStartString)
            {
                //INSERT
                using (MySqlCommand cmd2 = new
MySqlCommand("INSERT INTO Full_House_Web_Control.Logs VALUES
('" + CustomerID + "', @Alert, NOW(), '0', '0')", myConn2))
                {
                    cmd2.Parameters.AddWithValue("@Alert", AlertName);
                    cmd2.ExecuteNonQuery();
                }
            }
        }
        else
        {
            using (MySqlCommand cmd2 = new
MySqlCommand("INSERT INTO Full_House_Web_Control.Logs VALUES
('" + CustomerID + "', @Alert, NOW(), '0', '0')", myConn2))
            {
                cmd2.Parameters.AddWithValue("@Alert",
AlertName);
                cmd2.ExecuteNonQuery();
            }
        }
    }
}

```

```

    }
    myConn.Close ();
    myConn2.Close ();
    c++;
}

}

}

```

Αυτά συμβαίνουν όσον αφορά το πάτημα του κουμπιού έναρξης λειτουργίας. Με την ενεργοποίηση του δεξιού κουμπιού μπορούμε να θέσουμε σε λειτουργία καταγραφής το πρόγραμμα καταγραφής της κάμερας που είναι συνδεδεμένη στον υπολογιστή μας. Το πρόγραμμα καταγραφής έχει τον τίτλο OBS (Open Broadcaster Software) και είναι ελεύθερο προς λήψη στο διαδίκτυο. Μετά από την εγκατάστασή του γίνεται η ρύθμιση των παραμέτρων εγγραφής και η αποθήκευσή τους. Σε περίπτωση που ο χρήστης δεν θα χρησιμοποιήσει αυτόν τον τρόπο παρακολούθησης της οικίας του και έχει διαδικτυακές κάμερες (IP Cameras) αυτή η λειτουργία δεν χρησιμοποιείται.

Συνοψίζοντας με το πάτημα αυτού του κουμπιού ενεργοποιείται η καταγραφή μέσω του προγράμματος OBS εφόσον αυτό υπάρχει εγκατεστημένο και ρυθμισμένο σωστά στο σύστημά μας. Αν το κουμπί πατηθεί ξανά το πρόγραμμα OBS κλείνει. Σε περίπτωση που ο χρήστης κλείσει την εφαρμογή μας απενεργοποιεί και το OBS.

//Αν πατηθεί το δεξί πλήκτρο της εφαρμογής και είναι η πρώτη φορά αναζήτησε στον υπολογιστή του χρήστη το πρόγραμμα OBS και ξεκίνα το.

```

private void button2_Click_1(object sender, EventArgs e)
{
    if (SurveillanceButtonStatus == 0)
    {
        myProcess = Process.Start("Surveillance.bat");
        SurveillanceButtonStatus++;
        button2.Image = Image.FromFile("images/CCTV-CameraG2.png");
    }
}

```

//Αν το OBS είναι ήδη ενεργοποιημένο και το δεξί πλήκτρο πατηθεί ξανά τερμάτισε την εφαρμογή OBS

```

else
{
    Process[] processes = Process.GetProcessesByName("Obs");
    foreach (var process in processes)
    {
        process.Kill();
    }
}

```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
        button2.Image = Image.FromFile("images/CCTV-CameraG.png");
        SurveillanceButtonStatus = 0;
    }

}

//Με τον τερματισμό του προγράμματος κλείσε και την εφαρμογή OBS εφόσον
είναι ανοιχτή
void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    Process[] processes = Process.GetProcessesByName("Obs");
    foreach (var process in processes)
    {
        process.Kill();
    }
}
```

Παρακάτω βρίσκονται βοηθητικές λειτουργίες όπως η λειτουργία `public Form()` και χειριστές συμβάντος για την πλοήγηση στο κεντρικό μενού.

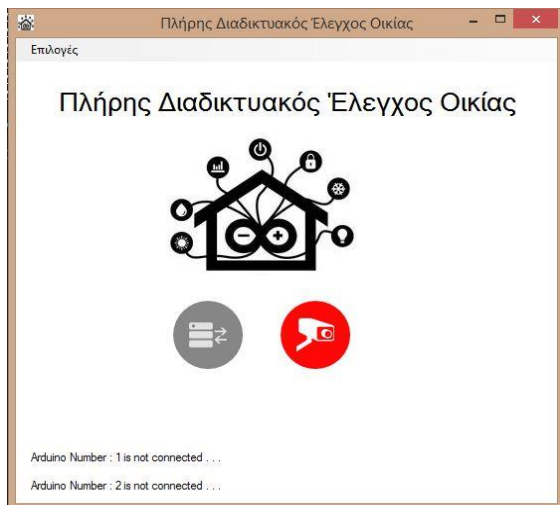
```
public Form1()
{
    InitializeComponent();
    this.FormClosed += Form1_FormClosed; //Event σε περίπτωση που ο
χρήστης κλείσει την εφαρμογή
    label5.Hide();
    ConnectArduino(); //Σύνδεση Arduino συσκευών
    ConnectArduino(); //Σύνδεση Arduino αισθητηρίων
}

//Εμφάνιση φόρμας about us με την επιλογή του Menu
private void aboutUsToolStripMenuItem_Click(object sender,
EventArgs e)
{
    var myForm = new Form2();
    myForm.Show();
}

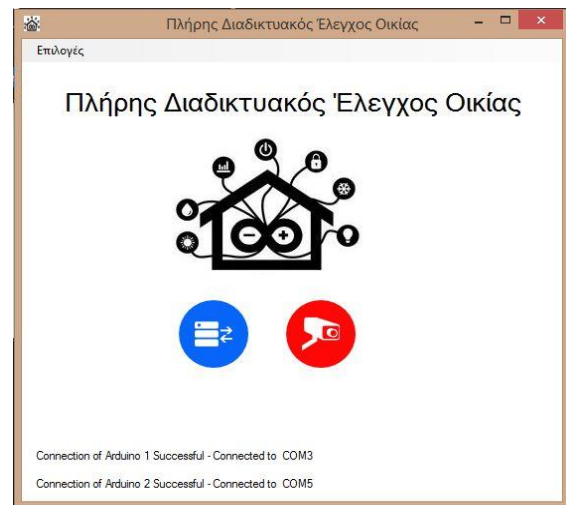
//Εμφάνιση φόρμας οδηγιών με την επιλογή του Menu
private void οδηγίεςToolStripMenuItem_Click(object sender,
EventArgs e)
{
    var myForm2 = new Form3();
    myForm2.Show();
}

//Εμφάνιση φόρμας ρυθμίσεων με την επιλογή του Menu
private void ρυθμίσειςToolStripMenuItem_Click(object sender,
EventArgs e)
{
    var myForm3 = new Form4();
    myForm3.Show();
}
```

Παρακάτω βλέπουμε δύο εικόνες του πρώτου παράθυρου της εφαρμογής σε περίπτωση επιτυχημένης σύνδεσης των δύο πλακετών Arduino με το πρόγραμμα.



Εικόνα 76: Επιτυχία σύνδεσης πλακετών



Εικόνα 77: Αποτυχία σύνδεσης πλακετών

## 8.6 Παράθυρο οδηγιών εφαρμογής

Το παράθυρο αυτό περιέχει οδηγίες ως προς τον σωστό τρόπο χρήσης της συνολικής μας εφαρμογής τις οποίες μπορεί εύκολα να διαβάσει ο χρήστης και να κατανοήσει.

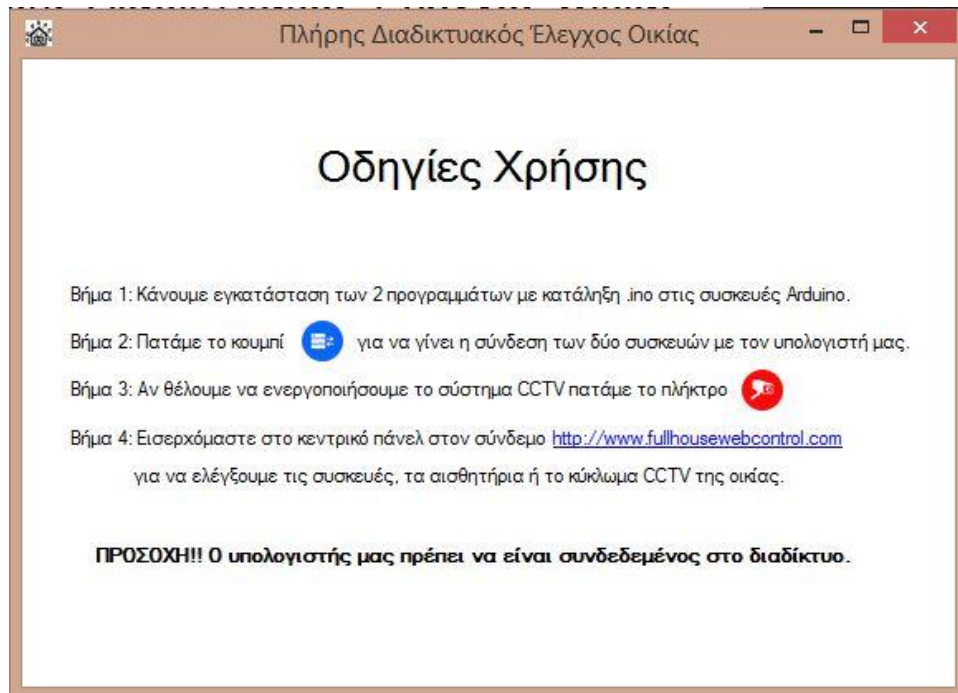
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace FullHouseWebControl
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        private void linkLabel1_LinkClicked(object sender,
        LinkLabelLinkClickedEventArgs e)
        {
            System.Diagnostics.Process.Start("http://www.fullhousewebcontrol.com");
        }
    }
}
```

//Με το πάτημα στην διεύθυνση άνοιξε στον προεπιλεγμένο περιηγητή του χρήστη αυτήν την σελίδα

```
}  
}  
}
```



Εικόνα 78: Παράθυρο οδηγιών χρήσης

## 8.7 Παράθυρο ρυθμίσεων εφαρμογής

Ο σκοπός ύπαρξης αυτού του παραθύρου είναι για να επιτρέψουμε στον χρήστη να αποφασίσει αν θα χρησιμοποιήσει την λειτουργία ελέγχου των διακοπών αφής. Επιπρόσθετα έχει προστεθεί ένα πεδίο το οποίο επιτρέπει με την εισαγωγή του μοναδικού κωδικού χρήστη τον έλεγχο της αντίστοιχης οικίας που ανήκει αυτός ο κωδικός. Αυτή η λειτουργία έχει προστεθεί μόνο για λόγους επίδειξης καθώς θα ήταν επικίνδυνο να επιτρέψουμε στον χρήστη απλά με την εισαγωγή ενός αριθμού να ελέγχει όποια οικία ήθελε.

Αυτή η σελίδα έχει την λειτουργία με το πάτημα του πλήκτρου επιβεβαίωση να κάνει έναν τυπικό έλεγχο για το αν υπάρχει χρήστης με τέτοιο κωδικό στο σύστημα και άλλη μία λειτουργία η οποία αποθηκεύει την προτίμηση ως προς τον έλεγχο των διακοπών αφής.



## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace FullHouseWebControl
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();

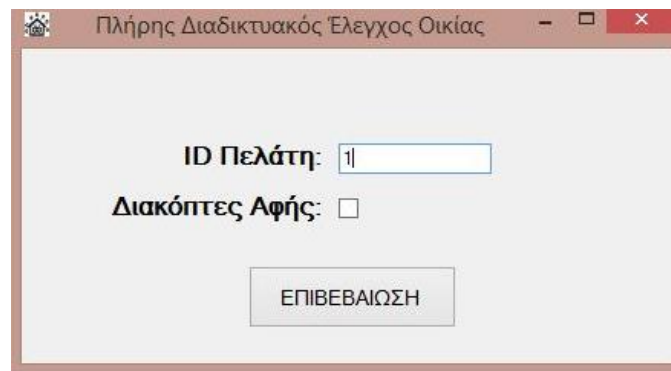
            //Με το πάτημα του κουμπιού το πρόγραμμα ελέγχει αν το περιεχόμενο
            του κουτιού υπάρχει. Έπειτα ελέγχει αν αυτό το ID χρήστη υπάρχει στην βάση
            δεδομένων και το ορίζει ως CustomerID
            private void button1_Click(object sender, EventArgs e)
            {
                if (String.IsNullOrEmpty(textBox1.Text))
                {
                    MessageBox.Show("Παρακαλώ πληκτρολογήστε ID πελάτη");
                }
                else {
                    string ConnectionString =
                    "datasource=83.212.120.81;port=3306;username=Arduino;password=ard1234";
                    MySqlConnection myConn = new
                    MySqlConnection(ConnectionString);
                    MySqlDataReader reader = null;

                    myConn.Open();
                    using (MySqlCommand cmd = new MySqlCommand("Select
                    * from Full_House_Web_Control.Users WHERE CustomerID=" + textBox1.Text,
                    myConn))
                    {
                        reader = cmd.ExecuteReader();

                        if (reader.Read())
                        {
                            Form1.CustomerID = textBox1.Text;
                            MessageBox.Show("Οι αλλαγές
                            πραγματοποιήθηκαν με επιτυχία.");
                            this.Close();
                        }
                        //Αν δεν υπάρχει εμφάνιση μηνύματος λάθους
                        else {

                            MessageBox.Show("Ο χρήστης που
                            πληκτρολογήσατε δεν υπάρχει στην βάση δεδομένων μας. Παρακαλώ προσπαθήστε
                            ξανά.", "Σφάλμα", MessageBoxButtons.OK, MessageBoxIcon.Error);
                        }
                    }
                }
            }
        }
    }
}
```

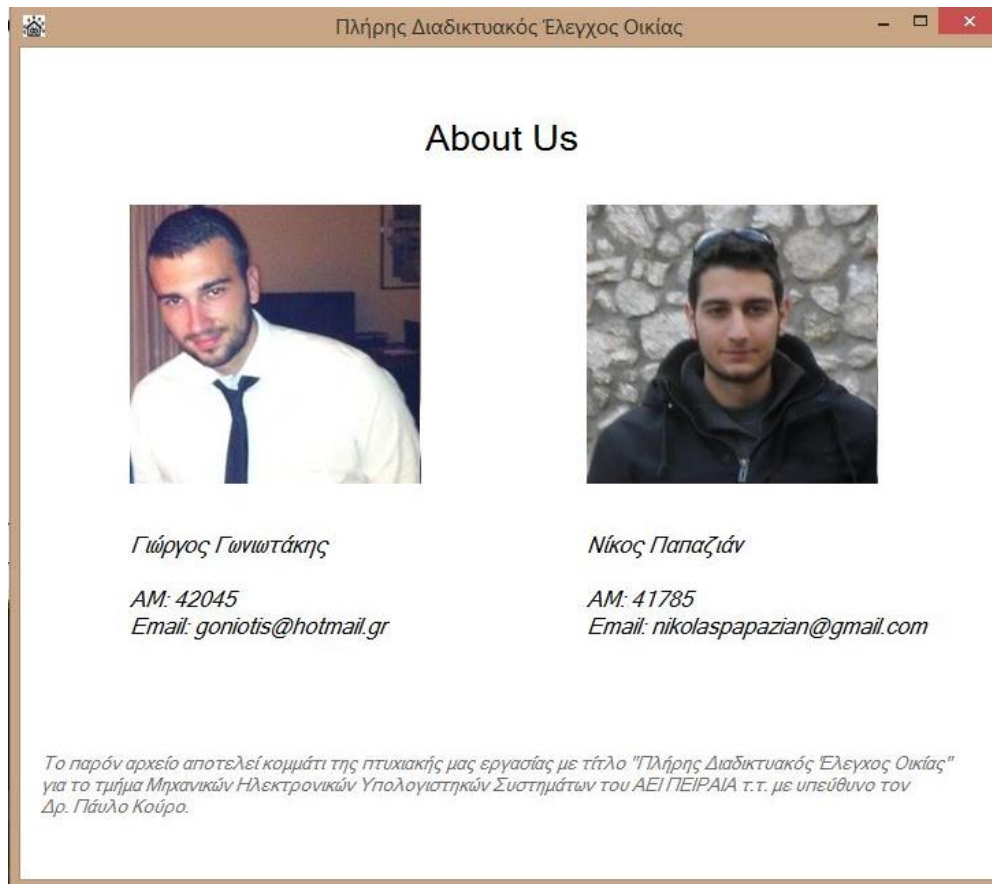
```
        }  
        myConn.Close ();  
    }  
  
    }  
    if (checkBox1.Checked)  
    {  
        Form1.ButtonPref = true;  
    }  
    else {  
        Form1.ButtonPref = false;  
    }  
    }  
    }  
}
```



Εικόνα 79: Παράθυρο ρυθμίσεων εφαρμογής

## 8.8 Παράθυρο πληροφοριών πτυχιακής εργασίας

Αυτό το παράθυρο έχει απλά ενημερωτικό χαρακτήρα. Σε αυτό εμπεριέχονται πληροφορίες σε σχέση με εμάς και την πτυχιακή μας εργασία.



Εικόνα 80: Παράθυρο πληροφοριών πρότζεκτ

## ΚΕΦΑΛΑΙΟ 9

### ΙΣΤΟΣΕΛΙΔΑ ΕΦΑΡΜΟΓΗΣ

#### 9.1 Σκοπός κατασκευής ιστοσελίδας

Θέλοντας να φτιάξουμε μια γραφική διεπαφή του χρήστη με τον έλεγχο της οικίας του ερευνήσαμε τις τεχνολογίες που υπάρχουν. Έτσι έπειτα από συζήτηση αποφανθήκαμε ότι σε περίπτωση που φτιάχναμε την εφαρμογή σε κάποια πλατφόρμα θα στερούσαμε την ελευθερία από τον χρήστη να έχει πρόσβαση του ελέγχου της οικίας του από όποια συσκευή ήθελε. Για παράδειγμα αν αποφασίζαμε να κατασκευάσουμε μία εφαρμογή για το λειτουργικό σύστημα Android σε περίπτωση που ο χρήστης δεν είχε κάποια συσκευή με αυτό το λογισμικό, δεν θα μπορούσε να χρησιμοποιήσει το σύστημά μας. Κοινή παραδοχή επίσης αποτελεί ότι οι χρήστες περνούν ένα μεγάλο κομμάτι της ημέρας τους στον υπολογιστή. Όμως στην περίπτωση που ο χρήστης της οικίας θα ήθελε να ενεργοποιήσει γρήγορα ένα φως θα έπρεπε να τρέξει στον υπολογιστή του ο οποίος μπορεί να μην λειτουργεί ή να μην είναι διαθέσιμος εκείνη την στιγμή.

Έτσι αποφασίσαμε να κατασκευάσουμε μία ιστοσελίδα η οποία θα είναι συμβατή με όλες τις συσκευές που χρησιμοποιούν περιηγητές (browsers) όπως υπολογιστές, ταμπλέτες (tablets), κινητές συσκευές, έξυπνες τηλεοράσεις κλπ. Με αυτόν τον τρόπο καταφέρνουμε να μετατρέψουμε οποιαδήποτε συσκευή μπορεί να συνδεθεί στο διαδίκτυο σε ένα χειριστήριο (controller) για το σύστημά μας. Ο χρήστης του συστήματός μας δεν χρειάζεται με αυτόν τον τρόπο να ανησυχεί αν για παράδειγμα η κινητή του συσκευή δεν έχει σήμα ή μπαταρία, από την στιγμή που μπορεί να χρησιμοποιήσει οποιαδήποτε συσκευή ως χειριστήριο.

#### 9.2 Τεχνολογίες που χρησιμοποιήθηκαν

Μία ιστοσελίδα όπως είναι γνωστό αποτελείται από κώδικα της Γλώσσας Σήμανσης Υπερκειμένου (HTML) ο οποίος μεταγλωττίζεται από το πρόγραμμα περιηγητή (browser) σε γραφικά και παρουσιάζεται στον χρήστη. Φυσικά η HTML μπορεί μόνο να περιγράψει τα βασικά αντικείμενα που μπορεί να δει ένας χρήστης (κουμπιά, κείμενο, εικόνες κ.α.) και όχι να τροποποιήσει την μορφή τους. Για αυτόν τον λόγο

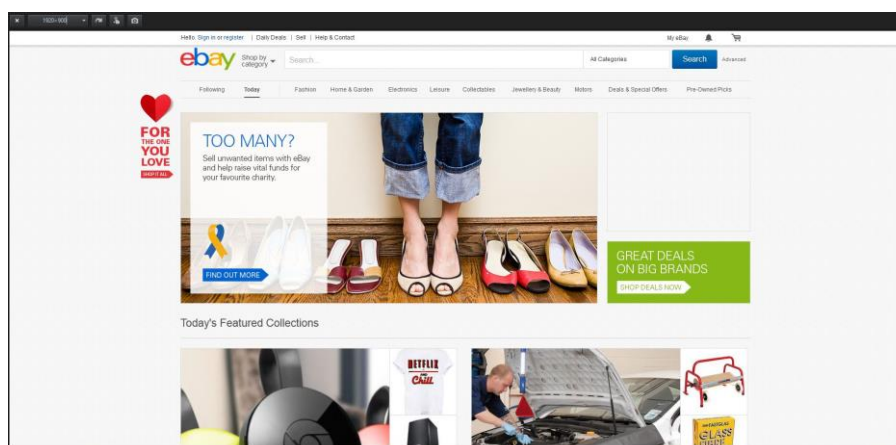
χρησιμοποιείται η CSS με την οποία μπορούμε να τροποποιήσουμε όλες τις οπτικές ιδιότητες ενός αντικειμένου π.χ. μέγεθος, σχήμα, χρώμα και άλλα.

Μέχρι στιγμής αναφέραμε πως μπορούμε να δημιουργήσουμε τα αντικείμενα και να τους δώσουμε το στυλ εμφάνισης που θέλουμε αλλά όλα αυτά δεν θα είχαν καμία ουσία αν δεν μπορούσαμε να τους αποδώσουμε περαιτέρω λειτουργικότητα από αυτήν που ήδη έχουν. Για αυτόν τον λόγο χρησιμοποιούμε την γλώσσα JavaScript για να αποδώσουμε λογική σε ότι αφορά τα στοιχεία που έχει πρόσβαση ο χρήστης (client side).

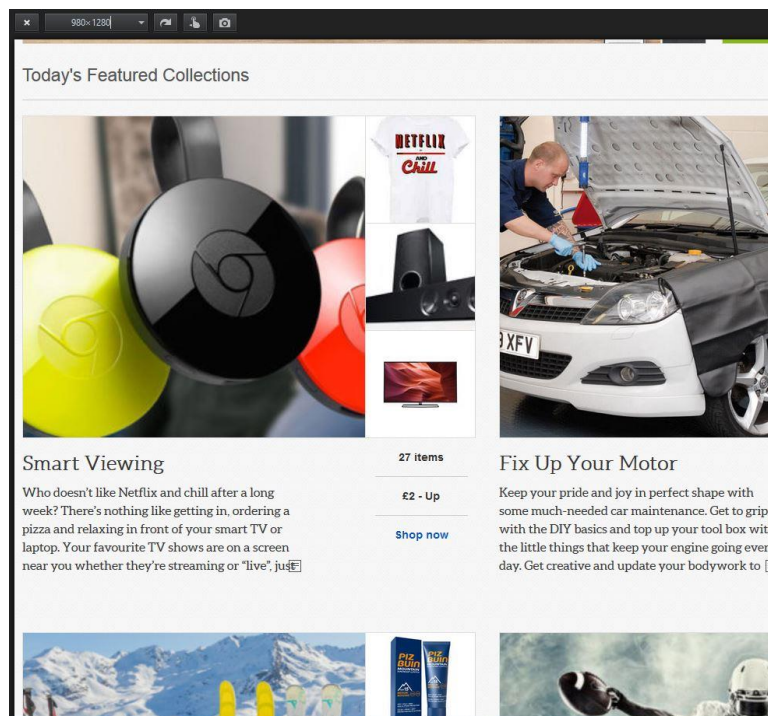
Όσον αφορά τα στοιχεία στα οποία δεν έχει πρόσβαση ο χρήστης (server side) χρησιμοποιήσαμε την γλώσσα προγραμματισμού PHP. Η γλώσσα αυτή θα μας βοηθήσει να παρέχουμε στον χρήστη δυναμικές σελίδες ενημέρωσης της κατάστασης της οικίας του. Επίσης η PHP θα αποτελέσει τον δίαυλο επικοινωνίας της ιστοσελίδας μας με την βάση δεδομένων.

### 9.3 Συλλογή εργαλείων ανοιχτού κώδικα – Bootstrap

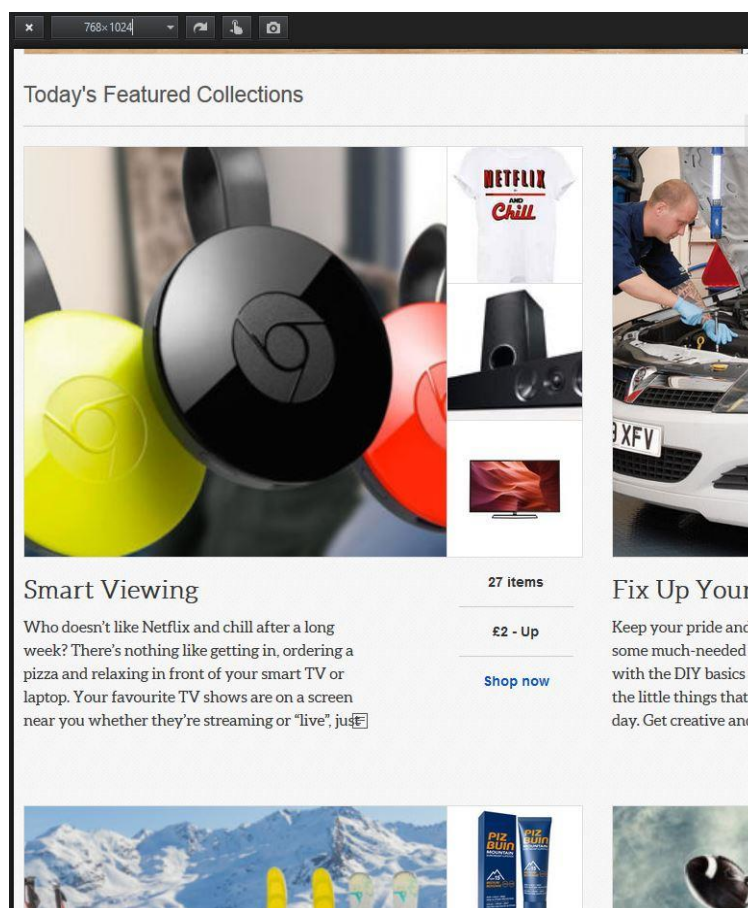
Όπως αναφέραμε και παραπάνω θέλαμε να φτιάξουμε μια ιστοσελίδα η οποία θα ήταν προσβάσιμη από όλες τις συσκευές που έχουν περιηγητές (browsers). Με τις παραπάνω γλώσσες προγραμματισμού αυτό θα ήταν εφικτό. Καθώς θα αναπτύσσαμε λοιπόν την ιστοσελίδα τα αντικείμενα θα είχαν σταθερή θέση και μέγεθος σε όλες τις συσκευές όπως συμβαίνει στην ιστοσελίδα του ηλεκτρονικού καταστήματος EBay παρακάτω.



Εικόνα 81: EBay.com σε ανάλυση οθόνης 1920 x 900

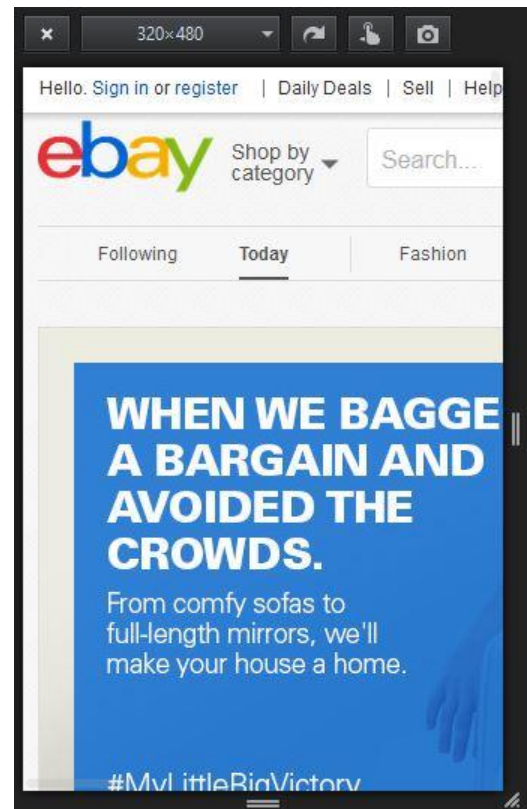
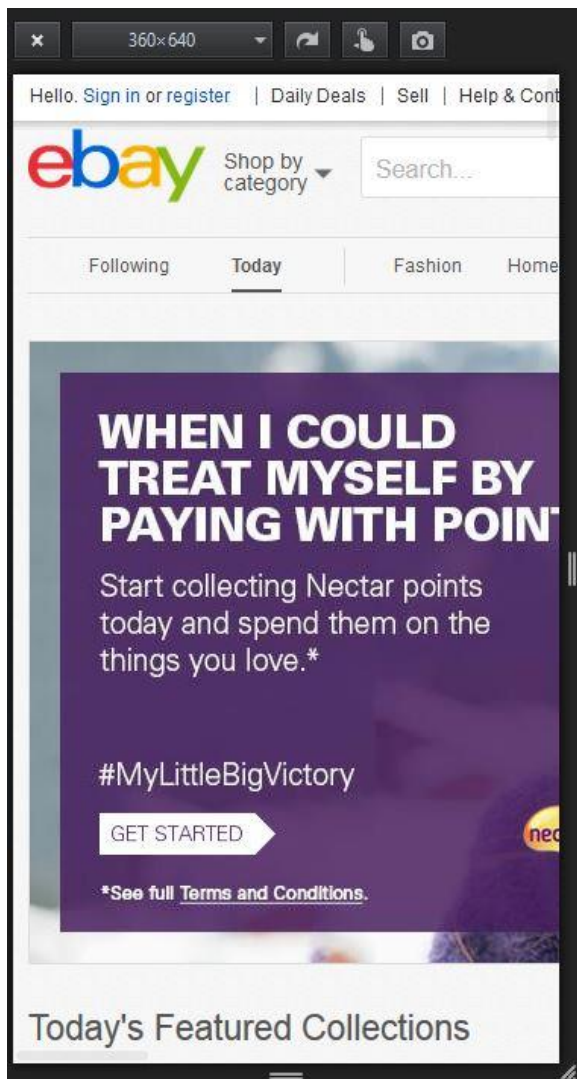


Εικόνα 82: EBay.com σε ανάλυση οθόνης 980 x 1280



Εικόνα 83: EBay.com σε ανάλυση οθόνης 768 x 1024





Εικόνα 84: EBay.com σε ανάλυση 360 x 640      Εικόνα 85: EBay.com σε ανάλυση 320 x 480

Μπορούμε πολύ εύκολα να παρατηρήσουμε ότι ήδη από την στιγμή που η οθόνη της συσκευής μας με την οποία θα προσπαθούσαμε να μπούμε σε αυτήν την ιστοσελίδα και έχει ανάλυση μικρότερη από 980pixels σε πλάτος, θα καθιστούσε την προβολή της σελίδας δύσκολη. Ο χρήστης θα έπρεπε να χρησιμοποιήσει τις μπάρες πλοήγησης του περιηγητή για να δει τα κομμάτια του περιεχομένου της σελίδας που δεν είναι ορατά αλλά πάντα θα έχανε κάτι. Φυσικά αυτό το γεγονός δεν προβληματίζει την εταιρία γιατί έχει κατασκευάσει και σε μορφή εφαρμογής το σύστημα της για όλες τις πλατφόρμες συσκευών. Φυσικά τέσσερις ξεχωριστές εφαρμογές έχουν κόστος κατασκευής, συντήρησης και υποστήριξης τετραπλάσιο από μία απλή ιστοσελίδα.

Για να σχεδιάσουμε μία σελίδα προσιτή σε όλες τις αναλύσεις οθονών, δηλαδή σε όλες τις συσκευές, θα χρησιμοποιήσουμε την συλλογή εργαλείων ανοιχτού κώδικα με την ονομασία Bootstrap.

Το Bootstrap αποτελεί ένα σύνολο, ανοιχτού κώδικα, βιβλιοθηκών οι οποίες περιέχουν κομμάτια κώδικα σε JavaScript και έτοιμα στυλ εμφάνισης σε CSS. Δημιουργήθηκε από δύο προγραμματιστές της εταιρίας κοινωνικής δικτύωσης Twitter και αποτελεί το πιο δημοφιλές έργο της κοινότητας GitHub. Ο λόγος που είναι τόσο δημοφιλές είναι επειδή εισάγει μέσω έτοιμων στυλ εμφάνισης τον ανταποκρίσιμο σχεδιασμό (responsive design) στον χώρο των ιστοσελίδων. Τον ανταποκρίσιμο σχεδιασμό, συμβατότητα προβολής της ιστοσελίδας σε όλες τις αναλύσεις, τον καταφέρνει επιτρέποντας στον προγραμματιστή να σχεδιάσει την ιστοσελίδα μέσω ενός συστήματος πλέγματος (grid). Αυτό το πλέγμα περιβάλλει τα τμήματα της ιστοσελίδας και αποτελείται από δώδεκα κομμάτια σε τέσσερις διαφορετικούς συνδυασμούς αναλύσεων. Κάθε ένας από αυτούς τους συνδυασμούς καλύπτει ένα ξεχωριστό εύρος αναλύσεων και έχει ξεχωριστή ονομασία. Αυτοί οι συνδυασμοί δεν είναι κάτι περισσότερο από στυλ εμφάνισης στην γλώσσα CSS τα οποία περιέχουν εντολές αυτόματου εντοπισμού της ανάλυσης της οθόνης του χρήστη (media queries) και προσαρμόζουν το μέγεθος του περιεχομένου μέσα στο πλέγμα. Έτσι ο προγραμματιστής έχει την δυνατότητα χρησιμοποιώντας αυτά τα στυλ να σχεδιάσει την ιστοσελίδα ταυτόχρονα σε τέσσερα εύρη αναλύσεων αντί ενός που κανονικά μπορούσε.

	<b>Extra small devices</b> Phones (<768px)	<b>Small devices</b> Tablets (≥768px)	<b>Medium devices</b> Desktops (≥992px)	<b>Large devices</b> Desktops (≥1200px)
<b>Grid behavior</b>	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
<b>Container width</b>	None (auto)	750px	970px	1170px
<b>Class prefix</b>	<code>.col-xs-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>

Εικόνα 86: Λειτουργία συστήματος πλέγματος



## Πλήρης Διαδικτυακός Έλεγχος Οικίας

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8										.col-md-4		
.col-md-4				.col-md-4				.col-md-4				
.col-md-6						.col-md-6						

```
<div class="row">
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
</div>
<div class="row">
  <div class="col-md-8">.col-md-8</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-6">.col-md-6</div>
  <div class="col-md-6">.col-md-6</div>
</div>
```

Εικόνα 87: Δομή συστήματος πλέγματος

.col-xs-12 .col-md-8		.col-xs-6 .col-md-4	
.col-xs-6 .col-md-4	.col-xs-6 .col-md-4	.col-xs-6 .col-md-4	
.col-xs-6		.col-xs-6	

```
<!-- Stack the columns on mobile by making one full-width and the other half-width -->
<div class="row">
  <div class="col-xs-12 col-md-8">.col-xs-12 .col-md-8</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>

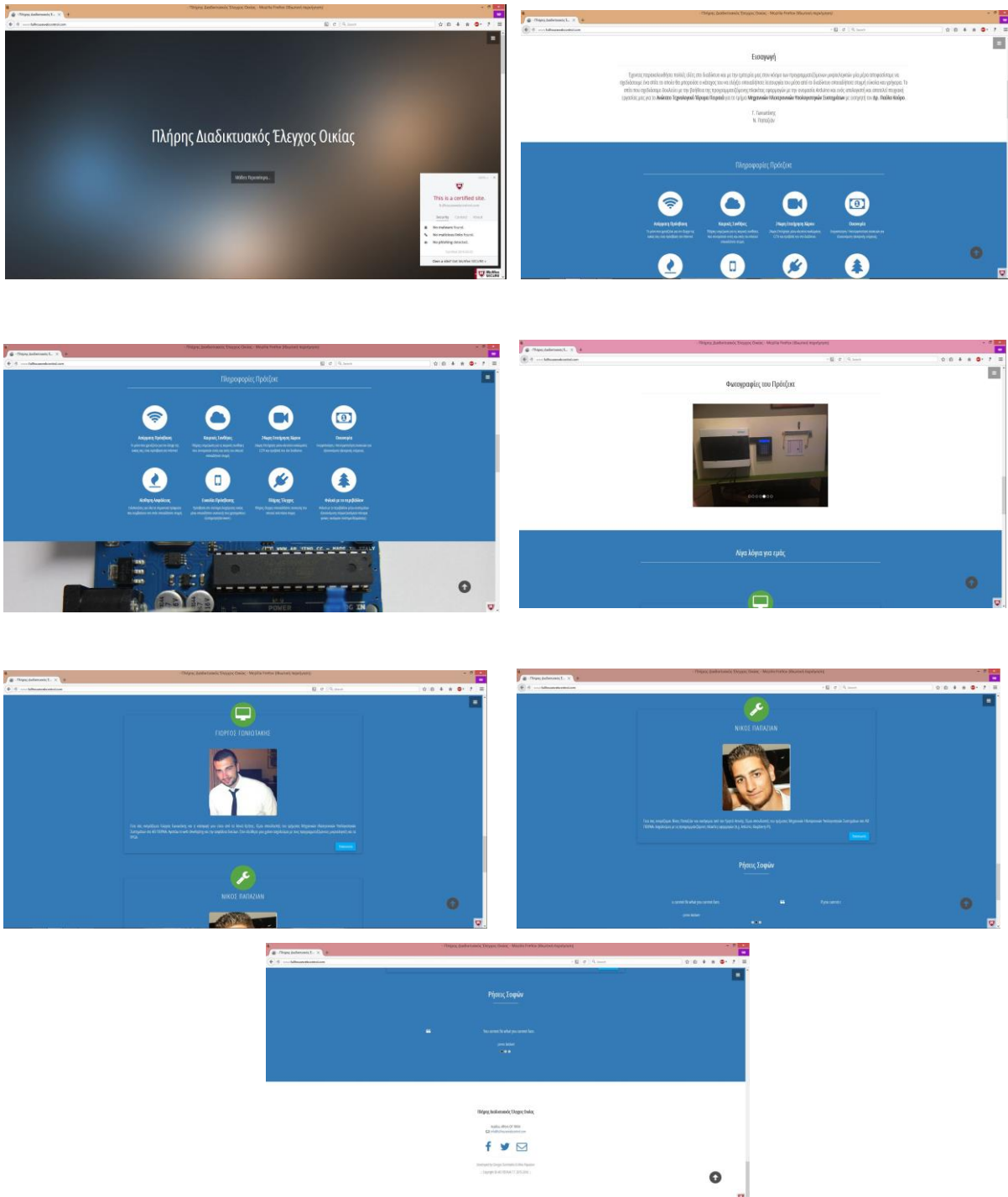
<!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
<div class="row">
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>

<!-- Columns are always 50% wide, on mobile and desktop -->
<div class="row">
  <div class="col-xs-6">.col-xs-6</div>
  <div class="col-xs-6">.col-xs-6</div>
</div>
```

Εικόνα 88: Σχεδίαση ιστοσελίδας για συσκευές διαφορετικών αναλύσεων οθόνης

### 9.4 Αρχική σελίδα εφαρμογής

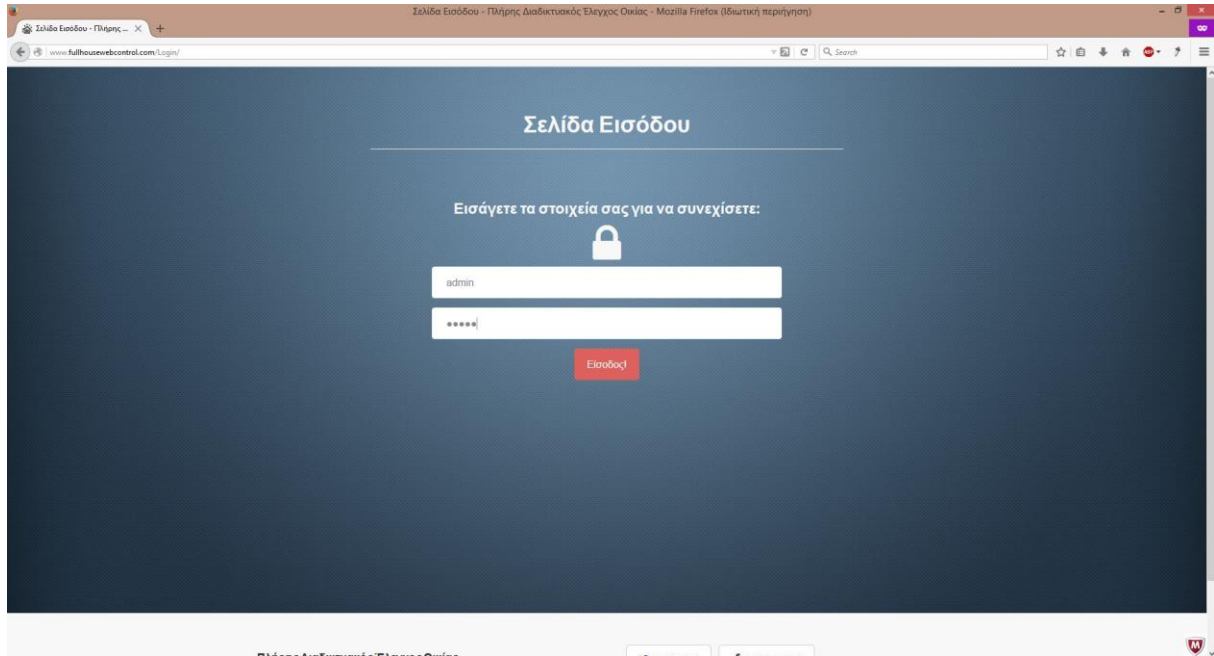
Στην αρχική σελίδα της εφαρμογής μας οι χρήστες μπορούν να μάθουν περισσότερες πληροφορίες για το πρότζεκτ και τα άτομα, να δουν φωτογραφίες της μακέτας και να διαβάσουν ένα σύντομο κείμενο οκτώ βασικών σημείων που αφορούν τα πλεονεκτήματα που προκύπτουν από την χρήση της εφαρμογής μας.



Εικόνες 89 – 95: Αρχική σελίδα εφαρμογής

## 9.5 Σελίδα εισόδου στο σύστημα

Σε αυτήν την σελίδα ο χρήστης μπορεί εισάγοντας τα διαπιστευτήρια που του έχουν δοθεί από εμάς να εισέλθει στο σύστημα για να αποκτήσει τον έλεγχο της οικίας του.



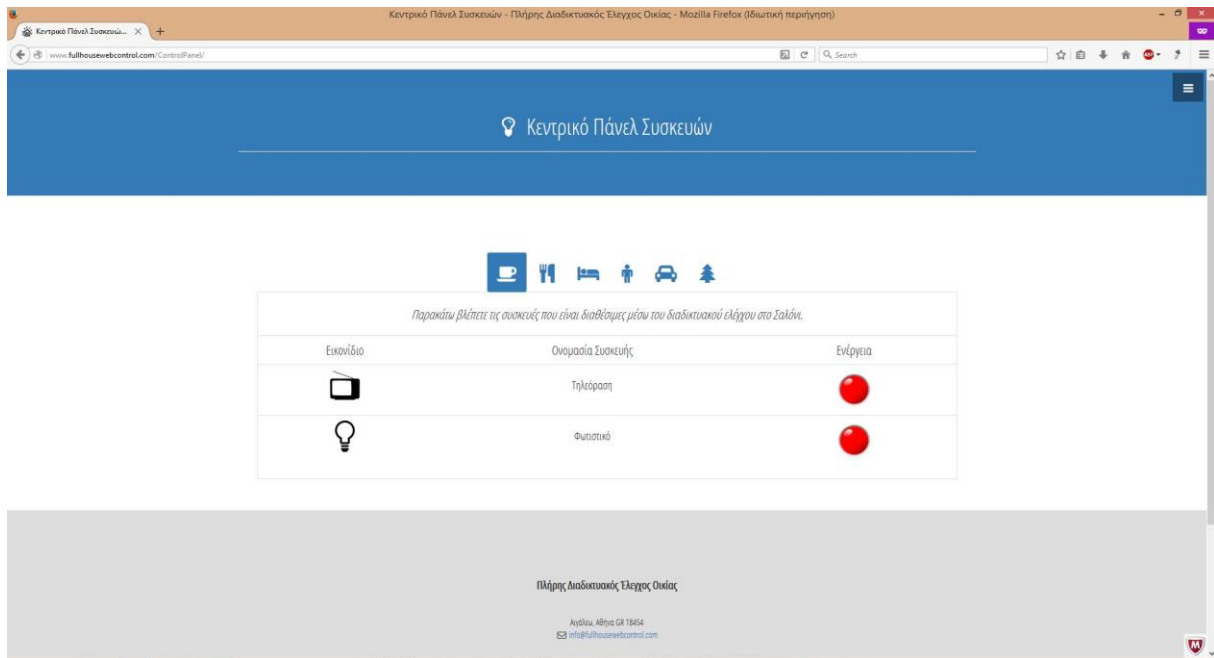
Εικόνα 96: Σελίδα εισόδου στο σύστημα

Έπειτα από την είσοδο των διαπιστευτηρίων του, όνομα χρήστη και κωδικό ασφαλείας, το πρόγραμμα συνδέεται με την βάση δεδομένων και αναζητά τον χρήστη που εφόσον τον βρει του επιτρέπει την είσοδο στο πάνελ ελέγχου της οικίας του. Σε περίπτωση εισαγωγής λάθος διαπιστευτηρίων τον ειδοποιεί και τον προτρέπει να προσπαθήσει ξανά.

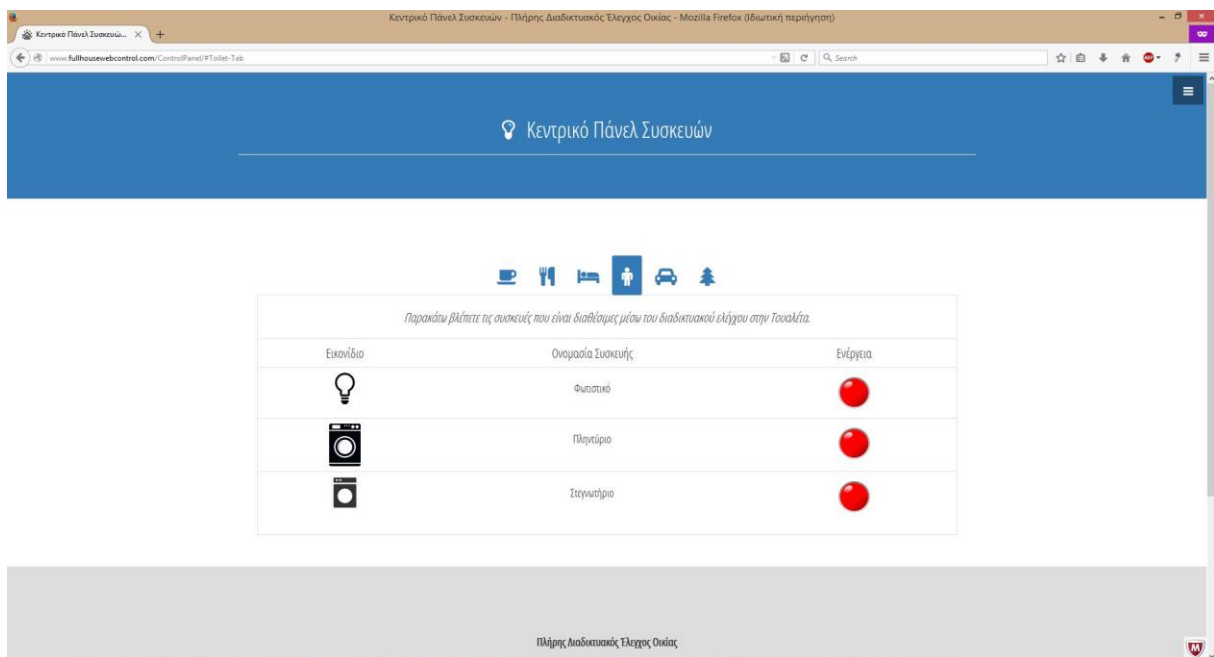
## 9.6 Κεντρική σελίδα συστήματος

Σε αυτήν την σελίδα ο χρήστης έχει την δυνατότητα να παρακολουθήσει την κατάσταση λειτουργίας των συσκευών της οικίας του και να την τροποποιήσει. Ο πίνακας έχει σχεδιασθεί με μινιμαλιστικό μοτίβο έτσι ώστε μέσω εικόνων να διευκολύνει τον χρήστη να ελέγχει πιο γρήγορα και πιο αποτελεσματικά τις συσκευές του. Μετά από την ενεργοποίηση/απενεργοποίηση μιας συσκευής, η βάση δεδομένων ενημερώνεται για την αλλαγή επιτρέποντας έτσι το πρόγραμμα διαμεσολάβησης να ενημερώσει την πλακέτα Arduino.

## Πλήρης Διαδικτυακός Έλεγχος Οικίας



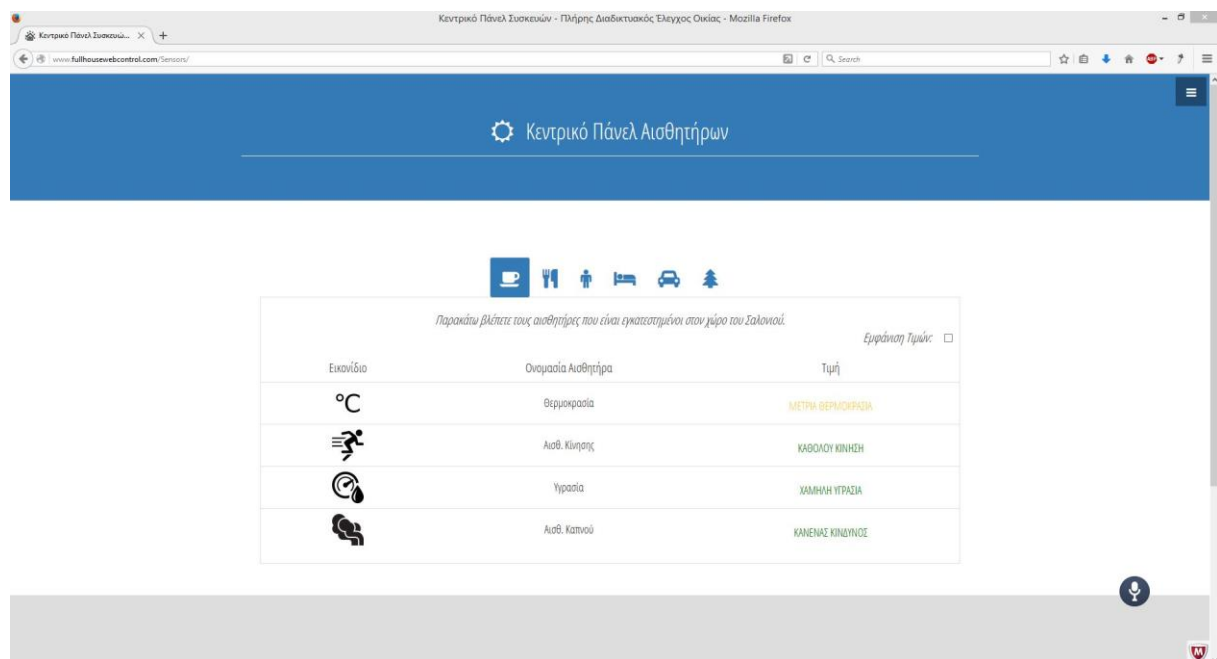
Εικόνα 97: Παράδειγμα ελέγχου συσκευών σαλονιού



Εικόνα 98: Παράδειγμα ελέγχου συσκευών τουαλέτας

## 9.7 Κεντρικό Πάνελ Αισθητηρίων

Στο κεντρικό πάνελ αισθητηρίων ο χρήστης έχει την δυνατότητα να ελέγξει την κατάσταση όλων των αισθητηρίων της οικίας του με δύο τρόπους. Ο πρώτος τρόπος είναι να ενημερωθεί μέσω των χαρακτηρισμών όπως για παράδειγμα χαμηλή ή υψηλή φωτεινότητα τους οποίους έχουμε αποδώσει για κάθε είδος αισθητηρίου ξεχωριστά. Με την ενεργοποίηση του πεδίου “Εμφάνιση Τιμών” ενεργοποιεί τον δεύτερο τρόπο ελέγχου ο οποίος του εμφανίζει τις αναλογικές και ψηφιακές τιμές των αισθητηρίων της οικίας του.

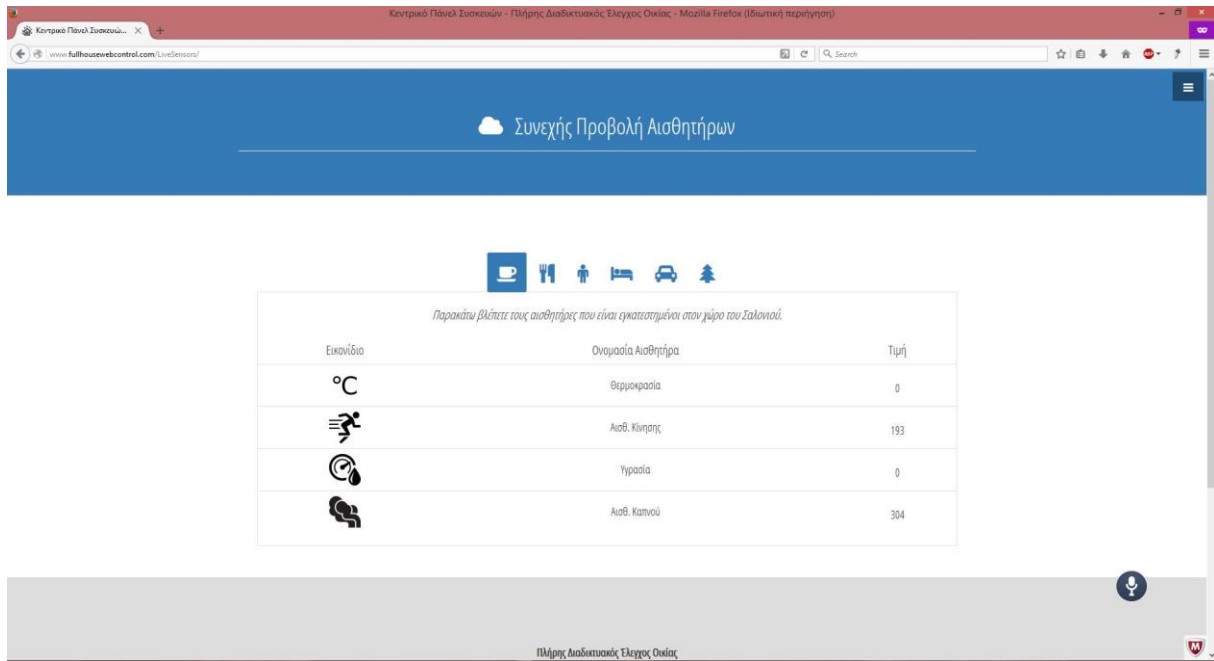


Εικόνα 99: Κεντρικό Πάνελ Αισθητηρίων

## 9.8 Συνεχής Προβολή Αισθητηρίων

Στην σελίδα αυτή ο χρήστης μπορεί να βλέπει την ζωντανή εξέλιξη των αισθητηρίων της οικίας του όπως αυτή προκύπτει από την αποστολή των τιμών της πλακέτας Arduino στο κεντρικό πρόγραμμα και από εκεί στην βάση δεδομένων. Η σελίδα μέσω κώδικα JavaScript ανανεώνει τον πίνακα των τιμών των αισθητηρίων ανά ένα δευτερόλεπτο. Ο λόγος του διαχωρισμού αυτής της λειτουργίας από την προηγούμενη είναι λόγω του ότι η συνεχόμενη ανανέωση της σελίδας αυτής δημιουργεί την λήψη μεγαλύτερου όγκου δεδομένων, πράγμα το οποίο θα προκαλούσε στον χρήστη πρόβλημα σε περίπτωση που δεσμευόταν από ένα συμβόλαιο παροχής Internet με χαμηλό διαθέσιμο όγκο κατανάλωσης δεδομένων. Έτσι για παράδειγμα όταν ο

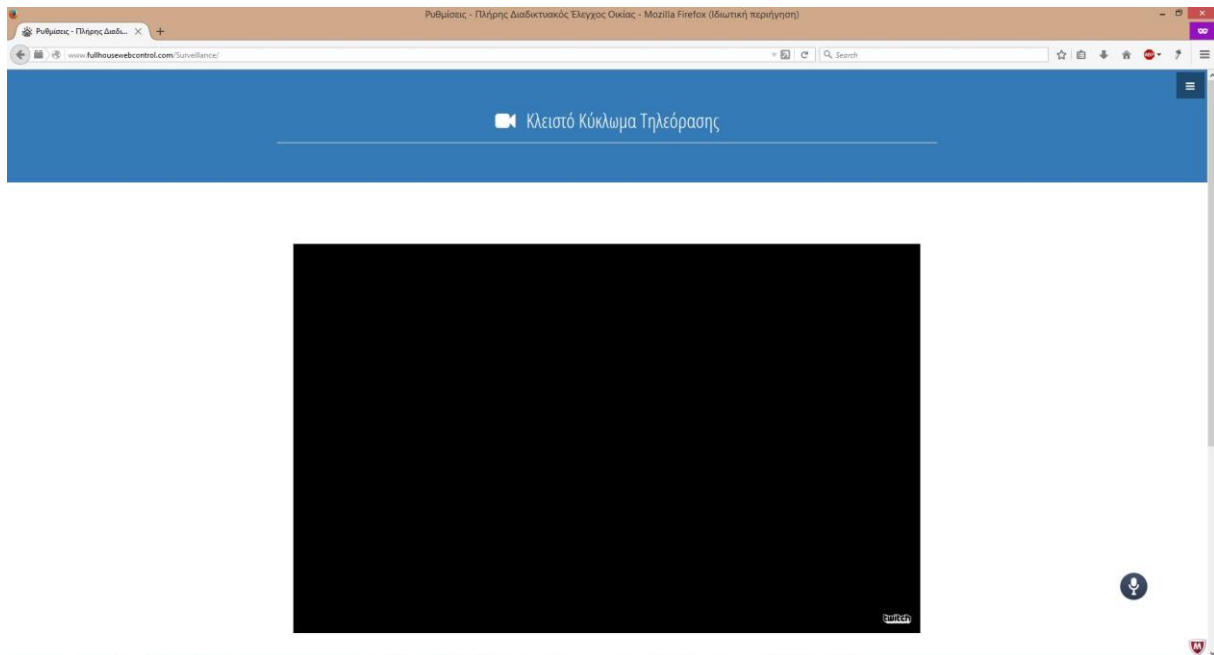
χρήστης εισέλθει από κινητή συσκευή για να δει τις τιμές των αισθητηρίων μπορεί να χρησιμοποιήσει το κεντρικό πάνελ αισθητηρίων ενώ εάν εισέλθει από τον προσωπικό υπολογιστή της οικίας ή του γραφείου του μπορεί να χρησιμοποιήσει την σελίδα συνεχούς προβολής αισθητηρίων.



Εικόνα 100: Σελίδα συνεχούς προβολής τιμών αισθητηρίων

### 9.9 Σελίδα παρακολούθησης κλειστού κυκλώματος παρακολούθησης (CCTV)

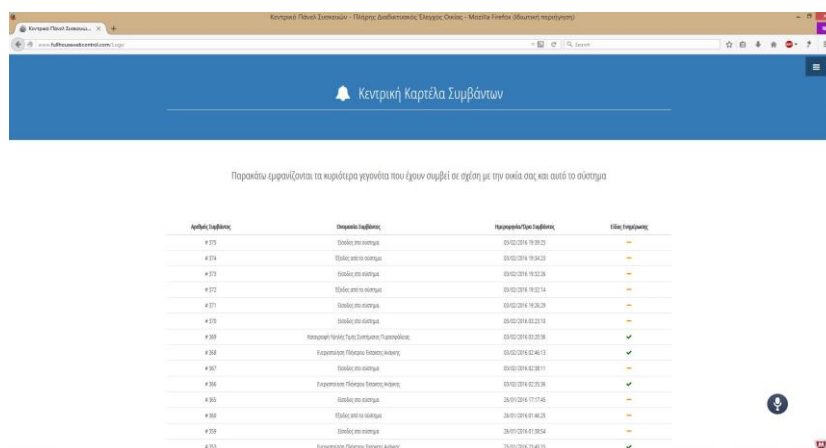
Αυτή η σελίδα περιέχει ένα αντικείμενο αναπαραγωγής ήχου και βίντεο (Player) ο οποίος χρησιμοποιεί τεχνολογία Flash. Αυτό το αντικείμενο έχει τοποθετηθεί για λόγους επίδειξης και είναι διαθέσιμο σε περιηγητές με υποστήριξη Flash. Στις περισσότερες κινητές συσκευές είναι πιθανόν να πρέπει ο χρήστης να προμηθευτεί έναν περιηγητή με υποστήριξη Flash για να δει την βιντεοσκόπηση. Η σελίδα αυτή περιέχει τον σύνδεσμο από την διαδικτυακή κάμερα βιντεοσκόπησης (IP Camera) την οποία έχει ο χρήστης και μας έχει δώσει. Έτσι ενσωματώνοντας τον σύνδεσμο από την κάμερα ασφαλείας του στο σύστημά μας, μαζεύουμε σε ένα σημείο οποιαδήποτε διαδικτυακή λειτουργία σχετικά με την οικία του. Η σελίδα αυτή συνδέεται στην βάση δεδομένων και παίρνει τον σύνδεσμο που μας έχει δώσει ο χρήστης και τον ενσωματώνει σε ένα τέτοιο αντικείμενο αναπαραγωγής εικόνας και ήχου.



Εικόνα 101: Σελίδα παρακολούθησης κλειστού κυκλώματος τηλεόρασης (CCTV)

## 9.10 Σελίδα προβολής συμβάντων

Κάθε σημαντική ενέργεια που συμβαίνει στην οικία του χρήστη καταγράφεται στην βάση δεδομένων και εμφανίζεται σε αυτήν την σελίδα. Εδώ ο χρήστης μπορεί να δει με αναλυτική χρονική περιγραφή τις ενέργειες εισόδου και εξόδου στο σύστημα με την χρήση των διαπιστευτηρίων του, την ενεργοποίηση και απενεργοποίηση συγκεκριμένων ρυθμίσεων του λογαριασμού του καθώς και έκτακτα συμβάντα όπως η ενεργοποίηση των συστημάτων πυρασφάλειας, έκτακτης ανάγκης και συναγερμού. Η σελίδα αυτή παίρνει δυναμικά το περιεχόμενό της από την βάση δεδομένων μας.



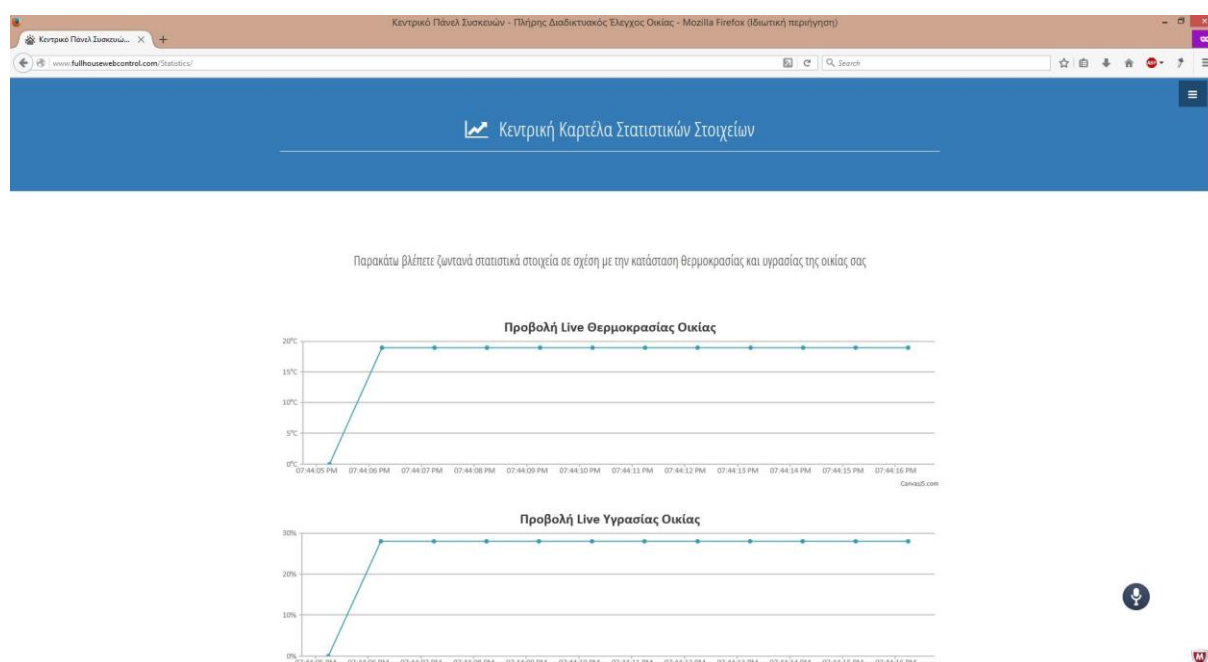
Εικόνα 102: Σελίδα προβολής συμβάντων



## 9.11 Κεντρική σελίδα προβολής στατιστικών στοιχείων

Αυτή η σελίδα παράγει δυναμικά μέσω των γλωσσών προγραμματισμού JavaScript και PHP δύο γραφήματα με τον μέσο όρο της θερμοκρασίας και υγρασίας της οικίας. Μετά από ένα σύντομο χρονικό διάστημα τοποθετεί στον οριζόντιο άξονα των γραφημάτων την ημερομηνία και ώρα της μέτρησης. Ο σχεδιασμός των σημείων προκύπτει από συνεχή ανάγνωση των τιμών των αισθητηρίων από την βάση δεδομένων καθώς η αποθήκευση όλων αυτών των δεδομένων για πολλούς ταυτόχρονους χρήστες θα χρειαζόταν μεγαλύτερη βάση δεδομένων, γεγονός το οποίο θα ανέβαζε την τιμή της εφαρμογής.

Η προβολή των γραφημάτων λόγω της κατασκευής τους στην γλώσσα JavaScript είναι συμβατή και μέσω κινητών συσκευών.



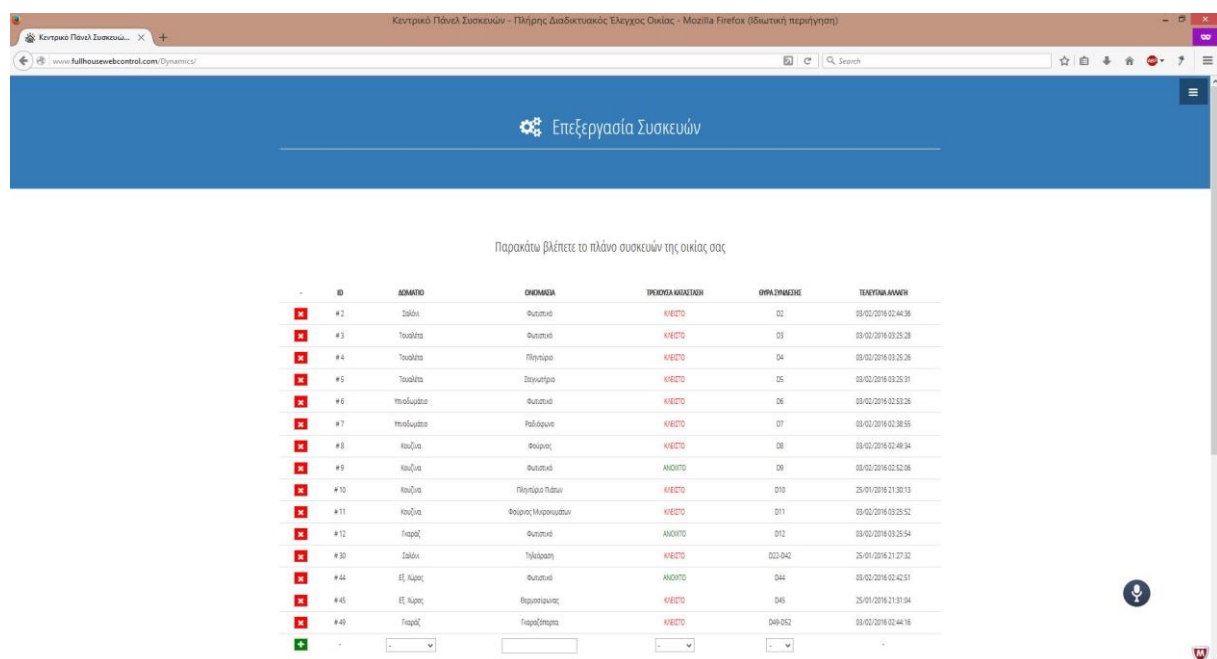
Εικόνα 103: Κεντρική σελίδα προβολής στατιστικών στοιχείων

## 9.12 Σελίδα επεξεργασίας συσκευών οικίας

Θέλοντας να επιτρέψουμε την απόλυτη ελευθερία για επεκτασιμότητα και διαχείριση από τον χρήστη δημιουργήσαμε την λειτουργία προσθήκης/αφαίρεσης συσκευών, που αφορούν την οικία του, στο σύστημα. Χρησιμοποιώντας αυτήν την λειτουργία ο χρήστης μπορεί να αφαιρέσει κάποια συσκευή η οποία ελέγχεται διαδικτυακά. Επίσης κάποιος χρήστης – γνώστης της λειτουργίας της πλακέτας Arduino μπορεί να δει ποιες ακίδες της πλακέτας είναι διαθέσιμες και να αναθέσει μία καινούργια συσκευή προς έλεγχο στο σύστημα. Έτσι πολύ εύκολα όπως θα



αναφέρουμε και στο κεφάλαιο της πραγματικής εφαρμογής, ο χρήστης θα μπορεί να κατασκευάσει και να διαχειριστεί μόνος του το υλικό που βρίσκεται στην οικία του. Η σελίδα παίρνει δυναμικά τις συσκευές του χρήστη μέσα από την βάση δεδομένων. Σε περίπτωση που ο χρήστης θέλει να προσθέσει μία συσκευή του δείχνει τις διαθέσιμες επιλογές που έχει και αφού τα καταχωρήσει κάνει έλεγχο ορθότητας και ενημερώνει την βάση δεδομένων. Αφού ολοκληρωθεί αυτή η διαδικασία και εφόσον η συσκευή είναι συνδεδεμένη πάνω στην πλακέτα Arduino ο χρήστης μπορεί να την ελέγξει μέσω της κεντρικής σελίδας συσκευών.

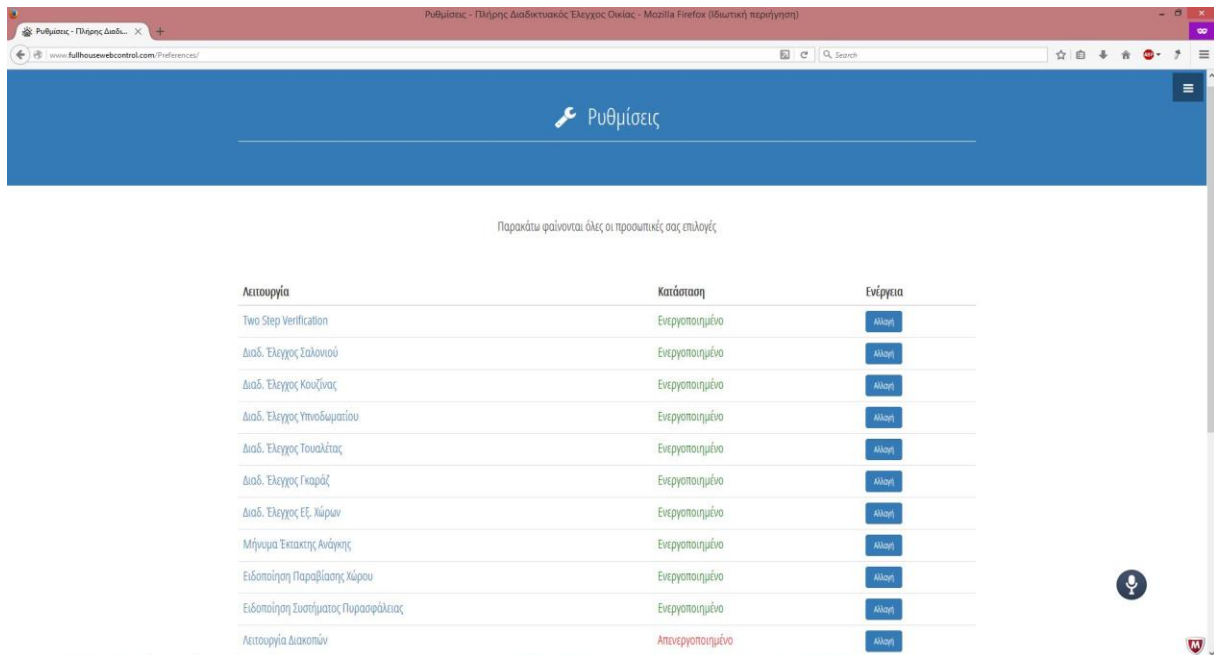


ID	ΔΩΜΑΤΙΟ	ΟΝΟΜΑΣΙΑ	ΤΡΟΠΟΣ ΚΑΤΑΣΤΑΣΗΣ	ΘΡΥΠΑ ΣΠΙΝΑΞΗΣ	ΤΙΜΗΜΑ ΛΗΨΗΣ
#1	Σάβανο	Φωτιστικό	ΚΙΣΤΟ	02	03/02/2016 02:44:36
#3	Τουαλέτα	Φωτιστικό	ΚΙΣΤΟ	03	03/02/2016 03:25:28
#4	Τουαλέτα	Πληγίτρο	ΚΙΣΤΟ	04	03/02/2016 03:25:36
#5	Τουαλέτα	Σταμπατέρ	ΚΙΣΤΟ	05	03/02/2016 03:25:31
#6	Παρόδιωμα	Φωτιστικό	ΚΙΣΤΟ	06	03/02/2016 02:53:26
#7	Παρόδιωμα	Ραδιόφωνο	ΚΙΣΤΟ	07	03/02/2016 02:36:55
#8	Κουζίνα	Φούρνος	ΚΙΣΤΟ	08	03/02/2016 02:48:34
#9	Κουζίνα	Φωτιστικό	ΑΚΙΣΤΟ	09	03/02/2016 02:52:36
#10	Κουζίνα	Πληγίτρο Πάτωμα	ΚΙΣΤΟ	010	25/01/2016 21:30:13
#11	Κουζίνα	Φούρνος Μικροκυμάτων	ΚΙΣΤΟ	011	03/02/2016 03:25:52
#12	Παράζ	Φωτιστικό	ΑΚΙΣΤΟ	012	03/02/2016 03:25:54
#30	Σάβανο	Τηλέφωνο	ΚΙΣΤΟ	033-042	25/01/2016 21:27:32
#44	Εξ. Άρρος	Φωτιστικό	ΑΚΙΣΤΟ	044	03/02/2016 02:42:51
#45	Εξ. Άρρος	Θερμοσίμανα	ΚΙΣΤΟ	045	25/01/2016 21:31:04
#49	Παράζ	Παραλόγισμα	ΚΙΣΤΟ	049-052	03/02/2016 02:44:16

Εικόνα 104: Σελίδα επεξεργασίας συσκευών οικίας

### 9.13 Σελίδα ρυθμίσεων

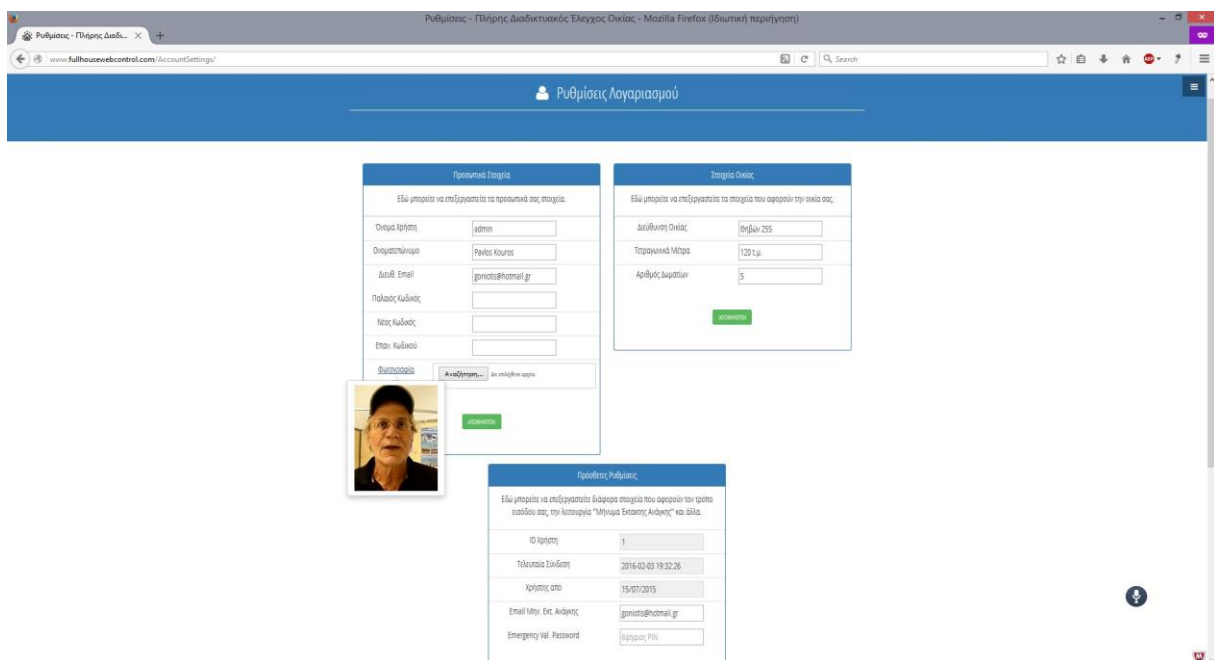
Σε αυτήν την σελίδα ο χρήστης μπορεί να ενεργοποιήσει/απενεργοποιήσει λειτουργίες του συστήματος όπως για παράδειγμα την είσοδο στο σύστημα μέσω επαλήθευσης δύο βημάτων (Two Step Verification) , θα αναλυθεί παρακάτω, τον διαδικτυακό έλεγχο κάποιου συγκεκριμένου δωματίου, την λειτουργία διακοπών και τις ειδοποιήσεις μέσω μηνύματος ηλεκτρονικού ταχυδρομείου σε περίπτωση έκτακτου συμβάντος. Με το πέρασμα του κέρσορα πάνω από κάθε ρύθμιση ο χρήστης μπορεί να διαβάσει ένα σύντομο κείμενο για την λειτουργία κάθε ρύθμισης.



Εικόνα 105: Σελίδα προσωπικών ρυθμίσεων

### 9.14 Σελίδα ρυθμίσεων λογαριασμού

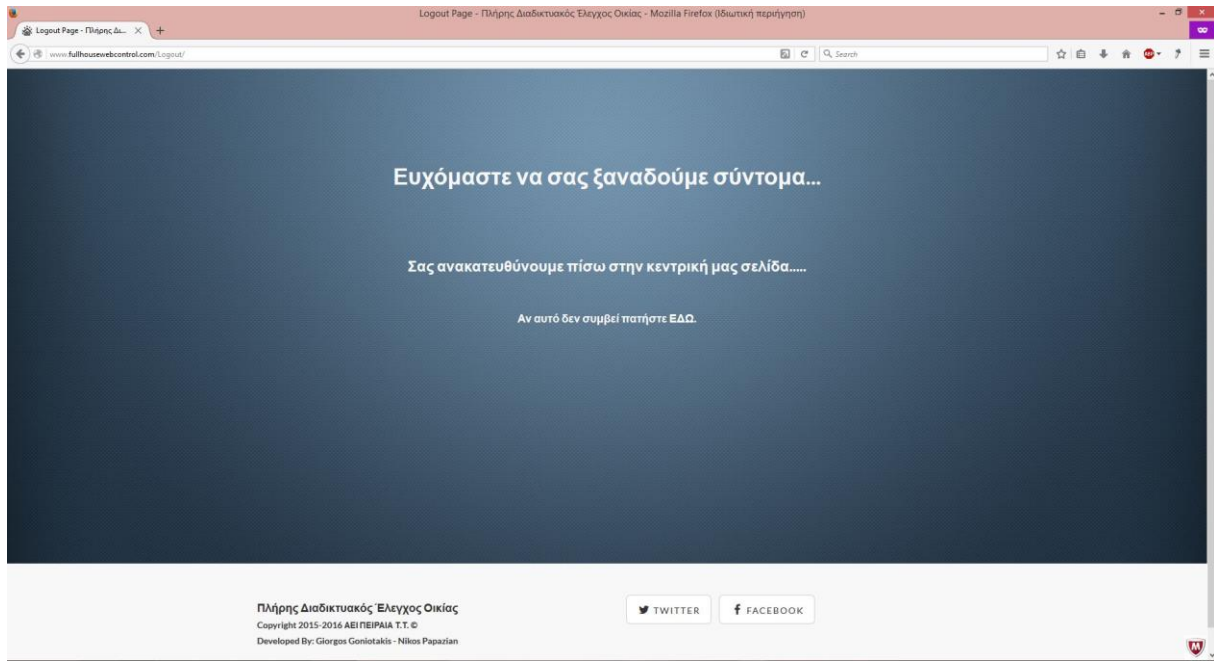
Σε αυτήν την σελίδα ο χρήστης μπορεί να τροποποιήσει τις προσωπικές του πληροφορίες όπως είναι η διεύθυνση της οικίας του, το ονοματεπώνυμό του, η διεύθυνση ηλεκτρονικού ταχυδρομείου αυτού και του ατόμου που θα ειδοποιηθεί σε περίπτωση έκτακτης ανάγκης, η φωτογραφία του και άλλα.



Εικόνα 106: Σελίδα ρυθμίσεων λογαριασμού

## 9.15 Σελίδα εξόδου από το σύστημα

Αυτή η σελίδα διαγράφει όλες τις μεταβλητές συνεδρίας από τον περιηγητή του χρήστη, καταγράφει την ημερομηνία και ώρα εξόδου του από το σύστημα και τον ανακατευθύνει μέσα σε 4 δευτερόλεπτα πίσω στην κεντρική ιστοσελίδα μας.



Εικόνα 107: Σελίδα εξόδου από το σύστημα

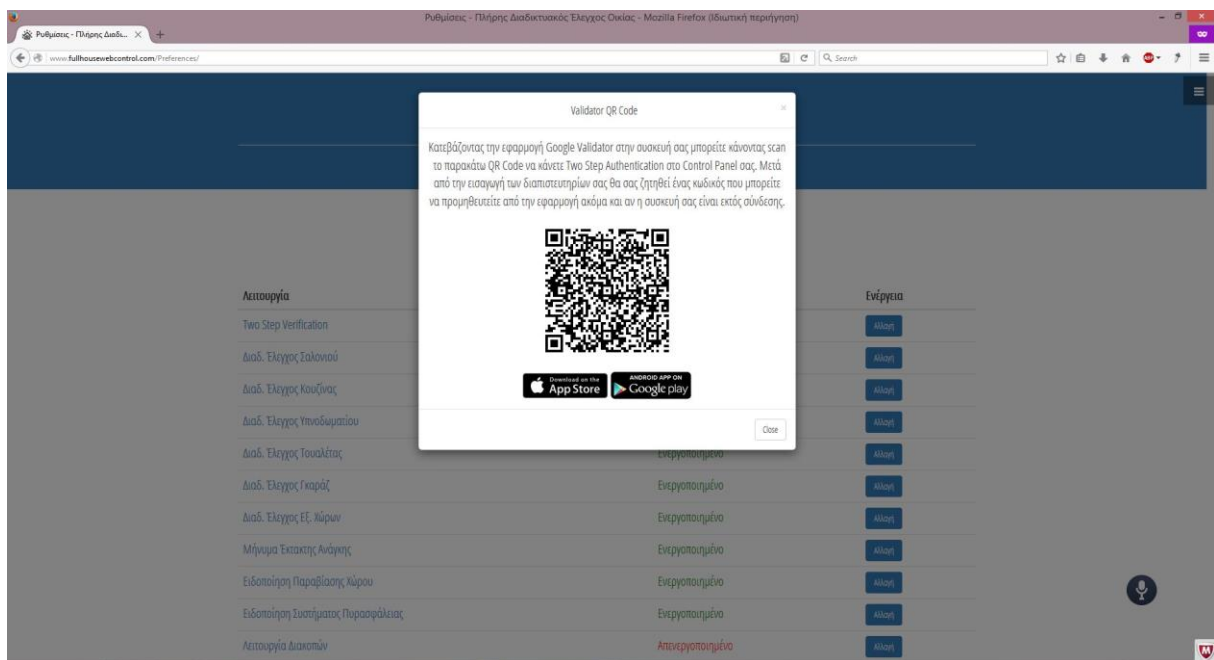
## 9.16 Είσοδος επαλήθευσης δύο σταδίων (Two Step Verification)

Σε περίπτωση που κάποιος θα κατάφερνε να αποσπάσει από τον χρήστη τα διαπιστευτήριά του οι συνέπειες θα ήταν δυσάρεστες καθώς θα μπορούσε να ελέγξει οποιαδήποτε λειτουργία της οικίας του. Για αυτόν τον λόγο ο χρήστης έχει την δυνατότητα μέσω των ρυθμίσεων να ενεργοποιήσει την λειτουργία επαλήθευσης δύο σταδίων.

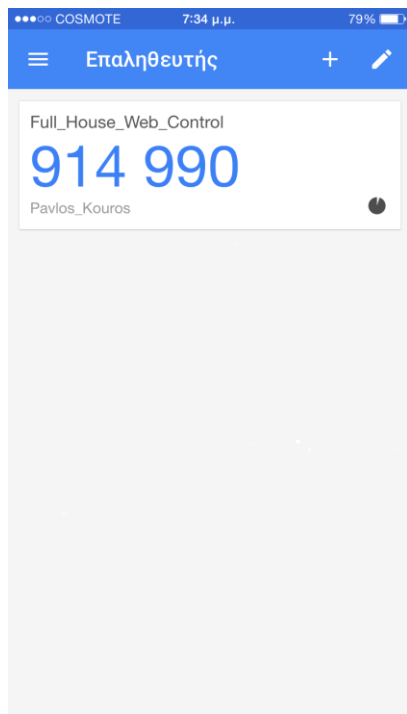
Η λειτουργία αυτή στηρίζεται σε έναν αλγόριθμο τον οποίο έχει κατασκευάσει η εταιρία Google ο οποίος δέχεται ένα 16ψήφιο μοναδικό κλειδί ασφαλείας το οποίο παρέχουμε εμείς στον χρήστη και βάσει της ώρας και ενός άλλου κλειδιού παράγει κωδικούς οι οποίοι καταστρέφονται μετά από δέκα δευτερόλεπτα. Τους κωδικούς αυτούς μπορεί να δει ο χρήστης μέσω της κινητής του συσκευής ή ταμπλέτας στην περίπτωση που εγκαταστήσει την εφαρμογή που ονομάζεται “Google Authenticator” και είναι δωρεάν. Η εφαρμογή παρέχεται για τις περισσότερες πλατφόρμες

λειτουργικών συστημάτων και δεν χρειάζεται να έχει πρόσβαση στο διαδίκτυο για να λειτουργήσει.

Αφού ο χρήστης εγκαταστήσει την εφαρμογή στην συσκευή του και περιηγηθεί στην καρτέλα ρυθμίσεων της ιστοσελίδας μας μπορεί πατώντας πάνω στην επιλογή “Two step verification” να διαβάσει αναλυτικές πληροφορίες για τον τρόπο συγχρονισμού της συσκευής του με το σύστημα. Η εικόνα τεχνολογίας QR Code παράγεται δυναμικά από το σύστημά μας και αφού ο χρήστης την σκανάρει με την εφαρμογή μέσω του κινητού του είναι έτοιμος να την χρησιμοποιήσει. Σε περίπτωση που ο χρήστης όμως δεν έχει στην διάθεση του την εφαρμογή και θέλει να πραγματοποιήσει είσοδο στο σύστημα μπορεί να το κάνει εφόσον έχει ορίσει έναν 6ψηφιο κωδικό (PIN) στην καρτέλα ρυθμίσεων λογαριασμού.

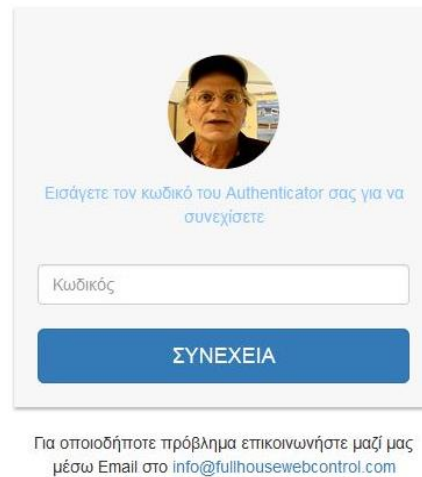


Εικόνα 108: Ενεργοποίηση επαλήθευσης δύο σταδίων



Εικόνα 109: Ρύθμιση εφαρμογής

## Two Step Verification



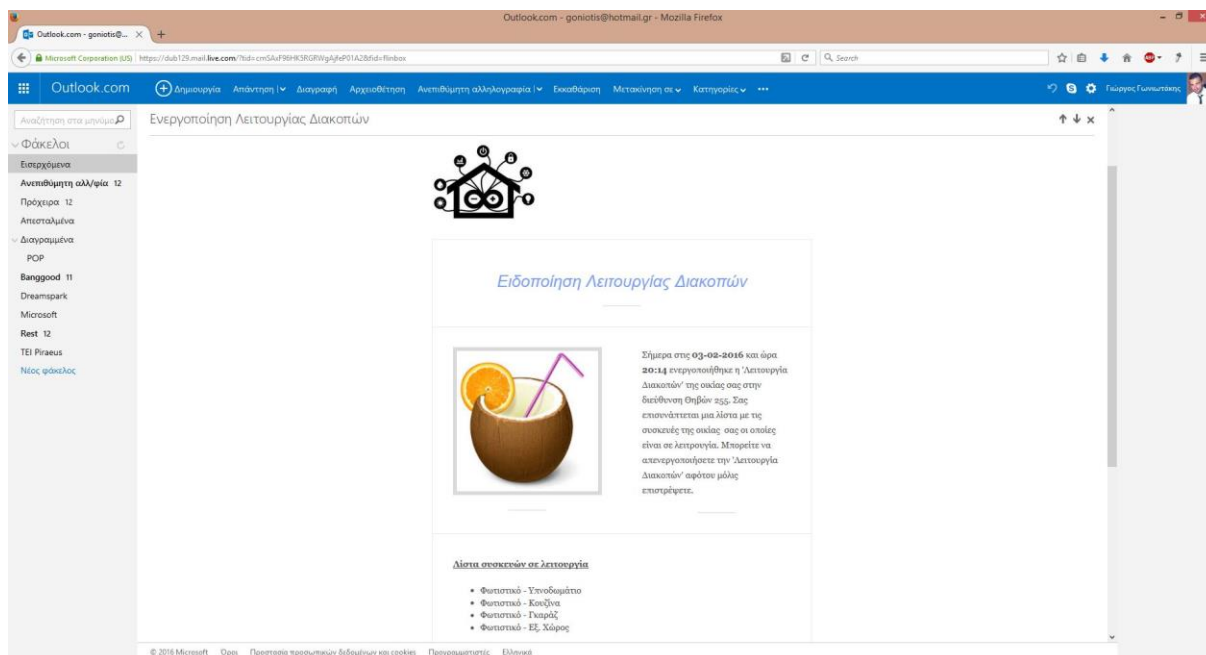
Εικόνα 110: Επαλήθευση δύο σταδίων

### 9.17 Ειδοποιήσεις συστήματος και ηλεκτρονικού ταχυδρομείου

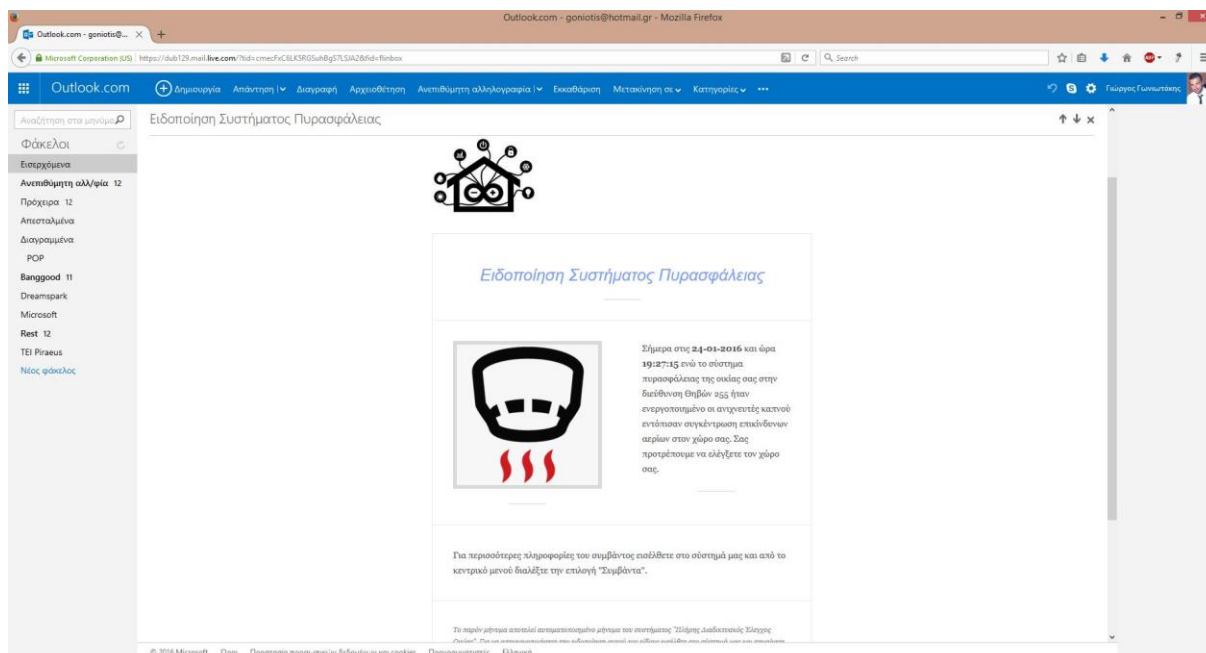
Στην περίπτωση που ο χρήστης του συστήματος θα ήθελε να εγκαταλείψει την οικία του για ένα μεγάλο χρονικό διάστημα και θέλει να βεβαιωθεί ότι δεν ξέχασε κάποια συσκευή ανοιχτή μπορεί να ενεργοποιήσει από τις ρυθμίσεις την λειτουργία διακοπών. Το πρόγραμμα κάνει έλεγχο στην βάση δεδομένων για τις συσκευές που είναι ενεργοποιημένες και μέσω του διακομιστή ηλεκτρονικής αλληλογραφίας μας (Email server) στέλνει ένα μήνυμα ηλεκτρονικού ταχυδρομείου στον χρήστη με την λίστα των συσκευών αυτών. Όταν ο χρήστης επιστρέψει στην οικία του μπορεί να απενεργοποιήσει την ρύθμιση αυτή.

Σε περίπτωση που η πλακέτα Arduino ενημερώσει το κύριο πρόγραμμα για κάποιο έκτακτο συμβάν όπως είναι η ενεργοποίηση του πλήκτρου έκτακτης ανάγκης, του συστήματος πυρασφάλειας και του συστήματος συναγερμού γίνεται μία καταχώρηση στην βάση δεδομένων. Έναν αυτοματοποιημένο πρόγραμμα εκτελείται στον διακομιστή (server) κάθε ένα δευτερόλεπτο και ελέγχει αυτά τα συμβάντα. Σε περίπτωση που ο χρήστης έχει ενεργοποιήσει την ενημέρωσή του μέσω μηνυμάτων ηλεκτρονικού ταχυδρομείου τον ειδοποιεί για το συμβάν αυτό. Επίσης αν βρίσκεται συνδεδεμένος στην ιστοσελίδα την στιγμή που συμβεί το συμβάν θα ειδοποιηθεί μέσω

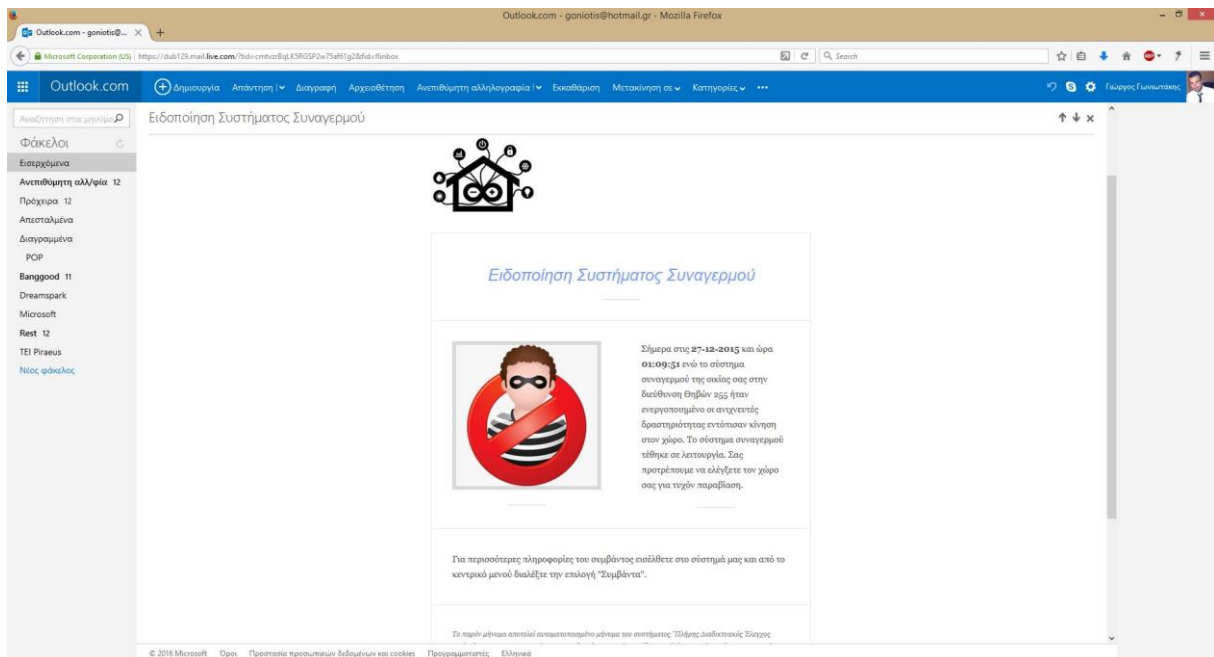
ηχητικού μηνύματος και γραπτού κειμένου (popup notification). Η ειδοποίηση θα εμφανιστεί στον χρήστη σε οποιαδήποτε κομμάτι της ιστοσελίδας βρίσκεται.



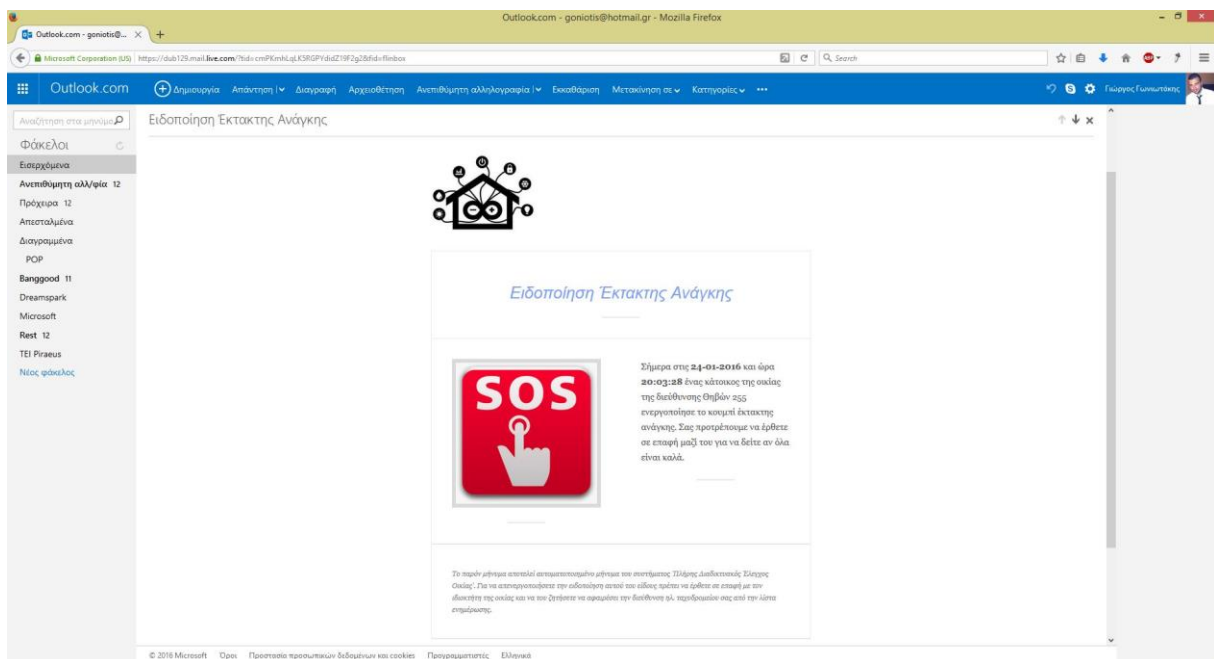
Εικόνα 111: Μήνυμα ηλεκτρονικού ταχυδρομείου σε περίπτωση ενεργοποίησης της λειτουργίας διακοπών



Εικόνα 112: Μήνυμα ηλεκτρονικού ταχυδρομείου σε περίπτωση ενεργοποίησης του συστήματος πυρασφάλειας

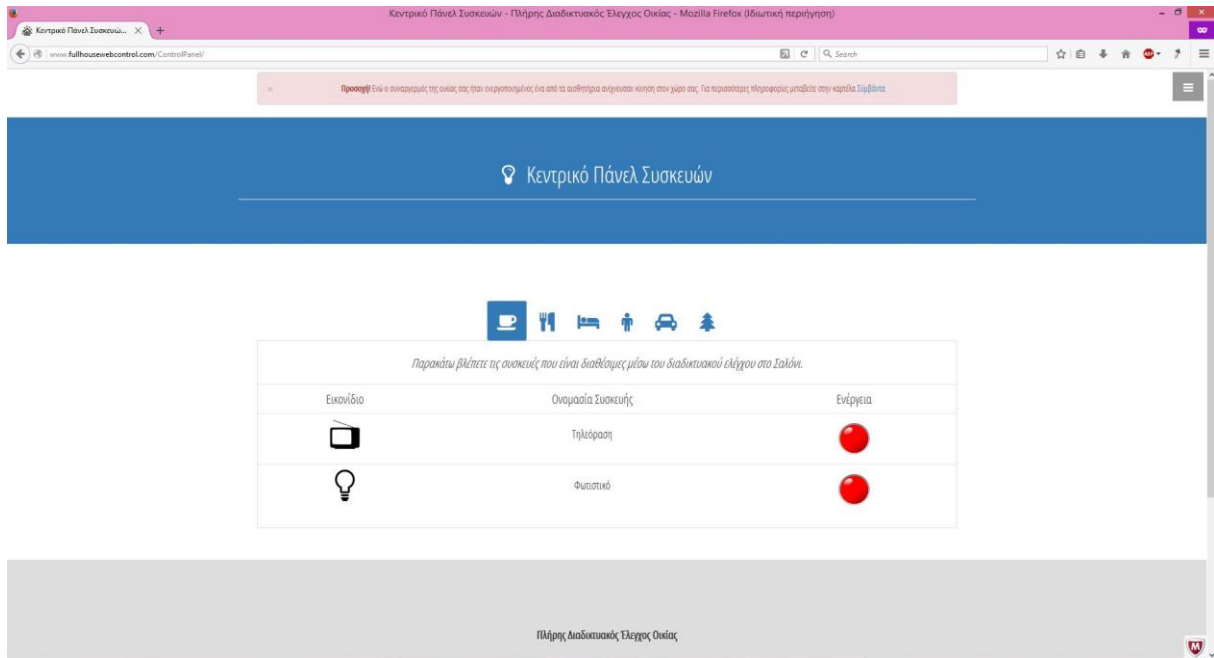


Εικόνα 113: Μήνυμα ηλεκτρονικού ταχυδρομείου σε περίπτωση ενεργοποίησης του συστήματος συναγερμού



Εικόνα 114: Μήνυμα ηλεκτρονικού ταχυδρομείου σε περίπτωση ενεργοποίησης του διακόπτη έκτακτης ανάγκης





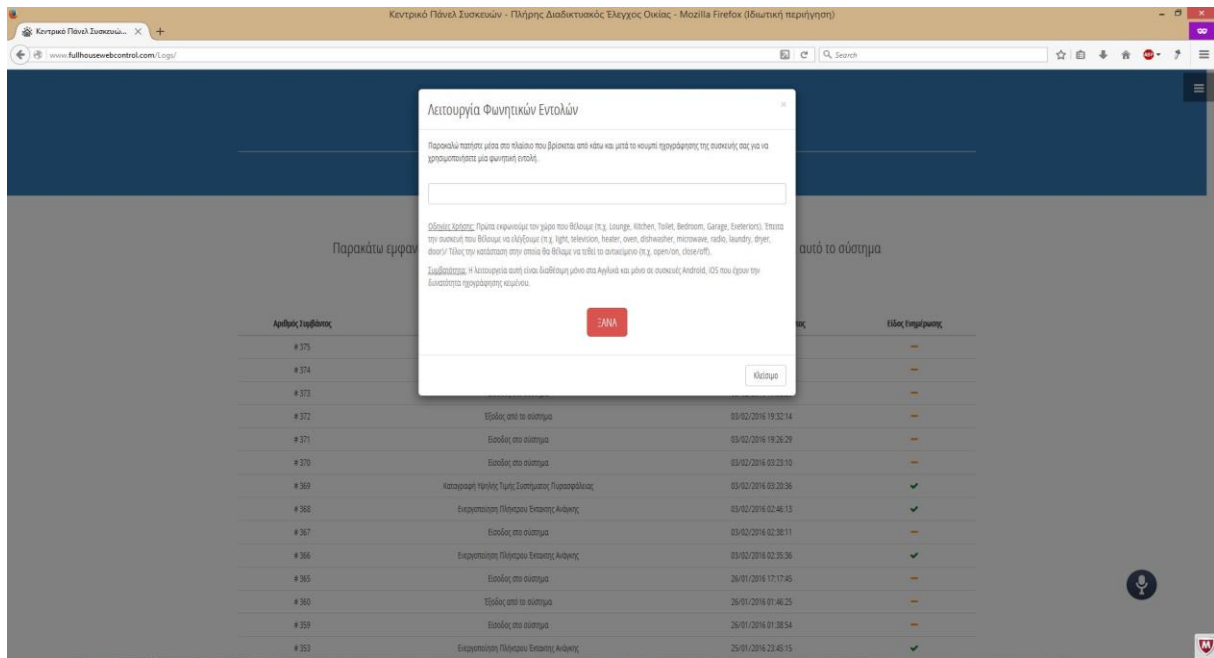
Εικόνα 115: Ειδοποίηση (Popup Notification) σε περίπτωση έκτακτου συμβάντος

### 9.18 Έλεγχος μέσω φωνητικών εντολών

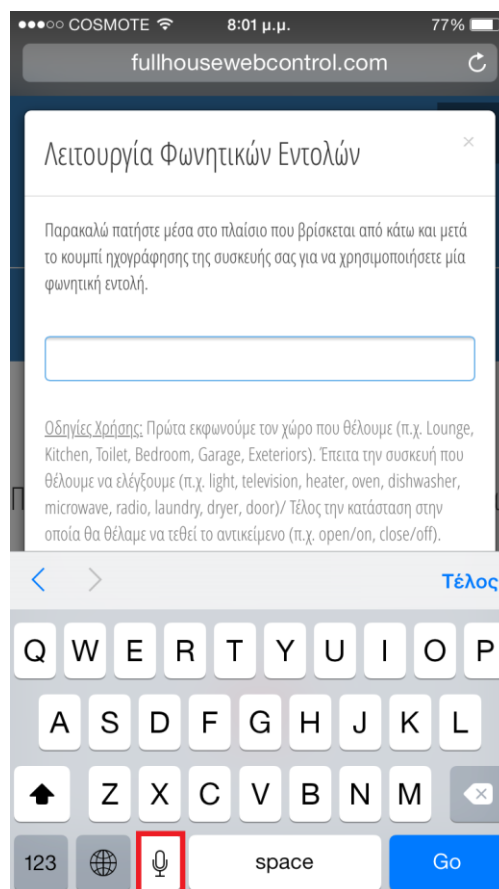
Στις περισσότερες σελίδες της διαδικτυακής μας εφαρμογής βρίσκεται στην κάτω δεξιά γωνία ένα εικονίδιο μικροφώνου. Σε περίπτωση που ο χρήστης το πατήσει εμφανίζεται ένα μενού με αναλυτικές οδηγίες που περιγράφει την χρήση των φωνητικών εντολών. Η λειτουργία αυτή στηρίζεται στο γεγονός ότι η συσκευή που χρησιμοποιεί ο χρήστης για να περιηγηθεί στην ιστοσελίδα υποστηρίζει την λειτουργία υπαγόρευσης κειμένου αντί πληκτρολόγησης. Ακολουθώντας τις οδηγίες ο χρήστης μπορεί να υπαγορεύσει στην αγγλική γλώσσα ένα δωμάτιο, μία συσκευή και αν επιθυμεί την ενεργοποίηση ή απενεργοποίησή της. Ακολουθεί ο έλεγχος ορθότητας μέσω JavaScript και εφόσον ολοκληρωθεί ακολουθεί η αποστολή σε αρχείο της γλώσσας PHP το οποίο κάνει αντιστοίχιση της φωνητικής εντολής στην κατάλληλη συσκευή. Έπειτα την αναζητά στην βάση δεδομένων και εφόσον την εντοπίσει την ενεργοποιεί/απενεργοποιεί και επιστρέφει ένα μήνυμα επιτυχίας στον χρήστη αλλιώς τον ενημερώνει ότι αυτή η συσκευή που ζήτησε δεν υπάρχει.



## Πλήρης Διαδικτυακός Έλεγχος Οικίας



Εικόνα 116: Παράθυρο φωνητικών εντολών



Εικόνα 117: Λειτουργία ηχογράφησης αντί για πληκτρολόγηση

### 9.19 Ασφάλεια διαδικτυακής εφαρμογής

Όσον αφορά την ασφάλεια του χρήστη εκτός του συστήματος επαλήθευσης δύο σταδίων (Two Step Verification) που αναφέραμε πριν έχουμε πραγματοποιήσει μια σειρά απαραίτητων ενεργειών.

Η ιστοσελίδα μας είναι συνδεδεμένη με το σύστημα McAfee Secure το οποίο την ελέγχει για οποιαδήποτε αθέμιτη ενέργεια μπορεί να προκληθεί μέσω της προσβολής της σελίδας από έναν ιό. Από την στιγμή που θα ανιχνευθεί κάτι στην ιστοσελίδα η εταιρία McAfee αναλαμβάνει να μας ειδοποιήσει μέσω μηνύματος ηλεκτρονικού ταχυδρομείου και να ενημερώσει τους επισκέπτες της σελίδας μας κατά την είσοδό τους.

Οι κωδικοί ασφαλείας των χρηστών κρυπτογραφούνται μέσω του αλγορίθμου MD5 και αποθηκεύονται στην βάση δεδομένων μας. Έτσι αν το περιεχόμενό της πέσει σε λάθος χέρια οι κωδικοί των χρηστών θα είναι άχρηστοι.

Κάθε χρήστης με την επιτυχή σύνδεση του στο σύστημα κατέχει ένα σύνολο από μεταβλητές συνεδρίας (Session Variables) οι οποίες αφαιρούνται από τον διακομιστή μας κάθε 30 λεπτά της ώρας. Έτσι σε περίπτωση που ο χρήστης αφήσει ανοιχτή την ιστοσελίδα σε κάποια συσκευή του χωρίς να το συνειδητοποιήσει και κάποιος άλλος προσπαθήσει να το εκμεταλλευτεί την στιγμή που θα προσπαθήσει να κινηθεί μέσα στην ιστοσελίδα μας θα αποσυνδεθεί αυτόματα.

Όλες οι εισοδοί στις φόρμες οι οποίες επιτρέπουν στον χρήστη να στείλει ένα αίτημα στον διακομιστή ελέγχονται για την ορθότητα του περιεχομένου τους και φιλτράρονται για ακατάλληλο περιεχόμενο μέσω του κώδικα λογικής που τρέχει πλευρά του διακομιστή (Server Side).

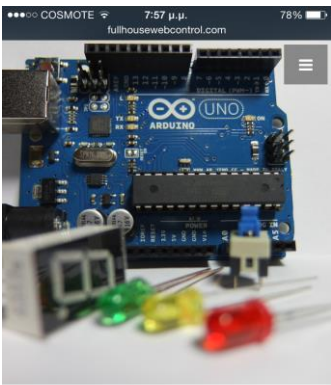
Τα αιτήματα που στέλνονται στην βάση δεδομένων προετοιμάζονται (prepared statements) και παραμετροποιούνται πριν σταλούν για εξυπηρέτηση. Έτσι αποφεύγονται φαινόμενα βασικών επιθέσεων όπως η MySQL Injection.

Τέλος να τονίσουμε ότι μια ιστοσελίδα για να είναι πραγματικά ασφαλής μια πολύ βασική προϋπόθεση είναι να είναι εξοπλισμένη με πιστοποιητικό ασφαλείας (SSL). Το πιστοποιητικό αυτό είναι απαραίτητο λόγω της κρυπτογράφησης των δεδομένων που παρέχει.

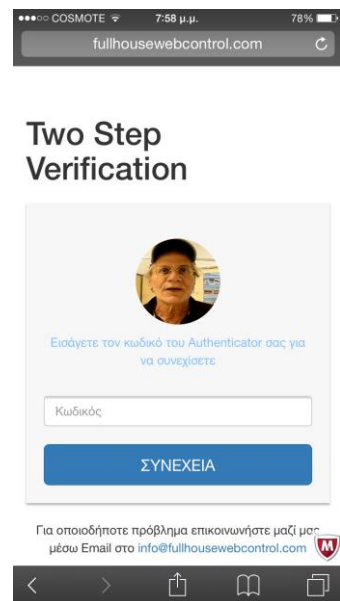
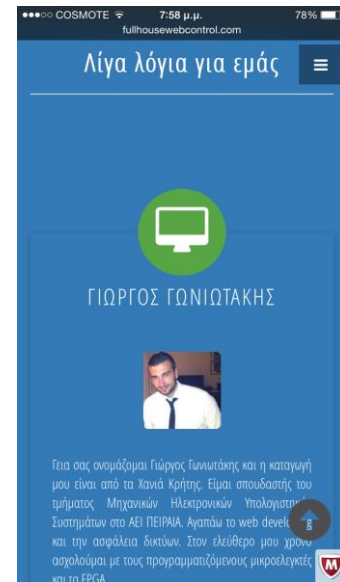
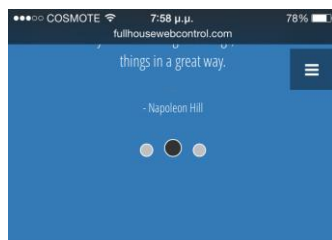
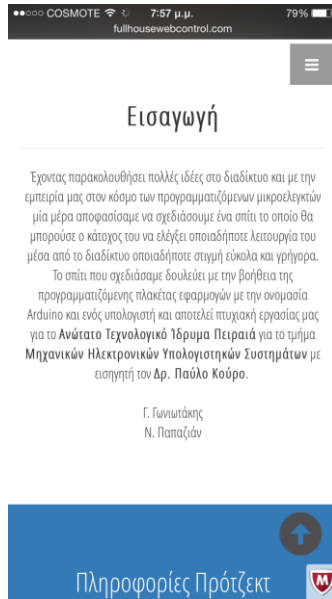
### 9.20 Συμβατότητα με φορητές συσκευές

Παρακάτω θα δούμε κάποιες φωτογραφίες για τον τρόπο εμφάνισης της ιστοσελίδας και των ειδοποιήσεων ηλεκτρονικού ταχυδρομείου σε κινητή συσκευή.

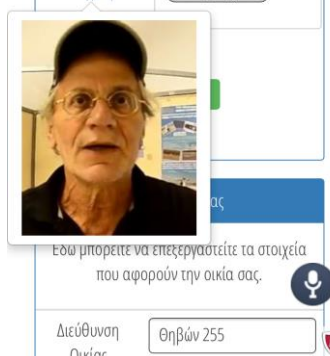
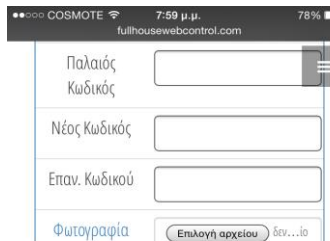
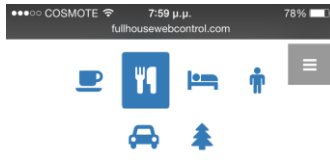
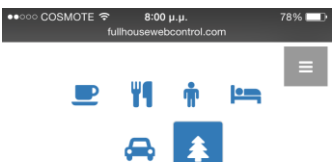
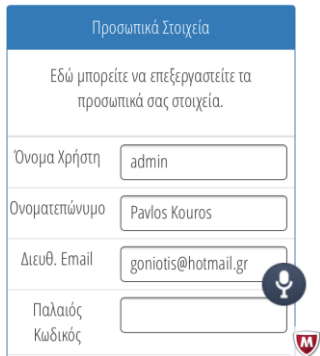
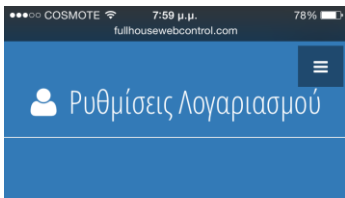
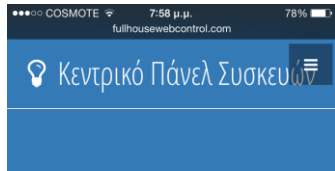
# Πλήρης Διαδικτυακός Έλεγχος Οικίας



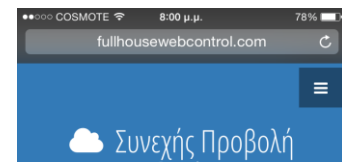
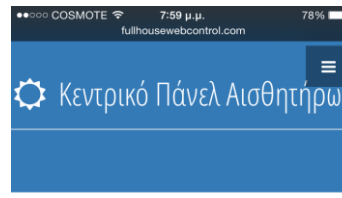
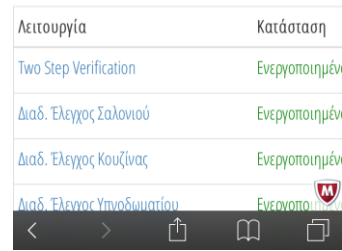
## Φωτογραφίες του Πρότζεκτ



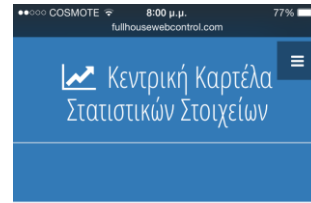
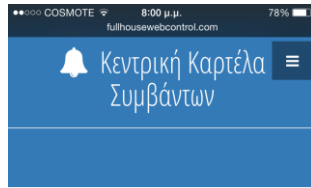
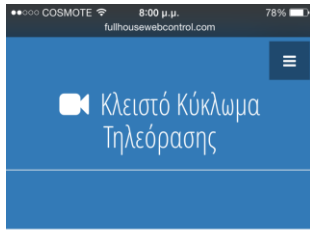
# Πλήρης Διαδικτυακός Έλεγχος Οικίας



Παρακάτω φαίνονται όλες οι προσωπικές σας επιλογές

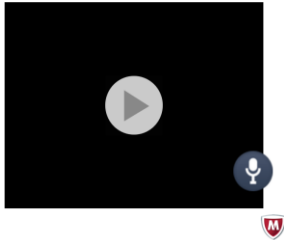


# Πλήρης Διαδικτυακός Έλεγχος Οικίας

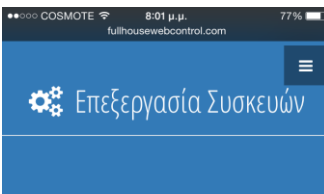


Παρακάτω εμφανίζονται τα κυριότερα γεγονότα που έχουν συμβεί σε σχέση με την οικία σας και αυτό το σύστημα

Παρακάτω βλέπετε ζωντανά στατιστικά στοιχεία σε σχέση με την κατάσταση θερμοκρασίας και υγρασίας της οικίας σας



Αριθμός Συμβάντος	Ονομασία Συμβάντος	Ημερομηνία/Ώρα Συμβάντος	Είδος Ενημέρωσης
# 381	Είσοδος στο σύστημα	03/02/2016 19:58:44	
# 380	Είσοδος στο σύστημα	03/02/2016 19:54:53	
# 379	Έξοδος από το σύστημα	03/02/2016	

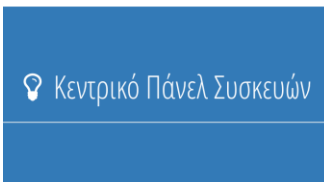
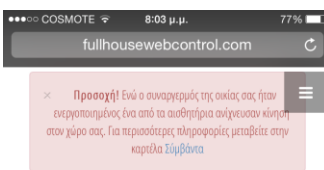
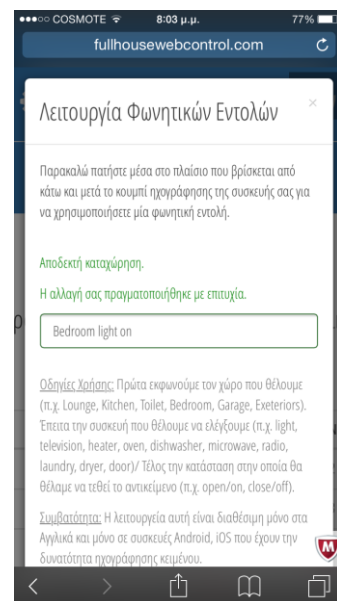


Παρακάτω βλέπετε το πλάνο συσκευών της οικίας σας

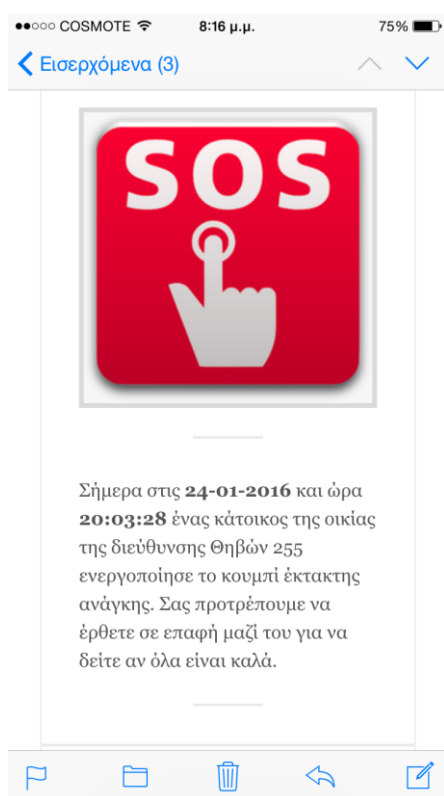
Στεγνωτήριο	ΚΛΕΙΣΤΟ	D6
Φωτιστικό	ΚΛΕΙΣΤΟ	D6
Ραδιόφωνο	ΚΛΕΙΣΤΟ	D7
Φούρνος	ΚΛΕΙΣΤΟ	D8
Φωτιστικό	ΑΝΟΙΧΤΟ	D9
Πληγτήριο Πιάτων	ΚΛΕΙΣΤΟ	D10
Φούρνος Μικροκυμάτων	ΚΛΕΙΣΤΟ	D11
Φωτιστικό	ΑΝΟΙΧΤΟ	D12
Τηλεόραση	ΚΛΕΙΣΤΟ	D22-D42
Φωτιστικό	ΑΝΟΙΧΤΟ	D44
Θερμοσίφωνα	ΚΛΕΙΣΤΟ	D45
Γκαραζόπορτα	ΚΛΕΙΣΤΟ	D49-D52



ID	ΔΩΜΑΤΙΟ	ΟΝΟΜΑΣΙΑ	ΤΕ
# 2	Σαλόνι	Φωτιστικό	
# 3	Τουαλέτα	Φωτιστικό	
# 4	Τουαλέτα	Πληγτήριο	
# 5	Τουαλέτα	Στεγνωτήριο	
# 6	Υπνοδωμάτιο	Φωτιστικό	
# 7	Υπνοδωμάτιο	Ραδιόφωνο	



## Πλήρης Διαδικτυακός Έλεγχος Οικίας



Εικόνες 118 – 149: Εμφάνιση μηνυμάτων ηλεκτρονικού ταχυδρομείου και ιστοσελίδας σε κινητή συσκευή

**(Κενό φύλλο)**

## ΚΕΦΑΛΑΙΟ 10

### ΜΑΚΕΤΑ ΕΦΑΡΜΟΓΗΣ

#### 10.1 Σκοπός κατασκευής μακέτας

Με σκοπό να παρουσιάσουμε στην πράξη την λειτουργία του συστήματός μας κατασκευάσαμε μία οικία σε μορφή μακέτας την οποία εξοπλίσαμε με κάποιες από τις συσκευές που πιθανόν να έχει μία πραγματική οικία. Επιπρόσθετα, εκτός από τις συσκευές τοποθετήσαμε και διάφορα αισθητήρια τα οποία θα εκμεταλλευτούμε για να παρουσιάσουμε τα πλεονεκτήματα που προσφέρει η εφαρμογή μας στην καθημερινότητα του χρήστη.

#### 10.2 Φυσικά χαρακτηριστικά μακέτας

Η πρώτη ύλη που επιλέξαμε για να δημιουργήσουμε την βάση, την οροφή και τους χώρους της οικίας είναι το ξύλο. Το ξύλο που χρησιμοποιούμε είναι είδους κόντρα πλακέ θαλάσσης το οποίο το κάνει ανθεκτικό στην υγρασία και αυξάνει την αντοχή του.

Η οικία αποτελείται από την οροφή, τον εξωτερικό χώρο και πέντε δωμάτια τα οποία είναι το σαλόνι, η κρεβατοκάμαρα, το μπάνιο, η κουζίνα και ο χώρος στάθμευσης. Οι διαστάσεις της μακέτας συμπεριλαμβανομένου και του εξωτερικού χώρου είναι 1.2 Μ x 1.1 Π x 0.9 Υ.

#### 10.3 Υλικά κατασκευής

Οι διαστάσεις των ξύλινων κομματιών τα οποία χρησιμοποιήθηκαν για την κατασκευή της μακέτας βρίσκονται παρακάτω:

Ποσότητα	Διαστάσεις (ΜΗΚΟΣ x ΠΛΑΤΟΣ x ΠΑΧΟΣ)
1	115 x 121 x 1
1	92.2 x 40 x 1
6	50.6 x 40 x 1
5	40.6 x 40 x 1
1	52.2 x 40 x 1
2	26 x 40 x 1



2	39 x 40 x 1
1	25 x 40 x 1
1	26.6 x 40 x 1
1	92.2 x 98.8 x 1
1	115 x 121 x 1

Πίνακας 14: Διαστάσεις Ξύλινων Κομματιών Μακέτας

Επίσης για την κατασκευή χρησιμοποιήθηκαν τα παρακάτω υλικά:

Τεμάχια	Ονομασία Υλικού
-	Μονωτική Ταινία
2	2m x 2m Τεχνητός χλοοτάπητας
3	Ξύλο μπάλασαμ 5mm
5	Ξύλο μπάλασαμ 3mm
5	Ξύλο μπάλασαμ σε σχήμα κύβου 1.5mm
1	Ξυλόκολλα
1	Πακέτο με αυτοκόλλητο χαρτί εκτύπωσης
1	Ρυζόχαρτο
-	Κομμάτια από διάφορα υφάσματα
5	Χρώματα για τον βάψιμο των τοίχων
1	Φύλλο αλουμινίου
2	Κομμάτια χαρτόνι
-	Σπάγκος
2	Γλάστρες με φυτά
10	Θήκες για ηλεκτρολογικό πίνακα

Πίνακας 15: Υλικά κατασκευής

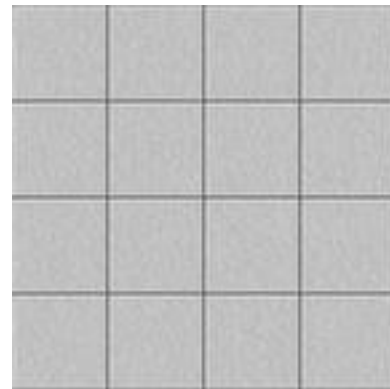


Εικόνα 150: Χρωματολόγιο οικίας

Για να φτιάξουμε το δάπεδο χρησιμοποιήσαμε καφέ χρώμα και εκτυπώσαμε, σε αυτοκόλλητο χαρτί, πλακάκια τα οποία τοποθετήσαμε στο μπάνιο και στην κουζίνα της οικίας. Επίσης με την ίδια μέθοδο εκτυπώσαμε παράθυρα, πόρτες δωματίων και εισόδου καθώς και την πόρτα του χώρου στάθμευσης της οικίας.



Εικόνα 151: Δάπεδο κουζίνας



Εικόνα 152: Δάπεδο μπάνιου



Εικόνα 153: Εσωτερική όψη παραθύρου



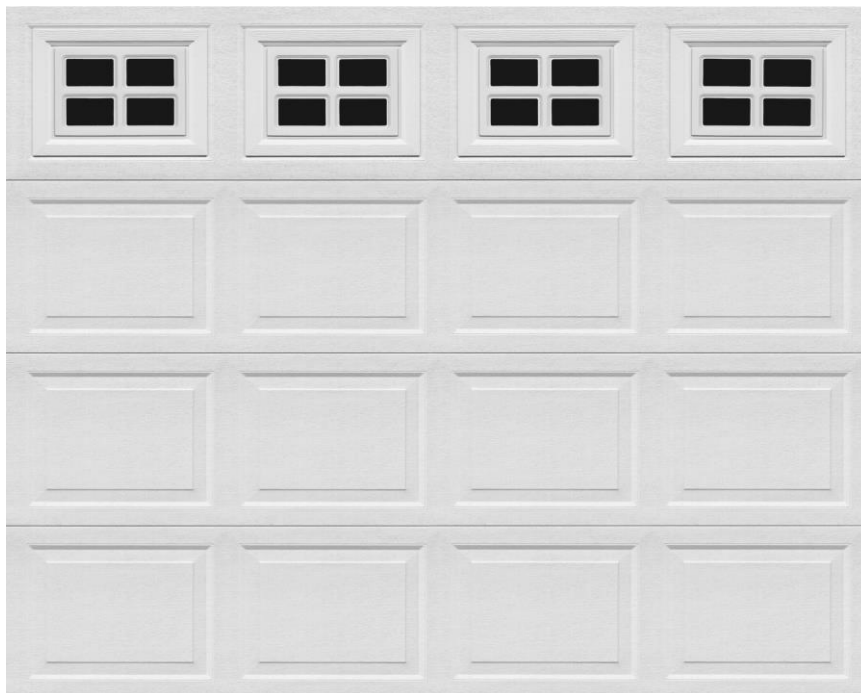
Εικόνα 154: Εξωτερική όψη παραθύρου



Εικόνα 155: Κεντρική πόρτα οικίας



Εικόνα 156: Πόρτα δωματίου οικίας



Εικόνα 157: Πόρτα χώρου στάθμευσης οικίας

#### 10.4 Συσσκευές και αισθητήρια μακέτας

Για να δείξουμε την λειτουργία της εφαρμογής μας χρησιμοποιήσαμε πλαστικές ηλεκτρικές συσκευές που βρήκαμε σε είδη παιχνιδιών τις οποίες τις εξοπλίσαμε με led λειτουργίας. Η λειτουργία και η χρήση των συσκευών και των αισθητηρίων που χρησιμοποιήσαμε περιγράφεται αναλυτικά στο κεφάλαιο 5.

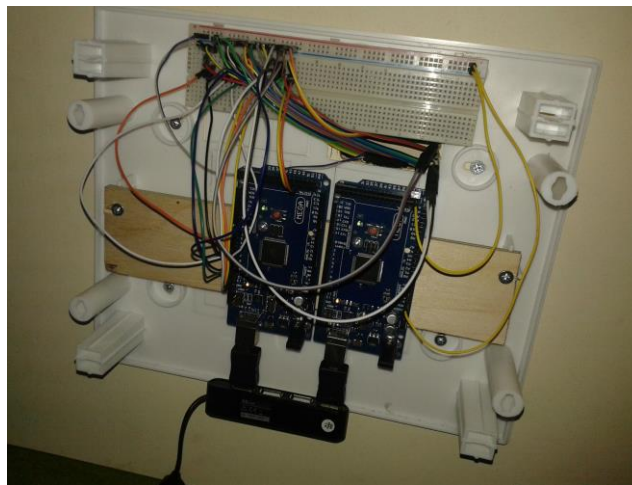
Παρακάτω βρίσκεται η λίστα του υλικού που χρησιμοποιήσαμε για να εξοπλίσουμε το σπίτι καθώς και να κατασκευάσουμε διάφορα συστήματα του όπως το σύστημα συναγερμού, η τέντα, το ανεμόμετρο και άλλα.

Ποσότητα	Ονομασία
2	Προγραμματιζόμενες πλακέτες Arduino MEGA
3	Κινητήρες συνεχούς
2	Βηματικούς κινητήρες
-	Led κόκκινου χρώματος
-	Led μπλε χρώματος
-	Led κίτρινου χρώματος
-	Led λευκού χρώματος
6	Σποτ των 4 led
2	Breadboard 800 υποδοχών
1	Διαδικτυακή κάμερα
1	Οθόνη 3.2''
3	Οθόνες 0.96''
1	Οθόνη 32 χαρακτήρων
1	Πληκτρολόγιο 12 ψηφίων
1	Διακόπτη push
2	Κανονικούς διακόπτες δύο καταστάσεων
2	Αντιστάσεις 4,7K
2	Ρελέδες
6	Αισθητήρια υγρασίας – θερμοκρασίας DHT11
1	Αισθητήριο ανίχνευσης βροχής
1	Αισθητήριο ανίχνευσης φωτεινότητας
2	Αισθητήρια ανίχνευσης υγρασίας χρώματος φυτών
3	Αισθητήρια ανίχνευσης επικίνδυνων αερίων MQ2

6	Διακόπτες αφής
2	Αισθητήρια ανίχνευσης κίνησης
1	Ηλεκτρολογικό πίνακα
1	Σειρήνα συναγερμού μικρού μεγέθους
-	Καλώδια breadboard
1	Μεταβλητή αντίσταση 10K
1	Ραδιοφωνάκι
2	Καλώδια σύνδεσης USB
1	Τροφοδοτικό 9V 1A

Πίνακας 16: Ηλεκτρικό – ηλεκτρονικό υλικό μακέτας

Οι συσκευές και τα αισθητήρια των δωματίων μέσω καλωδίων, τα οποία βρίσκονται ανάμεσα στους τοίχους, συνδέονται στις πλακέτες Arduino MEGA που βρίσκονται μέσα στον διαμορφωμένο ηλεκτρολογικό πίνακα της οικίας.



Εικόνα 158: Ηλεκτρολογικός Πίνακας Οικίας

Έπειτα μπορούμε από την κάτω πλευρά του ηλεκτρολογικού πίνακα να συνδέσουμε τις πλακέτες με καλώδιο USB και τροφοδοσία όπως φαίνεται παραπάνω.

### 10.5 Παρουσίαση χώρων

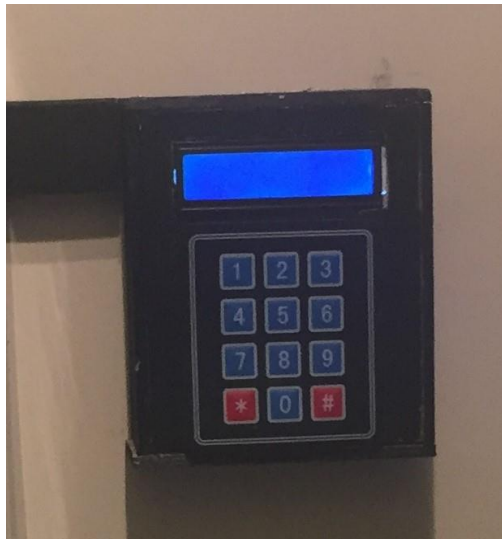
Στις παρακάτω φωτογραφίες θα δούμε κάποιες από τις συσκευές που δημιουργήσαμε και έπειτα θα προχωρήσουμε στους χώρους της οικίας.



Εικόνα 159: Τηλεόραση Οικίας



Εικόνα 160: Ηλεκτρολογικός Πίνακας



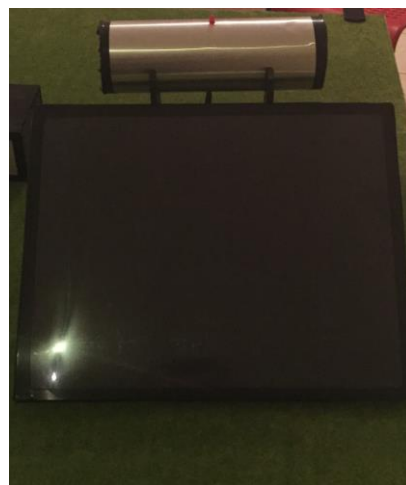
Εικόνα 161: Σύστημα συναγερμού



Εικόνα 162: Μηχανισμός τέντας



Εικόνα 163: Ανεμόμετρο



Εικόνα 164: Ηλιακός θερμοσίφωνας



Εικόνα 165: Μπροστά όψη πινακίδας logo



Εικόνα 166: Πίσω όψη πινακίδας logo



Εικόνα 167: Πάνελ ρύθμισης τέντας και ελέγχου διακόπτη έκτακτης ανάγκης

Παρακάτω θα παρουσιάσουμε κάποιες φωτογραφίες των δωματίων και του εξωτερικού χώρου της οικίας.





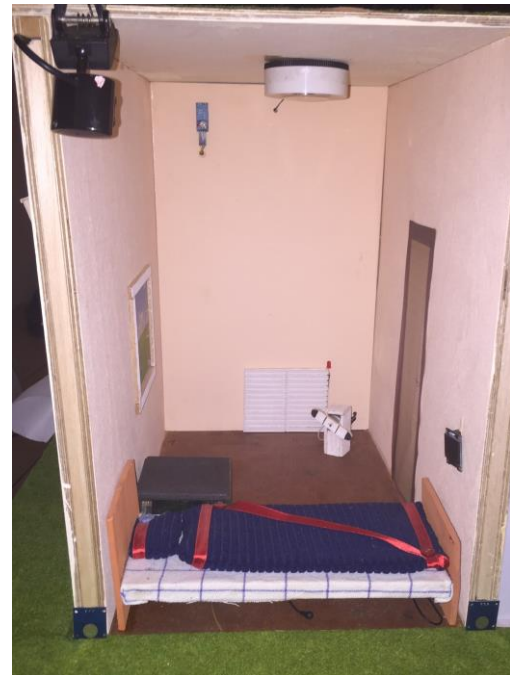
Εικόνα 168: Χώρος στάθμευσης



Εικόνα 169: Χώρος καθιστικού



Εικόνα 170: Χώρος μπάνιου

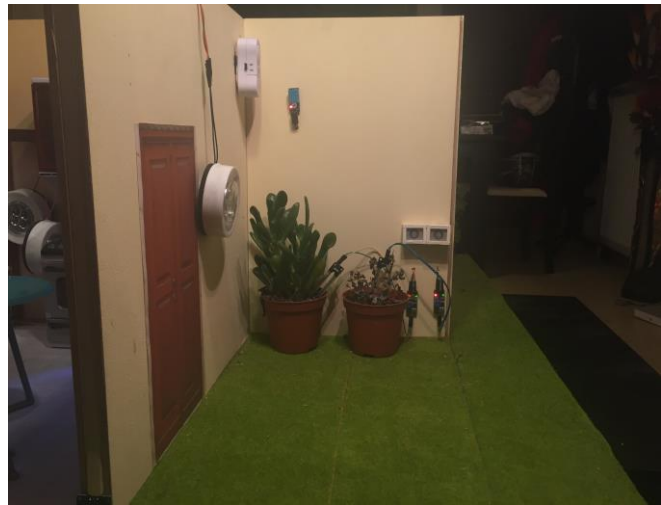


Εικόνα 171: Υπνοδωμάτιο





Εικόνα 172: Κουζίνα Εικόνα



Εικόνα 173: Είσοδος οικίας



Εικόνα 174: Μπροστά όψη οικίας



Εικόνα 175: Πίσω όψη οικίας



Εικόνα 176: Πλαϊνή όψη οικίας

**(Κενό φύλλο)**

## ΚΕΦΑΛΑΙΟ 11

### ΕΦΑΡΜΟΓΗ ΣΕ ΠΡΑΓΜΑΤΙΚΕΣ ΣΥΝΘΗΚΕΣ

#### 11.1 Υλικό πραγματικής εφαρμογής

Από πλευράς υλικού ο χρήστης για να ελέγξει τις συσκευές του θα χρειαστεί ρελέδες είτε σε μορφή έτοιμης πλακέτας (module) είτε όχι. Πάνω σε αυτούς θα συνδέει οποιαδήποτε πρίζα ή συσκευή θέλει να ελέγξει. Έπειτα με πολύ απλό τρόπο μπορεί να συνδέσει τις εισόδους της πλακέτες με τους ρελέδες πάνω στο Arduino της επιλογής του και είναι έτοιμος.

Όσον αφορά τα αισθητήρια δεν θα χρειαστεί να πραγματοποιηθεί καμία αλλαγή. Η προσθήκη επιπλέον αισθητηρίων είναι δυνατή σε περίπτωση που το αισθητήριο παράγει αναλογική μέτρηση και όχι ψηφιακή. Στην περίπτωση της ψηφιακής μέτρησης θα χρειαστεί να εγγραφεί νέα βιβλιοθήκη οδήγησης του αισθητηρίου. Στην περίπτωση της αναλογικής ο χρήστης μπορεί να συνδέσει το αισθητήριό του στην αναλογική ακίδα της πλακέτας που επιθυμεί.

#### 11.2 Πηγαίος κώδικας Arduino

Όσον αφορά τον κώδικα της πλακέτας Arduino που χειρίζεται τις συσκευές αυτός θα ελαττωθεί σημαντικά. Σε κανονικές συνθήκες θα θέλαμε η πλακέτα μόνο να ενεργοποιεί και να απενεργοποιεί μία συσκευή και όχι να λειτουργεί ως οδηγός υλικού για παράδειγμα να παράγει γραφικά για την τηλεόρασή μας, να χειρίζεται βηματικούς κινητήρες και άλλα. Για αυτό πολύ εύκολα το μόνο που έχουμε να κάνουμε είναι να αφαιρέσουμε τις επιπλέον λειτουργίες αυτές και να απελευθερώσουμε προς έλεγχο όλες τις ακίδες της πλακέτας.

Ο τροποποιημένος κώδικας της πλακέτας φαίνεται παρακάτω.

```
int LostConnection = 0; //Μεταβλητή αποθήκευσης κατάστασης σύνδεσης της
πλακέτας με το πρόγραμμα
byte inputByte[] = {0,0,0}; //Μεταβλητή αποθήκευσης μηνυμάτων σειριακής
επικοινωνίας
```

```
//Λειτουργία σύνδεσης πλακέτας με το πρόγραμμα C#
bool sindesi () {
```

```
    boolean sin = false;
```

```
    if (Serial.available())
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
{
  inputByte[0] = Serial.read();
  delay(10);
  inputByte[1] = Serial.read();
  delay(10);
  inputByte[2] = Serial.read();
  delay(10);
}

if(inputByte[0] == 10 && inputByte[1] == 20 && inputByte[2] == 30)
{
  Serial.print("Device 0");
  return(true);
}
}

void setup(){

  Serial.begin(9600); //Εναρξη σειριακής επικοινωνίας

  //Ορισμός λειτουργίας ψηφιακών ακίδων ως εξόδους
  for (int i=2; i<54; i++)
  {

    pinMode(i,OUTPUT);

  }

}

//Κεντρική λειτουργία προγράμματος
void loop() {

  //Αν επιτευχθεί σύνδεση με το πρόγραμμα
  if(sindesi() == true)
  {

    if (LostConnection == 0)
    {
      LostConnection=1;
    }

    if(inputByte[0] == 5)
    {

      if (inputByte[2] == 1){

        //Ενεργοποίηση της συσκευής που βρίσκεται στην ακίδα X όταν η
        αίτηση έχει την μορφή 5,X,1
        digitalWrite(inputByte[1], HIGH);

      }

      else if (inputByte[2]==0){
```

```
        //Απενεργοποίηση της συσκευής που βρίσκεται στην ακίδα X όταν η
        αίτηση έχει την μορφή 5,X,0
        digitalWrite(inputByte[1], LOW);
    }

    }

    inputByte[0] = 0;
    inputByte[1] = 0;
    inputByte[2] = 0;
}

//Σε περίπτωση διακοπής Internet ή κάποιου άλλου προβλήματος ενεργοποίησε
όλες τις συσκευές έτσι ώστε να μπορεί να τις χειριστεί ο χρήστης
else if(LostConnection==0){

    for (int i=2; i<54; i++)
    {
        digitalWrite(i,HIGH);
    }
}
}
```

### 11.3 Οδηγίες χρήσης πραγματικής εφαρμογής

Ξεκινώντας πρέπει να είμαστε σίγουροι ότι έχουμε τις δύο πλακέτες Arduino οι οποίες έχουν μέσα τα προγράμματά μας εγκατεστημένα. Έπειτα πρέπει να έχουμε καταχωρήσει έναν χρήστη στην βάση δεδομένων και να έχουμε εισάγει αναλυτικά τις πληροφορίες των αισθητηρίων της οικίας μας.

Αφού έχουμε ολοκληρώσει και αυτές τις ενέργειες συνδέουμε σε έναν ηλεκτρονικό υπολογιστή τις δύο πλακέτες Arduino μέσω USB καθώς και την τροφοδοσία τους. Χρησιμοποιούμε το πρόγραμμα το οποίο μας έχει δοθεί και είναι συνδεδεμένο με τον αριθμό πελάτη μας. Πατάμε εκκίνηση λειτουργίας και ανοίγουμε τον περιηγητή μας.

Για να κάνουμε τις συσκευές μας διαθέσιμες προς διαδικτυακό έλεγχο πηγαίνουμε στην σελίδα επεξεργασίας συσκευών και τις προσθέτουμε. Διαλέγουμε προσεκτικά τα στοιχεία της συσκευής μας και δίνουμε ιδιαίτερη προσοχή στο πεδίο ακίδας σύνδεσης. Σιγουρευόμαστε ότι η ακίδα σύνδεσης της συσκευής που θέλουμε να ελέγξουμε είναι ίδια με την ακίδα που είναι συνδεδεμένος ο ρελές.

Τέλος μεταβαίνοντας στην σελίδα διαχείρισης συσκευών μπορούμε να χειριστούμε την συσκευή που καταχωρήσαμε.

## ΠΑΡΑΡΤΗΜΑ Α΄

### Κώδικα πλακέτας Arduino συσκευών

```
#include <UTFT.h> //Βιβλιοθήκη οθόνης TFT
#include <avr/pgmspace.h> //Βιβλιοθήκη γενικής χρήσης
#include <Stepper.h> //Βιβλιοθήκη βηματικού κινητήρα

bool garagestatus = false; //Μεταβλητή αποθήκευσης κατάστασης γκαραζόπορτας
boolean tvstatus = false; //Μεταβλητή αποθήκευσης κατάστασης τηλεόρασης
int FirstTime = 0; //Μεταβλητή γκαραζόπορτας
int LostConnection = 0; //Μεταβλητή αποθήκευσης κατάστασης σύνδεσης της
πλακέτας με το πρόγραμμα
extern uint8_t SmallFont[]; //Δήλωση μικρού μεγέθους γραμματοσειράς οθόνης
TFT
extern uint8_t BigFont[]; //Δήλωση μεγάλου μεγέθους γραμματοσειράς οθόνης
TFT
extern unsigned int alt[0xD16]; //Δήλωση πίνακα λογότυπου προς εμφάνιση
στην οθόνη TFT
byte inputByte[] = {0,0,0}; //Μεταβλητή αποθήκευσης μηνυμάτων σειριακής
επικοινωνίας

Stepper myStepper(200, 49, 50, 51, 52); //Δήλωση οντότητας βηματικού
κινητήρα
UTFT myGLCD(SSD1289,38,39,40,41); //Δήλωση οντότητας οθόνης TFT

//Λειτουργία σύνδεσης πλακέτας με το πρόγραμμα C#
bool sindesi(){

    boolean sin = false;

    if (Serial.available())
    {
        inputByte[0] = Serial.read();
        delay(10);
        inputByte[1] = Serial.read();
        delay(10);
        inputByte[2] = Serial.read();
        delay(10);
    }

    if(inputByte[0] == 10 && inputByte[1] == 20 && inputByte[2] == 30)
    {
        Serial.print("Device 0");
        return(true);
    }
}

void setup(){

    myGLCD.InitLCD(); //Σύνδεση οθόνης TFT
    myStepper.setSpeed(60); //Ορισμός ταχύτητας βηματικού κινητήρα
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
Serial.begin(9600); //Έναρξη σειριακής επικοινωνίας

//Ορισμός λειτουργίας ψηφιακών ακίδων ως εξόδους
for (int i=2; i<14; i++)
{
    pinMode(i,OUTPUT);
}

for(int i=42; i<49; i++)
{
    pinMode(i,OUTPUT);
}

//Ορισμός των ψηφιακών ακίδων που βρίσκονται οι διακόπτες αφής ως
εισόδους
pinMode(16,INPUT);
pinMode(17,INPUT);
pinMode(18,INPUT);
pinMode(19,INPUT);
pinMode(20,INPUT);
pinMode(21,INPUT);
}

//Κεντρική λειτουργία προγράμματος
void loop() {

    //Αν επιτευχθεί σύνδεση με το πρόγραμμα
    if(sindesi() == true)
    {

        if (LostConnection == 0){
            LostConnection=1;
        }

        if(inputByte[0] == 5)
        {

            if (inputByte[2] == 1){

                //Αν η αίτηση είναι η 5,30,1 ενεργοποίησε την τηλεόραση
                if (inputByte[1] == 30){
                    televisionon();
                }

                //Αν η αίτηση είναι η 5,49,1 άνοιξε την πόρτα του γκαράζ
                else if(inputByte[1] == 49 && garagestatus==false){
                    if (FirstTime==0){
                        garagestatus=true;
                        FirstTime++;
                    }
                    else{
                        garageon();
                    }
                }
            }
        }
    }
}
```



## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    }
  }

  //Ενεργοποίηση φωτισμού δωματίων και απενεργοποίηση πινακίδων
Exit
  else if(inputByte[1] == 2){
    digitalWrite(2, HIGH);
    digitalWrite(46,LOW);
  }
  else if(inputByte[1] == 9){
    digitalWrite(9, HIGH);
    digitalWrite(47,LOW);
  }
  else if(inputByte[1] == 12){
    digitalWrite(12, HIGH);
    digitalWrite(48,LOW);
  }

  //Ενεργοποίηση της συσκευής που βρίσκεται στην ακίδα X όταν η
  αίτηση έχει την μορφή 5,X,1
  else{
    digitalWrite(inputByte[1], HIGH);
  }
}

else if (inputByte[2]==0){

  if (inputByte[1] == 30){
    televisionoff();
  }

  else if(inputByte[1] == 49 && garagestatus==true){

    if (FirstTime==0){
      garagestatus=false;
      FirstTime++;
    }

    else{
      garageoff();
    }
  }

  else if(inputByte[1] == 2){
    digitalWrite(2, LOW);
    digitalWrite(46,HIGH);
  }
  else if(inputByte[1] == 9){
    digitalWrite(9, LOW);
    digitalWrite(47,HIGH);
  }
  else if(inputByte[1] == 12){
    digitalWrite(12, LOW);
    digitalWrite(48,HIGH);
  }
  else{
    digitalWrite(inputByte[1], LOW);
  }
}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    }

    sendButtonState(); //Κλήση λειτουργίας ελέγχου κατάστασης διακοπών
αφής

    inputByte[0] = 0;
    inputByte[1] = 0;
    inputByte[2] = 0;

    }

    //Σε περίπτωση διακοπής Internet ή κάποιου άλλου προβλήματος ενεργοποίησε
όλες τις συσκευές έτσι ώστε να μπορεί να τις χειριστεί ο χρήστης
    else if(LostConnection==0){

        televisionon();

        for (int i=2; i<14; i++)
        {
            digitalWrite(i,HIGH);
        }

        for(int i=42; i<46; i++)
        {
            digitalWrite(i,HIGH);
        }
    }
}

//Λειτουργία ανοίγματος πόρτας γκαράζ
void garageon(){
myStepper.step(3500);
garagestatus =true;
}

//Λειτουργία κλεισίματος πόρτας γκαράζ
void garageoff(){
myStepper.step(-2000);
garagestatus=false;
}

//Λειτουργία ενεργοποίησης τηλεόρασης
void televisionon()
{

    unsigned long currentMillis = millis();
    unsigned long timeOfLastLedEvent = 0;
    int intervalOFF = 0;

    if (currentMillis - timeOfLastLedEvent > intervalOFF && tvstatus == false){

        timeOfLastLedEvent = currentMillis;
        tvstatus = true;
        digitalWrite(42,HIGH); //Ενεργοποίηση οπίσθιου φωτισμού τηλεόρασης
        myGLCD.clrScr(); //Καθαρισμός περιεχομένου οθόνης
        myGLCD.setColor(VGA_WHITE); //Ορισμός λευκού χρώματος γραμμάτων
        myGLCD.setBackColor(0, 0, 0); //Ορισμός μαύρου χρώματος background
        myGLCD.drawBitmap (90, 100, 67, 50, alt, 2); //Σχεδιασμός λογότυπου στην
οθόνη
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
myGLCD.setFont(BigFont); //Ορισμός μεγάλης γραμματοσειράς
myGLCD.print("House Web Control", CENTER, 40);
myGLCD.print("AEI Piraeus TT MHYS", CENTER, 65);
myGLCD.setFont(SmallFont); //Ορισμός μικρής γραμματοσειράς
myGLCD.print("By:",RIGHT , 200);
myGLCD.print("Giorgos Goniotakis & Nikos Papazian",RIGHT , 220);

}
}

//Λειτουργία απενεργοποίησης τηλεόρασης
void televisionoff(){
  myGLCD.clrScr(); //Καθαρισμός περιεχομένου οθόνης
  digitalWrite(42,LOW); //Απενεργοποίηση οπίσθιου φωτισμού
  tvstatus = false;
}

//Λειτουργία αποστολής κατάστασης πλήκτρων αφής
void sendButtonState(){

  //Αν έχει πατηθεί το πλήκτρο αφής στείλε αίτηση με αριθμοί 5,2,2
  if (digitalRead(16)==HIGH){
    if (inputByte[0]== 5 && inputByte[1]==2 && inputByte[2]==2 ){
      Serial.print("1");
    }
  }
  else if (digitalRead(17)==HIGH){
    if (inputByte[0]== 5 && inputByte[1]==2 && inputByte[2]==3 ){
      Serial.print("1");
    }
  }
  else if (digitalRead(18)==HIGH){
    if (inputByte[0]== 5 && inputByte[1]==2 && inputByte[2]==6 ){
      Serial.print("1");
    }
  }
  else if (digitalRead(19)==HIGH){
    if (inputByte[0]== 5 && inputByte[1]==2 && inputByte[2]==9 ){
      Serial.print("1");
    }
  }
  else if (digitalRead(20)==HIGH){
    if (inputByte[0]== 5 && inputByte[1]==2 && inputByte[2]==12 ){
      Serial.print("1");
    }
  }
  else if (digitalRead(21)==HIGH){
    if (inputByte[0]== 5 && inputByte[1]==2 && inputByte[2]==44 ){
      Serial.print("1");
    }
  }
}
}
```

## ΠΑΡΑΡΤΗΜΑ Β΄

Κώδικας πλακέτας Arduino αισθητηρίων

```

#include <SPI.h> //Βιβλιοθήκη Τεχνολογίας SPI
#include <Wire.h> //Βιβλιοθήκη Τεχνολογίας I2C
#include <Adafruit_GFX.h> //Βιβλιοθήκη οθονών OLED
#include <Adafruit_SSD1306.h> //Βιβλιοθήκη οθονών OLED
#include <dht.h> //Βιβλιοθήκη αισθητηρίων DHT11
#include <LiquidCrystal.h> //Βιβλιοθήκη οθόνης LCD 2X16
#include <Keypad.h> //Βιβλιοθήκη Πληκτρολογίου 4X3
#include <Stepper.h> //Βιβλιοθήκη Βηματικού Μοτέρ

#define DHTPINLIV 22 //Ακίδα σύνδεσης αισθητηρίου θερμοκρασίας-υγρασίας
σαλονιού
#define DHTPINBAT 23 //Ακίδα σύνδεσης αισθητηρίου θερμοκρασίας-υγρασίας
τουαλέτας
#define DHTPINBED 24 //Ακίδα σύνδεσης αισθητηρίου θερμοκρασίας-υγρασίας
υπνοδωματίου
#define DHTPINKIT 25 //Ακίδα σύνδεσης αισθητηρίου θερμοκρασίας-υγρασίας
κουζίνας
#define DHTPINOUT 26 //Ακίδα σύνδεσης αισθητηρίου θερμοκρασίας-υγρασίας
εξωτερικού χώρου
#define DHTPINROO 27 //Ακίδα σύνδεσης αισθητηρίου θερμοκρασίας θερμοσίφωνα

//Μεταβλητές Συστήματος
Ασφαλείας////////////////////////////////////
bool AlarmStatus=false; //Συναγερμός απενεργοποιημένος
int intruder=0; //Μηδενική Παραβίαση
int datacount=0; //Αριθμός ψηφίων που έχει πληκτρολογήσει ο χρήστης
char Code[5]={'7','7','7','7'}; //Σωστός κωδικός συναγερμού 7777
char tempCode[5]; //Μεταβλητή προσωρινής αποθήκευσης πληκτρολόγησης
const byte ROWS = 4; //4 γραμμές πληκτρολόγιο
const byte COLS = 3; //3 στήλες πληκτρολόγιο
byte rowPins[ROWS] = {8,9,10,11}; //Ακίδες σύνδεσης των τεσσάρων γραμμών
του πληκτρολογίου
byte colPins[COLS] = {12,14,15}; //Ακίδες σύνδεσης των τριών στηλών του
πληκτρολογίου

//Σχεδίαση διαθέσιμων ψηφίων του πληκτρολογίου
char hexaKeys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}};
};
////////////////////////////////////
////////////////////////////////////

//Γενικές
Μεταβλητές////////////////////////////////////
int Thermostat1 = 18; //Μεταβλητή θερμοστάτη πρώτου καλοριφέρ
int Thermostat2 = 18; //Μεταβλητή θερμοστάτη δεύτερου καλοριφέρ
int Thermostat3 = 30; //Μεταβλητή θερμοστάτη πρώτου air condition

```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
int Thermostat4 = 30; //Μεταβλητή θερμοστάτη δεύτερου air condition
int i; //Μεταβλητή εκάστοτε αναλογικού αισθητηρίου προς ανάγνωση
byte PinA[] = {A0,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10}; //Ακίδες που έχουν
συνδεδεμένες αναλογικά αισθητήρια
int TempHum[12]; //Πίνακας αποθήκευσης αποτελεσμάτων θερμοκρασίας και
υγρασίας
int flag = 0; //Μεταβλητή μέτρησης σελίδας για τις οθόνες OLED
unsigned long previousMillis = 0; //Μεταβλητή αποθήκευσης χρόνου
const long interval = 2000; //Χρονοκαθυστερήση πριν την αλλαγή σελίδας στις
οθόνες OLED
byte inputByte[]={0,0,0}; //Πίνακας αποθήκευσης μηνυμάτων σειριακής
επικοινωνίας
////////////////////////////////////
////////////////////////////////////

//Μεταβλητές Συστήματος Ρύθμισης
Τέντας////////////////////////////////////
int angle=0; //Προσωρινή Μεταβλητή Υπολογισμού
int lastpos=-6000; //Μεταβλητή αποθήκευσης τελευταίας θέσης
int pos1=0; //Θέση τέντας σε περίπτωση ανέμου
int pos2=-6000; //Θέση τέντας σε περίπτωση βροχής
int pos3=-5000; //Θέση τέντας σε περίπτωση υψηλής φωτεινότητας
int pos4=-3000; //Θέση τέντας σε περίπτωση μέτριας φωτεινότητας
int pos5=-1000; //Θέση τέντας σε περίπτωση χαμηλής φωτεινότητας
int DueToWind=0; //Μεταβλητή αποθήκευσης μετρήσεων ανέμου
int DueToRain=0; //Μεταβλητή αποθήκευσης μετρήσεων βροχής
int DueToLight1=0; //Μεταβλητή αποθήκευσης μετρήσεων υψηλής φωτεινότητας
int DueToLight2=0; //Μεταβλητή αποθήκευσης μετρήσεων μεσαίας φωτεινότητας
int DueToLight3=0; //Μεταβλητή αποθήκευσης μετρήσεων χαμηλής φωτεινότητας
int DueNotToWind=0; //Μεταβλητή αποθήκευσης μετρήσεων μη ύπαρξης ανέμου
int DueNotToRain=0; //Μεταβλητή αποθήκευσης μετρήσεων μη ύπαρξης βροχής
int TentIdle=0; //Τέντα απασχολημένη λόγω ανέμου
int TentIdle2=0; //Τέντα απασχολημένη λόγω βροχής
unsigned long previousMillis2 = 0; //Μεταβλητή αποθήκευσης χρονικής στιγμής
const long interval2 = 1000; //Έλεγχος καιρικών συνθηκών κάθε 1
δευτερόλεπτο
////////////////////////////////////
////////////////////////////////////

Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS,
COLS); //Εντολή δήλωσης οντότητας πληκτρολογίου
LiquidCrystal lcd(7,6,5,4,3,2); //Εντολή δήλωσης οντότητας οθόνης LCD
Adafruit_SSD1306 display(5); //Εντολή δήλωσης οντότητας OLED
dht DHT; //Εντολή δήλωσης οντότητας αισθητηρίου θερμοκρασίας-υγρασίας DHT11

//Λειτουργία ανταλλαγής σειριακών μηνυμάτων με το πρόγραμμα C#
bool sindesi(){
  boolean sin = false; //Αρχική κατάσταση ΜΗ ΣΥΝΔΕΔΕΜΕΝΟ

  //Εφόσον υπάρχουν διαθέσιμες πληροφορίες στο κανάλι επικοινωνίας διάβασέ
  τις
  if (Serial.available() > 2)
  {
    inputByte[0] = Serial.read();
    delay(10);
    inputByte[1] = Serial.read();
  }
}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    delay(10);
    inputByte[2] = Serial.read();
    delay(10);
}

//Μήνυμα επίτευξης σύνδεσης του Arduino αισθητηρίων
if(inputByte[0] == 11 && inputByte[1] == 21 && inputByte[2] == 31)
{
    Serial.print("Device 1");
    return(true);
}
}

void setup() {

Serial.begin(9600); //Έναρξη σειριακής επικοινωνίας
Measure(); //Κλήση λειτουργίας μέτρησης θερμοκρασίας και υγρασίας
display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //Έναρξη οθόνης OLED
display.clearDisplay(); //Καθάρισμα περιεχομένου της οθόνης OLED
UpdateScreen(); //Κλήση λειτουργίας ανανέωσης οθονών

//Διαχείριση ακίδων σύνδεσης του συστήματος ποτίσματος
pinMode(34,OUTPUT);
pinMode(35,OUTPUT);
digitalWrite(34,LOW);
digitalWrite(35,LOW);
////////////////////////////////////

//Διαχείριση ακίδων σύνδεσης του συστήματος Aircondition
pinMode(30,OUTPUT);
pinMode(32,OUTPUT);
digitalWrite(30,LOW);
digitalWrite(32,LOW);
////////////////////////////////////

//Διαχείριση ακίδων σύνδεσης του συστήματος θέρμανσης
pinMode(31,OUTPUT);
pinMode(33,OUTPUT);
digitalWrite(31,LOW);
digitalWrite(33,LOW);
////////////////////////////////////

//Διαχείριση ακίδων σύνδεσης του συστήματος Συναγερμού
pinMode(40,OUTPUT);
pinMode(13,OUTPUT);
digitalWrite(40,LOW);
digitalWrite(13,LOW);
lcd.begin(16, 2);
lcd.setCursor(0, 0);
lcd.print("Alarm System");
lcd.setCursor(0, 1);
lcd.print("Press * to enable");
////////////////////////////////////
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
//Διαχείριση συστήματος κουτιού του πίνακα ελέγχου
pinMode(42,INPUT);
pinMode(45,OUTPUT);
pinMode(43,OUTPUT);
pinMode(44,OUTPUT);
AwningBox(); //Κλήση λειτουργίας σεταρίσματος κουτιού του πίνακα ελέγχου
πριν την έναρξη εκτέλεσης του προγράμματος
////////////////////////////////////

}

//Κεντρική λειτουργία προγράμματος
void loop() {

  //Αν η πλακέτα έχει συνδεθεί με το πρόγραμμα
  if(sindesi() == true)
  {
    Measure();

    //Αν είναι ενεργοποιημένος ο έλεγχος της τέντας από το κουτί ελέγχου
    if (digitalRead(44) == LOW){
      unsigned long currentMillis2 = millis();
      if(currentMillis2 - previousMillis2 >= interval2 && digitalRead(43)==LOW)
      {
        previousMillis2 = currentMillis2;
        Awning();
      }
    }

    if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 11 )    {

      Serial.print(TempHum[1]);

    }
    else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 12 )
    {

      Serial.print(TempHum[2]);
    }
    else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 13 )
    {

      Serial.print(TempHum[3]);
    }
    else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 14 )
    {

      Serial.print(TempHum[4]);
    }
    else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 15 )
    {

      Serial.print(TempHum[5]);
    }
    else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 16 )
    {
```

```

        Serial.print(TempHum[6]);
    }
    else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 17 )
    {
        Serial.print(TempHum[7]);
    }
    else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 18 )
    {
        Serial.print(TempHum[8]);
    }
    else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 19 )
    {
        Serial.print(TempHum[9]);
    }
    else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 20 )
    {
        Serial.print(TempHum[10]);
    }
    else if(inputByte[0] == 2 && inputByte[1] == 1 && inputByte[2] == 21 )
    {
        Serial.print(TempHum[11]);
    }
    //Σε περίπτωση φωτιάς στείλε την απάντηση 1 στην αίτηση 7,1,96
    else if (inputByte[0]==7 && inputByte[1]==1 && inputByte[2]==96){
        if (analogRead(A1) > 600 || analogRead(A9) >600 ||
analogRead(A10) >240){
            Serial.print("1");
        }
        else{
            Serial.print("0");
        }
    }
    //Σε περίπτωση ενεργοποίησης του κουμπιού έκτακτης ανάγκης στείλε την
    απάντηση 1 στην αίτηση 7,1,97
    else if (inputByte[0]==7 && inputByte[1]==1 && inputByte[2]==97){
        if (digitalRead(44)==HIGH){
            Serial.print("1");
        }
        else if(digitalRead(44) == LOW){
            Serial.print("0");
        }
    }
}

```



## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    }
    //Σε περίπτωση παραβίασης του χώρου την ώρα που ήταν ενεργοποιημένος ο
    συναγερμός στείλε την απάντηση 1 στην αίτηση 7,1,98
    else if (inputByte[0]==7 && inputByte[1] == 1 && inputByte[2]==98){
        if(intruder==1){

            Serial.print("1");

        }
        else if(intruder==0){

            Serial.print("0");

        }
    }
    //Αλλιώς αν η αίτηση έχει την μορφή 2,1,X όπου X = 0-10 στείλε την
    αναλογική τιμή αισθητηρίου
    else if(inputByte[0] == 2 && inputByte[1] == 1 ){

        i=inputByte[2];
        Serial.print(analogRead(PinA[i]));

    }
    //Αν έχει άλλη μορφή το μήνυμα μην κάνεις τίποτα
    else{

    }

}
//Αν δεν έχει επιτευχθεί σύνδεση μην κάνεις τίποτα
else{

}

UpdateScreen(); //Κλήση λειτουργίας ανανέωσης οθονών
NeedWatering(); //Κλήση λειτουργίας ελέγχου ποτίσματος
Aircondition(); //Κλήση λειτουργίας ελέγχου κλιματισμού
Radiator(); //Κλήση λειτουργίας ελέγχου θέρμανσης
AlarmSystem(); //Κλήση λειτουργίας ελέγχου συστήματος συναγερμού

inputByte[0] = 0;
inputByte[1] = 0;
inputByte[2] = 0;

}

//Λειτουργία καταγραφής τιμών αισθητηρίων θερμοκρασίας - υγρασίας
void Measure(){

    TempHum[1] = DHT.read11(DHTPINLIV); //Σύνδεση με το αισθητήριο τύπου
DHT11
    TempHum[1] = (int)DHT.temperature; //Διάβασμα τιμής θερμοκρασίας
αισθητηρίου
    TempHum[2] = (int)DHT.humidity; //Διάβασμα τιμής υγρασίας αισθητηρίου
    TempHum[3] = DHT.read11(DHTPINBAT);
    TempHum[3] = (int)DHT.temperature;
    TempHum[4] = (int)DHT.humidity;
    TempHum[5] = DHT.read11(DHTPINBED);
    TempHum[5] = (int)DHT.temperature;
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
TempHum[6] = (int)DHT.humidity;
TempHum[7] = DHT.read11(DHTPINKIT);
TempHum[7] = (int)DHT.temperature;
TempHum[8] = (int)DHT.humidity;
TempHum[9] = DHT.read11(DHTPINOUT);
TempHum[9] = (int)DHT.temperature;
TempHum[10] = (int)DHT.humidity;
TempHum[11] = DHT.read11(DHTPINROO);
TempHum[11] = (int)DHT.temperature;
TempHum[12] = (int)DHT.humidity;

}

//Λειτουργία ενημέρωσης οθονών OLED
void UpdateScreen(){

//Αν έχουν περάσει 2 δευτερόλεπτα
unsigned long currentMillis = millis();
if(currentMillis - previousMillis >= interval) {

    previousMillis = currentMillis;

    //Αν βρίσκεσαι στην σελίδα 1 ενημέρωσε για τις τιμές των αισθητηρίων
    θερμοκρασίας του σαλονιού, του υπνοδωματίου και της τουαλέτας
    if (flag == 0)
    {
        display.clearDisplay();
        display.setTextSize(2);
        display.setTextColor(WHITE);
        display.setCursor(0,0);
        display.print("LIVIN: ");
        display.print(TempHum[1]);
        display.print("C");
        display.setCursor(0,25);
        display.print("BEDR: ");
        display.print(TempHum[5]);
        display.print("C");
        display.setCursor(0,50);
        display.print("BATH: ");
        display.print(TempHum[3]);
        display.print("C");
        display.display();

        flag=1;
    }

    //Αν βρίσκεσαι στην σελίδα 2 ενημέρωσε για τις τιμές των αισθητηρίων
    θερμοκρασίας της κουζίνας και του εξωτερικού χώρου
    else if (flag== 1){
        display.clearDisplay();
        display.setTextSize(2);
        display.setTextColor(WHITE);
        display.setCursor(0,0);
        display.print("KITCH: ");
        display.print(TempHum[7]);
        display.print("C");
        display.setCursor(0,25);
        display.print("EXTER: ");
        display.print(TempHum[9]);
        display.print("C");
        display.setCursor(0,50);
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    display.display();

    flag=0;
}

}

//Λειτουργία ελέγχου ποτίσματος
void NeedWatering(){

    //Ενεργοποίησε το σύστημα ποτίσματος ανάλογα με την τιμή του αισθητηρίου
του φυτού νούμερο 1
    if (analogRead(A3) > 500){

        digitalWrite(35,HIGH);

    }
    else{

        digitalWrite(35,LOW);

    }

    //Ενεργοποίησε το σύστημα ποτίσματος ανάλογα με την τιμή του αισθητηρίου
του φυτού νούμερο 2
    if (analogRead(A4) > 500){

        digitalWrite(34,HIGH);

    }
    else{

        digitalWrite(34,LOW);

    }
}

//Λειτουργία ελέγχου συστήματος κλιματισμού
void Aircondition(){

    //Αν η θερμοκρασία έχει φτάσει τους 30 βαθμούς ενεργοποίησε τον
κλιματισμού μέχρι η θερμοκρασία να φτάσει τους 28 βαθμούς
    if (TempHum[1] > Thermostat3){

        digitalWrite(30,HIGH);
        Thermostat3 = 28;

    }
    else{

        Thermostat3 = 30;
        digitalWrite(30,LOW);

    }

    //Αν η θερμοκρασία έχει φτάσει τους 30 βαθμούς ενεργοποίησε τον
κλιματισμού μέχρι η θερμοκρασία να φτάσει τους 28 βαθμούς
    if (TempHum[5] > Thermostat4){
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    digitalWrite(32,HIGH);
    Thermostat4 = 28;

}
else{

    Thermostat4 = 30;
    digitalWrite(32,LOW);

}
}

//Λειτουργία ελέγχου συστήματος θέρμανσης
void Radiator(){

    //Αν η θερμοκρασία έχει φτάσει τους 18 βαθμούς ενεργοποίησε την θέρμανση
    μέχρι η θερμοκρασία να φτάσει τους 20 βαθμούς
    if (TempHum[1] < Thermostat1){

        digitalWrite(31,HIGH);
        Thermostat1 = 20;

    }
    else{

        Thermostat1 = 18;
        digitalWrite(31,LOW);

    }

    //Αν η θερμοκρασία έχει φτάσει τους 18 βαθμούς ενεργοποίησε την θέρμανση
    μέχρι η θερμοκρασία να φτάσει τους 20 βαθμούς
    if (TempHum[5] < Thermostat2){

        digitalWrite(33,HIGH);
        Thermostat2=20;

    }
    else{

        Thermostat2=20;
        digitalWrite(33,LOW);

    }
}

//Λειτουργία συστήματος συναγερμού
void AlarmSystem(){

    char customKey = customKeypad.getKey(); //Σε περίπτωση που πατηθεί
    πλήκτρο αποθήκευσέ το

    //Αν πατηθεί το * ενεργοποίησε τον συναγερμό
    if (customKey == '*' && AlarmStatus == false){

        enableAlarm();

    }

}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    }
    //Αν είναι ενεργοποιημένος κάνε έλεγχο σωστού κωδικού ή παραβίασης
χώρου
    else if (AlarmStatus==true){

        checkAlarm();
        checkEntry();

    }
    //Αλλιώς μην κάνεις κάτι
    else{

    }

}

//Λειτουργία ελέγχου παραβίασης χώρου σε περίπτωση που το σύστημα
 συναγερμού είναι ενεργοποιημένο
void checkAlarm() {

    //Αν γίνει παραβίαση του χώρου ενεργοποίησε τον συναγερμό
    if (analogRead(A6) > 500 || analogRead(A8) > 500 && intruder==0){

        digitalWrite(40,HIGH);
        intruder=1;

    }

    //Αν είναι ήδη ενεργοποιημένος άφησέ τον ενεργοποιημένο
    else if (intruder==1){
        digitalWrite(40,HIGH);
    }

    //Αν δεν υπάρχει κίνηση άφησε τον συναγερμό απενεργοποιημένο
    else{
        digitalWrite(40,LOW);
    }

}

//Λειτουργία ελέγχου κωδικού
void checkPassword() {

    //Αν ο κωδικός είναι σωστός
    if(!strcmp(tempCode,Code)){

        AlarmStatus=false;
        intruder=0;
        datacount=0;
        digitalWrite(40,LOW);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Disarmed");
        delay(1000);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Alarm System");
        lcd.setCursor(0, 1);
        lcd.print("Press * to enable");
    }

}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
int rep;

for(rep=0;rep<4;rep++){
    tempCode[rep]='0';
}
}

//Αν ο κωδικός είναι λαθός, προσπάθεια ξανά
else{

    datacount=0;
    int rep;

    for(rep=0;rep<4;rep++){

        tempCode[rep]='0';

    }

    lcd.setCursor(0,1);
    lcd.print("Wrong Password");
    delay(1000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Enter Password...");
    lcd.setCursor(0, 1);
    lcd.blink();

}
}

//Λειτουργία ενεργοποίησης συστήματος συναγερμού
void enableAlarm(){

AlarmStatus=true;

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Enabling Alarm...");
delay(10000); //Περίμενε 10 δευτερόλεπτα
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Enter Password...");
lcd.setCursor(0, 1);
lcd.blink();

}

void checkEntry(){

char customKey = customKeypad.getKey();

//Αν πατηθεί πλήκτρο αποθήκευσε το
if(customKey){

    if (customKey!= '*' && customKey!='#'){

        lcd.print("*");
        tempCode[datacount] =customKey;

    }

}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    datacount++;

}

//Αν ο χρήστης εισάγει 4 χαρακτήρες έλεγξ τον κωδικό
if (datacount ==4){

    checkPassword();

}

//Αν πατήσει # καθαρισμός των ψηφίων που έχει εισάγει
else if (customKey == '#' && datacount<4){

    datacount=0;
    int rep;

    for (rep=0;rep<4;rep++){

        tempCode[rep]='0';

    }

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Enter Password...");
    lcd.setCursor(0, 1);
    lcd.blink();

}

}

}

//Λειτουργία ρύθμισης της τέντας από το κουτί που υπάρχει στον πίνακα
ελέγχου
void AwningBox(){

const int stepsPerRevolution = 200; //Βήματα ανά επανάληψη
Stepper myStepper(200,36,37,38,39); //Δήλωση βηματικού κινητήρα
myStepper.setSpeed(60); //Ορισμός ταχύτητας
int SetUp = 0; //Μεταβλητή αποθήκευσης κατάστασης διακόπτη Push Button
digitalWrite(45,HIGH); //Ενεργοποίηση LED ένδειξης κατάστασης

//Όσο δεν έχει πατηθεί το Push Button ρύθμιση της τέντας μέσω του
ποτενσιόμετρου
while (SetUp != 1){

    digitalWrite(52,HIGH);
    int Potentiometer = analogRead(A11);
    int PushButton = digitalRead(42);

    if (SetUp == 0){
        if (PushButton == HIGH){

            SetUp++;
            digitalWrite(45,LOW);
            delay(500);


```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
        digitalWrite(45,HIGH);
        delay(1000);
    }
    else if (Potentiometer > 820 && Potentiometer < 1023){

        myStepper.step(200);

    }
    else if (Potentiometer > 0 && Potentiometer <268){

        myStepper.step(-200);

    }
    else{
        //POTENTIOMETER MEDIUM
    }
}
}
digitalWrite(45,LOW);
}
```

```
void Awning() {
```

```
    Stepper myStepper(200,36,37,38,39); //Δήλωση οντότητας βηματικού κινητήρα
    myStepper.setSpeed(60);
    int WindSensor = analogRead(A5); //Αναλογική μέτρηση από τον αισθητήρα
    ανέμου
    int RainSensor = analogRead(A7); //Αναλογική μέτρηση από τον αισθητήρα
    βροχής
    int LightSensor = analogRead(A2); //Αναλογική μέτρηση από τον αισθητήρα
    φωτεινότητας
```

```
    //Αν έχει άνεμο
    if (WindSensor >= 70 && TentIdle!=1){
```

```
        DueToLight1=0;
        DueToLight2=0;
        DueToLight3=0;
        DueToWind++;
        DueNotToWind=0;
```

```
    }
```

```
    //Αν δεν έχει άνεμο
    else if(WindSensor<70 && TentIdle==1){
```

```
        DueNotToWind++;
        DueToWind=0;
```

```
    }
```

```
    //Αν πάρουμε 10 συνεχόμενες μετρήσεις ότι είχαμε άνεμο μετακινούμε την
    τέντα
```

```
    if (DueToWind >=10 && TentIdle==0){
```

```
        DueToLight1=0;
```



## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    DueToLight2=0;
    DueToLight3=0;
    TentIdle=1;
    angle=pos1-lastpos;
    lastpos=pos1;
    myStepper.step(angle);

}

//Αν πάρουμε 10 συνεχόμενες μετρήσεις ότι δεν έχει άνεμο αφήνουμε τον
έλεγχο της τέντιας στα υπόλοιπα αισθητήρια
else if (DueNotToWind ==10 && TentIdle==1){

    DueToLight1=0;
    DueToLight2=0;
    DueToLight3=0;
    TentIdle=0;

}

//Αν έχει βροχή και όχι άνεμο
if (RainSensor < 800 && TentIdle==0){

    DueToLight1=0;
    DueToLight2=0;
    DueToLight3=0;
    DueToRain++;
    DueNotToRain=0;

}

//Αν δεν έχει βροχή
else{

    DueToRain=0;
    DueNotToRain++;

}

}

//Αν πάρουμε 10 συνεχόμενες μετρήσεις ότι είχαμε βροχή και δεν είχαμε
άνεμο μετακινούμε την τέντια
if (DueToRain >= 10 && TentIdle!=1){

    DueToLight1=0;
    DueToLight2=0;
    DueToLight3=0;
    angle=pos2-lastpos;
    lastpos=pos2;
    TentIdle2=1;
    myStepper.step(angle);

}

//Αν πάρουμε 10 συνεχόμενες μετρήσεις ότι δεν έχει βροχή αφήνουμε τον
έλεγχο της τέντιας στα υπόλοιπα αισθητήρια
else if (DueNotToRain == 10 && TentIdle2==1){

    DueToLight1=0;
    DueToLight2=0;
    DueToLight3=0;
    TentIdle2=0;

}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
}

//Αν έχουμε υψηλή φωτεινότητα και δεν έχουμε άνεμο ή βροχή
if (LightSensor > 0 && LightSensor < 300 && TentIdle!=1 && TentIdle2!=1){

    DueToLight1++;
    DueToLight2=0;
    DueToLight3=0;

}

//Αν έχουμε μεσαία φωτεινότητα και δεν έχουμε άνεμο ή βροχή
else if(LightSensor >= 300 && LightSensor < 600 && TentIdle!=1 &&
TentIdle2 !=1){

    DueToLight1=0;
    DueToLight2++;
    DueToLight3=0;

}

//Αν έχουμε χαμηλή φωτεινότητα και δεν έχουμε άνεμο ή βροχή
else if (LightSensor >= 600 && LightSensor <= 1024 && TentIdle!=1 &&
TentIdle2!=1){

    DueToLight1=0;
    DueToLight2=0;
    DueToLight3++;

}

//Αν δεν συμβαίνει τίποτα από όλα αυτά μην κάνεις τίποτα
else{
}

//Αν έχουμε πάρει 10 μετρήσεις υψηλής φωτεινότητας και δεν υπάρχει άνεμος
και βροχή μετακίνησε την τέντα
if (DueToLight1>=10){

    angle=pos3-lastpos;
    lastpos=pos3;
    myStepper.step(angle);

}

//Αν έχουμε πάρει 10 μετρήσεις μεσαίας φωτεινότητας και δεν υπάρχει
άνεμος και βροχή μετακίνησε την τέντα
else if (DueToLight2>=10){

    angle=pos4-lastpos;
    lastpos=pos4;
    myStepper.step(angle);

}

//Αν έχουμε πάρει 10 μετρήσεις χαμηλής φωτεινότητας και δεν υπάρχει
άνεμος και βροχή μετακίνησε την τέντα
else if (DueToLight3>=10){

    angle=pos5-lastpos;
    lastpos=pos5;
    myStepper.step(angle);

}
```

```
    }  
    //Αλλιώς συνέχισε να μετράς  
    else{  
    }  
}
```

## ΠΑΡΑΡΤΗΜΑ Γ΄

### Κώδικας Arduino συσκευών πραγματικής εφαρμογής

```
int LostConnection = 0; //Μεταβλητή αποθήκευσης κατάστασης σύνδεσης της
πλακέτας με το πρόγραμμα
byte inputByte[] = {0,0,0}; //Μεταβλητή αποθήκευσης μηνυμάτων σειριακής
επικοινωνίας
```

```
//Λειτουργία σύνδεσης πλακέτας με το πρόγραμμα C#
```

```
bool sindesi() {
```

```
    boolean sin = false;
```

```
    if (Serial.available())
```

```
    {
```

```
        inputByte[0] = Serial.read();
```

```
        delay(10);
```

```
        inputByte[1] = Serial.read();
```

```
        delay(10);
```

```
        inputByte[2] = Serial.read();
```

```
        delay(10);
```

```
    }
```

```
    if(inputByte[0] == 10 && inputByte[1] == 20 && inputByte[2] == 30)
```

```
    {
```

```
        Serial.print("Device 0");
```

```
        return(true);
```

```
    }
```

```
}
```

```
void setup() {
```

```
    Serial.begin(9600); //Έναρξη σειριακής επικοινωνίας
```

```
    //Ορισμός λειτουργίας ψηφιακών ακίδων ως εξόδους
```

```
    for (int i=2; i<54; i++)
```

```
    {
```

```
        pinMode(i,OUTPUT);
```

```
    }
```

```
}
```

```
//Κεντρική λειτουργία προγράμματος
```

```
void loop() {
```

```
    //Αν επιτευχθεί σύνδεση με το πρόγραμμα
```

```
    if(sindesi() == true)
```

```
    {
```

```
if (LostConnection == 0)
{
LostConnection=1;
}

if(inputByte[0] == 5)
{

    if (inputByte[2] == 1){

        //Ενεργοποίηση της συσκευής που βρίσκεται στην ακίδα X όταν η
        αίτηση έχει την μορφή 5,X,1
        digitalWrite(inputByte[1], HIGH);

    }

    else if (inputByte[2]==0){

        //Απενεργοποίηση της συσκευής που βρίσκεται στην ακίδα X όταν η
        αίτηση έχει την μορφή 5,X,0
        digitalWrite(inputByte[1], LOW);

    }

}

inputByte[0] = 0;
inputByte[1] = 0;
inputByte[2] = 0;
}

//Σε περίπτωση διακοπής Internet ή κάποιου άλλου προβλήματος ενεργοποίησε
όλες τις συσκευές έτσι ώστε να μπορεί να τις χειριστεί ο χρήστης
else if(LostConnection==0){

    for (int i=2; i<54; i++)
    {
        digitalWrite(i,HIGH);
    }
}
}
```

## ΠΑΡΑΡΤΗΜΑ Δ΄

Κώδικας εφαρμογής διαμεσολάβησης

## Αρχείο Communicator.cs

```

using System;
using System.Collections.Generic;
using System.Text;
using System.IO.Ports;
using System.Threading;
using System.Data;

namespace FullHouseWebControl
{
    public class communicator
    {
        public string port = ""; //Ονομασία πόρτας επικοινωνίας USB
        public SerialPort currentPort; //Αριθμός πόρτας επικοινωνίας USB
        public bool portConnection = false; //Κατάσταση σύνδεσης πόρτας USB
        εξ ορισμού

        /*
            Λειτουργία επίτευξης σύνδεσης με τις πλακέτες Arduino. Αυτή η
            λειτουργία δέχεται ως όρισμα την ταχύτητα επικοινωνίας, μία συμβολοσειρά και
            τρεις αριθμούς
            και διατρέπει όλες τις διαθέσιμες πόρτες επικοινωνίας USB
            προσπαθώντας να δει αν κάποια συσκευή αναπαράγει αυτό το μήνυμα. Εφόσον την
            βρει επιστρέφει
            το όνομα και τον αριθμό της πόρτας επικοινωνίας της συσκευής.
        */
        public Boolean connect(int baud, string recognizeText, byte
paramone, byte paramtwo, byte paramthree)
        {
            try
            {
                byte[] buffer = new byte[3]; //Πίνακας αποθήκευσης των
τριών αριθμών
                buffer[0] = Convert.ToByte(paramone); //Μετατροπή του
πρώτου αριθμού σε byte
                buffer[1] = Convert.ToByte(paramtwo); //Μετατροπή του
δεύτερου αριθμού σε byte
                buffer[2] = Convert.ToByte(paramthree); //Μετατροπή του
τρίτου αριθμού σε byte

                int intReturnASCII = 0; //Μεταβλητή μέτρησης αριθμού ψηφίων
που θα διαβαστούν
                string[] ports = SerialPort.GetPortNames(); //Αποθήκευση
διαθέσιμων ονομάτων συσκευών USB σε έναν πίνακα
                foreach (string newport in ports) //Για κάθε μία πόρτα στην
λίστα
                {
                    currentPort = new SerialPort(newport, baud); //Επίτευξη
σύνδεσης στην πόρτα αυτή με ταχύτητα 9600
                    currentPort.Open(); //Άνοιγμα θύρας

```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
currentPort.Write(buffer, 0, 3); //Εγγραφή μηνύματος
Thread.Sleep(200); //Αναμονή 200ms
int count = currentPort.BytesToRead; //Ανάγνωση
mηνύματος που προέρχεται από την συσκευή USB
string returnMessage = ""; //Αποθήκευση σε πίνακα
while (count > 0) //Αν υπάρχει περιεχόμενο στο μήνυμα
{
    intReturnASCII = currentPort.ReadByte();
//Αποθήκευση σε μεταβλητή του περιεχομένου του μηνύματος
returnMessage = returnMessage +
Convert.ToChar(intReturnASCII); //Μετατροπή σε ASCII χαρακτήρες και προσθήκη
στο τέλος της μεταβλητής
    count--;
}

currentPort.Close(); //Κλείσιμο θύρας επικοινωνίας
port = newport; //Συνέχεια με την επόμενη διαθέσιμη
πόρτα της λίστας
if (returnMessage.Contains(recognizeText)) //Αν το
περιεχόμενο που δώσαμε και το μήνυμα της θύρας ήταν ίδια επέστρεψε αλήθεια
{
    return true;
}
return false; //Αλλιώς επέστρεψε λάθος
}
finally { }
}

//Λειτουργία επικοινωνίας μεταξύ θυρών στις οποίες έχει διασφαλισθεί
σύνδεση σύμφωνα με την μέθοδο που βρίσκεται παραπάνω
public string message(byte paramone, byte paramtwo, byte
paramthree)
{
    byte[] buffer = new byte[3]; //Πίνακας αποθήκευσης των τριών
αριθμών
    buffer[0] = Convert.ToByte(paramone); //Μετατροπή του πρώτου
αριθμού σε byte
    buffer[1] = Convert.ToByte(paramtwo); //Μετατροπή του δεύτερου
αριθμού σε byte
    buffer[2] = Convert.ToByte(paramthree); //Μετατροπή του τρίτου
αριθμού σε byte
    currentPort.Open(); //Άνοιγμα θύρας
    currentPort.Write(buffer, 0, 3); //Εγγραφή μηνύματος
    int intReturnASCII = 0; //Μεταβλητή μέτρησης αριθμού ψηφίων που
θα διαβαστούν
    Thread.Sleep(200); //Αναμονή 200ms
    int count = currentPort.BytesToRead; //Ανάγνωση μηνύματος που
προέρχεται από την συσκευή USB
    string returnMessage = ""; //Αποθήκευση σε πίνακα
    while (count > 0) //Αν υπάρχει περιεχόμενο στο μήνυμα
    {
        intReturnASCII = currentPort.ReadByte(); //Αποθήκευση σε
μεταβλητή του περιεχομένου του μηνύματος
        returnMessage = returnMessage +
Convert.ToChar(intReturnASCII); //Μετατροπή σε ASCII χαρακτήρες και προσθήκη
στο τέλος της μεταβλητής
        count--;
    }
    currentPort.Close(); //Κλείσιμο θύρας επικοινωνίας
```

```
        return returnMessage; //Επιστροφή μηνύματος από συσκευή USB
    }
}
}
```

## Αρχείο Form1.cs

```
using System;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Collections.Generic;
using System.Windows.Forms;
using System.IO.Ports; //Βιβλιοθήκη για τις συσκευές USB
using MySql.Data.MySqlClient; //Βιβλιοθήκη της βάσεως δεδομένων
using System.Threading; //Βιβλιοθήκη παραλληλοποίησης
using System.Threading.Tasks; //Βιβλιοθήκη παραλληλοποίησης
using System.Linq;
using System.Diagnostics;

namespace FullHouseWebControl
{
    public partial class Form1 : Form
    {
        public Form1 ()
        {
            InitializeComponent();
            this.FormClosed += Form1_FormClosed; //Event σε περίπτωση που ο
            χρήστης κλείσει την εφαρμογή
            label5.Hide();
            ConnectArduino(); //Σύνδεση Arduino συσκευών
            ConnectArduino(); //Σύνδεση Arduino αισθητηρίων
        }

        public static string CustomerID = "1"; //Κωδικός πελάτη 1 εξ
        ορισμού
        public static Boolean ButtonPref = false; //Κατάσταση λειτουργίας
        ελέγχου διακοπών αφής
        int ArduinoID = 0; //Μεταβλητή μέτρησης του αριθμού των πλακετών
        που θα συνδεθούν
        int SurveillanceButtonStatus = 0; //Μεταβλητή κατάστασης κουμπιού
        παρακολούθησης χώρου
        Process myProcess;

        List<communicator> Communicators = new List<communicator>();

        private void ConnectArduino ()
        {
```



## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
communicator comport = new communicator(); //Αίτηση για νέα
σύζευξη επικοινωνίας του προγράμματος με πλακέτα Arduino

//Μήνυμα σύνδεσης πλακέτας συσκευών: Device 0, 10, 20,30
//Μήνυμα σύνδεσης πλακέτας αισθητήριων: Device 1, 11, 21, 31

if (comport.connect(9600, "Device " + ArduinoID,
(byte) (ArduinoID + 10), (byte) (ArduinoID + 20), (byte) (ArduinoID + 30)))
{
    if (ArduinoID == 0)
    {
        label1.Text = "Connection of Arduino 1 Successful -
Connected to " + comport.port;
    }
    else if (ArduinoID == 1)
    {
        label2.Text = "Connection of Arduino 2 Successful -
Connected to " + comport.port;
    }
    comport.portConnection = true;

    Communicators.Add(comport);
    ArduinoID++;
}

//Σε περίπτωση μη σύνδεσης μιας από τις δύο πλακέτες ενημέρωση
του χρήστη
else
{
    if (ArduinoID == 0)
    {
        label1.Text = "Arduino Number : 1 is not connected . .
. ";
        comport.portConnection = false;
        button1.Enabled = false;
        ArduinoID++;
    }
    else if (ArduinoID == 1)
    {
        label2.Text = "Arduino Number : 2 is not connected . .
. ";
        comport.portConnection = false;
        button1.Enabled = false;
    }
}

}

public void SetUp()
{
    ρυθμίσειςToolStripMenuItem.Enabled = false; //Αν έχει ξεκινήσει
η σύνδεση το ID πελάτη δεν μπορεί να αλλάξει πια

    //Αν επιτευχθεί η σύνδεση με την πλακέτα συσκευών ενημέρωση
όλων των συσκευών σύμφωνα με την βάση δεδομένων
    if (Communicators[0].portConnection == true)
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
    {
        string ConnectionString =
"datasource=83.212.120.81;port=3306;username=Arduino;password=ard1234";
        MySqlConnection myConn = new
MySqlConnection(ConnectionString);
        MySqlDataReader reader = null;
        try
        {
            myConn.Open();
            using (MySqlCommand cmd = new MySqlCommand("Select
Id,State from Full_House_Web_Control.Appliances WHERE
CustomerID="+CustomerID, myConn))
            {

                reader = cmd.ExecuteReader();
                while (reader.Read())
                {

                    int Value = int.Parse(reader.GetString(1));
                    byte ID = (byte)int.Parse(reader.GetString(0));

                    //Αποστολή αίτησης 5,αριθμός συσκευής,1 για
ενεργοποίηση συσκευής και 5,αριθμός συσκευής,0 για απενεργοποίηση συσκευής
                    if (Value == 1)
                    {

                        Communicators[0].message(5, ID, 1);

                    }
                    else
                    {

                        Communicators[0].message(5, ID, 0);

                    }

                }

            }
            myConn.Close();
        }
        catch { }
        finally
        {
        }
    }
}
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
//Λειτουργία ενημέρωσης Arduino συσκευών για τυχόν αλλαγές που
προέκυψαν τα τελευταία 10 δευτερόλεπτα στην βάση δεδομένων
public async void CheckStatus ()
{
    //Παραλληλοποιημένη λειτουργία
    await Task.Run (() =>
    {
        if (Communicators[0].portConnection == true)
        {
            string ConnectionString =
"datasource=83.212.120.81;port=3306;username=Arduino;password=ard1234";
            MySqlConnection myConn = new
MySqlConnection(ConnectionString);
            MySqlDataReader reader = null;
            try
            {
                myConn.Open ();
                using (MySqlCommand cmd = new MySqlCommand("Select
Id,State from Full_House_Web_Control.Appliances WHERE update_time > (NOW()
- INTERVAL 10 SECOND) AND CustomerID="+CustomerID, myConn))
                {

                    reader = cmd.ExecuteReader ();
                    while (reader.Read ())
                    {

                        int Value = int.Parse(reader.GetString(1));
                        byte ID =
(byte)int.Parse(reader.GetString(0));

                        if (Value == 1)
                        {

                            Communicators[0].message(5, ID, 1);

                        }
                        else
                        {

                            Communicators[0].message(5, ID, 0);

                        }

                    }

                }

                myConn.Close ();
            }
        }
    }
}
```

```

        catch {

        }

        finally
        {

        }
        if (ButtonPref == true)
        {
            CheckButtons(); //Κλήση λειτουργίας ελέγχου για
            τυχόν πάτημα κάποιο αισθητήρα αφής
        }
    }
});

    CheckStatus(); //Κλήση της ίδιας λειτουργίας ξανά μετά το τέλος

}

//Λειτουργία αίτησης των τιμών των αισθητηρίων από το Arduino με τα
//αισθητήρια και ενημέρωση της βάσεως δεδομένων με τις τιμές αυτές
public async void CheckSensors ()
{
    //Παραλληλοποιημένη λειτουργία
    await Task.Run(() =>
    {

        if (Communicators[1].portConnection == true)
        {
            try
            {
                CheckAlerts(); //Κλήση λειτουργίας ελέγχου
                συστήματος συναγερμού, πυρασφάλειας και πλήκτρου έκτακτης ανάγκης
                string[] sensors = new string[22];

                string connectionString =
                "datasource=83.212.120.81;port=3306;username=Arduino;password=ard1234";
                MySqlConnection myConn = new
                MySqlConnection(connectionString);

                myConn.Open();
                //Ζητάει από το Arduino με τα αισθητήρια τις τιμές
                τους με την μορφή αίτησης: 2,1,αριθμός αισθητηρίου και ενημερώνει την βάση
                δεδομένων
                for (int i = 1; i < 22; i++)
                {
                    byte k = (byte)i;

                    sensors[i] = Communicators[1].message(2, 1, k);
                    using (MySqlCommand cmd = new
                    MySqlCommand("UPDATE Full_House_Web_Control.Sensors SET
                    Value=@Value,update_time = NOW() WHERE Id=@ID AND CustomerID="+CustomerID,
                    myConn))
                    {

                        cmd.Parameters.AddWithValue("@ID", i);
                    }
                }
            }
            catch { }
        }
    });
}

```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
sensors[i]);

cmd.Parameters.AddWithValue("@Value",

cmd.ExecuteNonQuery();

}

}

myConn.Close();

}
catch
{
}
finally
{
}

}

});

CheckSensors(); //Κλήση της ίδιας λειτουργίας ξανά μετά το
τέλος εκτέλεσής της
}

//Λειτουργία ελέγχου πατήματος κάποιου από τα πλήκτρα αφής
public void CheckButtons()
{

string ConnectionString =
"datasource=83.212.120.81;port=3306;username=Arduino;password=ard1234";
MySQLConnection myConn = new MySQLConnection(ConnectionString);

myConn.Open();
string[] buttons = new string[7];
int c = 0;
int j = 0;
for (int i = 0; i < 6; i++)
{
if (i == 0) { j = 2; }
if (i == 1) { j = 3; }
if (i == 2) { j = 6; }
if (i == 3) { j = 9; }
if (i == 4) { j = 12; }
if (i == 5) { j = 44; }

byte k = (byte)j;
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

/\*Κάνει αίτηση για την ανάγνωση της κατάστασης όλων των κουμπιών αφής και αν κάποιο είναι πατημένο ενημερώνει το αντίστοιχο φως του αντίστοιχου χώρου

στην βάση δεδομένων με αντίστροφη λογική. Πρακτικά αν ένας διακόπτης είναι πατημένος και το αντίστοιχο φως είναι απενεργοποιημένο η βάση θα ενημερωθεί

ότι το φως αυτού του χώρου είναι πια ενεργοποιημένο.

```
Η μορφή της αίτησης είναι η εξής: 5,2,αριθμός διακόπτη
*/
buttons[c] = Communicators[0].message(5, 2, k);
if (buttons[c] == "0")
{
}
else if (buttons[c] == "1")
{
    using (MySqlCommand cmd = new MySqlCommand("UPDATE
Full_House_Web_Control.Appliances SET State=NOT State,update_time = NOW()
WHERE Id=@ID AND CustomerID="+CustomerID, myConn))
    {
```

```
        cmd.Parameters.AddWithValue("@ID", j);
        cmd.ExecuteNonQuery();
```

```
    }
}
c++;
```

```
}
```

```
myConn.Close();
```

```
}
```

//Λειτουργία ελέγχου συστήματος πυρασφάλειας, κουμπιού έκτακτης ανάγκης και συστήματος συναγερμού

```
public void CheckAlerts()
{
```

```
    if (Communicators[1].portConnection == true)
    {
```

```
        string ConnectionString =
"datasource=83.212.120.81;port=3306;username=Arduino;password=ard1234";
        MySqlConnection myConn = new
MySqlConnection(ConnectionString);
        MySqlConnection myConn2 = new
MySqlConnection(ConnectionString);
```

```
        /*
```

Το πρόγραμμα ζητάει από το Arduino με τα αισθητήρια να του κοινοποιήσει αν στο σπίτι έχει ενεργοποιηθεί το σύστημα πυρασφάλειας, σύστημα συναγερμού ή

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

ο διακόπτης έκτακτης ανάγκης. Αν κάποιο από αυτά έχει ενεργοποιηθεί κοιτάει την βάση δεδομένων για να δει αν έχει αναφέρει κάποια ίδια ενέργεια στον

χρήστη τα τελευταία 10 λεπτά. Αν έχει αναφέρει δεν τον ειδοποιεί ξανά. Αν όμως δεν έχει αναφέρει ενημερώνει την βάση δεδομένων με αυτή την πληροφορία.

Τρόπος αίτησης για σύστημα πυρασφάλειας: 7,1,96

Τρόπος αίτησης για ενεργοποίηση πλήκτρου έκτακτης ανάγκης:

7,1,97

Τρόπος αίτησης για σύστημα συναγερμού: 7,1,98

```
*/
string[] AlertMessages = { "Empty Alert", "Fire Alert",
"Emergency Alert", "Burglar Alert" };
string[] alarm = new string[4];
int c = 1;

MySQLDataReader reader = null;

for (int i = 96; i < 99; i++)
{
    byte k = (byte)i;

    alarm[c] = Communicators[1].message(7, 1, k);
    if (alarm[c] == "1")
    {
        string AlertName = AlertMessages[c];

        using (MySQLCommand cmd = new MySQLCommand("Select
Timestamp from Full_House_Web_Control.Logs WHERE IncidentName='" +
AlertName + "' AND CustomerID='"+CustomerID+"' ORDER BY ID DESC LIMIT 1",
myConn))
        {
            //SELECT
            myConn.Open();
            myConn2.Open();
            reader = cmd.ExecuteReader();
            if (reader.Read())
            {

                DateTime Timestamp = reader.GetDateTime(0);
                DateTime dtStartString = DateTime.Now;
                dtStartString = dtStartString.AddMinutes(-
10);

                if (Timestamp < dtStartString)
                {

                    //INSERT
                    using (MySQLCommand cmd2 = new
MySQLCommand("INSERT INTO Full_House_Web_Control.Logs VALUES
('"+CustomerID+"',@Alert,NOW(),'0','0')", myConn2))
                    {

                        cmd2.Parameters.AddWithValue("@Alert", AlertName);
                        cmd2.ExecuteNonQuery();
                    }
                }
            }
        }
    }
}
```

```

        }
    }

    }
    else
    {
        using (MySqlCommand cmd2 = new
        MySqlCommand("INSERT INTO Full_House_Web_Control.Logs VALUES
        ('"+CustomerID+"',@Alert,NOW(), '0', '0')", myConn2))
        {
            cmd2.Parameters.AddWithValue("@Alert",
            AlertName);
            cmd2.ExecuteNonQuery();
        }
    }
}

}

}
myConn.Close();
myConn2.Close();
c++;
}

}

}

//Αν πατηθεί το αριστερό κουμπί της εφαρμογής ξεκίνα το πρόγραμμα
private void button1_Click(object sender, EventArgs e)
{
    label1.Hide();
    label2.Hide();
    label5.Show();
    button1.Enabled = false;
    SetUp();

    CheckStatus();
    CheckSensors();
}

//Αν πατηθεί το δεξί πλήκτρο της εφαρμογής και είναι η πρώτη φορά
αναζήτησε στον υπολογιστή του χρήστη το πρόγραμμα OBS και ξεκίνα το.
private void button2_Click_1(object sender, EventArgs e)
{
    if (SurveillanceButtonStatus == 0)

```



```

        {
            myProcess = Process.Start("Surveillance.bat");
            SurveillanceButtonStatus++;
            button2.Image = Image.FromFile("images/CCTV-CameraG2.png");
        }

        //Αν το OBS είναι ήδη ενεργοποιημένο και το δεξί πλήκτρο
        πατηθεί ξανά τερμάτισε την εφαρμογή OBS
        else
        {
            Process[] processes = Process.GetProcessesByName("Obs");
            foreach (var process in processes)
            {
                process.Kill();
            }
            button2.Image = Image.FromFile("images/CCTV-CameraG.png");
            SurveillanceButtonStatus = 0;
        }

    }

    //Εμφάνιση φόρμας about us με την επιλογή του Menu
    private void aboutUsToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        var myForm = new Form2();
        myForm.Show();
    }

    //Εμφάνιση φόρμας οδηγιών με την επιλογή του Menu
    private void οδηγίεςToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        var myForm2 = new Form3();
        myForm2.Show();
    }

    //Εμφάνιση φόρμας ρυθμίσεων με την επιλογή του Menu
    private void ρυθμίσειςToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        var myForm3 = new Form4();
        myForm3.Show();
    }

    //Με τον τερματισμό του προγράμματος κλείσε και την εφαρμογή OBS
    εφόσον είναι ανοιχτή
    void Form1_FormClosed(object sender, FormClosedEventArgs e)
    {
        Process[] processes = Process.GetProcessesByName("Obs");
        foreach (var process in processes)
        {
            process.Kill();
        }
    }

}
}

```

## Αρχείο Form2.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace FullHouseWebControl
{
    public partial class Form2 : Form
    {
        public Form2 ()
        {
            InitializeComponent ();
        }
    }
}
```

## Αρχείο Form3.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace FullHouseWebControl
{
    public partial class Form3 : Form
    {
        public Form3 ()
        {
            InitializeComponent ();

            private void linkLabel1_LinkClicked(object sender,
            LinkLabelLinkClickedEventArgs e)
            {
                System.Diagnostics.Process.Start ("http://www.fullhousewebcontrol.com");
                //Με το πάτημα στην διεύθυνση άνοιξε στον προεπιλεγμένο περιηγητή του
                χρήστη αυτήν την σελίδα
            }
        }
    }
}
```

## Αρχείο Form4.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace FullHouseWebControl
{
    public partial class Form4 : Form
    {
        public Form4 ()
        {
            InitializeComponent ();
        }

        //Με το πάτημα του κουμπιού το πρόγραμμα ελέγχει αν το περιεχόμενο
        //του κουτιού υπάρχει. Έπειτα ελέγχει αν αυτό το ID χρήστη υπάρχει στην βάση
        //δεδομένων και το ορίζει ως CustomerID
        private void button1_Click(object sender, EventArgs e)
        {
            if (String.IsNullOrEmpty(textBox1.Text))
            {
                MessageBox.Show("Παρακαλώ πληκτρολογήστε ID πελάτη");
            }
            else {
                string ConnectionString =
                "datasource=83.212.120.81;port=3306;username=Arduino;password=ard1234";
                MySqlConnection myConn = new
                MySqlConnection(ConnectionString);
                MySqlDataReader reader = null;

                myConn.Open ();
                using (MySqlCommand cmd = new MySqlCommand("Select
                * from Full_House_Web_Control.Users WHERE CustomerID=" + textBox1.Text,
                myConn))
                {
                    reader = cmd.ExecuteReader ();

                    if (reader.Read ())
                    {
                        Form1.CustomerID = textBox1.Text;
                        MessageBox.Show("Οι αλλαγές
                        πραγματοποιήθηκαν με επιτυχία.");
                        this.Close ();
                    }
                    //Αν δεν υπάρχει εμφάνιση μηνύματος λάθους
                }
            }
        }
    }
}

```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
        MessageBox.Show("Ο χρήστης που  
πληκτρολογήσατε δεν υπάρχει στην βάση δεδομένων μας. Παρακαλώ προσπαθήστε  
ξανά.", "Σφάλμα", MessageBoxButtons.OK, MessageBoxIcon.Error);  
    }  
    myConn.Close();  
}  
  
}  
  
if (checkBox1.Checked)  
{  
    Form1.ButtonPref = true;  
}  
else {  
    Form1.ButtonPref = false;  
}  
  
}  
}  
}
```

## ΠΑΡΑΡΤΗΜΑ Ε΄

```
CREATE TABLE `Users` (
  `Username` varchar(255) DEFAULT NULL,
  `Password` varchar(255) DEFAULT NULL,
  `Name` varchar(255) DEFAULT NULL,
  `Image` varchar(255) DEFAULT NULL,
  `Email` varchar(255) DEFAULT NULL,
  `CustomerFrom` varchar(100) DEFAULT NULL,
  `HomeAdr` varchar(200) DEFAULT NULL,
  `LastCon` varchar(200) DEFAULT NULL,
  `HomeImg` varchar(255) DEFAULT NULL,
  `Dimensions` varchar(10) DEFAULT NULL,
  `Rooms` varchar(10) DEFAULT NULL,
  `SmartSystem` varchar(5) DEFAULT NULL,
  `CustomerID` varchar(10) DEFAULT NULL,
  `AuthenticatorKey` varchar(20) DEFAULT NULL,
  `EmergencyKey` varchar(10) DEFAULT NULL,
  `EmergencyPerson` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `Appliances` (
  `CustomerID` varchar(50) DEFAULT NULL,
  `Gate` varchar(50) DEFAULT NULL,
  `Id` varchar(50) DEFAULT NULL,
  `Name` varchar(200) DEFAULT NULL,
  `Area` varchar(100) DEFAULT NULL,
  `State` varchar(100) DEFAULT NULL,
  `ImageOn` varchar(255) DEFAULT NULL,
  `ImageOff` varchar(255) DEFAULT NULL,
  `update_time` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `Logs` (
  `CustomerID` varchar(10) DEFAULT NULL,
  `IncidentName` varchar(100) DEFAULT NULL,
  `Timestamp` varchar(100) DEFAULT NULL,
  `InformedClient` varchar(100) DEFAULT NULL,
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=366 DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `Preferences` (
  `CustomerID` varchar(10) DEFAULT NULL,
  `TypeOfPreference` varchar(100) DEFAULT NULL,
  `RequiresAction` varchar(10) DEFAULT NULL,
  `Title` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `Sensors` (
  `CustomerID` varchar(100) DEFAULT NULL,
  `Gate` varchar(50) DEFAULT NULL,
  `Id` varchar(50) DEFAULT NULL,
  `Name` varchar(100) DEFAULT NULL,
```

## Πλήρης Διαδικτυακός Έλεγχος Οικίας

```
`Area` varchar(100) DEFAULT NULL,  
`Value` varchar(100) DEFAULT NULL,  
`Image` varchar(255) DEFAULT NULL,  
`update_time` varchar(100) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `Surveillance` (  
`CustomerID` varchar(100) DEFAULT NULL,  
`SurveillanceLink` varchar(200) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <https://en.wikipedia.org/wiki/Microcontroller>
- [2] <https://el.wikipedia.org/wiki/Arduino>
- [3] <http://ww1.microchip.com/downloads/en/DeviceDoc/61143H.pdf>
- [4] <http://store.digilentinc.com/chipkit-max32-microcontroller-board-with-mega-r3-headers/>
- [5] <http://www.intel.com/content/www/us/en/embedded/products/galileo/galileo-overview.html>
- [6] [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)
- [7] <https://el.wikipedia.org/wiki/FPGA>
- [8] <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- [9] <http://www.micropik.com/PDF/dht11.pdf>
- [10] <https://www.seeedstudio.com/depot/datasheet/MQ-2.pdf>
- [11] <http://www.engineersgarage.com/insight/how-gas-sensor-works>
- [12] <https://www.seeedstudio.com/depot/datasheet/MQ-2.pdf>
- [13] <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>
- [14] <https://en.wikipedia.org/wiki/Photoresistor>
- [15] [http://pcbheaven.com/wikipages/How\\_a\\_Touch\\_Button\\_works/](http://pcbheaven.com/wikipages/How_a_Touch_Button_works/)
- [16] <http://playground.arduino.cc/Csharp/SerialCommsCSharp>
- [17] <https://www.youtube.com/watch?v=scpvrDLa1oY>

**(Κενό φύλλο)**



