

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
(ΠΡΩΗΝ ΤΕΙ ΔΥΤΙΚΗΣ ΕΛΛΑΔΟΣ -ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.)



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1733

**ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΑΥΤΟΜΑΤΙΣΜΩΝ
ΓΙΑ ΕΞΥΨΗΝΟ ΣΠΙΤΙ ΚΑΙ ΒΡΑΧΙΟΛΑΚΙ
ΕΚΤΑΚΤΗΣ ΑΝΑΓΚΗΣ ΓΙΑ ΗΛΙΚΙΩΜΕΝΟΥΣ,
ΒΑΣΙΣΜΕΝΑ ΣΤΗΝ ΠΛΑΤΦΟΡΜΑ ARDUINO,
ΜΕ ΤΟΠΙΚΟ Η ΑΠΟΜΑΚΡΥΣΜΕΝΟ ΕΛΕΓΧΟ.**

ΣΠΟΥΔΑΣΤΗΣ 1: ΔΡΙΜΗΣ ΣΩΤΗΡΙΟΣ (7068)

ΣΠΟΥΔΑΣΤΗΣ 2: ΚΑΜΙΝΕΛΗΣ ΔΗΜΗΤΡΙΟΣ (7080)

ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ: ΤΟΠΑΛΗΣ ΕΥΑΓΓΕΛΟΣ

ΠΑΤΡΑ 2019

Ευχαριστίες

Πρώτα από όλα, θα θέλαμε να ευχαριστήσουμε τον εισηγητή της πτυχιακής μας εργασίας, κ. Ευάγγελο Τοπάλη, για την πολύτιμη βοήθεια, την καθοδήγηση και το ενδιαφέρον που έδειξε καθ' όλη την διάρκεια εκπόνησης της πτυχιακής μας εργασίας. Επίσης θα θέλαμε, να ευχαριστήσουμε τον κ. Λουκά Χαδέλλη και τον κ. Βασίλειο Καψάλη για την ανάγνωση της πτυχιακής ως μέλη της εξεταστικής επιτροπής και για την παραχώρηση του εργαστηριακού χώρου, που μας βοήθησε στην υλοποίηση της πτυχιακής εργασίας. Τέλος θα θέλαμε να εκφράσουμε θερμές ευχαριστίες προς τις οικογένειες μας, για την αγάπη και την υποστήριξη τους όλα αυτά τα χρόνια. Αφιερώνουμε αυτή την πτυχιακή εργασία στους γονείς μας.

Περίληψη

Η παρούσα πτυχιακή εργασία ασχολείται με την υλοποίηση αυτοματισμών για έξυπνο σπίτι καθώς και βραχιολάκι έκτακτης ανάγκης για ηλικιωμένους. Επιτυγχάνεται μέσω της πλατφόρμας Arduino και αισθητήρων, είτε με τοπικό είτε με απομακρυσμένο έλεγχο.

Αρχικά θα μελετηθεί η χρησιμότητα και η εφαρμογή του έξυπνου σπιτιού και βραχιολιού έκτακτης ανάγκης στην καθημερινότητα. Στη συνέχεια θα γίνει ανάλυση και αναφορά στους μικροελεγκτές Arduino, στην γλώσσα προγραμματισμού Wiring C , στο πρόγραμμα xampp στη βάση δεδομένων MySQL, στις γλώσσες και τα εργαλεία προγραμματισμού διαδικτύου HTML, PHP, JavaScript, CSS και στην παρουσίαση γραφικών παραστάσεων με google charts και thingspeak.

Στην συνέχεια θα γίνει ανάλυση του υλικού και του λογισμικού μέρους του συστήματος. Πιο συγκεκριμένα στο υλικό μέρος θα επεξηγηθεί η λειτουργία του κάθε αισθητήρα και στο λογισμικό μέρος τα σενάρια αυτοματισμών, οι βιβλιοθήκες η συνδεσμολογία , η σύνδεση με το διαδίκτυο και η απεικόνιση των γραφημάτων.

Τέλος θα γίνει παράθεση συμπερασμάτων, όσον αφορά το τελικό αποτέλεσμα και θα αναφερθούν ιδέες και σκέψεις για μελλοντικές επεκτάσεις και βελτιώσεις του συστήματος που υλοποιήθηκε.

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη.....	3
Περιεχόμενα.....	4
Κεφάλαιο 1:Εισαγωγή.....	6
1.1 Γενικά στοιχεία.....	6
1.2 Έξυπνο σπίτι.....	6
1.3 Βραχιόλι έκτακτης ανάγκης για ηλικιωμένους ή ατόμων με ιατρικές παθήσεις....	8
1.4Ιδέα - Σύστημα προς υλοποίηση.....	9
Κεφάλαιο 2 : Τεχνολογίες και Εργαλεία Ανάπτυξης που Χρησιμοποιήθηκαν.....	10
2.1 Προγραμματισμός διαδικτύου.....	10
2.1.1 Javascript.....	11
2.1.2 HTML.....	12
2.1.3 CSS.....	14
2.1.4 PHP.....	15
2.1.5 AJAX.....	16
2.1.6 MySQL.....	18
2.2 Εργαλεία ανάπτυξης λογισμικού.....	19
2.2.1 Notepad.....	19
2.2.2 Xampp.....	20
2.2.3 Fritzing.....	22
2.2.4 Google charts.....	23
2.2.5 IFTTT.....	24
2.2.6 Thingspeak.....	25
2.2.7 Arduino IDE.....	25
2.3 Προγραμματισμός Arduino.....	27
2.3.1 Βασικά στοιχεία προγράμματος.....	27
2.3.2 Βασικές εντολές.....	30
2.3.3 Καταχώρηση βιβλιοθηκών.....	33
Κεφάλαιο 3 : Υλικό μέρος συστήματος.....	34
3.1 Πλατφόρμα Arduino.....	34
3.1.1 Γενικές πληροφορίες.....	34
3.1.2 Εκδόσεις Arduino.....	36
3.2 Arduino Shields	37
3.2.1 Ethernet Shield.....	37
3.3 Αισθητήρες και συσκευές δράσης που χρησιμοποιήθηκαν.....	38
3.3.1 Mini IR Infrared Pyroelectric PIR Body Motion Human Sensor Detector Module.....	38
3.3.2 MQ-135 Gas Sensor Module.....	39
3.3.3 Infrared IR Receiver Module Wireless Remote	40
3.3.4 Temperature and humidity sensor module.....	40
3.3.5 5V Relay TTL Signal 1 Channel Module High Level Expansion Board For Arduino.....	41
3.3.6 Flame Sensor Module IR Sensor Detector.....	41
3.3.7 DS1302 Real Time Clock Module.....	42

3.3.8 HC-SR04 Ultrasonic Module Distance Measuring Transducer.....	42
3.3.9 3 Colour LED Module.....	43
3.3.10 KY-018 PHOTORESISTOR MODULE.....	43
3.3.11 Soil moisture detector module.....	44
3.3.12 KY-017 Mercury Tilt Switch Module.....	44
3.3.13 Water Level Sensor.....	44
3.3.14 KY-012 Active Buzzer Module.....	45
3.3.15 DFRobot Capacitive Touch Sensor.....	45
3.3.16 Fan 60X60X25 EB60251S1 SUNON.....	46
3.3.17 DC Mini Motor.....	46
3.3.18 MPU-6050 module.....	47
3.3.19 Geekcreit® 1.3 Inch 4Pin White OLED LCD Display 12864 IIC I2C Interface Module For Arduino.....	48
3.3.20 Pulsesensor Pulse Heart Rate Sensor.....	48
3.3.21 Esp8266 Wifi Module.....	49

Κεφάλαιο 4 : Σενάρια αυτοματισμών και κύκλωμα.....50

4.1 Σενάριο ασφάλειας εξωτερικής πόρτας.....	50
4.2 Σενάριο αυτόματης αντλίας ανελκυστήρα.....	54
4.3 Σενάριο αυτόματου ποτίσματος.....	56
4.4 Σενάριο ελέγχου γκαραζόπορτας.....	58
4.5 Σενάριο ελέγχου κλιματισμού και ποιότητας αέρα.....	60
4.6 Σενάριο ανίχνευσης φωτιάς και πλημμύρας.....	62
4.7 Σενάριο ελέγχου παντζουριών και εξωτερικού φωτισμού.....	65
4.8 Σενάριο ελέγχου φωτισμού αίθουσας ψυχαγωγίας.....	67
4.9 Τελικό κύκλωμα έξυπνου σπιτιού.....	71
4.10 Σενάριο βραχιολιού έκτακτης ανάγκης για ηλικιωμένους ή άτομων με ιατρικές παθήσεις.....	74

Κεφάλαιο 5: Λογισμικό μέρος.....82

5.1 Σύνδεση Ethernet και Βάση Δεδομένων.....	82
5.1.1 Σύνδεση Ethernet.....	82
5.1.2 Βάση Δεδομένων.....	83
5.2 Ένδειξη Αναλογικών-ψηφιακών εισόδων και έλεγχος ψηφιακών εξόδων με Ajax μέσω κάρτας SD.....	93
5.3 ThingSpeak.....	102
5.3.1 Έξυπνο σπίτι.....	103
5.3.2 Βραχιόλι έκτακτης ανάγκης για ηλικιωμένους ή άτομων με ιατρικές παθήσεις.....	109
5.3.2.1 Διαγράμματα.....	109
5.3.2.2 Αυτόματες ειδοποιήσεις.....	112
5.4 Google Charts.....	118
5.5 Δημιουργία ιστοσελίδας.....	129

Κεφάλαιο 6 : Συμπεράσματα – Μελλοντικές επεκτάσεις.....142

Βιβλιογραφία	143
Παράρτημα Α.....	145
Παράρτημα Β.....	160

Κεφάλαιο 1: Εισαγωγή

1.1 Γενικά στοιχεία

Οι τεχνολογικές εξελίξεις, υπήρξαν πάντοτε ένας καθοριστικός παράγοντας, που σηματοδοτούσε σημαντικές αλλαγές στην πορεία της ανθρωπότητας, με πολλαπλές επιδράσεις σε μία σειρά τομέων της ζωής του ανθρώπου. Η εξέλιξη στο τομέα της πληροφορικής, αυξάνεται με ρυθμούς γεωμετρικής προόδου. Η εμφάνιση του διαδικτύου, διευκόλυνε σε μεγάλο βαθμό την ζωή των ανθρώπων, με αποτέλεσμα την αυτοματοποίηση των συσκευών, την εμφάνιση έξυπνων σπιτιών, κινητών και αυτοκινήτων καθώς και βραχιόλια έκτακτης ανάγκης για την έγκαιρη παροχή βοήθειας προς τους ηλικιωμένους.

Αρχικά, οι επιτακτικοί ρυθμοί του σύγχρονου τρόπου ζωής δημιουργούν διαρκώς νέες ανάγκες, οι οποίες απαιτούν την διαχείριση τους, από ένα σύστημα αυτοματισμών και ελέγχου. Η φράση “έξυπνο σπίτι” χρησιμοποιείται για οποιαδήποτε οικία, ενσωματώνει τη δυνατότητα ρύθμισης και ελέγχου ορισμένων ηλεκτρομηχανολογικών εγκαταστάσεων. Επίσης, το σπίτι με νοημοσύνη σκέπτεται και ενεργεί βάση των καθημερινών αναγκών και συνηθειών, δίνοντας τον απόλυτο έλεγχο σε συστήματα ασφάλειας, θέρμανσης, φωτισμού, ηλεκτρικών συσκευών, «περιεχομένων multimedia» κ.α. Οι λειτουργίες αυτές, ελέγχονται με το πάτημα ενός κουμπιού, είτε ο ιδιοκτήτης βρίσκεται εντός του κτιρίου είτε βρίσκεται σε κάποια απομακρυσμένη περιοχή.

Σε ότι αφορά, το βραχιόλι έκτακτης ανάγκης για ηλικιωμένους, έχει σχεδιαστεί με σκοπό να εξασφαλιστεί, ότι οι ηλικιωμένοι ή άτομα όλων των ηλικιών με ιατρικές παθήσεις θα λαμβάνουν τη άμεση φροντίδα σε περίπτωση έκτακτης ανάγκης. Φορώντας ένα βραχιόλι έκτακτης ανάγκης προσφέρεται μια επιπλέον σιγουριά, καθώς σε περίπτωση που πατηθεί το κουμπί έκτακτης ανάγκης και ανά τακτά χρονικά διαστήματα θα στέλνονται μετρήσεις από τις ζωτικές ενδείξεις ακόμα και μία επικείμενη πτώση του ατόμου, σε μία βάση δεδομένων, η οποία θα διαβάζεται από ειδικό ιατρικό προσωπικό.

Τέλος, στα επόμενα υποκεφάλαια που ακολουθούν θα αναλυθούν εκτενέστερα, το έξυπνο σπίτι, το βραχιόλι έκτακτης ανάγκης και η ιδέα για την υλοποίηση της πτυχιακής.

1.2 Έξυπνο σπίτι

Το έξυπνο σπίτι, ή αλλιώς σύστημα αυτοματισμού, είναι μια τεχνολογική υπεροχή, της οποίας οι δυνατότητες ελέγχου, τηλε-εποπτείας και τηλεχειρισμού μιας κατοικίας ή ενός κτιρίου, μέσω ενός κεντρικού συστήματος, είναι τεράστιες, εξελίξιμες και πλήρως επεκτεινόμενες.

Οι αυτοματισμοί σπιτιού δεν είναι μια πρόσφατη τεχνολογική ανακάλυψη, αφού τα πρώτα έξυπνα σπίτι εμφανίστηκαν για πρώτη φορά πριν από δύο δεκαετίες! Ωστόσο η δημοτικότητά τους αυξήθηκε περισσότερο τα τελευταία 2-3 χρόνια, εξαιτίας της τεράστιας εξέλιξης που σημειώθηκε στους τομείς, των ηλεκτρονικών υπολογιστών, των ψηφιακών συστημάτων ψυχαγωγίας, αλλά και του διαδικτύου, εξέλιξη που σε συνδυασμό με την αύξηση της ζήτησης για απλότητα και άνεση στην καθημερινή ζωή, λειτούργησε καταλυτικά.

Οι σημερινές τεχνολογίες αυτοματισμών προσφέρουν πολύ περισσότερα από τους απλούς προγραμματιζόμενους χρονοδιακόπτες. Ένα πραγματικά αυτοματοποιημένο σπίτι

χαρακτηρίζεται από τη δυνατότητα δημιουργίας πολύπλοκων συνθηκών φωτισμού, εξελιγμένου ελέγχου ασφαλείας, διανομής εικόνας και ήχου, και διαχείρισης ενέργειας.

Το πολύ-σύστημα «έξυπνο σπίτι» παρέχει ένα πλήθος πληροφοριών χειρισμού, ειδοποίησης και ενημέρωσης, σε κινητά και σταθερά τηλέφωνα, αλλά και σε ηλεκτρονικούς υπολογιστές. Συγκεκριμένα παρέχει ενημέρωση για την κατάσταση της κατοικίας και χειρισμό ασφαλείας του προσωπικού σας χώρου, μέσω τηλεπικοινωνιακών συσκευών.

Τρόποι εφαρμογής

1. Χειροκίνητα μέσω της οθόνης γραφικών
2. Μέσω χρονοπρογράμματος
3. Μέσω τηλεφώνων ή ηλεκτρονικών υπολογιστών

Οι λειτουργίες του «έξυπνου σπιτιού»

Η πιο σημαντική λειτουργία που παρέχει το «έξυπνο σπίτι», είναι η εξοικονόμηση θερμικής και ηλεκτρικής ενέργειας, η οποία επιτυγχάνεται με τον πλήρη έλεγχο της θέρμανσης του σπιτιού ανά σώμα (θερμική ενέργεια), και με την εφαρμογή των διαφόρων σεναρίων φωτισμού τα οποία τίθενται σε λειτουργία, είτε ο ιδιοκτήτης βρίσκεται στο σπίτι, είτε απουσιάζει (ηλεκτρική ενέργεια). Σε περίπτωση απουσίας, το σύστημα αυτό μπορεί να αναβοσβήσει τα φώτα του σπιτιού ή του κήπου, μέσω ενός μηνύματος από το κινητό του ιδιοκτήτη, έτσι ώστε να δημιουργείται η εντύπωση ότι υπάρχει κινητικότητα στην κατοικία, και να αποτρέπεται οποιαδήποτε πιθανή διάρρηξη.

Από την άλλη υπάρχει ένας «έξυπνος ελεγκτής», που δίνει τη δυνατότητα να παρακολουθείται την κατανάλωση νερού και οικιακών συσκευών, και την δυνατότητα να βλέπει τα δεδομένα για οποιαδήποτε χρονική περίοδο επιθυμεί. Αυτό το σύστημα ελέγχου βρίσκεται σε επιφυλακή και ενημερώνει για κάθε περίπτωση υπερβολικής κατανάλωσης, απ' όπου κι αν προέρχεται. Εξάλλου, «οι έξυπνοι θερμοστάτες» ελέγχουν συνεχώς το επίπεδο θερμοκρασίας, καθώς και την ομαλή λειτουργία θέρμανσης και κλιματισμού. Αποτέλεσμα των παραπάνω είναι η εξοικονόμηση ενέργειας και η μείωση των περιβαλλοντικών ρύπων.

Ο έλεγχος ηλεκτρολογικού ή μηχανολογικού εξοπλισμού και η ενεργοποίηση των σεναρίων για την επιλογή συγκεκριμένων λειτουργιών και προγραμμάτων θέρμανσης, ψύξης, εξαερισμού, εξασφαλίζουν τη διαβίωση σε ευχάριστες και άνετες περιβαλλοντικές συνθήκες. Κοντά στο σπίτι, μέσα από το αυτοκίνητο μπορεί να γίνει κλήση με το κινητό τηλέφωνο και να ενεργοποιηθεί το σενάριο «επιστροφή στο σπίτι», το οποίο με τη σειρά του ενεργοποιεί τη θέρμανση ή την ψύξη, τα περιμετρικά φώτα και το άνοιγμα της γκαραζόπορτας.

Ο χρήστης θα νιώθει ασφάλεια, ακόμα και την νύχτα που κοιμάται, γνωρίζοντας ότι το σπίτι του ελέγχεται ηλεκτρονικά. Η ενεργοποίηση και απενεργοποίηση του συστήματος ασφαλείας μπορεί να γίνει από οπουδήποτε. Η ειδοποίηση μέσω του κινητού τηλεφώνου, για οποιαδήποτε έκτακτη ανάγκη, ακόμα κι όταν ο εισβολέας κόψει την τηλεφωνική γραμμή. Σε περίπτωση που ο χρήστης ξεχάσει να οπλίσει το σύστημα, αυτό θα γίνει αυτόματα μετά την παρέλευση κάποιου χρονικού διαστήματος, εφόσον βέβαια δεν βρίσκεται κανείς στο σπίτι.

Άλλη μια λειτουργία όπου καλύπτεται από το πολύ-σύστημα «έξυπνο σπίτι», είναι η πυρανίχνευση και η αυτόματη κατάσβεση. Για τη λειτουργία αυτή μπορούν να συνδεθούν στο σύστημα, αισθητήρες ορατού καπνού ή απότομης αύξησης της θερμοκρασίας, και θερμοδιαφορικοί.

Το αυτόματο πότισμα, είναι ακόμη μια έξυπνη λειτουργία, η οποία ενεργοποιείται και απενεργοποιείται χειροκίνητα από την οθόνη αφής, ή αυτόματα μέσω του χρονοπρογράμματος, δηλώνοντας την επιθυμητή ώρα εκκίνησης και διάρκειας ποτίσματος.

Ανοίγμα και κλείσιμο οποιασδήποτε πηγής φωτισμού σε προγραμματισμένα διαστήματα, σύμφωνα με την ύπαρξη συγκεκριμένων συνθηκών ή τυχαία. Όταν για παράδειγμα ο χρήστης βρίσκεται σ' ένα δωμάτιο, όταν υπάρχουν καλεσμένοι, κατά τη διάρκεια του γεύματος, ή στην παρακολούθηση μιας ταινίας, τα φώτα αλλάζουν σύμφωνα με τη διάθεση των ατόμων.

Για την πρόληψη παραβίασης, υπάρχει η δυνατότητα προσομοίωσης της ανθρώπινης παρουσίας κατά τη διάρκεια απουσίας από την οικία. Επίσης, μπορούν να ελέγχονται πολλές διαφορετικές ζώνες φωτισμού, τόσο από κοντά (μέσω ενός κεντρικού διακόπτη), όσο και εξ αποστάσεως (μέσω της χρήσης ΗΥ, ή μέσω σταθερής και κινητής τηλεφωνίας).

Υπάρχει η δυνατότητα ελέγχου ομάδων συσκευών από ένα σημείο (διακόπτη). Όταν ο χρήστης θα ξυπνάει το πρωί θα μπορεί να έχει ζεστό νερό για να απολαύσει το μπάνιο του, καθώς και την αγαπημένη του μουσική να διαχέεται στους χώρους που έχει επιλέξει. Όταν θα λείπει από το σπίτι, οι κουρτίνες και οι τέντες θα ανοίγουν και θα κλείνουν σε τυχαίες χρονικές στιγμές για να δηλώσουν εικονική παρουσία. Μπορεί εξάλλου να γίνεται χειρισμός κι άλλων οικιακών συσκευών, όπως του θερμοσίφωνα, των πλυντηρίων, της κουζίνας, του στεγνωτήριου, του λέβητα, των σωμάτων καλοριφέρ, του air condition, της πισίνας, της γκαραζόπορτας κλπ.

Ασφαλής και εύκολος τρόπος για να μπαίνει στο σπίτι χωρίς τη χρήση κλειδιών. Στα οικεία πρόσωπα θα δίνεται η δυνατότητα με μια προσωπική κάρτα πρόσβασης, να ελέγχεται κάθε στιγμή η είσοδος και έξοδος από το σπίτι.

Τέλος, η διασκέδαση, με τη διανομή ήχου και εικόνας σε κάθε χώρο που επιθυμεί, από μια κεντρική πηγή (στερεοφωνικό, TV, DVD, video). Αν ξεχαστεί να προγραμματίσει το video για να μαγνητοσκοπήσει την αγαπημένη του ταινία, μπορεί να το κάνει απ' όποιο σημείο κι αν βρίσκεται, μέσω του προσωπικού του υπολογιστή ή της τηλεφωνικής του συσκευής.

1.3 Βραχιολάκι έκτακτης ανάγκης για ηλικιωμένους ή ατόμων με ιατρικές παθήσεις

Το βραχιόλι έκτακτης ανάγκης είναι μία ολοκαίνουργια επιλογή για ηλικιωμένα άτομα ή άτομα όλων των ηλικιών που αντιμετωπίζουν δυσκολίες στην καθημερινότητά τους. Είναι ένα διακριτικό και ευκολοφόρετο σύστημα ιατρικής προειδοποίησης. Επίσης, η σχεδιάσή του είναι τέτοια, ώστε να μπορεί και ένα ηλικιωμένο άτομο να το χρησιμοποιήσει. Διαθέτει τα συνήθη χαρακτηριστικά ενός έξυπνου ρολογιού, όπως η παρακολούθηση της υγείας και της φυσικής κατάστασης.

Με το κουμπί έκτακτης ανάγκης, την ανίχνευση πτώσης, την παρακολούθηση των καρδιακών παλμών και γενικότερα των ζωτικών ενδείξεων προσφέρεται μία επιπλέον σιγουριά σε αυτά τα άτομα. Οι πληροφορίες του ατόμου που φοράει το βραχιόλι στέλνονται σε μία βάση δεδομένων, όπου διαβάζονται από εξειδικευμένο ιατρικό προσωπικό.

1.4 Ιδέα - Σύστημα προς υλοποίηση

Η αρχική ιδέα ήταν η υλοποίηση ενός οικιακού αυτοματοποιημένου σεναρίου χωρίς να υπάρχει η απαραίτητη υποδομή. Η ιδέα αυτή εξελίχθηκε με την προσθήκη αρκετών αισθητήρων με αποτέλεσμα την υλοποίηση 8 οικιακών αυτοματοποιημένων σεναρίων. Τα σενάρια αυτά είναι η ασφάλεια της εξωτερικής πόρτας, η αυτόματη αντλία ανελκυστήρα, το αυτόματο πότισμα , ο έλεγχος της γκαραζόπορτας, ο κλιματισμός και η ποιότητα του αέρα, η ανίχνευση της φωτιάς και της πλημμύρας, ο έλεγχος των πατζουριών και του εξωτερικού φωτισμού και ο έλεγχος του φωτισμού της αίθουσας ψυχαγωγίας.

Έπειτα, ήρθε η ιδέα για την υλοποίηση ενός βραχιολιού έκτακτης ανάγκης, το οποίο θα βοηθήσει ηλικιωμένους ή άτομα με σοβαρές ιατρικές παθήσεις. Το βραχιόλι αποτελείται από τον αισθητήρα καρδιακών παλμών, την ανίχνευση πτώσης, τον αισθητήρα θερμοκρασίας υγρασίας, την πλακέτα ρολογιού και την οθόνη, που δείχνει τους καρδιακούς παλμούς , την ημερομηνία και ώρα και την θερμοκρασία και υγρασία.

Επίσης, θα γίνεται καταγραφή των περισσότερων μετρήσεων σε μία βάση δεδομένων και θα παρουσιάζονται σε γραφήματα ως προς το χρόνο έτσι ώστε να είναι εύκολα κατανοητά στο χρήστη. Ο έλεγχος των οικιακών αυτοματοποιημένων σεναρίων και του βραχιολιού έκτακτης ανάγκης για ηλικιωμένους θα επιτυγχάνεται μέσω της ιστοσελίδας που υλοποιήθηκε για την πτυχιακή χρησιμοποιώντας είτε έναν υπολογιστή είτε ένα κινητό τηλέφωνο.

Επιπλέον θα μπορεί να βλέπει σε πραγματικό χρόνο τις τιμές των αισθητήρων και να πραγματοποιεί χειρισμούς είτε σε τοπικό είτε σε απομακρυσμένο περιβάλλον .

Κεφάλαιο 2: Τεχνολογίες και εργαλεία ανάπτυξης που χρησιμοποιήθηκαν

Το κεφάλαιο αυτό αποτελείται από τις τεχνολογίες και τα εργαλεία ανάπτυξης, που χρησιμοποιήθηκαν για την εκπόνηση της πτυχιακής εργασίας. Γίνεται αναφορά στις γλώσσες και τα εργαλεία ανάπτυξης που χρησιμοποιήθηκαν για τον σχεδιασμό και την υλοποίηση της πτυχιακής εργασίας. Το κεφάλαιο απαρτίζεται από δύο ενότητες και ασχολείται με τον προγραμματισμό του διαδικτύου και τις πλατφόρμες ανάπτυξης λογισμικού.

2.1 Προγραμματισμός διαδικτύου

Ο προγραμματισμός του διαδικτύου έχει να κάνει με την εισαγωγή στις τεχνολογίες διαδικτύου, στον προγραμματισμό διαδικτυακών εφαρμογών και τις πλατφόρμες διαχείρισης περιεχομένου στο διαδίκτυο, που επιλέχθηκαν για την υλοποίηση της πτυχιακής εργασίας, τα οποία είναι πολύ σημαντικά για την σωστή λειτουργία του συστήματος καταγραφής μετρήσεων και εποπτείας των αυτοματοποιημένων σεναρίων τόσο στο έξυπνο σπίτι όσο και στο βραχιόλι έκτακτης ανάγκης για ηλικιωμένους. Στην ενότητα αυτή αναφέρονται οι γλώσσες προγραμματισμού, που χρησιμοποιήθηκαν για την επίτευξη της πτυχιακής εργασίας.



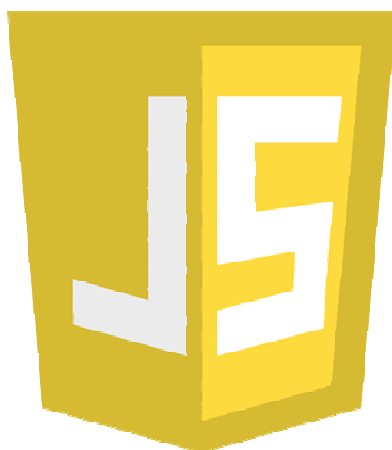
Εικόνα 1: Λογότυπα γλωσσών προγραμματισμού διαδικτύου

2.1.1 Javascript

Η **JavaScript (JS)** είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων. Τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).



Εικόνα 2: Λογότυπο Javascript

Ο κώδικας Javascript μιας σελίδας περικλείεται από τις ετικέτες της HTML `<script type="text/javascript">` και `</script>`.

Για παράδειγμα, ο ακόλουθος κώδικας Javascript εμφανίζει ένα πλαίσιο διαλόγου με το κείμενο "Hello world!":

```
<scripttype="text/javascript">  
alert("Hello world!");  
</script>
```

Εικόνα 3: Παράδειγμα κώδικα Javascript

2.1.2 HTML

Η **HTML** (αρχικοποίηση του αγγλικού **HyperText Markup Language**, ελλ. Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ. Ο σκοπός ενός web browser είναι να διαβάζει τα έγγραφα HTML και να τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να παρουσιάσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML και από στατικές τις κάνουν διαδραστικές.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει

τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.



Εικόνα 4: Λογότυπο HTML

Η σήμανση HTML αποτελείται από μερικά βασικά συστατικά, συμπεριλαμβανομένων των στοιχείων (και των ιδιοτήτων τους), τους βασισμένους σε χαρακτήρες τύπους δεδομένων, τις αναφορές χαρακτήρων και τις αναφορές οντοτήτων. Ένα ξεχωριστό σημαντικό συστατικό είναι η δήλωση τύπου εγγράφου (document type declaration), η οποία ορίζει στο πρόγραμμα περιήγησης (browser) τον τρόπο εμφάνισης της σελίδας.

Για παράδειγμα το πρόγραμμα Hello world που ακολουθεί στην HTML, είναι ένα συνηθισμένο πρόγραμμα υπολογιστή που χρησιμεύει για τη σύγκριση γλωσσών προγραμματισμού, γλωσσών σεναρίων και γλωσσών σήμανσης, φτιάχεται με 9 γραμμές κώδικα, παρότι οι νέες γραμμές είναι προαιρετικές στην HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello
HTML</title>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

Εικόνα 5: Παράδειγμα κώδικα HTML

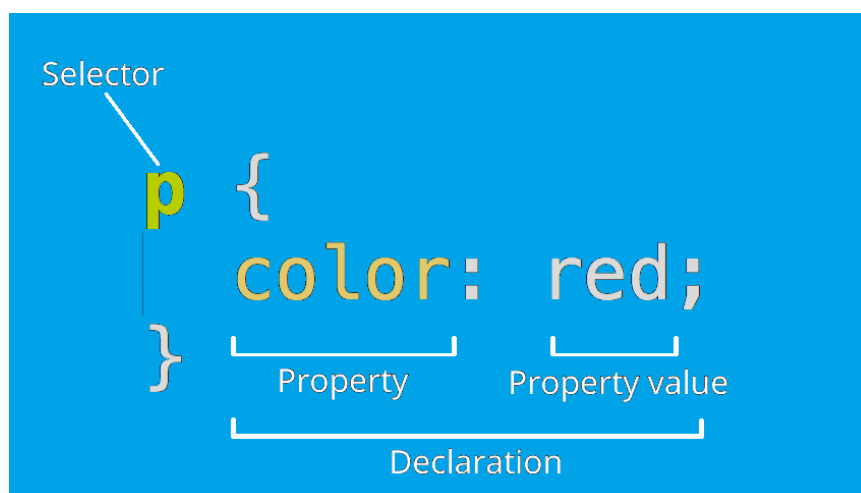
2.1.3 CSS

Η CSS (Cascading Style Sheets - Διαδοχικά Φύλλα Ύφους) ή (αλληλουχία φύλλων ύφους) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.



Εικόνα 6: Λογότυπο CSS

Ένας κανόνας CSS αποτελείται από δύο κύρια μέρη. Το πρώτο μέρος είναι ο επιλογέας (selector) και το δεύτερο μέρος είναι οι δηλώσεις (declaration). Ο επιλογέας αφορά το στοιχείο το οποίο είναι προς μορφοποίηση και είναι είτε μία ετικέτα html είτε κάποιο συγκεκριμένο αντικείμενο html. Οι δηλώσεις αποτελούνται από τις ιδιότητες που χαρακτηρίζουν τον επιλογέα και τις τιμές που αυτές παίρνουν. Παρακάτω απεικονίζεται ένα πρότυπο σύνταξης της CSS:



Εικόνα 7: Κανόνας σύνταξης CSS

```
body {  
    width: 80%;  
    margin: 0 auto;  
    background: white;
```

Εικόνα 8: Παράδειγμα σύνταξης CSS

2.1.4 PHP

Η **PHP** (PHP: Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργασθεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script.

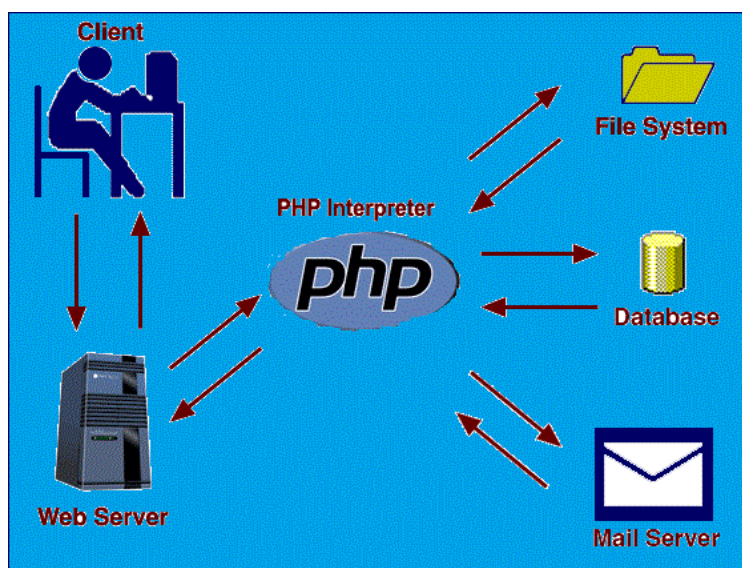
Η PHP αποτελεί μια από τις πιο διαδεδομένες τεχνολογίες στο Παγκόσμιο Ιστό, καθώς χρησιμοποιείται από πληθώρα εφαρμογών και ιστότοπων. Διάσημες εφαρμογές που κάνουν εκτενή χρήση της PHP είναι το γνωστό Σύστημα Διαχείρισης Περιεχομένου (Content Management System, WordPress και το Drupal).

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα πελάτη. Όπως αναφέρθηκε παραπάνω η γλώσσα php περιλαμβάνει κάποιες βασικές μεταβλητές οι οποίες έχουν σκοπό να βοηθούν στην ανάπτυξη διαδικτυακών δυναμικών εφαρμογών. Όπως αναφέρθηκε παραπάνω η γλώσσα php περιλαμβάνει κάποιες βασικές μεταβλητές οι οποίες έχουν σκοπό να βοηθούν στην ανάπτυξη διαδικτυακών δυναμικών εφαρμογών.



Εικόνα 9: Λογότυπο PHP

Έχει το ιδιαίτερο χαρακτηριστικό ότι ο κώδικάς της πρώτα μεταγλωττίζεται στον server και μετά φορτώνεται σαν ένα κανονικό html έγγραφο, χωρίς ο χρήστης να είναι σε θέση να δει τον αρχικό κώδικα. Η PHP είναι δωρεάν για χρήση, και είναι η δημοφιλέστερη για τους hosts Unix και Linux, αν και υπάρχουν εκδόσεις διαθέσιμες για τα Windows.



Εικόνα 10: Αρχιτεκτονική PHP

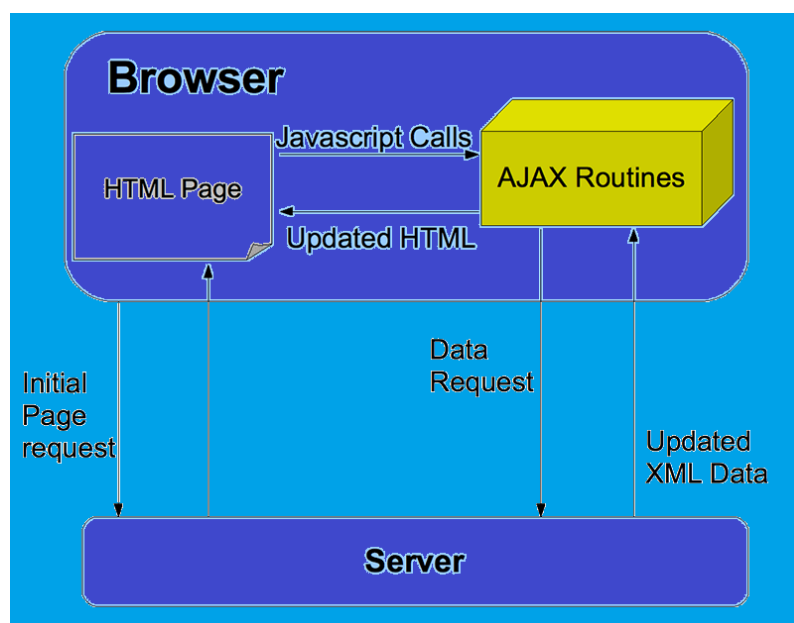
2.1.5 AJAX

Η **Ajax** (επίσης **AJAX** συντομογραφία για "Ασύγχρονη JavaScript και XML") είναι ένα σύνολο από Web development τεχνικές που χρησιμοποιούν πολλές τεχνολογίες του διαδικτύου από την πλευρά του πελάτη για να δημιουργήσουν ασύγχρονες Web εφαρμογές. Με Ajax, οι Web εφαρμογές μπορούν να στέλνουν και να ανακτούν δεδομένα από έναν διακομιστή (server) ασύγχρονα (τρέχοντας στο παρασκήνιο), χωρίς να παρεμβαίνουν στην εμφάνιση και τη συμπεριφορά της υπάρχουσας σελίδας. Με την αποσύνδεση του επιπέδου των δεδομένων που έχουν την δυνατότητα αλλαγής από το επίπεδο παρουσίασης της σελίδας, η Ajax επιτρέπει σε Web σελίδες, και κατ' επέκταση σε Web εφαρμογές, να αλλάζουν το περιεχόμενο τους δυναμικά, χωρίς να χρειάζεται να φορτωθεί εκ νέου ολόκληρη η σελίδα. Στην πράξη, οι σύγχρονες εφαρμογές συνήθως χρησιμοποιούν JSON, αντί για XML, λόγω των πλεονεκτημάτων του JSON που υπάρχουν εκ φυσικού στην JavaScript.



Εικόνα 11: Λογότυπο AJAX

Η Ajax δεν είναι μια ενιαία τεχνολογία, αλλά μάλλον μια ομάδα τεχνολογιών. Η HTML και η CSS μπορεί να χρησιμοποιηθούν σε συνδυασμό για να επισημάνουν και να δώσουν ειδικό στυλ στις πληροφορίες. Η ιστοσελίδα μπορεί στη συνέχεια να τροποποιηθεί με την JavaScript για εμφανίζει δυναμικά – και να επιτρέπει στο χρήστη να αλληλεπιδρά με τα νέα στοιχεία. Το ενσωματωμένο XMLHttpRequest αντικείμενο εντός της JavaScript χρησιμοποιείται συνήθως για να εκτελέσει την Ajax στις ιστοσελίδες επιτρέποντας σε ιστοσελίδες να φορτώσουν το περιεχόμενο τους πάνω στην οθόνη, χωρίς να ανανεώσουν τη σελίδα. Η Ajax δεν είναι μια νέα τεχνολογία, ή κάποια διαφορετική γλώσσα, αλλά υπάρχουσες τεχνολογίες που χρησιμοποιούνται με νέους τρόπους.



Εικόνα 12: Αρχιτεκτονική AJAX

2.1.6 MySQL

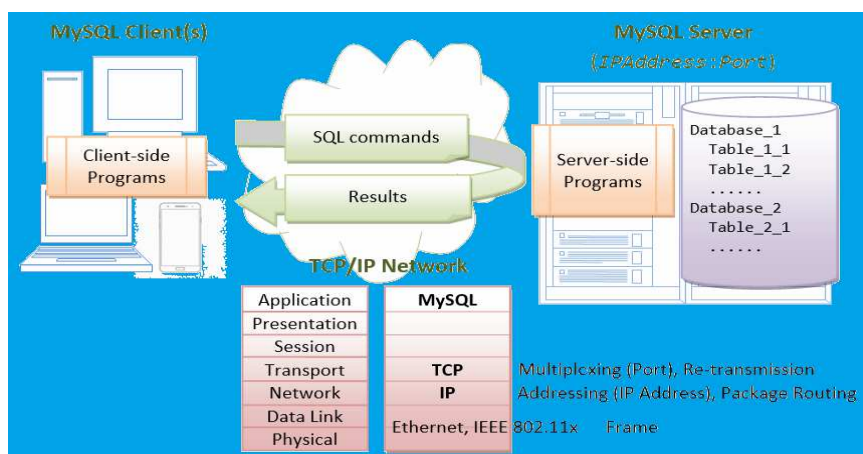
Η **MySQL** είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι (αγγλ. My). Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.



Εικόνα 13: Λογότυπο MySQL

Τα δεδομένα στη MySQL αποθηκεύονται σε αντικείμενα βάσης δεδομένων, τα οποία ονομάζεται πίνακες. Ένας πίνακας, είναι μια συλλογή από σχετικές καταχωρήσεις δεδομένων και αποτελείται από στήλες και γραμμές. Οι βάσεις δεδομένων είναι χρήσιμες για την αποθήκευση πληροφοριών σε κατηγορίες. Η διαχείριση των δεδομένων στις βάσεις, γίνονται μέσω των SQL εντολών, στη MySQL. Η πιο σημαντική ίσως κατηγορία εντολών που χρησιμοποιείται είναι τα queries. Ένα query είναι ένα ερώτημα ή ένα αίτημα. Με τη MySQL, επιτρέπεται η διερεύνηση σε μια βάση δεδομένων και η επιστροφή ζητηθέντων πληροφοριών. Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες.



Εικόνα 14: Αρχιτεκτονική MySQL

2.2 Εργαλεία ανάπτυξης λογισμικού

Για την ανάπτυξη του λογισμικού χρησιμοποιήθηκαν εργαλεία ανοιχτού λογισμικού. Όσον αφορά την συγγραφή του κώδικα και την κατασκευή της ιστοσελίδας αυτή υλοποιήθηκε με το Notepad++. Επίσης, έγινε χρήση του τοπικού διακομιστή Xampp για την άντληση των τιμών των αυτοματοποιημένων σεναρίων του έξυπνου σπιτιού. Επιπροσθέτως, χρησιμοποιήθηκε το λογισμικό arduino IDE για τον προγραμματισμό και την φόρτωση του κώδικα του arduino. Τέλος, η σχεδίαση των κυκλωμάτων του έξυπνου σπιτιού και βραχιολιού έκτακτης ανάγκης για ηλικιωμένους σε μορφή εικόνων υλοποιήθηκε με το πρόγραμμα Fritzing. Παρακάτω θα γίνει ανάλυση των προαναφερθέντων εργαλείων.



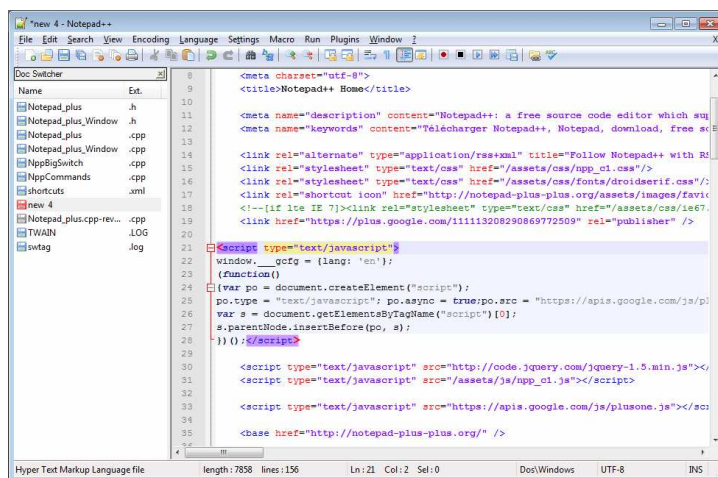
Εικόνα 15: Λογότυπα εργαλείων ανάπτυξης λογισμικού

2.2.1 Notepad++

Πρόκειται για ανοιχτού λογισμικού πρόγραμμα το οποίο χρησιμοποιείται για συγγραφή κώδικα Πρόκειται για τον πιο γνωστό επεξεργαστή κειμένου (text editor). Είναι απαραίτητο εργαλείο συγγραφής των προγραμματιστών, καθώς αποτελεί ένα ελαφρύ πρόγραμμα το οποίο όμως έχει πάρα πολλές δυνατότητες. Το περιβάλλον εργασίας είναι εξαιρετικά απλό και λειτουργικό. Στο επάνω μέρος του υπάρχει μία εργαλειοθήκη με τις κυριότερες λειτουργίες. Όλο το υπόλοιπο μέρος του καταλαμβάνεται από τον χώρο συγγραφής του κώδικα. Γύρω από τον χώρο αυτόν, μπορούν να εμφανιστούν διάφορα πλαίσια τα οποία παρουσιάζουν διάφορες πληροφορίες ανάλογα με τα plugins από τα οποία προέρχονται. Το σημαντικό του χαρακτηριστικό είναι το λεγόμενο syntax highlighting, το οποίο χρωματίζει τον κώδικα ανάλογα με την δομή του και διευκολύνει την ανάπτυξη του. Για κάθε γλώσσα προγραμματισμού υπάρχει διαφορετικό χρωματικό στίλ. Το πρόγραμμα έχει έτοιμα στίλ για πολλές γλώσσες και δίνει την δυνατότητα να τα ρυθμίσεις σύμφωνα με τις ανάγκες του προγραμματιστή. Στην παρούσα εργασία χρησιμοποιήθηκε για τη συγγραφή όλων των script σε PHP, Javascript, HTML, CSS, Ajax και MySQL.



Εικόνα 16: Λογότυπο Notepad++



Εικόνα 17: Περιβάλλον Notepad++

2.2.2 Xampp

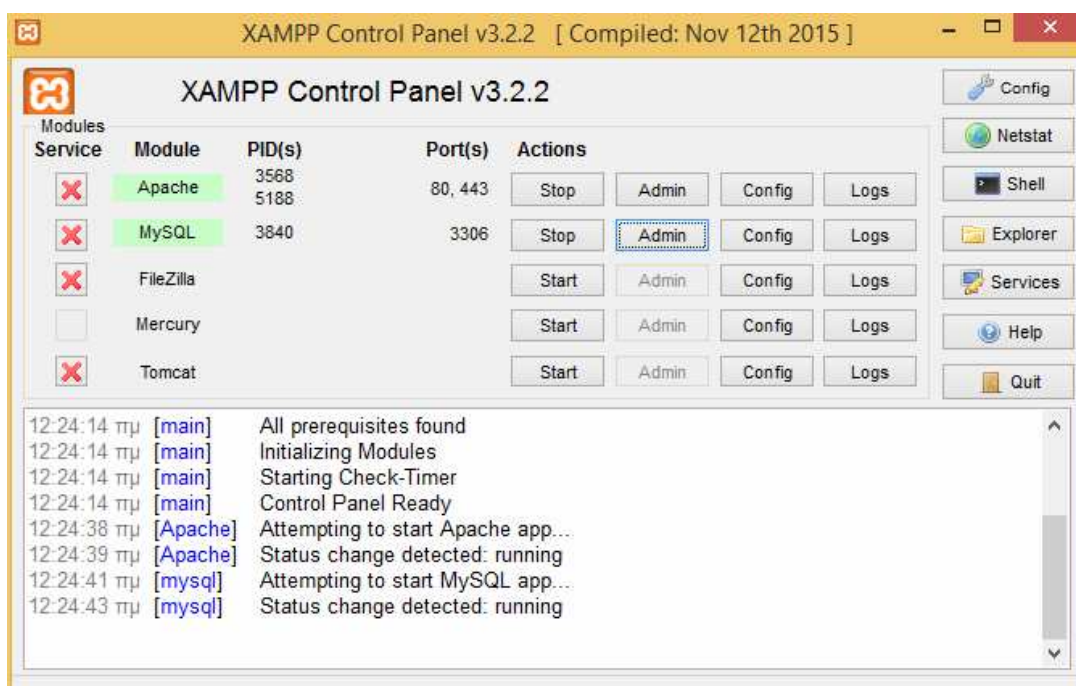
Το XAMPP είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα διεργασμένο για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl.

Το XAMPP είναι ελεύθερο λογισμικό το οποίο περιέχει ένα εξυπηρετητή ιστοσελίδων το οποίο μπορεί να εξυπηρετεί και δυναμικές ιστοσελίδες τεχνολογίας PHP/MySQL. Είναι ανεξάρτητο πλατφόρμας και τρέχει σε Microsoft Windows, Linux, Solaris, and Mac OS X και χρησιμοποιείται ως πλατφόρμα για την σχεδίαση και ανάπτυξη ιστοσελίδων με την τεχνολογίες όπως PHP, JSP και Servlets. Το XAMPP συμπεριλαμβάνει τα πακέτα OpenSSL και το phpMyAdmin και προϋποθέτει μόνο τα λογισμικά συμπίεσης αρχείων zip, tar, 7z ή exe κατά την διάρκεια της εγκατάστασης. Επίσης, έχει την δυνατότητα αναβάθμισης σε νέες εκδόσεις του εξυπηρετητή ιστοσελίδων http Apache, της βάσης δεδομένων MySQL, της γλώσσας PHP και Perl.



Εικόνα 18: Λογότυπο XAMPP

Επίσημα οι σχεδιαστές του XAMPP προόριζαν το λογισμικό ως εργαλείο ανάπτυξης και δοκιμής ιστοσελίδων τοπικά στον υπολογιστή χωρίς να είναι απαραίτητη η σύνδεση στο διαδίκτυο. Για να είναι δυνατή η χρήση του, πολλές σημαντικές λειτουργίες ασφάλειας έχουν απενεργοποιηθεί. Στην πράξη το XAMPP ορισμένες φορές χρησιμοποιείται και για την φιλοξενία ιστοσελίδων. Υπάρχει ειδικό εργαλείο το οποίο περιέχεται στο XAMPP για την προστασία με κωδικό των σημαντικών μερών. Το XAMPP υποστηρίζει την δημιουργία και διαχείριση βάσεων δεδομένων τύπου MySQL και SQLite. Όταν το XAMPP εγκατασταθεί στον τοπικό υπολογιστή διαχειρίζεται τον localhost ως ένα απομακρυσμένο κόμβο, ο οποίος συνδέεται με το πρωτόκολλο μεταφοράς αρχείων FTP. Η σύνδεση στον localhost μέσω του FTP μπορεί να γίνει με το όνομα χρήστη «newuser» και το κωδικό «wampp». Για την βάση δεδομένων MySQL υπάρχει ο χρήστης «root» χωρίς κωδικό πρόσβασης.



Εικόνα 19: Περιβάλλον XAMPP

2.2.3 Fritzing

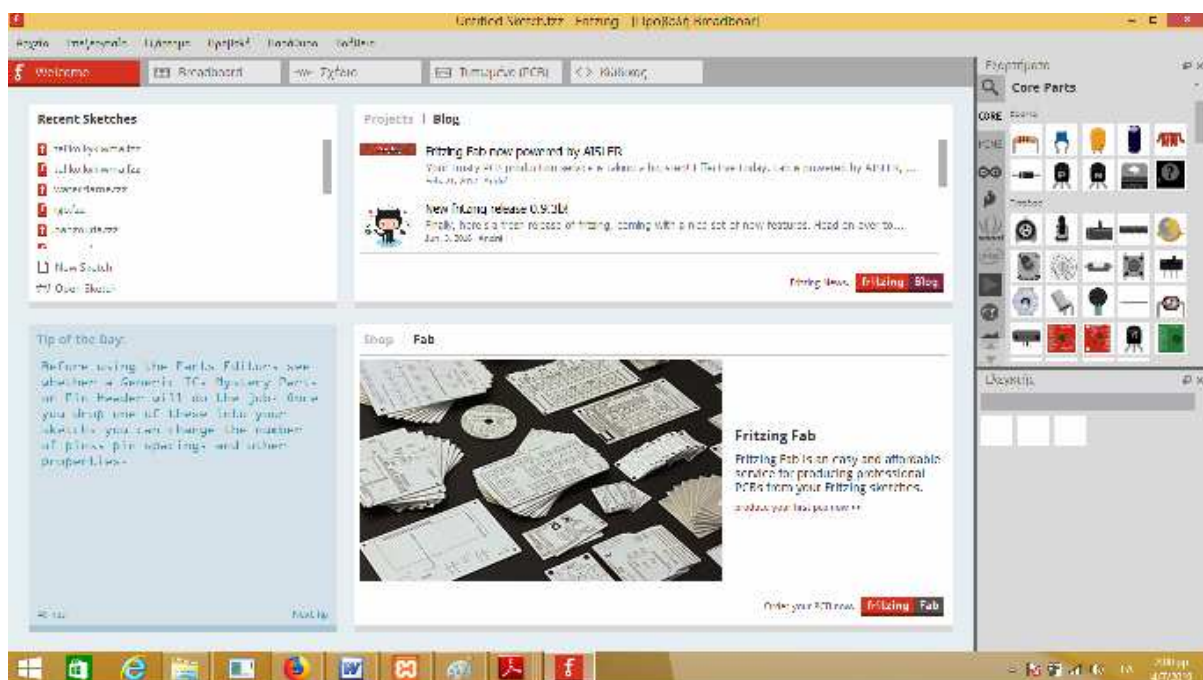
Το Fritzing είναι ένα ανοιχτού κώδικα σχεδιαστικό πρόγραμμα. Έχει δημιουργηθεί στο πνεύμα της γλώσσας Processing του Arduino και επιτρέπει στον σχεδιαστή να μετατρέψει το πρωτότυπο κύκλωμα που έχει υλοποιηθεί με την πλατφόρμα Arduino σε σχέδιο PCB. Το πρόγραμμα αναπτύχθηκε στο πανεπιστήμιο εφαρμοσμένων επιστημών του Πότσταμ.



Εικόνα 20: Λογότυπο Fritzing

Πρόκειται ίσως για τον πιο εύκολο τρόπο σχεδίασης κυκλωμάτων πλακετών. Διάφορες εταιρίες ηλεκτρονικών διαθέτουν ελεύθερες βιβλιοθήκες με αισθητήρες και ηλεκτρονικά εξαρτήματα και εργαλεία που μπορούν να προσθέτουν στα σχέδια. Επίσης ο κάθε χρήστης έχει τη δυνατότητα να σχεδιάσει μόνος του κάποιο εξάρτημα ή κάποιο αισθητήρα.

Το πρόγραμμα μπορούμε να το κατεβάσουμε ελεύθερα από την επίσημη ιστοσελίδα <http://fritzing.org/home/>. Όλα τα σχέδια που αφορούν το Arduino και παρουσιάζονται στο υπόλοιπο μέρος της πτυχιακής εργασίας έχουν σχεδιαστεί στο Fritzing.



Εικόνα 21: Περιβάλλον Fritzing

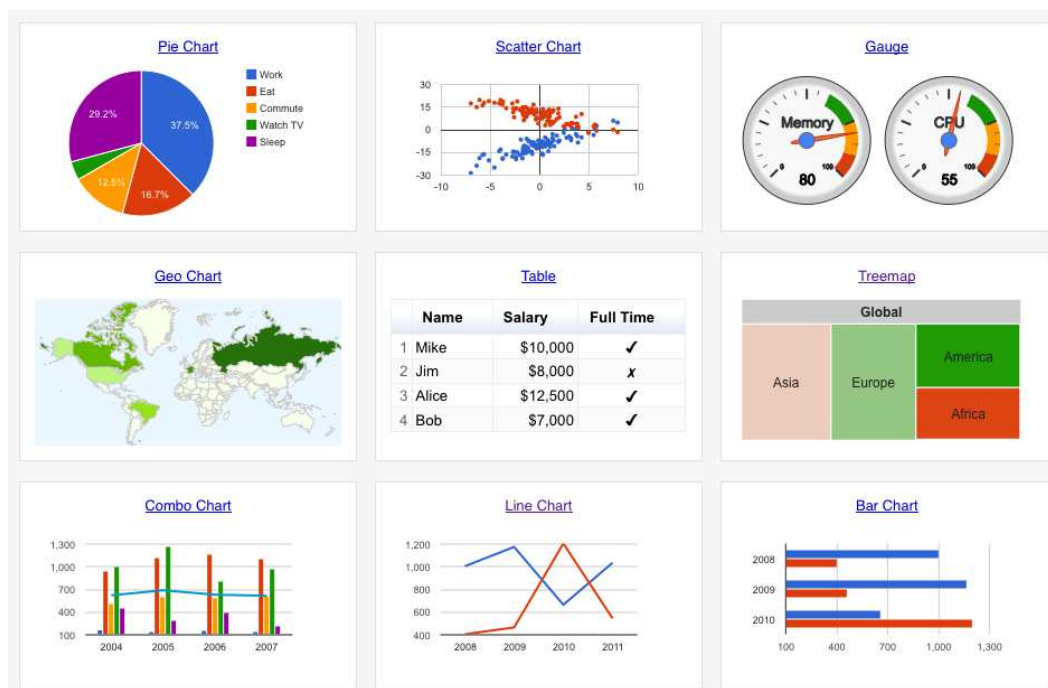
2.2.4 Google Charts

Το Google Charts είναι μία διαδραστική υπηρεσία Web η οποία δημιουργεί γραφικά γραφήματα από τις πληροφορίες που παρέχει ο χρήστης. Επίσης παρέχει τα δεδομένα και ένα αρχείο Javascript ενσωματωμένο σε μία ιστοσελίδα. Σε απάντηση η υπηρεσία του στέλνει μία εικόνα του διαγράμματος.

Η υπηρεσία Google Charts διαφέρει από την παλαιότερη υπηρεσία που παρείχε το API του Google Charts. Τα διαγράμματα δεν υποστηρίζουν την μέθοδο αιτήματος HTTP αλλά αναβαθμίστηκαν και πλέον υποστηρίζουν το Javascript. Η επικοινωνία του γίνεται απευθείας με τη βάση δεδομένων.



Εικόνα 22: Λογότυπο Google Charts



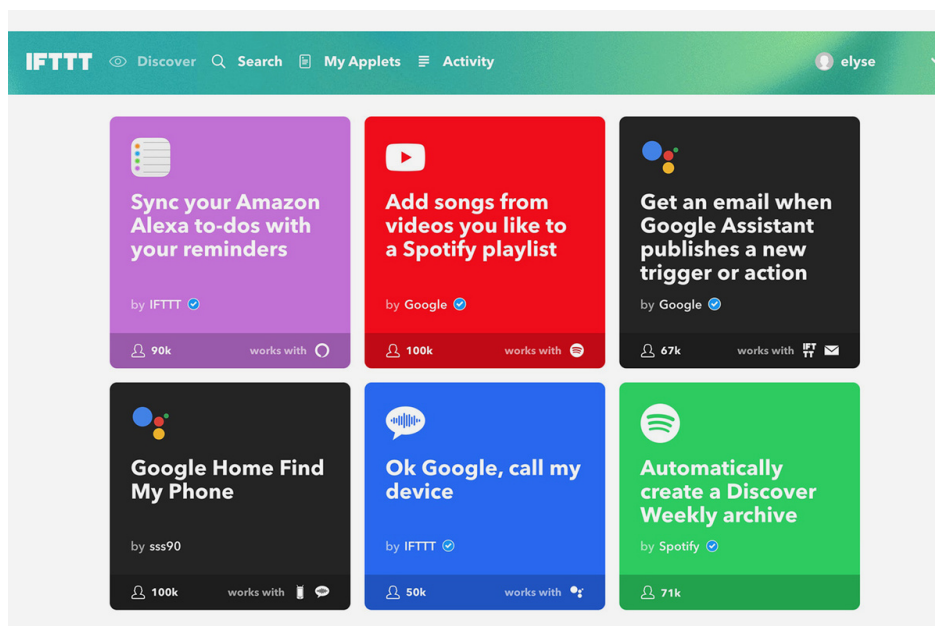
Εικόνα 23: Διάφορα γραφήματα Google Charts

2.2.5 IFTTT

Το IFTTT είναι μια υπηρεσία ιστού που επιτρέπει την δημιουργία applets που λειτουργούν ως απάντηση σε άλλη ενέργεια. Η υπηρεσία IFTTT Webhooks μπορεί να χρησιμοποιηθεί για την δημιουργία αιτημάτων ιστού για την ενεργοποίηση μια ενέργειας. Η εισερχόμενη ενέργεια είναι ένα αίτημα HTTP στον διακομιστή ιστού και η εξερχόμενη ενέργεια είναι ειδοποίηση στην εφαρμογή IFTTT μίας συσκευής .



Εικόνα 24: Λογότυπο IFTTT



Εικόνα 25: Περιβάλλον IFTTT

2.2.6 Thingspeak

Σύμφωνα με τους προγραμματιστές του, το ThingSpeak είναι μια εφαρμογή ανοιχτού κώδικα Internet of Things (IoT) και API για την αποθήκευση και ανάκτηση δεδομένων από στοιχεία που χρησιμοποιούν το πρωτόκολλο HTTP μέσω Internet ή μέσω τοπικού δικτύου. , εφαρμογές παρακολούθησης τοποθεσίας και ένα κοινωνικό δίκτυο πραγμάτων με ενημερώσεις κατάστασης.

Επίσης έχει ενσωματώσει υποστήριξη από το λογισμικό αριθμητικών υπολογιστών MATLAB από την MathWorks , επιτρέποντας στους χρήστες του ThingSpeak να αναλύουν και να απεικονίζουν δεδομένα που μεταφορτώνονται χρησιμοποιώντας το Matlab χωρίς να απαιτείται η αγορά άδειας χρήσης Matlab από τη Mathworks.

Προσφέρει δυνατότητες συλλογής δεδομένων σε πραγματικό χρόνο, οπτικοποίηση των συλλεγόμενων δεδομένων με τη μορφή διαγραμμάτων, δυνατότητα δημιουργίας προσθηκών και εφαρμογών για συνεργασία με υπηρεσίες ιστού, κοινωνικά δίκτυα και άλλα API.



Εικόνα 26: Λογότυπο Thingspeak



Εικόνα 27: Περιβάλλον Thingspeak

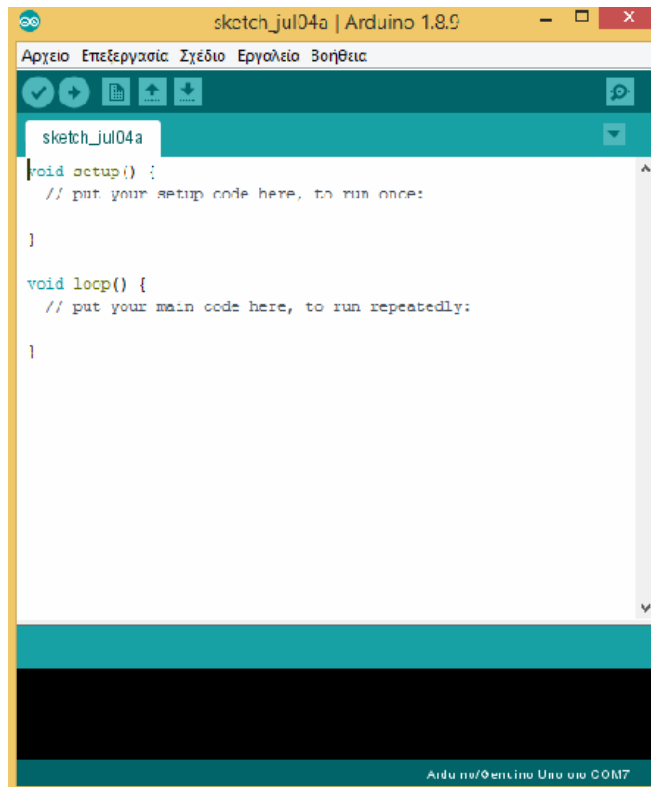
2.2.7 Arduino IDE

Το περιβάλλον ανάπτυξης του Arduino (IDE) είναι μία εφαρμογή μεταξύ των πλατφορμών (για Windows, MacOS, Linux) που είναι γραμμένη στη γλώσσα προγραμματισμού Java. Χρησιμοποιείται για την εγγραφή και τη μεταφόρτωση προγραμμάτων συμβατές με Arduino, χρησιμοποιώντας πυρήνες τριών μερών.



Εικόνα 28: Λογότυπο Arduino IDE

Ο πηγαίος κώδικας για τον IDE πηγάει από την GNU General Public License, έκδοση 2. Το Arduino IDE υποστηρίζει τις γλώσσες C και C ++ χρησιμοποιώντας ειδικούς κανόνες για τη δομή του κώδικα. Το Arduino IDE παρέχει πλήθος βιβλιοθηκών λογισμικού, το οποίο παρέχει πολλές κοινές διαδικασίες εισόδου και εξόδου. Ο κώδικας που έχει γραφτεί από τον χρήστη απαιτεί μόνο δύο βασικές εντολές για την εκκίνησή του, οι οποίες συντάσσονται και συνδέονται με το κύριο πρόγραμμα. Επίσης, με το IDE διατίθεται και η εργαλειοθήκη GNU. Το IDE του Arduino χρησιμοποιεί το πρόγραμμα avrdude για να μετατρέψει τον εκτελέσιμο κώδικα σε αρχείο κειμένου σε δεκαεξαδική μορφή το οποίο φορτώνεται στον πίνακα Arduino από το πρόγραμμα φορτωτή στο firmware του πίνακα.



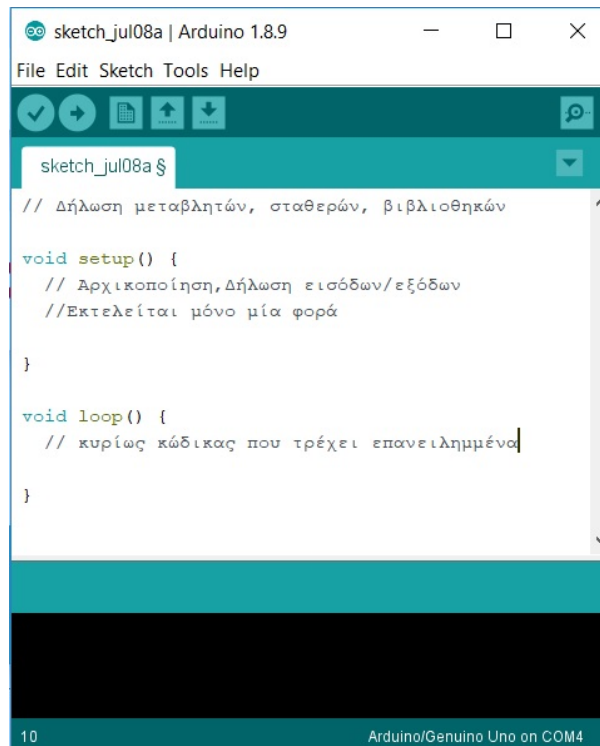
Εικόνα 29: Περιβάλλον Arduino IDE

2.3 Προγραμματισμός Arduino

2.3.1 Βασικά στοιχεία προγράμματος

Σε αυτήν την υποενότητα θα αναφερθούν κάποια από τα βασικά στοιχεία του προγράμματος arduino τα οποία είναι απαραίτητα για την χρήση του.

Περιβάλλον προγράμματος και βασικά μέρη του κώδικα



```
sketch_jul08a | Arduino 1.8.9
File Edit Sketch Tools Help
sketch_jul08a $
// Δήλωση μεταβλητών, σταθερών, βιβλιοθηκών

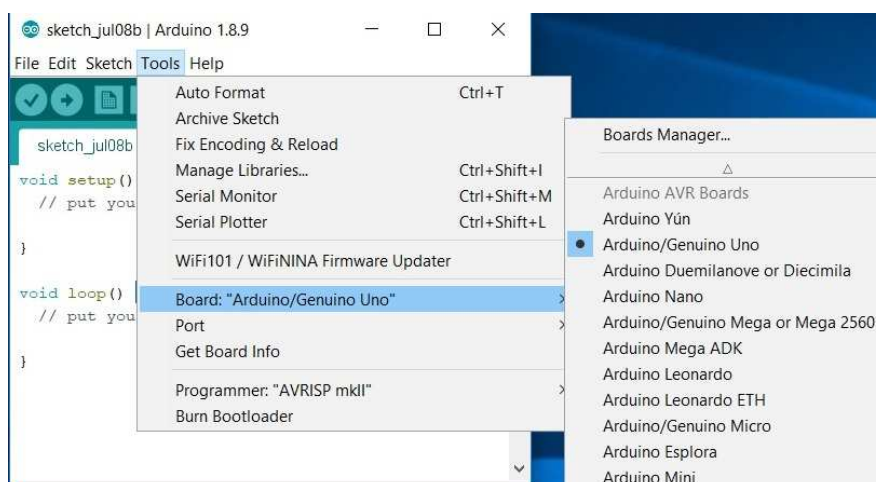
void setup() {
  // Αρχικοποίηση, Δήλωση εισόδων/εξόδων
  //Εκτελείται μόνο μία φορά
}

void loop() {
  // κυρίως κώδικας που τρέχει επανειλημμένα
}

10 Arduino/Genuino Uno on COM4
```

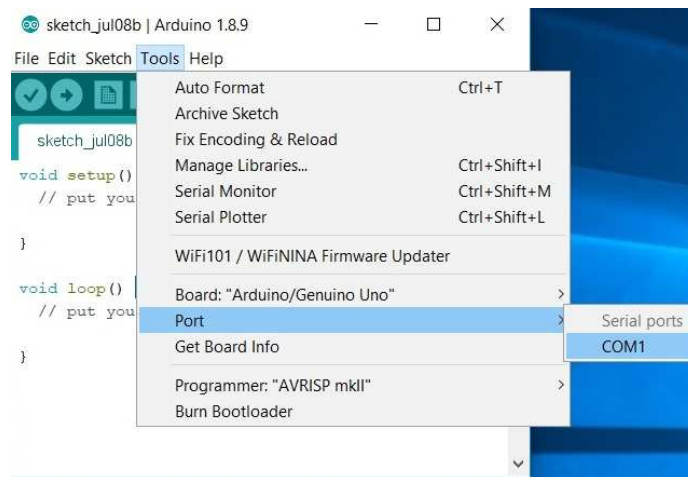
Εικόνα 30: Βασικά μέρη του προγράμματος

Επιλογή πλακέτας arduino



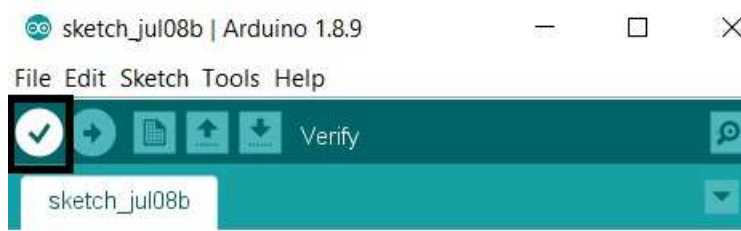
Εικόνα 31: Επιλογή πλακέτας Arduino

Επιλογή θύρας usb την οποία είναι συνδεδεμένη η πλακέτα Arduino



Εικόνα 32: Επιλογή θύρας Arduino

Έλεγχος του προγράμματος για συντακτικά λάθη



Εικόνα 33: Έλεγχος για συντακτικά λάθη

Ανέβασμα του κώδικα στην πλακέτα Arduino



Εικόνα 34: Ανέβασμα του κώδικα

Άνοιγμα σειριακής οθόνης

Η σειριακή οθόνη είναι ο «σύνδεσμος» μεταξύ του υπολογιστή και του Arduino - Επιτρέπει την αποστολή και την λήψη μηνυμάτων κειμένου, εύχρηστα για αποσφαλμάτωση ή για έλεγχο του Arduino από το πληκτρολόγιο.



Εικόνα 35: Σειριακή οθόνη

Ένδειξη σφαλμάτων που μπορεί να προκύψουν είτε κατά την μεταμόρφωση είτε κατά το ανέβασμα



Εικόνα 36: Ένδειξη σφαλμάτων

2.3.2 Βασικές εντολές

Η γλώσσα προγραμματισμού Arduino μπορεί να χωριστεί σε τρία κύρια μέρη:

1. Συναρτήσεις
2. Τιμές (μεταβλητές, σταθερές)
3. Δομή

Συναρτήσεις	
Digital I/O	
digitalRead()	Διαβάζει την τιμή από έναν ψηφιακό ακροδέκτη, είτε HIGH είτε LOW.
digitalWrite()	Γράφει μια τιμή HIGH ή LOW σε έναν ψηφιακό ακροδέκτη.
pinMode()	Ρυθμίζει ένα pin ώστε να συμπεριφέρεται ως είσοδος ή έξοδος
Analog I/O	
analogRead()	Διαβάζει την τιμή από έναν αναλογικό ακροδέκτη, είτε HIGH είτε LOW.
analogWrite()	Γράφει μια τιμή HIGH ή LOW σε έναν αναλογικό ακροδέκτη.
Time	
delay()	Παύει το πρόγραμμα για ένα χρονικό διάστημα (σε χιλιοστά του δευτερολέπτου)
millis()	Επιστρέφει τον αριθμό των χιλιοστών του δευτερολέπτου που πέρασαν από την έναρξη λειτουργίας του τρέχοντος προγράμματος Arduino.

Τιμές (μεταβλητές, σταθερές)	
Constants	
<u>Integer Constants</u>	Οι ακέραιες σταθερές είναι αριθμοί που χρησιμοποιούνται απευθείας στον κώδικα, όπως 123.
<u>HIGH</u> <u>LOW</u>	Κατά την εγγραφή ή ανάγνωση σε ένα ψηφιακό pin υπάρχουν δύο τιμές που μπορεί να πάρει: HIGH και LOW
<u>INPUT</u> <u>OUTPUT</u>	Δήλωση ενός ψηφιακού pin ως είσοδος ή έξοδος
<u>true</u> <u>false</u>	Σταθερές που δηλώνουν αλήθεια ή ψέμα
Data Types	
<u>byte</u>	Ένα byte αποθηκεύει έναν 8-bit μη υπογεγραμμένο αριθμό, από 0 έως 255.
<u>char</u>	Ένας τύπος δεδομένων που χρησιμοποιείται για την αποθήκευση μιας τιμής χαρακτήρα.
<u>array</u>	Ένας πίνακας είναι μια συλλογή μεταβλητών στον οποίο γίνεται πρόσβαση με έναν αριθμό ευρετηρίου.
bool	Ένα bool έχει δύο ερμηνείες τιμών αξίες, αληθής ή ψευδής. (Κάθε μεταβλητή bool καταλαμβάνει ένα byte μνήμης.)
<u>float</u>	Αριθμός που περιέχει δεκαδικό ψηφίο
<u>int</u>	Ακέραιος αριθμός

<u>long</u>	Μεταβλητές μεγάλου μεγέθους για την αποθήκευση αριθμών. Καταλαμβάνουν 32 bits (4 bytes), από -2,147,483,648 έως 2,147,483,647.
<u>unsigned long</u>	Σε αντίθεση με τις standard longs, οι unsigned longs δεν αποθηκεύουν αρνητικούς αριθμούς, κάνοντας το φάσμα από 0 έως 4.294.967.295.
<u>void</u>	Χρησιμοποιείται στις δηλώσεις συναρτήσεων.
<u>double</u>	Αριθμός κινητής υποδιαστολής δύο δεκαδικών. Στο Uno και σε άλλες πλακέτες που βασίζονται στο ATMEGA, καταλαμβάνει 4 byte.

Δομή	
Sketch	
<u>loop()</u>	Τοποθετείται μετά την void setup και τρέχει επανειλημμένα (Ενεργός συνεχής έλεγχος του Arduino)
<u>setup()</u>	Η συνάρτηση setup() καλείται στην αρχή ενός sketch. Χρησιμοποιείται για αρχικοποίηση μεταβλητών, pin modes, την χρησιμοποίηση βιβλιοθηκών κ.λπ. Η λειτουργία setup () θα εκτελείται μόνο μία φορά, μετά από κάθε ενεργοποίηση ή επανεκκίνηση του Arduino.
Control Structure	
break	Έξοδος από for, while, do...while loop και switch case
continue	Η εντολή continue παραλείπει την υπόλοιπη τρέχουσα επανάληψη ενός βρόχου
do...while	Η do ... while loop λειτουργεί με τον ίδιο τρόπο η while, με την εξαίρεση ότι ο έλεγχος γίνεται στο τέλος του βρόχου, οπότε ο βρόχος do θα τρέχει τουλάχιστον μία φορά.
for	Δομή επανάληψης όπου χρησιμοποιείται μετρητής αύξησης για την αύξηση και τον τερματισμό του βρόχου.
<u>if</u> (else)	έλεγχος μίας συνθήκης (πολλαπλών συνθηκών)
<u>while</u>	Ένας βρόχος while θα εκτελείται συνεχώς, και απεριόριστα, μέχρι η έκφραση μέσα στην παρένθεση () να γίνει ψευδής.
switch...case	Μία switch statement συγκρίνει την τιμή μιας μεταβλητής με τις τιμές που καθορίζονται στις case statements. Όταν εντοπιστεί μια case statement της οποίας η τιμή ταιριάζει με εκείνη της μεταβλητής, εκτελείται ο κώδικας.
Further Syntax	
<u>#define</u> (define)	Απόδοση ονόματος σε μία σταθερά
<u>#include</u> (include)	Χρησιμοποιείται για να συμπεριλάβει εξωτερικές βιβλιοθήκες στον κώδικα.
<u>/* */</u> (block comment)	Σχόλια πολλών γραμμών. Τα σχόλια είναι γραμμές στο πρόγραμμα που χρησιμοποιούνται για την διευκρίνιση του τρόπου

	λειτουργίας του προγράμματος.
// (single line comment)	σχόλιο μίας γραμμής
;(semicolon)	μπαίνει στο τέλος μίας δήλωσης ή εντολής
{(curly braces)	Άνοιγμα/ κλείσιμο δομών επανάληψης, συναρτήσεων και δομών επιλογής
Arithmetic Operators	
*	(πολλαπλασιασμός)
+	(πρόσθεση)
-	(αφαίρεση)
/	(διαίρεση)
=	(ισότητα)
%	(υπόλοιπο ακέραιας διαίρεσης)
Comparison Operators	
!=	(όχι ίσο)
<	(μικρότερο)
<=	(μικρότερο ή ίσο)
==	(ίσο με)
>	(μεγαλύτερο)
>=	(μεγαλύτερο ή ίσο)
Boolean Operators	
!	(λογικό όχι)
&&	(λογικό και)
	(λογικό ή)

Πίνακας 1: Κύρια μέρη γλώσσας προγραμματισμού Arduino

2.3.3 Καταχώρηση βιβλιοθηκών

Στην αρχή κάθε προγράμματος γίνεται εισαγωγή των βιβλιοθηκών για την εύρυθμη λειτουργία του κώδικα και την αναγνώριση των αισθητήρων. Οι βιβλιοθήκες που χρησιμοποιήθηκαν στους κώδικες είναι οι εξής:

BIBΛΙΟΘΗΚΗ	ΕΠΕΞΗΓΗΣΗ
SPI	Χρησιμοποιείται για την σειριακή επικοινωνία μεταξύ περιφερειακών συσκευών. Στην περίπτωσή μας του Ethernet Shield με το Arduino ως Master.
ETHERNET	Επιτρέπει τη σύνδεση του Arduino με το Internet. Είτε Αποδέχοντας συνδέσεις ως server είτε αποστέλοντας ως client.
SD	Επιτρέπει την εγγραφή και την ανάγνωση από την κάρτα μνήμης SD.
THINGSPEAK	Είναι η βιβλιοθήκη επικοινωνίας της υπηρεσίας Thingspeak με τον κώδικα.
IR REMOTE	Επιτρέπει να στέλνει και να δέχεται σήματα υπεράυθρων με πολλαπλά πρωτόκολλα.
DS1302	Είναι η βιβλιοθήκη για την αναγνώριση της πλακέτας ρολογιού.
WIRE	Επιτρέπει την επικοινωνία με συσκευές που υποστηρίζουν το πρωτόκολλο I2C.
U8GLIB	Είναι η βιβλιοθήκη της OLED οθόνης με πρωτόκολλο επικοινωνίας I2C.
SOFTWARE SERIAL	Επιτρέπει την σειριακή επικοινωνία με άλλες ψηφιακές ακίδες του Arduino, χρησιμοποιώντας λογισμικό για την αναπαραγωγή της λειτουργικότητας (εξ ου και το όνομα "SoftwareSerial").
DHT	Είναι η βιβλιοθήκη για την αναγνώριση του αισθητήρα θερμοκρασίας και υγρασίας.

Πίνακας 2: Βιβλιοθήκες που χρησιμοποιήθηκαν

ΚΕΦΑΛΑΙΟ 3: ΥΛΙΚΟ ΜΕΡΟΣ ΣΥΣΤΗΜΑΤΟΣ

3.1 Πλατφόρμα Arduino

3.1.1 Γενικές πληροφορίες

Το **Arduino** είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες· το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.

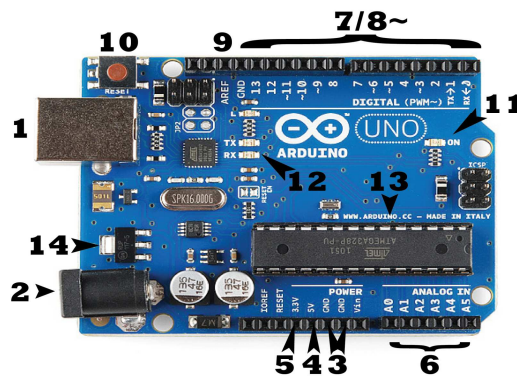
Το πρόγραμμα Arduino έλαβε τιμητική μνεία στην κατηγορία *Digital Communities* στο *Prix Ars Electronica* το 2006.

Το υλικό και το λογισμικό Arduino σχεδιάστηκε για καλλιτέχνες, σχεδιαστές, χομπίστες, χάκερ, αρχάριους και όσους ενδιαφέρονται να δημιουργήσουν διαδραστικά αντικείμενα ή περιβάλλοντα. Το Arduino μπορεί να αλληλεπιδράσει με κουμπιά, LED, κινητήρες, ηχεία, μονάδες GPS, κάμερες, Internet, έξυπνα τηλέφωνα ή τηλεόρασεις. Αυτή η ευελιξία σε συνδυασμό με το γεγονός ότι το λογισμικό Arduino είναι δωρεάν, οι πλακέτες είναι αρκετά φθηνές και το λογισμικό και το υλικό είναι εύκολο να κατανοηθεί οδήγησε σε μια μεγάλη κοινότητα χρηστών που έχουν συνεισφέρει κώδικα και έχουν εκδώσει οδηγίες για μια τεράστια ποικιλία έργων με βάση το Arduino.

Υλικό (Hardware)

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωσή του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε κάποιες παραλλαγές). Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Βασικά στοιχεία που βρίσκονται πάνω σε μία πλατφόρμα Arduino:



Εικόνα 37: Βασικά στοιχεία πλατφόρμας Arduino

Τροφοδοσία/ φόρτωση κώδικα

Σύνδεση USB (1)
τροφοδοσία Barrel Jack (2)

Υποδοχές-Ακίδες (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

Οι υποδοχές στο Arduino είναι τα σημεία όπου συνδέονται τα καλώδια προκειμένου να κατασκευαστεί ένα κύκλωμα (πιθανώς σε συνδυασμό με ένα breadboard και μερικά καλώδια.)

GND (3)	Γείωση. Υπάρχουν αρκετές υποδοχές GND στο Arduino, οι οποίες μπορούν να χρησιμοποιηθούν για τη γείωση του κυκλώματός.
5V (4) & 3.3V (5)	Όπως ίσως μαντέψατε, ο 5V ακροδέκτης παρέχει 5 βολτ ισχύος και η 3,3V καρφίτσα τροφοδοτεί 3,3 βολτ ισχύος. Τα περισσότερα από τα απλά εξαρτήματα που χρησιμοποιούνται με το Arduino τρέχουν ευτυχώς μακριά από 5 ή 3,3 βολτ.
Αναλογικές (6)	'Analog In' (A0 έως A5 στο Uno) Αυτές οι ακίδες μπορούν να διαβάσουν το σήμα από έναν αναλογικό αισθητήρα (όπως ένας αισθητήρας θερμοκρασίας) και να το μετατρέψουν σε ψηφιακή τιμή που μπορούμε να διαβάσουμε.
Ψηφιακές (7)	Πέρα από τις αναλογικές ακίδες υπάρχουν οι ψηφιακές ακίδες (0 έως 13 στο Uno). Αυτές οι καρφίτσες μπορούν να χρησιμοποιηθούν τόσο για ψηφιακή είσοδο όσο και για ψηφιακή έξοδο (όπως τροφοδοσία LED).
PWM (8) (~)	Ακίδες (3, 5, 6, 9, 10 και 11 στο UNO). Αυτές οι ακίδες λειτουργούν ως κανονικές ψηφιακές ακίδες, αλλά μπορούν επίσης να χρησιμοποιηθούν και για διαμόρφωση πλάτους παλμών Pulse-Width Modulation (PWM). Αυτές οι ακίδες είναι σε θέση να προσομοιώνουν την αναλογική έξοδο.
AREF (9)	Υποστηρίζει αναλογική αναφορά. Χρησιμοποιείται μερικές φορές για να ρυθμίσετε μια εξωτερική τάση αναφοράς (μεταξύ 0 και 5 Volt) ως το ανώτερο όριο για τους ακροδέκτες αναλογικής εισόδου.
Κουμπί Reset (10)	Πιέζοντας αυτό θα συνδέσει προσωρινά τον ακροδέκτη επαναφοράς στη γείωση και θα επανεκκινήσει οποιονδήποτε κώδικα που έχει φορτωθεί στο Arduino.

Πίνακας 3: Υποδοχές- Ακίδες Arduino

Δείκτης LED λειτουργίας (ON) (11).

TX RX LEDs (12) Ακίδες που είναι υπεύθυνες για σειριακή επικοινωνία. Αυτά τα LED θα μας δώσουν κάποιες οπτικές ενδείξεις όποτε το Arduino μας λαμβάνει ή μεταδίδει δεδομένα (όπως όταν φορτώνουμε ένα καινούργιο πρόγραμμα στην πλατφόρμα).

Κύριο ολοκληρωμένο κύκλωμα (13)

Ρυθμιστής τάσης (14)

Name	Processor	Operating/In put Voltage	CPU Spee d	Analo g In/Out	Digital IO/PW M	EEPRO M [kB]	SRA M [kB]	Flash [kB]	USB	UART
<u>Mega</u> <u>2560</u>	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regul ar	4
<u>Uno</u>	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regul ar	1
<u>Nano</u>	ATmega168P ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	0.512 1	1 2	16 32	Mini	1

Πίνακας 4: Χαρακτηριστικά διάφορων τύπων Arduino

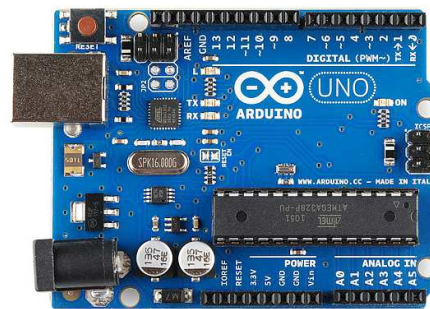
3.1.2 Εκδόσεις Arduino

Υπάρχουν διάφοροι τύποι πλακετών arduino οι οποίες έχουν δημιουργηθεί για να εξυπηρετήσουν πληθώρα εφαρμογών. Για την συγκεκριμένη πτυχιακή χρησιμοποιήθηκαν οι πλακέτες arduino uno και mega

Επίσης θα αναφερθεί και το arduino nano το οποίο προοριζόταν για το έξυπνο βραχιόλι, αλλά τελικά δεν χρησιμοποιήθηκε λόγω προβλήματος πλακέτας.

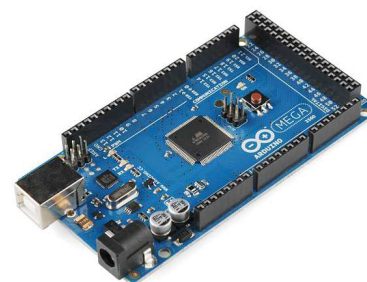
Τεχνικά χαρακτηριστικά Arduino

Το arduino uno είναι η πιο διαδεδομένη πλακέτα καθώς και η αρχική έκδοση που δημιουργήθηκε. Μπορεί να χρησιμοποιηθεί σαν αρχική πλακέτα εκμάθησης αλλά και για πιο εξειδικευμένες εφαρμογές. Διαθέτει 14 ψηφιακές εισόδους/εξόδους (6 PWM) και έξι αναλογικές.



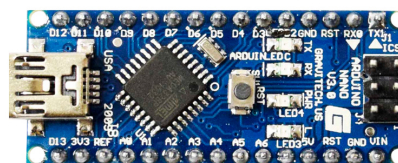
Εικόνα 38: Arduino Uno R3

Το Arduino Mega είναι σαν ο μεγάλος αδελφός του Uno. Έχει 54 ψηφιακές Ακίδες εισόδου / εξόδου (14 PWM), 16 αναλογικές εισόδους. Ο μεγάλος αριθμός ακίδων το καθιστούν αυτήν την πλακέτα πολύ βολική για έργα που απαιτούν πολλές ψηφιακές εισόδους ή εξόδους (όπως πολλά LED, κουμπιά, αισθητήρες).



Εικόνα 39: Arduino Mega 2560

Το arduino nano έχει περίπου τα ίδια χαρακτηριστικά με το arduino uno με τη διαφορά ότι είναι πολύ μικρότερο σε μέγεθος. Για αυτόν τον λόγο μπορεί να χρησιμοποιηθεί σε εφαρμογές με μικρότερο όγκο όπως για παράδειγμα στο έξυπνο βραχιόλι.



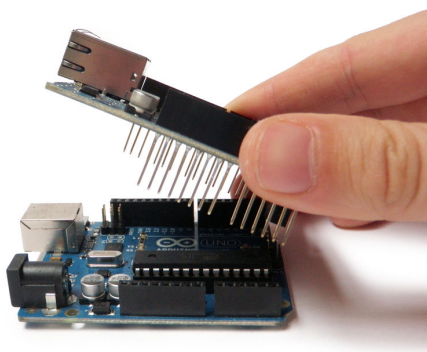
Εικόνα 40: Arduino Nano

3.2 Arduino shields

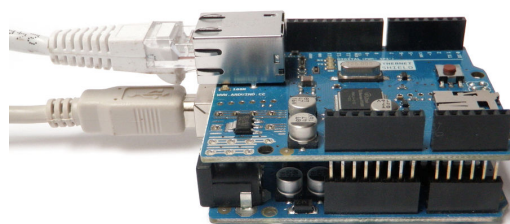
Οι Arduino shields είναι προ-κατασκευασμένες πλακέτες κυκλωμάτων οι οποίες συνδέονται πάνω σε συμβατές πλακέτες arduino. Αυτές παρέχουν πρόσθετες δυνατότητες όπως σύνδεση στο διαδίκτυο, έλεγχος κινητήρα, παροχή ασύρματης επικοινωνίας, έλεγχος οθόνης LCD κ.λπ.

3.2.1 Ethernet shield

Για την συγκεκριμένη πτυχιακή χρησιμοποιήθηκε η Ethernet shield η οποία δίνει την δυνατότητα ενσύρματης σύνδεσης στο διαδίκτυο



Εικόνα 41: σύνδεση shield με arduino



Εικόνα 42: σύνδεση shield με διαδίκτυο (καλώδιο RJ45)

Η Ethernet shield βασίζεται στο τσιπ W5100, το οποίο διαθέτει εσωτερική μνήμη 16K. Έχει ταχύτητα σύνδεσης μέχρι 10 / 100Mb. Στηρίζεται στη βιβλιοθήκη Arduino Ethernet, η οποία συνοδεύεται από το περιβάλλον ανάπτυξης.

Υπάρχει επίσης μια ενσωματωμένη υποδοχή micro SD, η οποία επιτρέπει την αποθήκευση πολλών δεδομένων για την υποστήριξη ακόμα και για ολόκληρους ιστότοπους. Αυτό απαιτεί τη χρήση εξωτερικής βιβλιοθήκης SD, η οποία δεν συνοδεύεται από το

λογισμικό. Επίσης το W5100 παρέχει δίκτυο (IP) ικανό για TCP και για UDP και υποστηρίζει μέχρι και τέσσερις ταυτόχρονες συνδέσεις socket.

Ένα σημαντικό πράγμα που πρέπει να σημειωθεί εδώ είναι ότι τόσο η κάρτα SD όσο και το W5100 μοιράζονται την SPI bus καθώς το Arduino επικοινωνεί μέσω θύρας SPI, οπότε μπορούμε να χρησιμοποιήσουμε μόνο ένα από αυτά. Αν θέλουμε να χρησιμοποιήσουμε και τα δύο, πρέπει να ελέγξουμε την αντίστοιχη βιβλιοθήκη.

Χρησιμοποιούμενες ακίδες για τις πλακέτες uno/mega

Το Arduino επικοινωνεί τόσο με την κάρτα W5100 όσο και με την κάρτα SD χρησιμοποιώντας το δίαυλο SPI (μέσω ICSP).

Αυτό συμβαίνει στις ψηφιακές ακίδες 11, 12 και 13 στο Uno και στις ακίδες 50, 51 και 52 στο Mega. Και στις δύο πλακέτες, ο ακροδέκτης 10 χρησιμοποιείται για την επιλογή του W5100 και ο ακροδέκτης 4 για την κάρτα SD. Αυτές οι καρφίτσες δεν μπορούν να χρησιμοποιηθούν ως είσοδοι έξοδοι. Στο Mega, ο ακροδέκτης υλικού SS 53, δεν χρησιμοποιείται για την επιλογή είτε της κάρτας W5100 είτε της κάρτα SD, αλλά πρέπει να διατηρείται ως έξοδος αλλιώς η διεπαφή SPI δεν λειτουργεί.

Αυτή η shield διαθέτει πολλές ενδεικτικές λυχνίες LED. Οι λειτουργίες αυτών παρατίθενται παρακάτω:

PWR: Για την ένδειξη τροφοδοσίας της πλακέτας.

LINK: Για την ένδειξη της παρουσίας σύνδεσης στο δίκτυο. Αναβοσβήνει όταν λαμβάνονται ή διαβιβάζονται δεδομένα.

FULLD: Για την ένδειξη της σύνδεσης αμφίδρομης σύνδεσης δικτύου

100M: Για σύνδεση δικτύου 100 MB / s

RX: Όταν η shield λαμβάνει δεδομένα αρχίζει να αναβοσβήνει

TX: Όταν η shield στέλνει δεδομένα αρχίζει να αναβοσβήνει

COLL: Αναβοσβήνει στην περίπτωση ανίχνευσης συγκρούσεων δικτύου

3.3 Αισθητήρες και συσκευές δράσης που χρησιμοποιήθηκαν

3.3.1 Mini IR Infrared Pyroelectric PIR Body Motion Human Sensor Detector Module

Όταν ο αισθητήρας εντοπίσει κίνηση εντός της εμβέλειας του θα ενεργοποιήσει την έξοδο του και θα παραμείνει ενεργή μέχρι να περάσει η περίοδος καθυστέρησης και μέχρι να σταματήσει να υπάρχει κίνηση.

Τεχνικές παράμετροι:

1. Τάση λειτουργίας: DC 2.7-12V
2. Στατική κατανάλωση ενέργειας: <0,1mA
3. Χρόνος καθυστέρησης: 2 δευτερόλεπτα
4. Ο χρόνος αποκλεισμού: 2 δευτερόλεπτα
5. Trigger: μπορεί να επαναληφθεί

6. Εύρος ανίχνευσης: $\leq 100^\circ$ γωνία κώνου, 3-5 μέτρα. (απαιτείται ανάλογα με τον φακό)
7. Θερμοκρασία λειτουργίας: -20 - + 60 °C
8. Διαστάσεις PCB: 10mm x 8mm
9. Συνολικό μέγεθος: Περίπου. 12 mm x 25 mm
10. Φακός: Μικρός φακός



Εικόνα 43: Αισθητήρας κίνησης PIR

3.3.2 MQ-135 Gas Sensor Module

Χρησιμοποιούνται σε εξοπλισμούς ελέγχου ποιότητας αέρα για κτίρια / γραφεία και είναι κατάλληλοι για ανίχνευση των NH₃, NO_x, αλκοόλης, βενζολίου, καπνού, CO₂, κλπ.

Specifications:

1. Ένδειξη ισχύος και ένδειξη σήματος εξόδου TTL.
2. Έχει έξοδο σήματος διακόπτη DO (TTL) και έξοδο αναλογικού σήματος AO.
4. Όσο μεγαλύτερη είναι η συγκέντρωση αερίου, τόσο μεγαλύτερη είναι η τάση της αναλογικής εξόδου.
5. Έχει καλή ευαισθησία στην ανίχνευση αερίων και φυσικού αερίου.
6. Διαθέτει τέσσερις οπές βιδών για εύκολη τοποθέτηση.
7. διαστάσεις προϊόντος: 32 (L) * 20 (W) * 22 (H)
8. έχει μεγάλη διάρκεια ζωής και αξιόπιστη σταθερότητα
9. χαρακτηριστικά ταχείας απόκρισης



Εικόνα 44: Αισθητήρας ατμόσφαιρας

Ηλεκτρική απόδοση:

Ανίχνευση αερίου: Ατμοί αμμωνίας / σουλφίδια / βενζόλιο

Ανίχνευση συγκέντρωσης: 10-1000ppm

Τάση εισόδου: DC 5V

Κατανάλωση ρεύματος: 150mA

DO έξοδος: TTL ψηφιακές τιμές 0 και 1 (0,1 και 5V)

AO έξοδος: 0,1-0,3V (σχετικά μη ρυπογόνο), η υψηλότερη τάση συγκέντρωσης είναι περίπου 4V

Χρησιμοποιούμενο τσιπ: AT89S52

Crystal: 11.0592MHZ

Ρυθμός μετάδοσης(Baud) : 9600

Περιβάλλον σύνταξης: Keil

3.3.3 Infrared IR Receiver Module Wireless Remote Control Kit For Arduino

Το kit ασύρματης επικοινωνίας arduino αποτελείται από εξαιρετικά λεπτό τηλεχειριστήριο υπερύθρων και δέκτη υπερύθρων 38KHz. Το τηλεχειριστήριο υπερύθρων διαθέτει 20 πλήκτρα λειτουργιών. Οι απόσταση μετάδοσης είναι έως και 8 μέτρα. Ιδανικό για χειρισμό διάφορων συσκευών σε εσωτερικούς χώρους.

Χαρακτηριστικά:

Απόσταση μετάδοσης: έως 8m (ανάλογα με το περιβάλλον, την ευαισθησία του δέκτη κλπ)
Μπαταρία: Μπαταρία κουμπί CR2025
Χωρητικότητα μπαταρίας: 160mAh
Αποτελεσματική γωνία: 60 °
Υλικό συγκόλλησης: 0,125mmPET
Αποτελεσματική ζωή χρήσης: 20.000 φορές
Στατικό ρεύμα: 3uA - 5uA
Δυναμικό ρεύμα: 3mA - 5mA

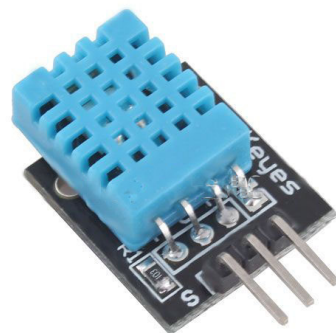


Εικόνα 45: Αισθητήρας υπερύθρων

3.3.4 Temperature and humidity sensor module

Χαρακτηριστικά:

Μοντέλο αισθητήρα: DHT11
Εύρος μετάδοσης σήματος: 20 μέτρα
Εύρος μέτρησης υγρασίας: 20 - 95% RH
Σφάλμα μέτρησης υγρασίας: +/- 5%
Εύρος μέτρησης θερμοκρασίας: 0 - 50 ° C
Σφάλμα μέτρησης θερμοκρασίας: + / -2 ° C
Τάση λειτουργίας: 3.3 - 5V
Τύπος εξόδου: ψηφιακή έξοδος
Μικρές πλάκες PCB μέγεθος: 3.2 x 1.4cm
Ένδειξη LED: κόκκινο
Καθαρό βάρος: 8g



Εικόνα 46: Αισθητήρας θερμοκρασίας υγρασίας

3.3.5 5V Relay TTL Signal 1 Channel Module High Level Expansion Board For Arduino

Περιγραφή:

Λειτουργία ενεργοποίησης: Υψηλό επίπεδο

5V μονάδα ρελέ

Μπορεί να χρησιμοποιηθεί για την ανάπτυξη μικροελεγκτών,

μονάδα ελέγχου ή έλεγχο οικιακών συσκευών

5V - 12V σήμα ελέγχου TTL

Μπορεί να ελέγξει σήμα DC ή AC, επίσης φορτίο 220V AC

Κανονικά ανοικτή / κλειστή επαφή

Ένδειξη λειτουργίας

Οι ακροδέκτες ελέγχου έχουν pull down κύκλωμα για την αποφυγή δυσλειτουργίας του ρελέ



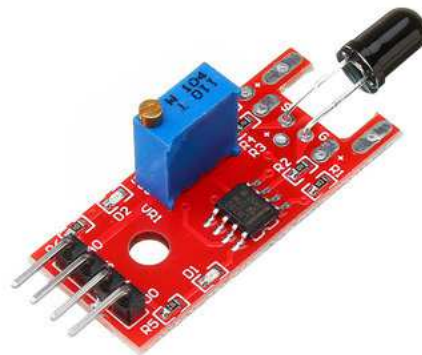
Εικόνα 47: Πλακέτα Ρελέ

3.3.6 Flame Sensor Module IR Sensor Detector

Η ηλεκτρονική μονάδα αισθητήρα φλόγας KY-026 για το Arduino ανιχνεύει το υπέρυθρο φως που εκπέμπεται από τη φωτιά. Η μονάδα διαθέτει τόσο ψηφιακές όσο και αναλογικές εξόδους και ένα ποτενσιόμετρο για τη ρύθμιση της ευαισθησίας. Συνήθως χρησιμοποιείται σε συστήματα ανίχνευσης πυρκαγιάς.

Περιγραφή:

- Δέκτης IR υψηλής ευαισθησίας
 - Εξαιρετικά ευαίσθητο σε κύματα μεταξύ 760-1100nm
 - Με λυχνία ένδειξης τροφοδοσίας ρεύματος και λυχνία ένδειξης εξόδου συγκριτικού
 - ΑΟ, σήμα εξόδου τάσης θερμοστορ σε πραγματικό χρόνο και εξόδους σήματος υψηλής / χαμηλής ηλεκτρικής στάθμης. Παραγωγή αναλογικής ποσότητας
 - Έξοδος ηλεκτρικής στάθμης καταφλίου
Το κατώφλι ρυθμίζεται από το ποτενσιόμετρο
- Εύρος ανίχνευσης: περίπου 60 μοίρες
Τροφοδοσία: 0-15 V DC
Μέγεθος (L x W): 36 x 16 χιλιοστά



Εικόνα 48: Αισθητήρας φωτιάς

3.3.7 DS1302 Real Time Clock Module

Στοιχεία:

Τα ρολόγια πραγματικού χρόνου έχουν τη δυνατότητα να υπολογίζουν δευτερόλεπτα, λεπτά, ώρες, ημέρες, εβδομάδες, μήνες και χρόνια πριν από το έτος 2100, καθώς και τη δυνατότητα προσαρμογής των δίσεκτων χρόνων.

Μνήμη RAM αποθήκευσης δεδομένων 318 bit.

Τάση λειτουργίας ευρείας κλίμακας 2- 5.5V.

Όταν η τάση λειτουργίας είναι 2.0V, το ρεύμα είναι λιγότερο από 300nA.

Όταν διαβάζονται / γράφονται δεδομένα ρολογιού ή RAM, υπάρχουν δύο τρόποι μετάδοσης: μετάδοση ενός byte και μεταφορά συνόλου χαρακτήρων πολλών byte.

Απλό interface 3 γραμμών.

TTL συμβατότητα με $V_{cc} = 5V$.

Προαιρετική βιομηχανική περιοχή θερμοκρασιών -40 +85.



Εικόνα 49: Πλακέτα ρολογιού

Προδιαγραφές:

Μέγεθος PCB πάνελ: 44mm * 23mm * 1.6mm

Η μπαταρία αναμονής είναι γνήσια CR2032, τάσης 3V, ρεύματος 260mAh, μη επαναφορτιζόμενη. Τα θεωρητικά δεδομένα διατηρούνται για περισσότερα από 10 χρόνια.

Με 4 οπές τοποθέτησης, διαμέτρου 3,1mm

Ταλαντωτής κρυστάλλου 32.768KHz, 6pF, μέγεθος 2 * 6mm

DS1302 με 8 πόδια για σύνδεση στο εσωτερικό τσιπ, τσιπ με IC seat, εύκολο να αντικατασταθεί.

Η τάση λειτουργίας: 3.3V / 5V

Θερμοκρασία λειτουργίας: 0 - 70 ° C

3.3.8 HC-SR04 Ultrasonic Module Distance Measuring Transducer

Ο ψηφιακός αισθητήρας υπερήχων δημιουργεί ηχητικά κύματα και διαβάζει την ηχώ τους για τον εντοπισμό και τη μέτρηση της απόστασης από αντικείμενα.

Περιγραφή:

Τάση λειτουργίας: 5V (DC)

Στατικό ρεύμα: Λιγότερο από 2mA.

Σήμα εξόδου: Ηλεκτρικό σήμα συχνότητας, υψηλό 5V, χαμηλό 0V.

Γωνία αισθητήρα: Το πολύ 15 μοίρες.

Απόσταση ανίχνευσης: 2cm-450cm.

Ακρίβεια: Μέχρι 0,3 εκατοστά

Σήμα ενεργοποίησης εισόδου: 10us TTL impulse

Echo Signal: έξοδος σήματος TTL PWL

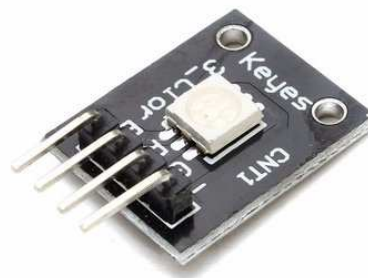


Εικόνα 50: Ultrasonic

3.3.9 3 Colour LED Module

Το συγκεκριμένο module διαθέτει τρία βασικά χρώματα (πράσινο, κόκκινο, μπλε) και μέσω διαμόρφωσης pwm μπορούν να παραχθούν πολλά χρώματα

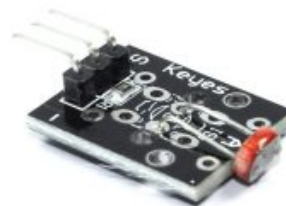
Χρήση πλήρους έγχρωμης λυχνίας LED
RGB τριχρωματική αντίσταση περιορισμού για την πρόληψη της εξουθένωσης(burnout).
Μέσω της ρύθμισης PWM τρία βασικά χρώματα μπορούν να αναμιχθούν σε διαφορετικά χρώματα.
Με μια ποικιλία διεπαφών ενός τσιπ.
Τάση λειτουργίας: 5V
LED drive mode: common cathode driver



Εικόνα 51: Πλακέτα RGB

3.3.10 KY-018 PHOTORESISTOR MODULE

Αυτή η μονάδα αποτελείται από μία φωτοαντίσταση και μία αντίσταση 10 kΩ συνδεδεμένα σε σειρά. Η αντίσταση της φωτοαντίστασης μειώνεται όσο αυξάνεται το φως ενώ μειώνεται όταν το φως μειώνεται. Η αναλογική έξοδος καθορίζει την ένταση του φωτός



Εικόνα 52: Πλακέτα φωτοαντίστασης

3.3.11 Soil moisture detector module

Χαρακτηριστικά

Τάση λειτουργίας: 3.3V ~ 5V.

Ρυθμιζόμενη ευαισθησία (μπλε ποτενσιόμετρο)

Λειτουργία διπλής εξόδου, αναλογική έξοδος ακριβέστερη.

Με μία οπή για εύκολη εγκατάσταση.

Δείκτης λειτουργίας (κόκκινο) και την ένδειξη ψηφιακής εξόδου (πράσινο).

Τσιπ σύγκρισης: LM393

Διάσταση PCB: 3cm x 1.5cm.

Διάσταση probe εδάφους: 6cm x 2cm.

Μήκος καλωδίου: 21εκ.

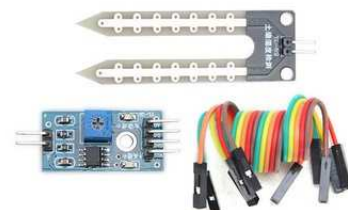
Pins:

VCC: 3.3V-5V.

GND: GND.

DO: Διεπαφή ψηφιακής εξόδου (0 και 1).

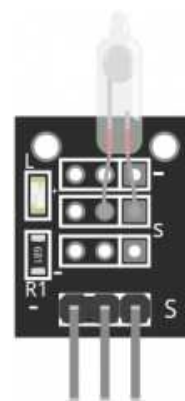
AO: Διεπαφή αναλογικής εξόδου.



Εικόνα 53: Αισθητήρας υγρασίας χώματος

3.3.12 KY-017 Mercury Tilt Switch Module

Ο KY-017 διακόπτης υδραργύρου χρησιμοποιεί μία μπάλα υδραργύρου για να εντοπίσει αλλαγή στην κλίση. Όταν ο υδράργυρος αλλάξει θέση τότε ο αισθητήρας στέλνει λογικό 1 στην έξοδό του και επίσης ανάβει η ενσωματωμένη προειδοποιητική λυχνία led.



Εικόνα 54: Αισθητήρας κλίσης(υδραργύρου)

3.3.13 Water Level Sensor

Προδιαγραφές

Τάση λειτουργίας: 5V DC

Ρεύμα λειτουργίας: λιγότερο από 35mA

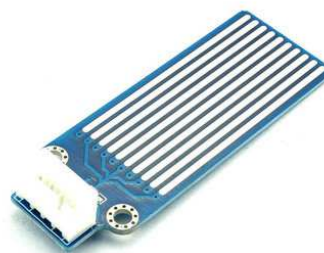
Τύπος: Αναλογικός αισθητήρα

Περιοχή δοκιμής: 40 mm x 18 mm

Θερμοκρασία λειτουργίας: 10 °C έως 30 °C

Υγρασία λειτουργίας: 10% - 90%

Μέγεθος οπών στερέωσης: Φ3mm
Μέγεθος προϊόντος: 60mm x 22mm
Παράδειγμα Εφαρμογής: Σχεδιασμός συναγερμού
στάθμης νερού



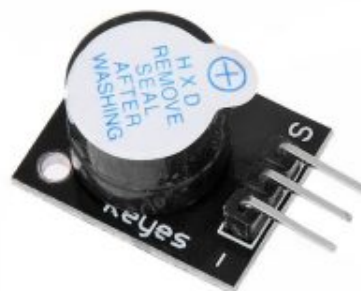
Εικόνα 55: Αισθητήρας νερού

3.3.14 KY-012 Active Buzzer Module

Ο ενεργός πιεζοηλεκτρικός βομβητής KY-012 παράγει έναν μονοτονικό ήχο όταν δέχεται ψηφιακό σήμα 1. Επίσης ο αντίστοιχος παθητικός βομβητής μπορεί να παράγει πολυτονικούς ήχους.

Προδιαγραφές:

Τάση λειτουργίας: 3.5V ~ 5.5V
Μέγιστο ρεύμα: 30mA / 5VDC
Συχνότητα συντονισμού: 2500Hz ± 300Hz
Θερμοκρασία λειτουργίας: -20 ° C ~ 70 ° C
Θερμοκρασία αποθήκευσης: -30 ° C ~ 105 ° C
Διαστάσεις: 18.5mm x 15mm



Εικόνα 56: Πλακέτα Buzzer

3.3.15 DFRobot Capacitive Touch Sensor

Ο χωρητικός αισθητήρας αφής μπορεί να αισθάνεται την ανθρώπινη αφή ή τα μέταλλα και να στείλει στην έξοδο του σήμα 1 ή 0 αντίστοιχα. έχει πολύ μεγάλη ευαισθησία καθώς μπορεί να ανιχνεύσει αφή και μέσα από κάποιο ύφασμα.

Χαρακτηριστικά:

Τάση τροφοδοσίας: 3.3 V- 5V
Ψηφιακή έξοδος
Μέγεθος: 22x 30 mm



Εικόνα 57: Αισθητήρας αφής

3.3.16 Fan 60X60X25 EB60251S1 SUNON

Τεχνικά Χαρακτηριστικά

- Κωδικός εργοστασίου : 01.031.0160
- Διαστάσεις: 60x60x25 mm
- Κίνηση ανεμιστήρα : ρουλεμάν απλό
- Κατανάλωση ισχύος : 1.62W
- Τάση : 12V DC
- Ένταση ρεύματος : 0.135 A
- Στροφές Ανεμιστήρα (Max) : 4.500 RPM
- Επίπεδα Θορύβου (Max) : 33.5 dBA
- Ροή Αέρα (Max) : 23.5 CFM
- Βάρος : 57.0 gr



Εικόνα 58: Ανεμιστήρας DC

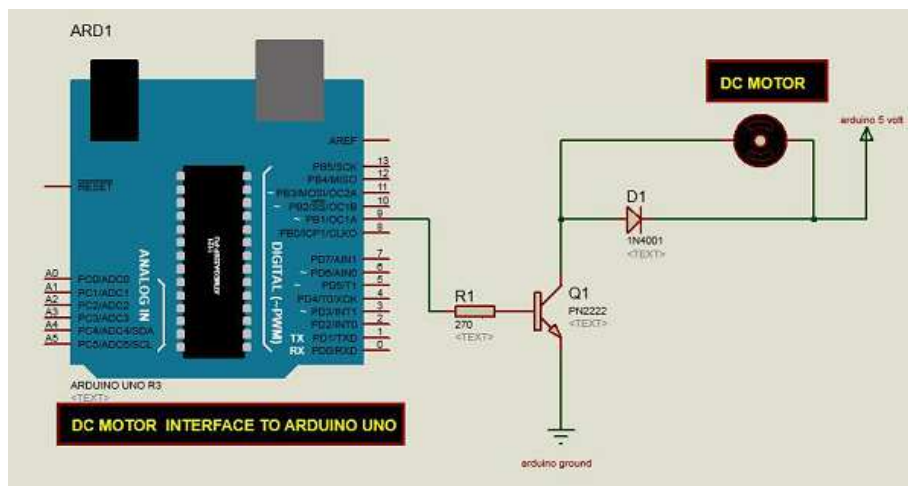
3.3.17 DC Mini Motor

Προδιαγραφές:

Μέγεθος κινητήρα: 38x20x15mm
Διάμετρος άξονα: 2mm
Αξονικό μήκος: 9mm
Τάση: 3-6V
Ρεύμα (στάση): 600mA
Ταχύτητα: 18000RPM
Ροπή: 0,55kg.cm



Εικόνα 59: DC μοτέρ



Εικόνα 60: Λειτουργία DC μοτέρ

Για την συνδεσμολογία του μοτέρ και του ανεμιστήρα δεν αρκεί απλά να συνδεθούν στην έξοδο του arduino και στη γείωση γιατί αυτό μπορεί να καταστρέψει την πλακέτα. Για αυτόν τον λόγο χρειάζεται ένα κύκλωμα οδήγησης (παραπάνω εικόνα)

Τα στοιχεία που χρησιμοποιήθηκαν για το κύκλωμα είναι:

- 1x PN2222 Transistor
- 1x 1N4001 diode
- 1x 270 Ω Resistor

Το τρανζίστορ λειτουργεί ως διακόπτης για την τροφοδοσία του μοτέρ.

Επίσης τοποθετείται η δίοδος παράλληλα στο μοτέρ. Αυτό γίνεται ώστε αν δημιουργηθεί αντίθετο ρεύμα από το μοτέρ να περάσει μέσα από αυτό και να μην κατευθυνθεί προς το arduino ή το τρανζίστορ.

3.3.18 MPU-6050 module

Ο αισθητήρας MPU-6050 περιέχει ένα επιταχυνσιόμετρο MEMS και ένα γυροσκόπιο MEMS σε ένα ενιαίο τσιπ. Είναι πολύ ακριβής, αφού περιέχει υλικό 16-bit μετατροπής αναλογικό σε ψηφιακό, ανά κανάλι. Για αυτό, συλλαμβάνει ταυτόχρονα τα κανάλια x, y και z.

Εφαρμογή:

Παιχνίδια ανίχνευσης κίνησης

Εικονική πραγματικότητα

Ηλεκτρονική σταθεροποίηση εικόνας (EIS: Electronic Image Stabilization)

Οπτική σταθεροποίηση εικόνας (OIS: Optical Image Stabilization)

Πλοήγηση πεζών

Συντομεύσεις χειρονομίας

Προδιαγραφές:

16bit AD converter -chip, έξοδος δεδομένων 16-bit

Χρησιμοποιούμενο τσιπ: MPU-6050

Τροφοδοσία: 3-5v (εσωτερικός ρυθμιστής χαμηλής αποκοπής)

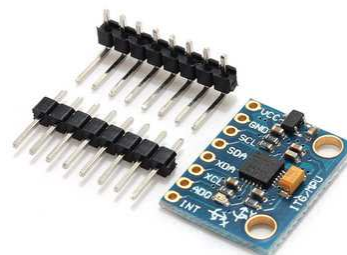
Επικοινωνία: Πρότυπο πρωτόκολλο επικοινωνίας IIC

Εύρος γυροσκόπιου: ± 250 500 1000 2000 ° / s

Εύρος επιταχυνσιόμετρου: ± 2 ± 4 ± 8 ± 16 g

Gold PCB

Μέγεθος: 2 x 1,6 x 0,1 cm



Εικόνα 61: Επιταχυνσιόμετρο MPU-6050

3.3.19 Geekcreit® 1.3 Inch 4Pin White OLED LCD Display 12864 IIC I2C Interface Module For Arduino

Χαρακτηριστικά:

Ανάλυση: 128 * 64
Τσιπ ελέγχου: SSH1106
Περιοχή εμφάνισης: 29,42 x 14,7 mm
Τάση οδήγησης: 3.3-5V
Θερμοκρασία λειτουργίας: -40 C έως 70 C
Τύπος διεπαφής: Διεπαφή IIC / I2C

Ορισμοί ακίδων:

GND (Γείωση)
VCC (Τροφοδοσία)
SCL (Γραμμή ρολογιού)
SDA (Γραμμή δεδομένων)



Εικόνα 62: Οθόνη OLED display

3.3.20 PulseSensor Pulse Heart Rate Sensor Module For Arduino Pulse Sensor

Ο αισθητήρας παλμών είναι ένας αισθητήρας που χρησιμοποιείται για τον έλεγχο της καρδιακής συχνότητας. Μαθητές, καλλιτέχνες, αθλητές, δημιουργοί, παιχνιδιών ή προγραμματιστές κινητών μπορούν να αναπτύξουν διαδραστικές εργασίες που σχετίζονται με τον καρδιακό ρυθμό. Ο αισθητήρας μπορεί να φορεθεί σε ένα δάχτυλο ή στο λοβό και να συνδεθεί με το Arduino μέσω ενός καλωδίου. Έχει επίσης μια εφαρμογή ανοιχτού κώδικα που εμφανίζει τον καρδιακό ρυθμό σε πραγματικό χρόνο. Στην ουσία είναι ένας αισθητήρας οπτικού καρδιακού ρυθμού ο οποίος ενσωματώνει ένα κύκλωμα ενίσχυσης και ένα κύκλωμα ακύρωσης θορύβου.

Τάση τροφοδοσίας: 3V ή 5V.



Εικόνα 63: Αισθητήρας καρδιακών παλμών

3.3.21 Esp8266 Wifi Module

Η ηλεκτρονική μονάδα ESP8266 WiFi είναι μια αυτόνομη SOC με ενσωματωμένη στοίβα πρωτοκόλλων TCP / IP που μπορεί να δώσει σε κάθε μικροελεγκτή πρόσβαση σε κάποιο WiFi δίκτυο.



Εικόνα 64: ESP-8266 Wifi

ΠΡΟΔΙΑΓΡΑΦΕΣ

- 802,11 b / g / n
- Wi-Fi Direct (P2P), soft-AP
- Ενσωματωμένη στοίβα πρωτοκόλλων TCP / IP
- Ενσωματωμένος διακόπτης TR, balun, LNA, ενισχυτής ισχύος και δίκτυο που ταιριάζει
- Ενσωματωμένα PLL, ρυθμιστές, DCXO και μονάδες διαχείρισης ενέργειας
- + Ισχύς εξόδου 19,5dBm σε λειτουργία 802.11b
- Απενεργοποίηση για ρεύμα διαρροής <10uA
- Η ενσωματωμένη CPU 32 bit με χαμηλή ισχύ μπορεί να χρησιμοποιηθεί ως επεξεργαστής εφαρμογών
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1 × 1 MIMO, 2 × 1 MIMO
- Συσσωμάτωση A-MPDU & A-MSDU & διάστημα προστασίας 0,4ms
- Αφύπνιση και μετάδοση πακέτων σε <2ms
- Κατανάλωση ισχύος σε αναμονή <1.0mW (DTIM3)

Κεφάλαιο 4: Σενάρια αυτοματισμών και κύκλωμα

Το κεφάλαιο αυτό αποτελείται από τα σενάρια αυτοματισμών που υλοποιήθηκαν και τα κυκλώματα τους. Θα αναλυθεί η χρησιμότητα του κάθε σεναρίου στην καθημερινότητα και ο κώδικας που χρησιμοποιήθηκε για κάθε σενάριο ξεχωριστά. Επίσης η σχεδίαση του τελικού κυκλώματος και των επιμέρους κυκλωμάτων επιτεύχθηκε με τη βοήθεια του προγράμματος fritzing. Το κεφάλαιο απαρτίζεται από οχτώ σενάρια για το έξυπνο σπίτι (επτά τρέχουν στο Arduino Mega και ένα στο Arduino Uno) και ένα σενάριο για το βραχιόλι έκτακτης ανάγκης για ηλικιωμένους(τρέχει στο δικό του Arduino Uno).

Τα σενάρια αυτά είναι τα παρακάτω:

- Σενάριο ασφάλειας εξωτερικής πόρτας.
- Σενάριο αυτόματης αντλίας ανελκυστήρα.
- Σενάριο αυτόματου ποτίσματος.
- Σενάριο ελέγχου γκαραζόπορτας.
- Σενάριο ελέγχου κλιματισμού και ποιότητας αέρα.
- Σενάριο ανίχνευσης φωτιάς και πλημμύρας.
- Σενάριο ελέγχου παντζουριών και εξωτερικού φωτισμού.
- Σενάριο ελέγχου φωτισμού αίθουσας ψυχαγωγίας.
- Σενάριο βραχιολιού έκτακτης ανάγκης για ηλικιωμένους ή άτομων με ιατρικές παθήσεις.

4.1 Σενάριο ασφάλειας εξωτερικής πόρτας

Η ασφάλεια της εξωτερικής πόρτας πραγματοποιήθηκε με τον συνδυασμό μυστικού κωδικού και με το πάτημα προκαθορισμένης χρονικής διάρκειας του αισθητήρα αφής. Το σενάριο αυτό υλοποιήθηκε για την διευκόλυνση του χρήστη στο ξεκλείδωμα της εξωτερικής πόρτας χωρίς κλειδιά.

Οι αισθητήρες που χρησιμοποιήθηκαν για την ασφάλεια εξωτερικής πόρτας είναι οι παρακάτω:

1. Mini IR Infrared Pyroelectric PIR Body Motion Human Sensor Detector Module
2. Infrared IR Receiver Module Wireless Remote Control Kit For Arduino
3. DFRobot Capacitive Touch Sensor
4. 1 x Button
5. 4 x Leds (φωτισμού, αισθητήρα αφής, κωδικού, πόρτας)

Η βιβλιοθήκη που χρειάζεται για την λειτουργία του σεναρίου είναι η IRremote.h.

Στην συνέχεια θα επεξηγηθεί η λειτουργία του σεναρίου.

Αρχικά έχει χρησιμοποιηθεί ο αισθητήρας pir για την διευκόλυνση του χρήστη σε συνθήκες έλλειψης φωτισμού στον χώρο της εξωτερικής πόρτας. Όταν ο αισθητήρας εντοπίσει κίνηση ανάβει το led φωτισμού και παραμένει αναμμένο για όση ώρα βλέπει κίνηση. Θα σβήσει δύο δευτερόλεπτα αφού σταματήσει η κίνηση στον χώρο.

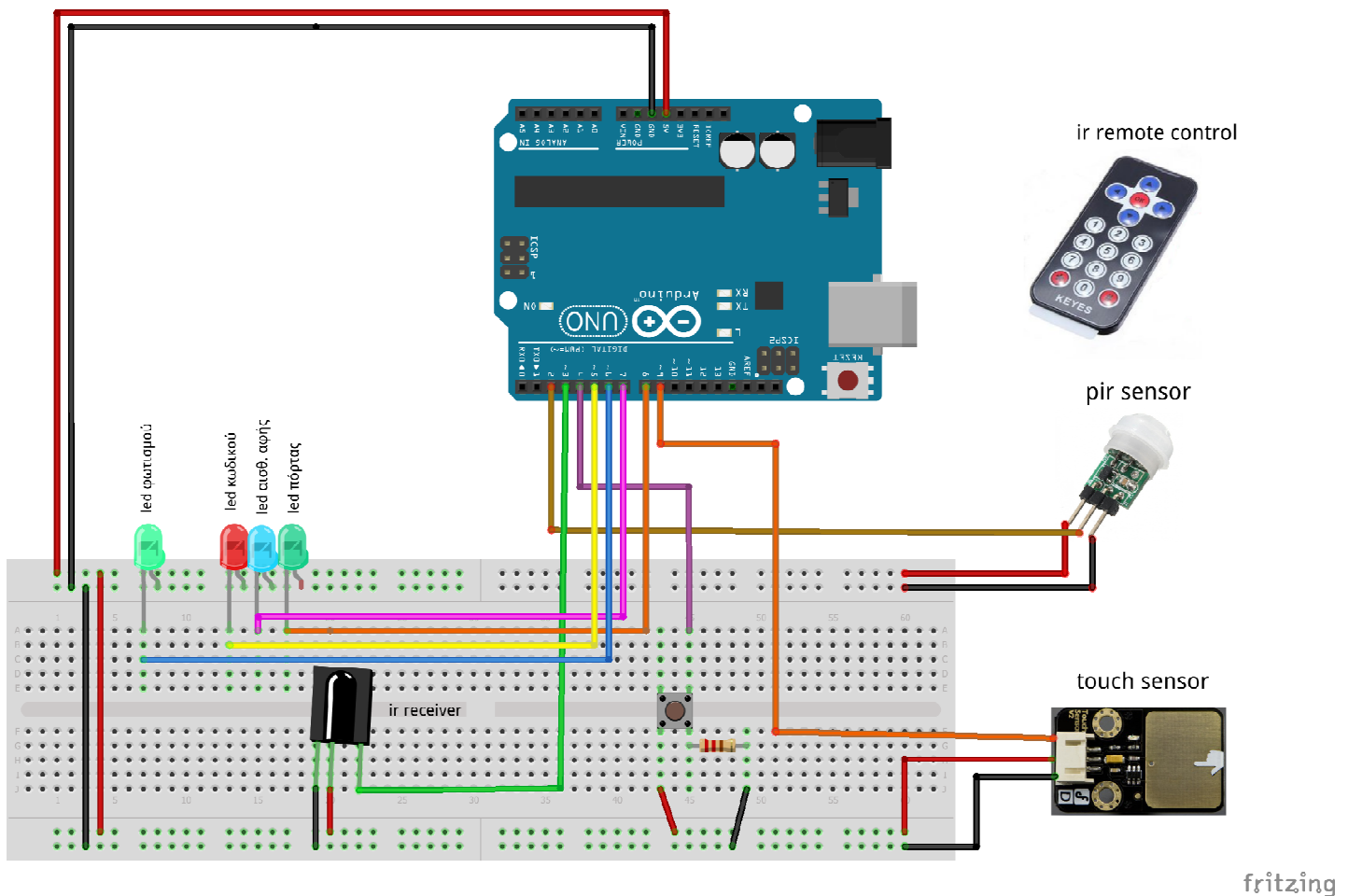
Έπειτα έχει τοποθετηθεί το Button για να υποδηλώνει το αν είναι ανοιχτή ή κλειστή η πόρτα. Όταν το button είναι πατημένο η πόρτα είναι κλειστή.

Για να λειτουργήσει το αυτόματο σενάριο θα πρέπει η εξωτερική πόρτα να είναι κλειστή. Προκειμένου να ξεκλειδώσει η πόρτα πρέπει να ισχύουν συγχρόνως δύο συνθήκες (λογικό και).

1. Ο χρήστης να πατήσει τον αισθητήρα αφής για ένα συγκεκριμένο χρονικό διάστημα (π.χ. 2 δευτερόλεπτα) ώστε να ενεργοποιηθεί η πρώτη συνθήκη. Σε περίπτωση που ο αισθητήρας πατηθεί για λιγότερο ή περισσότερο χρόνο η συνθήκη απενεργοποιείται. Για την διευκόλυνση του πειράματος έχει τοποθετηθεί ενδεικτική λυχνία led κωδικού (κόκκινο) η οποία δείχνει πότε επιτυγχάνουμε τον χρόνο. Στην πραγματικότητα δεν θα υπάρχει αυτό το led.
2. Επίσης με την χρήση του τηλεχειριστήριου να πατηθεί ένας τριψήφιος κωδικός ο οποίος ενεργοποιεί την δεύτερη συνθήκη και ενεργοποιείται συγχρόνως και η δεύτερη λυχνία led αισθ. αφής (μπλε).

Τέλος έχει οριστεί ένα χρονικό όριο είκοσι δευτερολέπτων μέσα στο οποίο θα πρέπει ο χρήστης να ενεργοποιήσει τις δύο συνθήκες για να ανοίξει η πόρτα. Αν μία από τις δύο δεν είναι σωστή η διαδικασία ξεκινάει από την αρχή. Αν το πρόγραμμα δει και τις δύο συνθήκες σωστές τότε η πόρτα ξεκλειδώνει και ενεργοποιείται η τρίτη λυχνία led πόρτας(πράσινο).

Το κύκλωμα για το σενάριο ασφάλειας εξωτερικής πόρτας είναι το παρακάτω:



Εικόνα 65: Σχεδίαση σεναρίου ασφάλειας με το fritzing

Ο κώδικας για το σενάριο ασφάλειας εξωτερικής πόρτας (τρέχει στο Arduino Uno του έξυπνου σπιτιού):

```
//asfaleia
int led1 = 6;           // pin led φωτισμού
int sensor = 2;        // pir sensor pin
int state = LOW;       // by default, no motion detected
int val = 0;           // variable to store the sensor status
                        (value)
#define ctsPin 9        // touch sensor pin
int flag1=0;
int flag=0;
int counter=0;
int led2=7;            // pin led αισθ. αφής
int led3=8;            // pin led πόρτας
int period = 10000;
unsigned long time_now = 0;
int led=5;             // pin led κωδικού
int a=0;
int b=0;
int c=0;
const int buttonPin = 4; // pin button
int buttonState = 0;
#include <IRremote.h>    // βιβλιοθήκη IR
int RECV_PIN = 3;       // pin ir receiver

IRrecv irrecv(RECV_PIN);

decode_results results;
void setup()
{
  pinMode(led1, OUTPUT); // initialize LED as an output
  //asfaleia
  pinMode(sensor, INPUT); // initialize sensor as an input

  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(ctsPin, INPUT);

  pinMode(led,OUTPUT);

  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}
void loop()
{
  if (irrecv.decode(&results)) //asfaleia
  {
    Serial.println(results.value, HEX);
    translateIR();
    for (int z=0; z<2; z++) // ignore 2nd and 3rd signal repeat
    {
      irrecv.resume(); // receive the next value
    }
  }
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    if (millis() > (time_now + period)){
      if (flag+flag1==2) digitalWrite(led3,HIGH);
    }
  }
}
```



```

    else digitalWrite(led3,LOW);
    counter=0;

Serial.println("-----");
    time_now = millis();
    a=0; b=0; c=0;
    flag1=0;
    digitalWrite(led,LOW);
    }
    else{
        if (digitalRead(ctsPin)==HIGH) counter=counter+1;

        if (counter<5 &&counter>1 ){
            digitalWrite(led2,HIGH);
            flag=1;
        }
        else {
            flag=0;
            digitalWrite(led2,LOW);}

        if (a+b+c==3){
            digitalWrite(led,HIGH);
            flag1=1;
        }
    }
}

Serial.println(counter);

val = digitalRead(sensor); // read sensor value
if (val == HIGH) {        // check if the sensor is HIGH
    digitalWrite(led1, HIGH); // turn LED ON
    delay(100);             // delay 100 milliseconds

    if (state == LOW) {
        Serial.println("Motion detected!");
        state = HIGH;      // update variable state to HIGH
    }
}
else {
    digitalWrite(led1, LOW); // turn LED OFF
    delay(200);             // delay 200 milliseconds

    if (state == HIGH){
        Serial.println("Motion stopped!");
        state = LOW;      // update variable state to LOW
    }
}
}

void translateIR() // takes action based on IR code received
// describing Sony IR codes on LCD module
{
switch(results.value)
{
case 0xFFA25D:

a=1;
break;
case 0xFF629D:

b=1;
break;

```

```
case 0xFFE21D:

c=1;
break;

default:
Serial.println("OTHER");
}
delay(200);
}
```

4.2 Σενάριο αυτόματης αντλίας ανελκυστήρα

Η αυτόματη αντλία ανελκυστήρα έχει μεγάλη χρησιμότητα για τα νερά που μαζεύονται στα φρεάτια των ανελκυστήρων λόγω διαφόρων διαρροών. Αυτό το πρόβλημα μπορεί να γίνει πολύ χειρότερο τους χειμερινούς μήνες που υπάρχουν έντονες βροχοπτώσεις. Για αυτό το λόγο και για την σωστή λειτουργία του ανελκυστήρα είναι απαραίτητο να γίνεται άντληση των νερών και έλεγχος της στάθμης τους.

Οι αισθητήρες που χρησιμοποιήθηκαν για την αυτόματη αντλία ανελκυστήρα είναι:

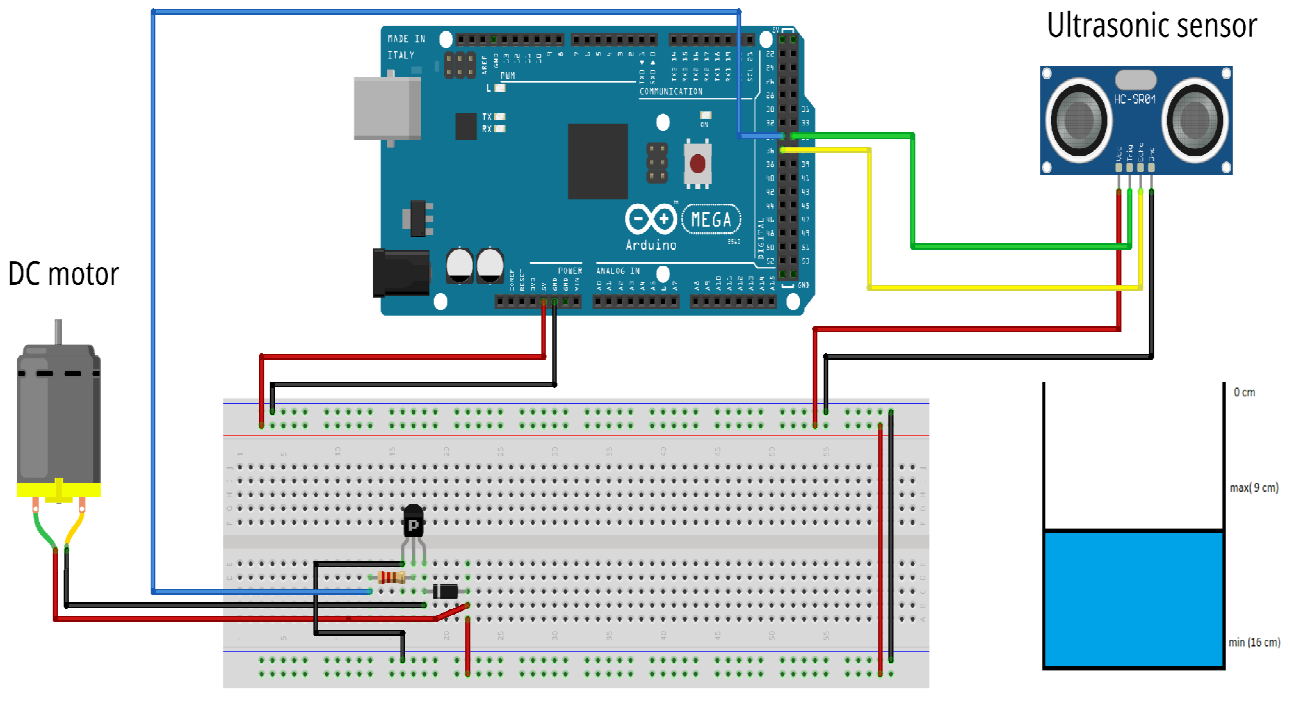
1. HC-SR04 Ultrasonic Module Distance Measuring Transducer
2. DC Mini Motor
3. 1x PN2222 Transistor
4. 1x N4001 diode
5. 1x 270 Ω Resistor

Το συγκεκριμένο σενάριο δεν χρειάζεται κάποια βιβλιοθήκη για την λειτουργία του.

Η λειτουργία του σεναρίου είναι η εξής:

Ο αισθητήρας ultrasonic μετράει την απόσταση από το σημείο που εκπέμπει μέχρι το σημείο του αντικειμένου(στην περίπτωση μας το νερό του φρεατίου) Όταν ο αισθητήρας ultrasonic εντοπίσει στάθμη νερού μικρότερη ή ίση με 9 cm(πάνω από το max) τότε θα ενεργοποιηθεί το dc motor (αντλία) και θα συνεχίσει να λειτουργεί μέχρι η στάθμη του νερού να γίνει μεγαλύτερη ή ίση με 16cm (κάτω από το min) .

Το κύκλωμα για το σενάριο αυτόματης αντλίας ανελκυστήρα είναι το εξής:



Εικόνα 66: Σχεδίαση αυτόματης αντλίας με το fritzing

Ο κώδικας για το σενάριο αυτόματης αντλίας ανελκυστήρα(τρέχει στο Arduino Mega του έξυπνου σπιτιού):

```
int ANTLIA=34;           // dc motor pin           //asanser
#define trigPin 35      // ultrasonic trigger pin
#define echoPin 36     // ultrasonic echo pin
int flag=0;

void setup()
{
  pinMode(ANTLIA,OUTPUT);           //asanser
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop()
{
  int antlia =digitalRead(ANTLIA); //asanser
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
```

```

if (distance <=9) flag=1;
if (distance >=16) flag=0;
if (flag==1)
{
  do{
    digitalWrite (ANTLIA,HIGH) ;
    Serial.println("antlia on"); }
  while (distance>=16) ;
}
else
{
  Serial.println("antlia off");
  digitalWrite (ANTLIA,LOW) ;
}
}

```

4.3 Σενάριο αυτόματου ποτίσματος

Το αυτόματο πότισμα είναι πολύ χρήσιμο στην καθημερινότητα καθώς απαλλάσσει τον χρήστη από την έννοια του ποτίσματος και ειδικά σε φυτά που χρειάζονται καθημερινό πότισμα. Επίσης μπορεί να διατηρήσει την υγρασία του χώματος σε επιθυμητά επίπεδα, ανάλογα με τις απαιτήσεις του κάθε φυτού. Τέλος μπορεί να ρυθμιστεί ώστε να ποτίζει τις ώρες που δεν υπάρχει έντονη ηλιοφάνεια για την προστασία των φυτών.

Οι αισθητήρες που χρησιμοποιήθηκαν για το αυτόματο πότισμα είναι:

1. Soil moisture detector module
2. KY-018 PHOTORESISTOR MODULE
3. 1 x led (ηλεκτροβόνα)

Το συγκεκριμένο σενάριο δεν χρειάζεται κάποια βιβλιοθήκη για την λειτουργία του.

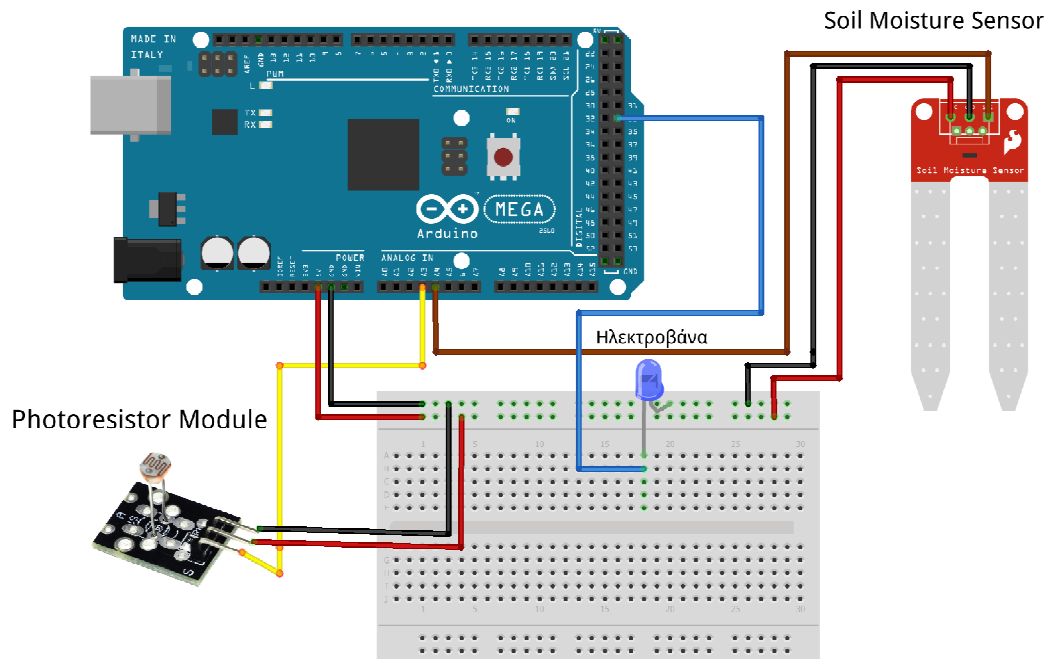
Λειτουργία αυτόματου σεναρίου:

Για την ενεργοποίηση του αυτόματου ποτίσματος (η μπλε λυχνία led υποδηλώνει την ηλεκτροβόνα) χρειάζεται να ισχύουν δύο συνθήκες(λογικό και) :

1. Διαβάζεται η τιμή της υγρασίας του χώματος από τον αισθητήρα soil moisture sensor. Αν εντοπιστεί ποσοστό υγρασίας μικρότερο του 40% ή (or) μικρότερο του 80% ενεργοποιείται η πρώτη συνθήκη. Άρα θα ενεργοποιήσει το αυτόματο πότισμα μόλις εντοπίσει υγρασία <40% και θα ποτίζει μέχρι να φτάσει =80%. Με αυτόν τον τρόπο διατηρούμε την υγρασία σε επιθυμητό επίπεδο(ούτε πολύ υγρό ούτε πολύ ξερό)
2. Διαβάζεται η τιμή της ηλιοφάνειας με τον αισθητήρα φωτοαντίστασης. Η τιμή πρέπει να είναι μικρότερη από 60% για να ισχύει η δεύτερη συνθήκη.

Αυτό γίνεται για να αποτρέψουμε το πότισμα κατά την έντονη ηλιοφάνεια, το οποίο μπορεί να προκαλέσει ασθένειες στα φυτά.

Το κύκλωμα που υλοποιήθηκε για το σενάριο αυτόματου ποτίσματος είναι το παρακάτω:



fritzing

Εικόνα 67: Σχεδίαση αυτόματου ποτίσματος με το fritzing

Ο κώδικας για το σενάριο αυτόματου ποτίσματος(τρέχει στο Arduino Mega του έξυπνου σπιτιού):

```
int sensorPin2 = A4; // soil moisture sensor pin //potisma
int sensorPin1 = A3; // photoresistor module pin
int potisma=33; // pin led ηλεκτροβάνας
int water_value ;
int sun_Value1 = 0; // variable to store the value coming from the
sensor

void setup()
{
pinMode(potisma,OUTPUT);
}
void loop()
{
water_value= analogRead(sensorPin1); // potisma
sun_Value1 = analogRead(sensorPin2);
water_value = map(water_value,350,1023,120,0);
sun_Value1 = map(sun_Value1,15,800,100,0);

if(sun_Value1<60 && (water_value<40 ||water_value<80 ))

{digitalWrite(potisma,HIGH); }

else
{digitalWrite(potisma,LOW); }
```

```

Serial.println(water_value);
Serial.println(sun_Value1);
}

```

4.4 Σενάριο ελέγχου γκαραζόπορτας

Ο έλεγχος της γκαραζόπορτας πραγματοποιήθηκε με τον αισθητήρα κλίσης (υδραργύρου). Το σενάριο αυτό υλοποιήθηκε για την έγκαιρη προειδοποίηση του χρήστη τότε η γκαραζόπορτα είναι ανοιχτή ή όχι. Η προειδοποίηση του χρήστη γίνεται μέσω της ηχητικής ένδειξης με την χρήση του Buzzer.

Οι αισθητήρες που χρησιμοποιήθηκαν για τον έλεγχο της γκαραζόπορτας είναι οι παρακάτω:

1. KY-017 Mercury Tilt Switch Module
2. 1x Buzzer
3. 2x Leds (garaze open, garaze closed)

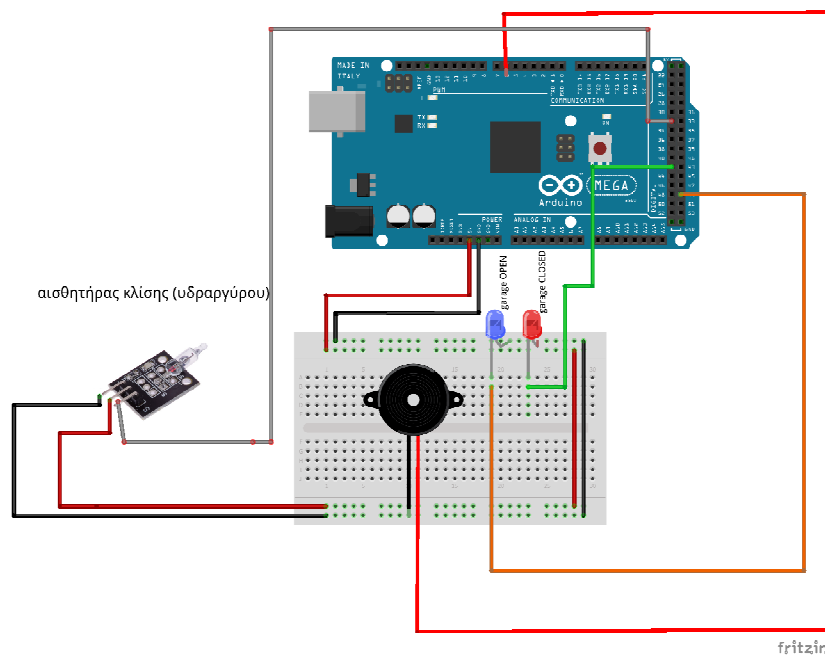
Το συγκεκριμένο σενάριο δεν χρειάζεται κάποια βιβλιοθήκη για την λειτουργία του.

Στη συνέχεια θα επεξηγηθεί η λειτουργία του σεναρίου.

Στο σενάριο ελέγχου της γκαραζόπορτας ο αισθητήρας κλίσης (υδραργύρου) λειτουργεί με τον εξής τρόπο:

1. Όταν η γκαραζόπορτα είναι κλειστή, τότε ο αισθητήρας θα πάρει την τιμή λογικό 0 και θα ανάψει η λυχνία led (κόκκινο).
2. Όταν η γκαραζόπορτα παραμείνει ανοιχτή από λάθος του χρήστη ή επιχειρήσει κάποιος να την παραβιάσει, τότε ο αισθητήρας θα πάρει την τιμή λογικό 1, θα ανάψει η λυχνία led (μπλέ) και θα ενεργοποιηθεί η ηχητική ένδειξη (Buzzer).

Το κύκλωμα του σεναρίου ελέγχου γκαραζόπορτας:



Εικόνα 68: Σχεδίαση ελέγχου γκαραζόπορτας με το fritzing

Ο κώδικας του σεναρίου για τον έλεγχο της γκαραζόπορτας(τρέχει στο Arduino Mega του έξυπνου σπιτιού):

```
int tiltSwitch = 32; // pin Tilt Switch
int garage;        // variable to store tilt input
int buzzer=6;      // pin buzzer
int greenled1=49;  // pin led garage open
int redled1=42;    // pin led garage closed
void setup ()
{
  pinMode(buzzer,OUTPUT);
  pinMode(greenled,OUTPUT);
  pinMode(redled,OUTPUT);
  Serial.begin(9600);

  pinMode (tiltSwitch, INPUT) ;// define the mercury tilt switch
  sensor output interface
}
void loop ()
{
  garage = digitalRead (tiltSwitch) ;// read the switch value
  if (garage == LOW) // Means we've tilted
  {
    digitalWrite(buzzer,HIGH);
    delay(100);
    digitalWrite(buzzer,LOW);
    delay(100);
    digitalWrite(redled1,HIGH);
    digitalWrite(greenled1,LOW);
    Serial.println("*garage open");
  }

  else
  {
    digitalWrite(greenled1,HIGH);
    digitalWrite(redled1,LOW);
    digitalWrite(buzzer,LOW);
    Serial.println("garage closed");
  }
  delay(1000);
}
```

4.5 Σενάριο ελέγχου κλιματισμού και ποιότητας αέρα

Το συγκεκριμένο σενάριο πραγματοποιήθηκε για τον έλεγχο της θερμοκρασίας μέσα στο σπίτι και για τον έλεγχο της ποιότητας του αέρα. Πιο συγκεκριμένα είναι επιθυμητό να υπάρχει ένα συγκεκριμένο εύρος θερμοκρασίας για καλύτερες συνθήκες διαβίωσης και ειδικά σε περίπτωση παρουσίας ηλικιωμένων ατόμων όπου δεν είναι εύκολος ο χειροκίνητος έλεγχος των συσκευών.

Επίσης έχει προστεθεί σύστημα εξαερισμού με την βοήθεια αισθητήρα επιβλαβών αερίων και επικίνδυνων ουσιών, προκειμένου να υπάρχει όσο το δυνατόν καλύτερη ποιότητα αέρα και καθαρής ατμόσφαιρας, το οποίο βοηθάει στην αποφυγή αναπνευστικών προβλημάτων και σε πιο ευνοϊκές συνθήκες διαβίωσης.

Οι αισθητήρες που χρησιμοποιήθηκαν για τον έλεγχο της γκαραζόπορτας είναι οι παρακάτω:

1. MQ-135 Gas Sensor Module
2. Temperature and humidity sensor module
3. 16 Fan 60X60X25 EB60251S1 SUNON
4. 1x PN2222 Transistor
5. 1x N4001 diode
6. 2x 270 Ω Resistor
7. 1 x Button
8. 1 x Led (κλιματιστικό)

Η βιβλιοθήκη που χρειάζεται για την λειτουργία του σεναρίου είναι η dht.h.

Λειτουργία αυτόματου σεναρίου:

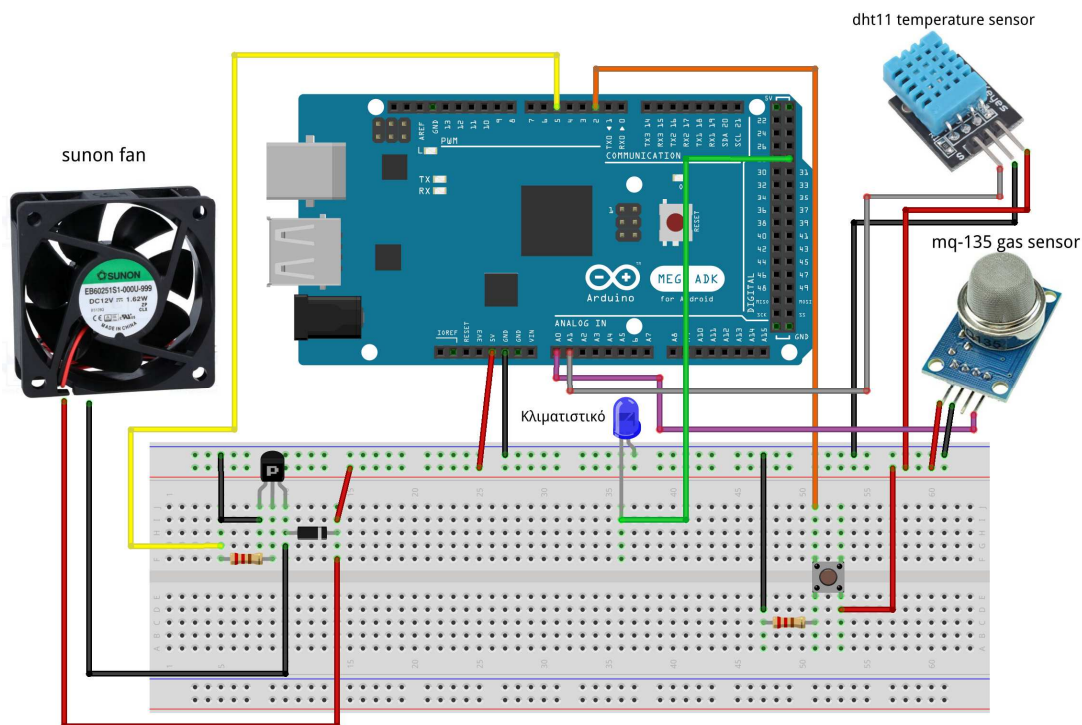
Για το σύστημα ποιότητας αέρα έχει χρησιμοποιηθεί ένας αισθητήρας ανίχνευσης επιβλαβών αερίων και ένας ανεμιστήρας που εξυπηρετεί στον εξαερισμό του σπιτιού. Όταν ο αισθητήρας εντοπίσει συγκέντρωση αερίων μεγαλύτερη της τιμής που έχουμε ορίσει (αναλογική τιμή 400), τότε θα ενεργοποιηθεί ο εξαερισμός μέχρι να καθαρίσει η ατμόσφαιρα και η τιμή να πέσει κάτω από 400.

Για το σύστημα κλιματισμού υπάρχουν δύο επιλογές λειτουργίας

1. Αν η θερμοκρασία είναι μέσα στο εύρος από 21 μέχρι και 24 βαθμούς κελσίου τότε θερμοκρασία θεωρείται φυσιολογική και ο χρήστης μπορεί να ενεργοποιήσει το κλιματιστικό χειροκίνητα με τη βοήθεια του button αν αυτό είναι επιθυμητό.
2. Αν η θερμοκρασία είναι μεγαλύτερη από 24 βαθμούς (για τους καλοκαιρινούς μήνες) τότε το κλιματιστικό θα είναι ρυθμισμένο στην ψύξη και θα ενεργοποιείται αυτόματα μέχρι η θερμοκρασία να φτάσει κάτω από 24 βαθμούς.
Αν η θερμοκρασία είναι μικρότερη από 22 βαθμούς (για τους χειμερινούς μήνες) τότε το κλιματιστικό θα είναι ρυθμισμένο στην θέρμανση και θα ενεργοποιείται αυτόματα μέχρι η θερμοκρασία να φτάσει πάνω από 22 βαθμούς.

Η μπλε λυχνία led υποδηλώνει τη λειτουργία του κλιματιστικού

Το κύκλωμα που υλοποιήθηκε για το σενάριο ελέγχου κλιματισμού και ποιότητας αέρα είναι το παρακάτω:



fritzing

Εικόνα 69: Σχεδίαση ελέγχου κλιματισμού και ποιότητας αέρα με το fritzing

Ο κώδικας του σεναρίου ελέγχου κλιματισμού και ποιότητας αέρα (τρέχει στο Arduino Mega του έξυπνου σπιτιού):

```
int sensorValue; //αισθητήρας ρύπανσης //atmosfaira
int pinButton = 2; // pin button
int stateLED = HIGH;
int stateButton;
int previous = LOW;
long time = 0;
long debounce = 200;
int FAN=5; // pin ανεμιστήρα
int AC=29; // pin led κλιματισμού
#include <dht.h> // βιβλιοθήκη αισθητήρα θερμοκρασίας
#define dht_dpın A1 // pin αισθητήρα θερμοκρασίας
dht dht;

void setup()
{
pinMode(pinButton, INPUT); //atmosfaira
```

```

pinMode(FAN, OUTPUT);
Serial.println("Air Pollution Level\n\n");
pinMode(AC,OUTPUT);
Serial.println("Humidity and temperature\n\n");
void loop()
{
stateButton = digitalRead(pinButton); //atmosfaira
sensorValue = analogRead(0); // read analog input pin 0
int fan =digitalRead(FAN);
if(sensorValue>400)
{
digitalWrite(FAN,HIGH );
}
else
digitalWrite(FAN,LOW );
Serial.print("air pollution = ");
Serial.println(sensorValue, DEC); // prints the value read

if ((dht.temperature>28 || dht.temperature>24)|| (dht.temperature<15
|| dht.temperature<22))
{digitalWrite(AC,HIGH );
}
else digitalWrite(AC,LOW );

if(dht.temperature>=21 && dht.temperature<=24) {

if(stateButton == HIGH && previous == LOW && millis() - time >
debounce) {
if(stateLED == HIGH){
stateLED = LOW;
} else {
stateLED = HIGH;
}
time = millis();
}

digitalWrite(AC, stateLED);
previous = stateButton;}

dht.read11(dht_dp);

Serial.print("Current humidity = ");
Serial.print(dht.humidity);
Serial.print("% ");
Serial.print("temperature = ");
Serial.print(dht.temperature);
Serial.println("C ");
}

```

4.6 Σενάριο ανίχνευσης φωτιάς και πλημμύρας

Η ανίχνευση της φωτιάς και της πλημμύρας είναι πολύ σημαντική στις μέρες μας καθώς δίνεται η δυνατότητα στο χρήστη να ενημερώνεται άμεσα για τις συνθήκες που επικρατούν στο σπίτι του, ώστε να εξασφαλιστεί η έγκαιρη επέμβαση όποτε αυτή κριθεί αναγκαία.

Οι αισθητήρες που χρησιμοποιήθηκαν για την ανίχνευση της φωτιάς και της πλημμύρας είναι οι παρακάτω:

1. Flame Sensor Module IR Sensor Detector
2. Water Level Sensor
3. 1x Buzzer
4. 2x Leds (fire, flood)

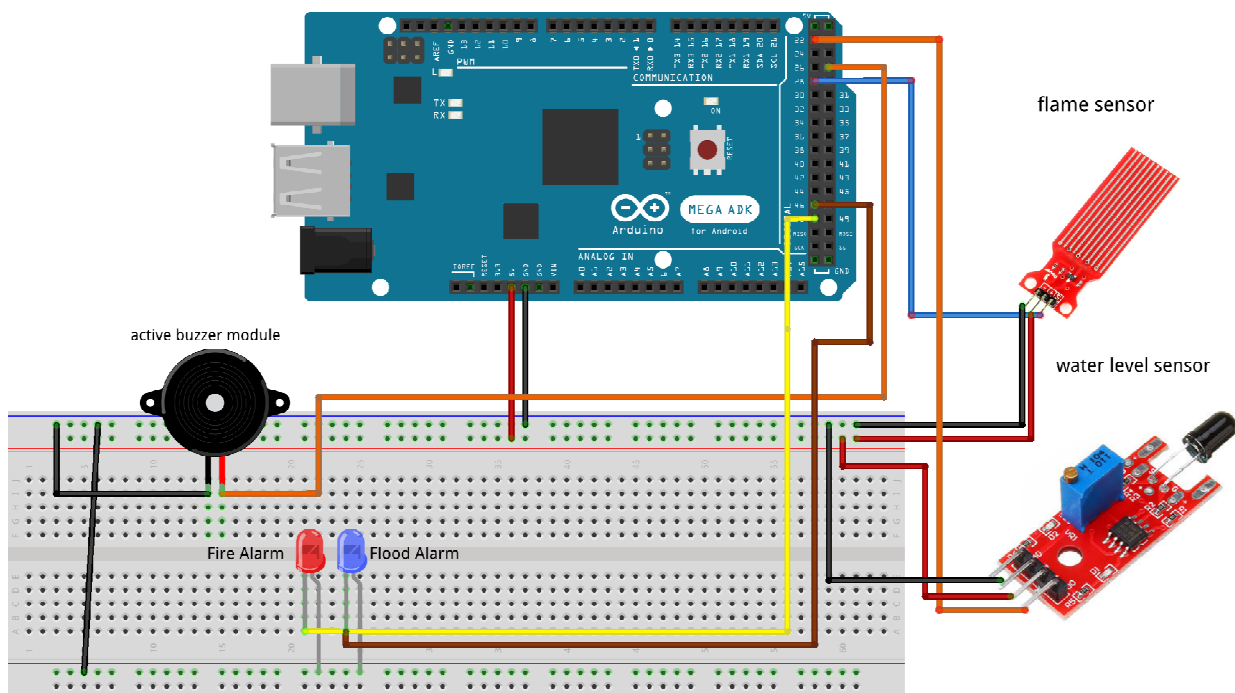
Το συγκεκριμένο σενάριο δεν χρειάζεται κάποια βιβλιοθήκη για την λειτουργία του.

Στη συνέχεια θα επεξηγηθεί η λειτουργία του σεναρίου.

Η λειτουργία του σεναρίου ανίχνευσης της φωτιάς και της πλημμύρας είναι η εξής:

1. Όταν ο αισθητήρας νερού ανιχνεύσει πλημμύρα σε κάποιο χώρο του σπιτιού, τότε θα πάρει την τιμή λογικό 1, θα ανάψει η λυχνία led Flood Alarm (μπλέ) και θα ενεργοποιηθεί η ηχητική ένδειξη (Buzzer), με καθορισμένη ηχητική σήμανση.
2. Όταν ο αισθητήρας φωτιάς ανιχνεύσει ακτινοβολία σε κάποιο χώρο του σπιτιού, τότε θα πάρει την τιμή λογικό 1, θα ανάψει η λυχνία led Fire Alarm (κόκκινο) και θα ενεργοποιηθεί η ηχητική ένδειξη (Buzzer), με καθορισμένη ηχητική σήμανση. Επίσης, ο αισθητήρας φωτιάς μπορεί να ρυθμιστεί με την βοήθεια του ποτενσιομέτρου που διαθέτει, ανάλογα με τον χώρο λειτουργίας του (π.χ ψησταριά, οικεία κ.τ.λ).
3. Σε περίπτωση που και οι δύο αισθητήρες ανιχνεύσουν πλημμύρα και ακτινοβολία ταυτόχρονα σε κάποιο χώρο του σπιτιού, τότε θα πάρουν την τιμή λογικό 1, θα ανάψουν οι λυχνίες led Flood & Fire Alarm (μπλέ, κόκκινο) ταυτόχρονα και θα ενεργοποιηθεί η ηχητική ένδειξη (Buzzer) με καθορισμένη ηχητική σήμανση.

Το κύκλωμα που υλοποιήθηκε για το σενάριο ανίχνευσης φωτιάς και πλημμύρας είναι το παρακάτω:



Εικόνα 70: Σχεδίαση ανίχνευσης φωτιάς και πλημμύρας με το fritzing

Ο κώδικας του σεναρίου για την ανάγνωση φωτιάς και πλημμύρας(τρέχει στο Arduino Mega του έξυπνου σπιτιού):

```
int Grove_Water_Sensor=28; // pin αισθητήρα στάθμης νερού
int FLAME=22; // pin αισθητήρα φωτιάς
int ALARM=27; // pin led buzzer
int FLOOD=48; // pin led Flood Alarm
int FIRE=46; // pin led Fire Alarm

void setup()
{
  Serial.println("Fire and Flood test"); // WATER-FLAME
  pinMode(FLAME, INPUT); //define FLAME input pin
  pinMode(ALARM, OUTPUT); //define ALARM output pin
  // Flame sensor code for Robojax.com
  pinMode (FLOOD,OUTPUT);
  pinMode(Grove_Water_Sensor, INPUT); // The Water Sensor is an
  Input
  pinMode(FIRE,OUTPUT ); // The LED is an Output
  void loop()
  {
    int flood=digitalRead(Grove_Water_Sensor);
    int fire = digitalRead(FLAME); // read FLAME sensor //
    WATER-FLAME
    if (flood==HIGH && fire == LOW ) {
      digitalWrite(ALARM,HIGH);
      digitalWrite(FIRE,HIGH);
      digitalWrite(FLOOD,HIGH);
      Serial.println("Flood fire !"); }

    else if (digitalRead(Grove_Water_Sensor)==HIGH) {
      digitalWrite(ALARM,HIGH);
      delay(200);
      digitalWrite(ALARM,LOW);
      delay(200);
      Serial.println("Flood!");
      digitalWrite(FLOOD,HIGH);
    }
    else if( fire == LOW)
  {
    Serial.println("Fire! Fire!");
    digitalWrite(ALARM,HIGH); // set the buzzer ON
    delay(50);
    digitalWrite(ALARM,LOW);
    delay(50);
    digitalWrite(FIRE,HIGH);
  }
  else {
    digitalWrite(ALARM,LOW); // Set the buzzer OFF
    digitalWrite(FLOOD,LOW);
    Serial.println("Peace");
    Serial.println("Not Flood");
    digitalWrite(FIRE,LOW); }
  }
}
```

4.7 Σενάριο ελέγχου παντζουριών και εξωτερικού φωτισμού

Στο σενάριο αυτό έχει αυτοματοποιηθεί η λειτουργία των εξωτερικών φώτων ανάλογα με την φωτεινότητα καθώς και ο έλεγχος των παντζουριών ανάλογα με την ώρα της ημέρας και την φωτεινότητα. Ο έλεγχος του εξωτερικού φωτισμού μπορεί να συντελέσει στην εξοικονόμηση ενέργειας για τις ώρες που υπάρχει επαρκής φωτισμός και ο έλεγχος των παντζουριών προσφέρει ευκολία στον χρήστη και εξοικονόμηση θερμικής ενέργειας ειδικά σε κατοικίες με μεγάλο αριθμός ανοιγμάτων.

Οι αισθητήρες που χρησιμοποιήθηκαν για τον έλεγχο παντζουριών και εξωτερικού φωτισμού είναι:

1. DS1302 Real Time Clock Module
2. 1x Photoresistor
3. 2x Leds (παντζούρια, εξωτερικός φωτισμός)
4. 1x Resistor 10000Ω

Η βιβλιοθήκη που χρειάζεται για την λειτουργία του σεναρίου είναι η DS1302.h.

Η λειτουργία του σεναρίου είναι η εξής:

Εξωτερικός φωτισμός (πράσινη λυγία led)

Όταν ο αισθητήρας φωτοαντίστασης εντοπίσει τιμή φυσικού φωτισμού κάτω από 50% (δύση ηλίου) τότε ενεργοποιούνται τα εξωτερικά φώτα. Αντίστοιχα θα σβήσουν όταν ο φωτισμός ξεπεράσει το 40% κατά το ξημέρωμα.

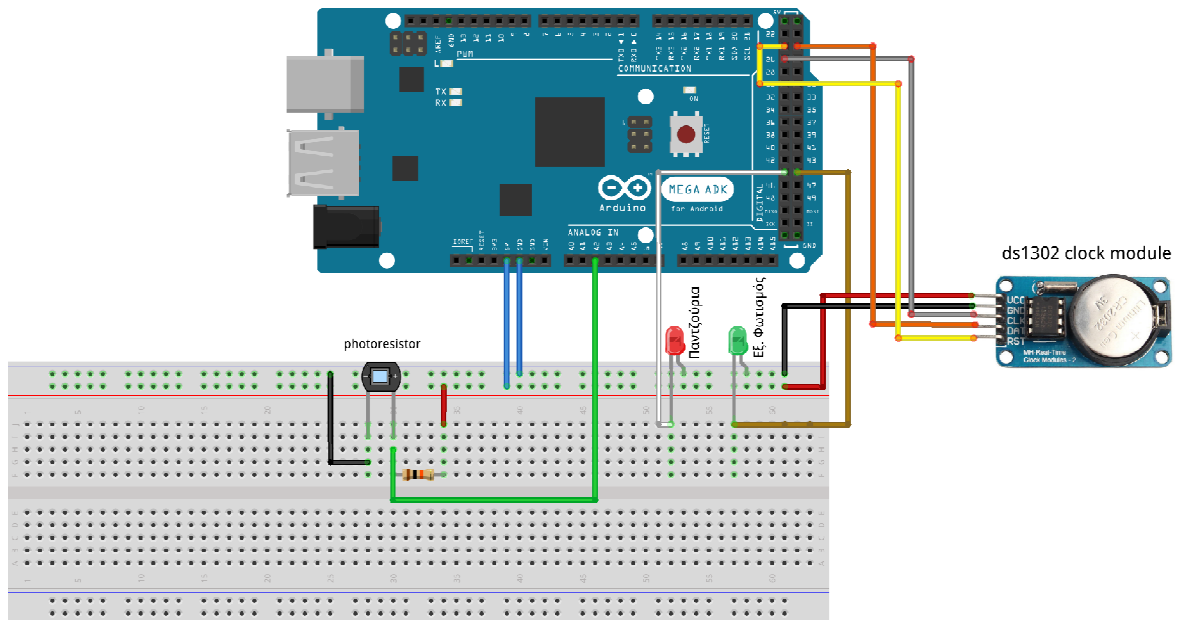
Παντζούρια (κόκκινη λυγία led)

Για τα παντζούρια εκτός από την παράμετρο του εξωτερικού φωτισμού έχει οριστεί και η παράμετρος της ώρας της ημέρας. Αυτό έχει πραγματοποιηθεί ώστε τους μήνες όπου η μέρα είναι μικρότερη να μπορεί να οριστεί η ώρα κλεισίματος των παντζουριών ανάλογα με τις προτιμήσεις του χρήστη. Η ώρα διαβάζεται με την βοήθεια του αισθητήρα DS1302 Real Time Clock Module.

Αρα αν η φωτεινότητα πέσει κάτω από το 50% και η ώρα είναι αυτή που έχει οριστεί (π.χ. 21:00μμ) τότε τα παντζούρια κλείνουν.

Αντίστοιχα εάν η φωτεινότητα ξεπεράσει το 40% και η ώρα είναι η καθορισμένη (π.χ. 7:00πμ) τότε τα παντζούρια ανοίγουν.

Το κύκλωμα του σενάριου ελέγχου παντζουριών και εξωτερικού φωτισμού είναι:



fritzing

Εικόνα 71: Σχεδίαση ελέγχου παντζουριών και εξωτερικού φωτισμού με το fritzing

Ο κώδικας για το σενάριο ελέγχου παντζουριών και εξωτερικού φωτισμού (τρέχει στο Arduino Mega του έξυπνου σπιτιού):

```

//eks.fwta- patzouria
#include <DS1302.h> // βιβλιοθήκη ds1302 clock module
int led=45; // pin led εξ. φωτισμού
int sensorPin = A2; // pin photoresistor
int sun_Value = 0;
int led1=44; // pin led παντζουριών
DS1302 rtc(24,25,26); // pin clock module
Time t;
const int OnHour = 00; // ώρα ανοίγματος παντζουριών
const int OnMin = 04;
const int OffHour = 00; // ώρα κλεισίματος παντζουριών
const int OffMin = 06;

void setup()
{
  rtc.halt(false); //eks.fwta- patzouria
  rtc.writeProtect(false);
  pinMode (led,OUTPUT);
  pinMode (led1,OUTPUT);
}

```

```

void loop()
{
t = rtc.getTime(); //eks.fwta- patzouria
Serial.print(t.hour);
Serial.print(" hour(s), ");
Serial.print(t.min);
Serial.print(" minute(s)");
Serial.println(" ");

sun_Value = analogRead(sensorPin);
sun_Value = map(sun_Value,100,800,100,0);
if(sun_Value<50){
digitalWrite(led,HIGH);
Serial.println("LIGHT ON");
if (t.hour == OnHour && t.min == OnMin)
{digitalWrite(led1,HIGH);
Serial.println("pantzouri kleisto");}
}

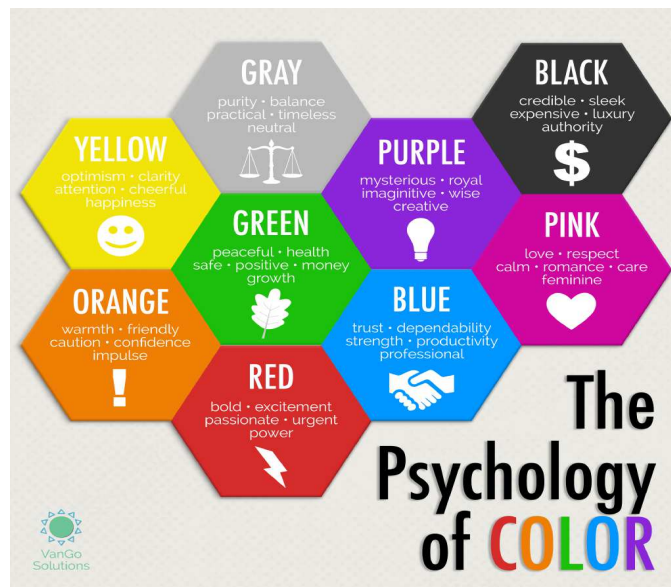
else if(sun_Value>40){
digitalWrite(led,LOW);
Serial.println("LIGHT OFF");
if (t.hour == OffHour && t.min == OffMin)
{digitalWrite(led1,LOW);
Serial.println("pantzouri anoixto");}
}
}
}

```

4.8 Σενάριο ελέγχου φωτισμού αίθουσας ψυχαγωγίας

Το σενάριο αυτό υλοποιήθηκε για τον έλεγχο της απόχρωσης του φωτισμού σε μία αίθουσα ψυχαγωγίας όπως για παράδειγμα το σαλόνι ή ένα playroom. Τα διαφορετικά χρώματα μπορούν να επηρεάσουν την ψυχολογία του ανθρώπου και επίσης να έχουν διάφορες επιδράσεις πάνω στο ανθρώπινο σώμα. Με αυτό το δεδομένο δίνεται η δυνατότητα αναλόγως την δραστηριότητα που είναι επιθυμητή (άθληση, κινηματογραφική ταινία, χαλάρωση) να ρυθμιστεί αναλόγως και η θερμοκρασία του χρώματος.

Παρακάτω φαίνονται δύο πίνακες που δείχνουν τι επίδραση έχει το κάθε χρώμα στο σώμα και στην ψυχολογία.



Εικόνα 72: Η ψυχολογία των χρωμάτων



Εικόνα 73: Πως τα χρώματα επηρεάζουν το σώμα

Οι αισθητήρες που χρησιμοποιήθηκαν για το σενάριο είναι οι παρακάτω:

1. 3 Colour LED Module
2. Infrared IR Receiver Module Wireless Remote Control Kit For Arduino

Η βιβλιοθήκη που χρειάζεται για την λειτουργία του σεναρίου είναι η IRremote.h.

Η λειτουργία του σεναρίου περιγράφεται παρακάτω:

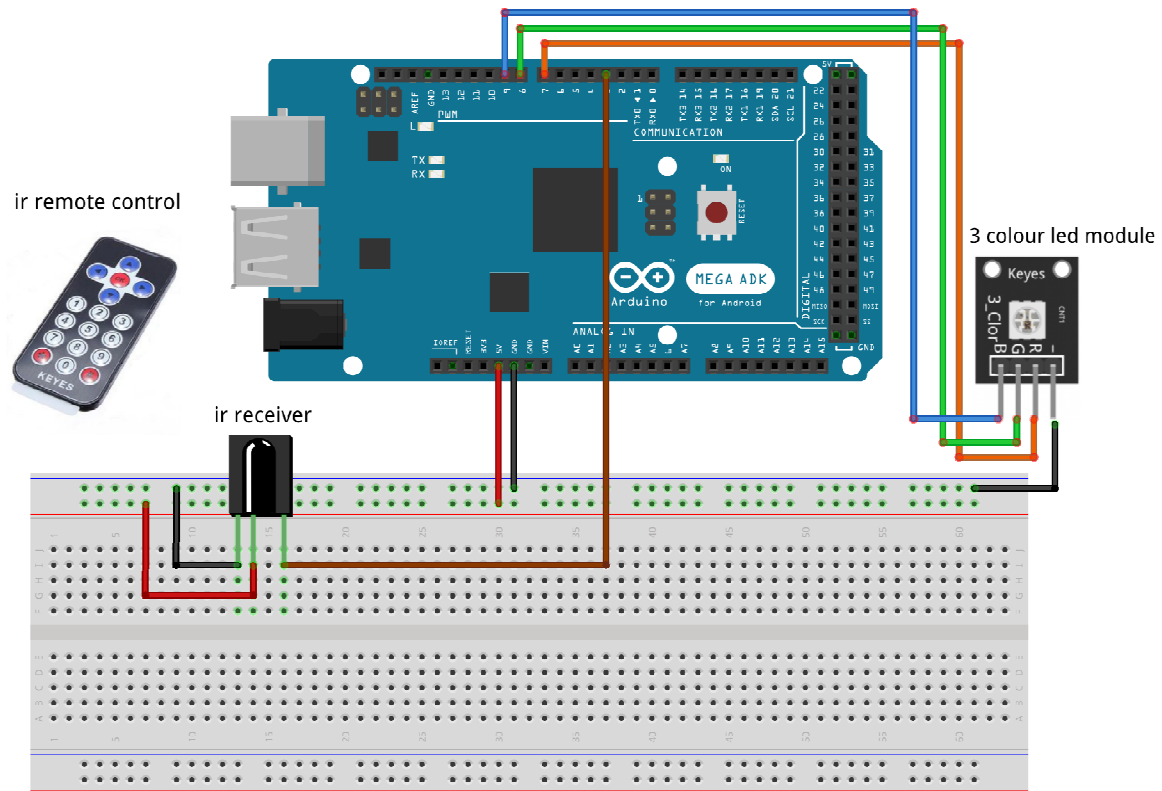
Με το τηλεχειριστήριο μπορεί να ελεγχθεί το χρώμα που θα πάρει το 3 Colour LED Module.

Πιο συγκεκριμένα:

- Πλήκτρο 4: Πράσινο χρώμα
- Πλήκτρο 5: Κόκκινο χρώμα
- Πλήκτρο 6: Μπλε χρώμα
- Πλήκτρο 7: Λευκό χρώμα
- Πλήκτρο 8: Πορτοκαλί χρώμα
- Πλήκτρο 9: Κανένα χρώμα

Στην κανονική υλοποίηση για το έξυπνο σπίτι θα χρησιμοποιηθεί rgb ταινία led η οποία πρέπει να υπολογιστεί ώστε να καλύπτει τις απαιτήσεις φωτισμού του χώρου. Από τον κώδικα και αναλόγως την επιθυμητή δραστηριότητα μπορεί να ρυθμιστεί και η αντίστοιχη απόχρωση.

Το κύκλωμα για το σενάριο ελέγχου φωτισμού αίθουσας ψυχαγωγίας είναι το παρακάτω:



fritzing

Εικόνα 74: Σχεδίαση ελέγχου φωτισμού αίθουσας ψυχαγωγίας με το fritzing

Ο κώδικας για το σενάριο ελέγχου φωτισμού αίθουσας ψυχαγωγίας (τρέχει στο Arduino Mega του έξυπνου σπιτιού):

```

//RGB
#include <IRremote.h> // βιβλιοθήκη IR
int RECV_PIN = 3; // pin ir receiver
int redPin = 7; // red rgb pin
int greenPin = 8; // green rgb pin
int bluePin = 9; // blue rgb pin
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup()
{
  pinMode(redPin,OUTPUT); //RGB
  pinMode(greenPin,OUTPUT);
  pinMode(bluePin,OUTPUT);

```

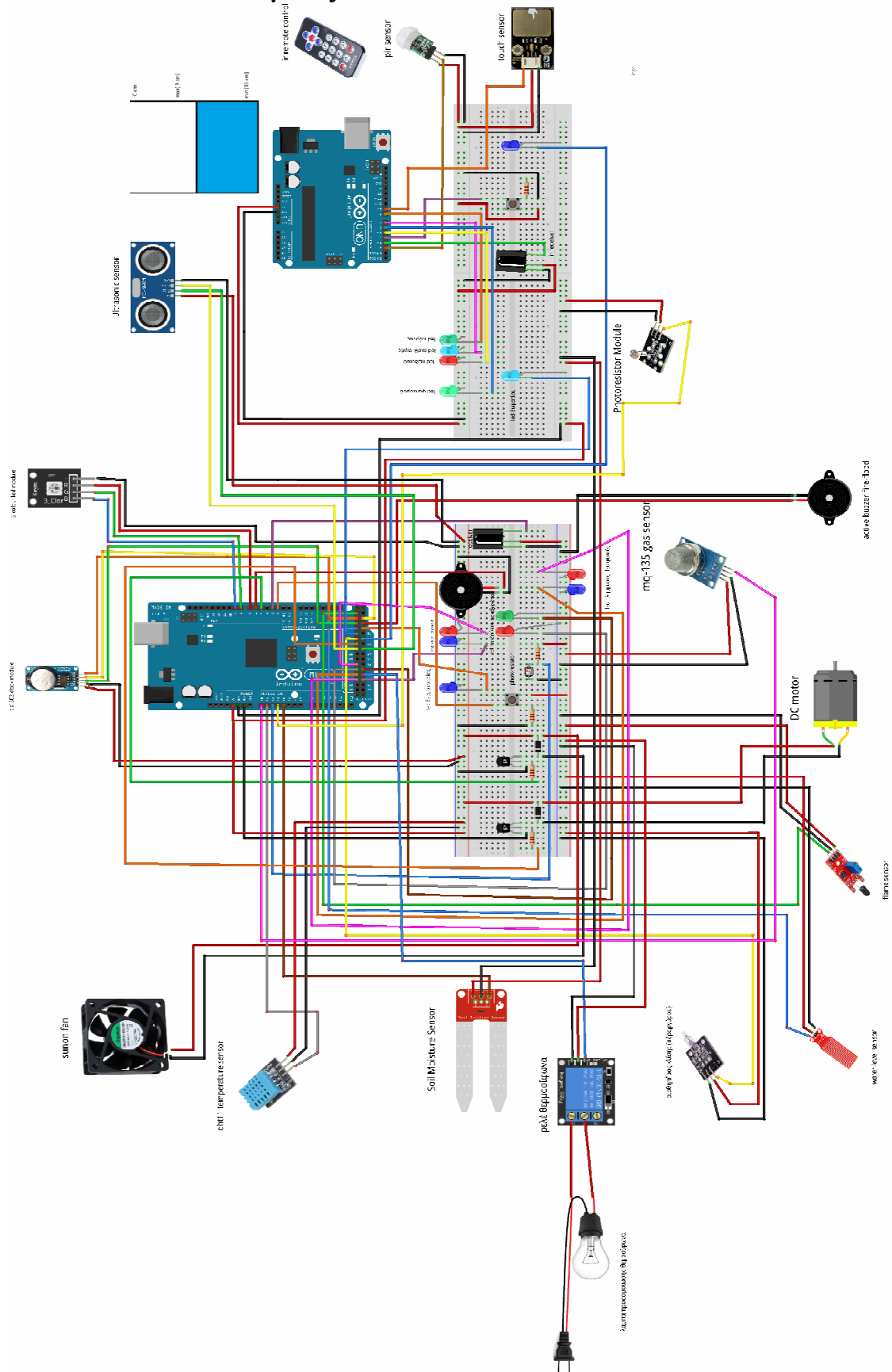
```

irrecv.enableIRIn(); // Start the receiver
void loop()
{
  if (irrecv.decode(&results)) //RGB
  {
    Serial.println(results.value, HEX);
    translateIR();
    for (int z=0; z<2; z++) // ignore 2nd and 3rd signal repeat
    {
      irrecv.resume(); // receive the next value
    } }
  void translateIR() // takes action based on IR code received
  // describing Sony IR codes on LCD module
  {
    switch(results.value)
    {
      case 0xFF22DD:
        setColor(255, 0, 0); //Green Color
        break;
      case 0xFF02FD:
        setColor(0, 255, 0); // Red Color
        break;
      case 0xFFC23D:
        setColor(0, 0, 255); // Blue Color
        break;
      case 0xFFE01F:
        setColor(255, 255, 255); // White Color
        break;
      case 0xFFA857:
        setColor(50, 255, 200); // Orange Color
        break;
      case 0xFF906F:
        setColor(0, 0, 0); // off
        break;

      default:
        Serial.println("OTHER");
    }
    //delay(200);
  }
  void setColor(int redValue, int greenValue, int blueValue) {
    analogWrite(redPin, redValue);
    analogWrite(greenPin, greenValue);
    analogWrite(bluePin, blueValue);}
}

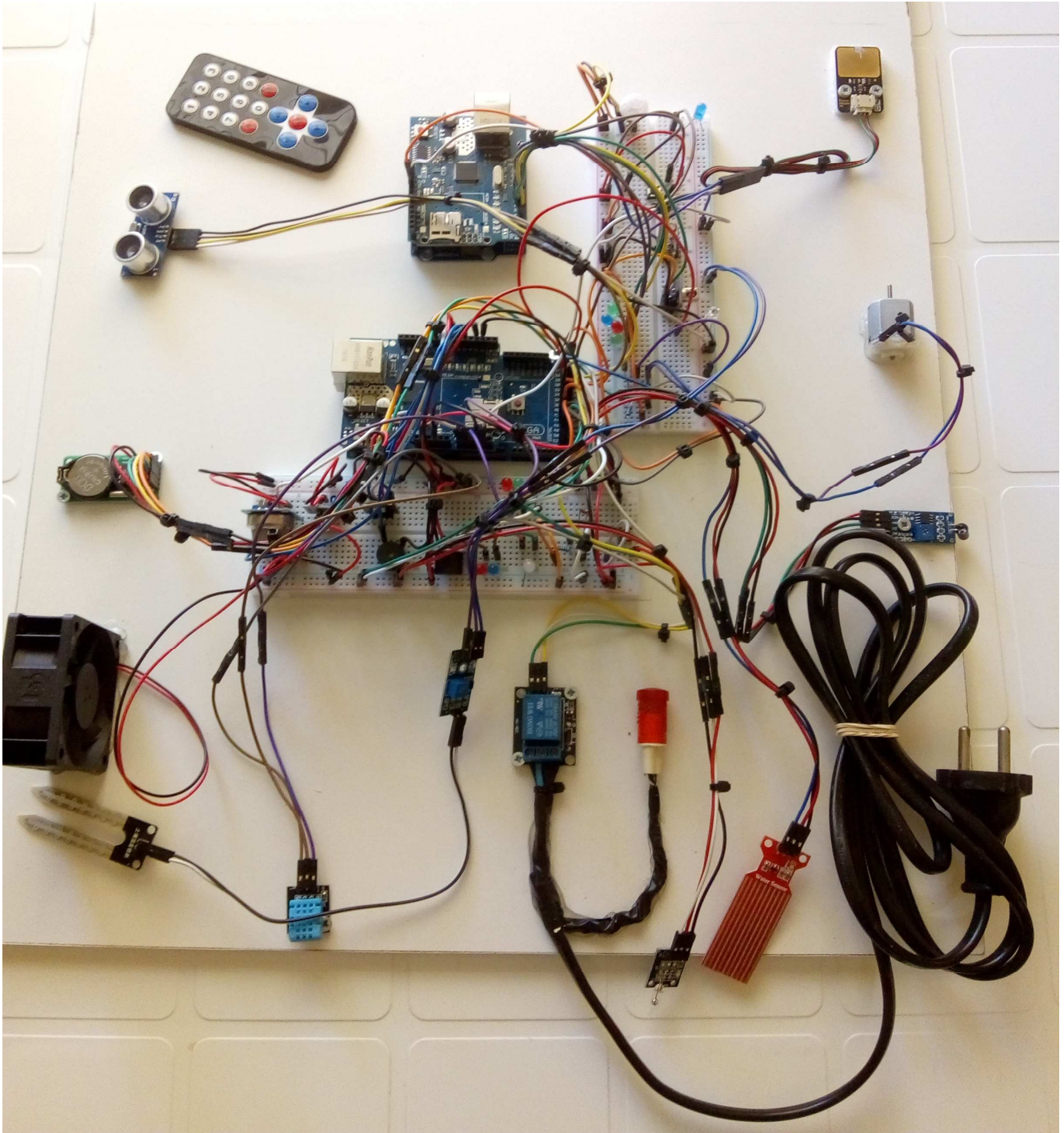
```

4.9 Τελικό κύκλωμα έξυπνου σπιτιού

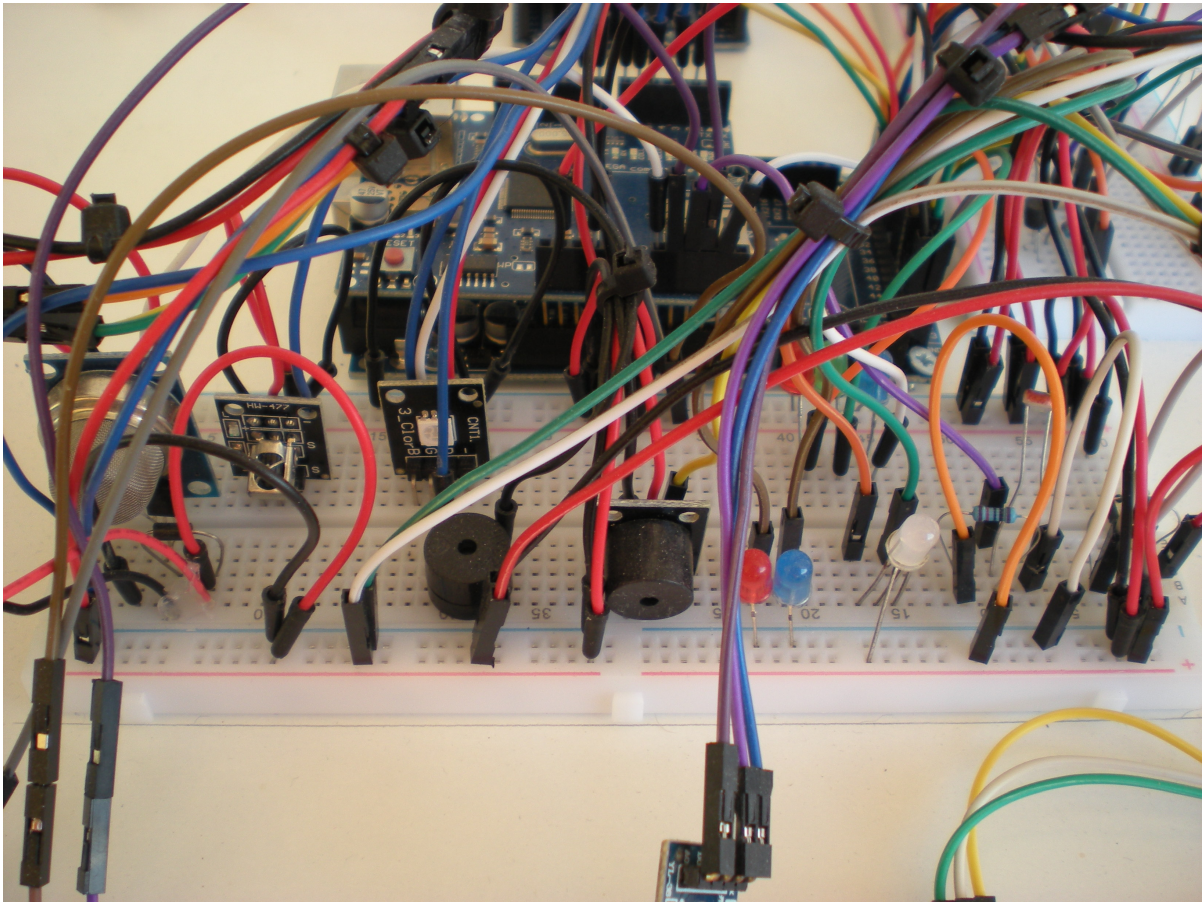


fritzing

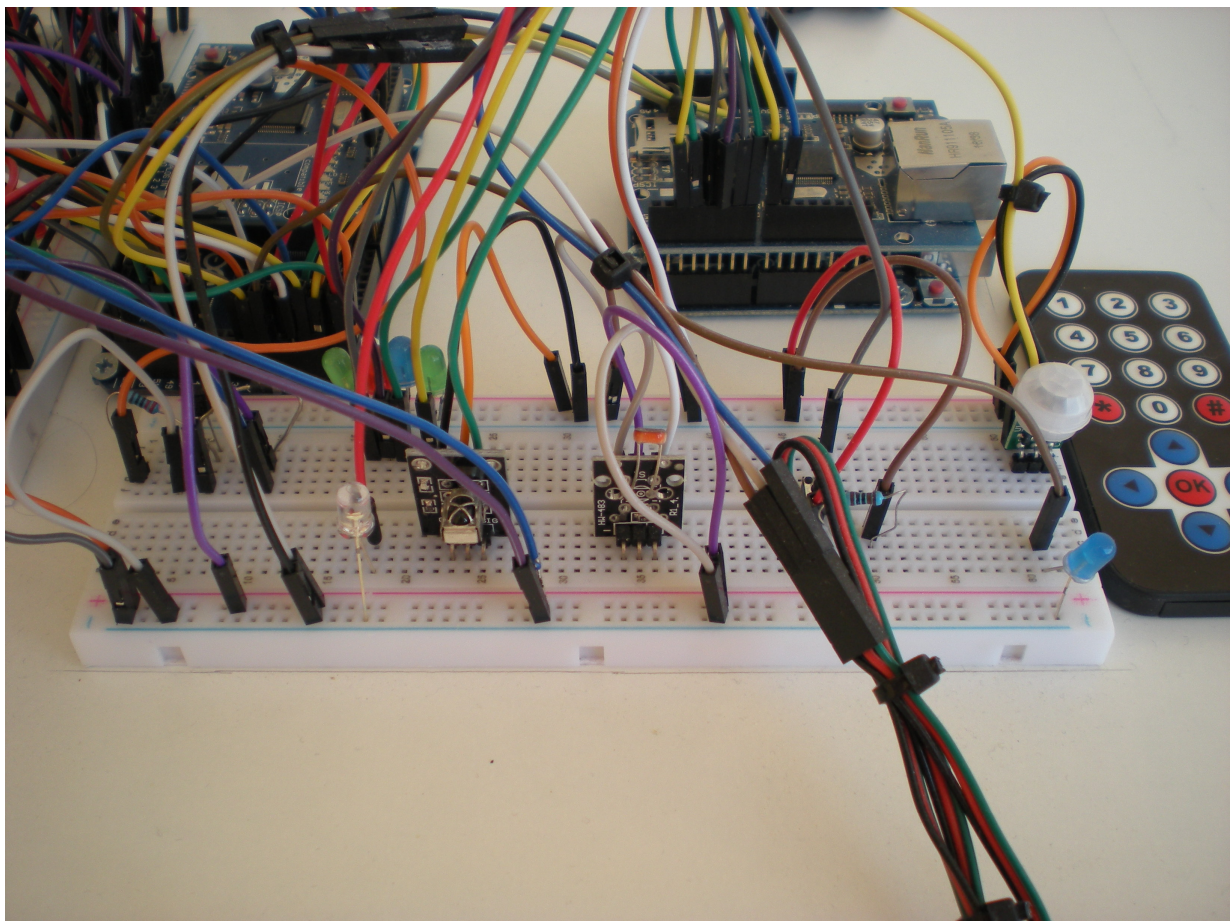
Εικόνα 75: Σχεδίαση τελικού κυκλώματος με το fritzing



Εικόνα 76: Η μακέτα του έξυπνου σπιτιού



Εικόνα 77: Μακέτα(Arduino Mega & 1^ο breadboard)



Εικόνα 78: Μακέτα(Arduino Uno & 2^ο breadboard)

4.10 Σενάριο βραχιολιού έκτακτης ανάγκης για ηλικιωμένους ή ατόμων με ιατρικές παθήσεις.

Το βραχιόλι έκτακτης ανάγκης για ηλικιωμένους ή ατόμων με ιατρικές παθήσεις κατασκευάστηκε για να εξυπηρετήσει ηλικιωμένα άτομα ή άτομα όλων των ηλικιών με ιατρικές παθήσεις, τα οποία βρίσκονται μέσα στο σπίτι και είναι αναγκαία η παρακολούθηση τους τις ώρες που λείπουν τα άτομα που είναι υπεύθυνα για την φροντίδα τους. Πιο συγκεκριμένα μπορεί να υπάρξει απομακρυσμένη ειδοποίηση για πιθανή πτώση, αυξημένους καρδιακούς παλμούς και μπουτόν έκτακτης ανάγκης(SOS) καθώς και ένδειξη καρδιακών παλμών και συνθηκών περιβάλλοντος(θερμοκρασία-υγρασία).

Επίσης έχει τοποθετηθεί και οθόνη η οποία δίνει τη δυνατότητα στον ίδιο τον χρήστη να έχει ένδειξη κάποιων χρήσιμων πληροφοριών όπως οι καρδιακοί παλμοί του, η ώρα, η θερμοκρασία και η υγρασία.

Οι αισθητήρες που χρησιμοποιήθηκαν για το βραχιόλι έκτακτης ανάγκης ηλικιωμένων είναι:

1. MPU-6050 module
2. esp8266 wifi module
3. Pulsesensor Pulse Heart Rate Sensor Module For Arduino
4. Geekcreit® 1.3 Inch 4Pin White OLED LCD Display 12864 IIC I2C Interface Module For Arduino
5. DS 1302 clock module
6. DHT 11 temperature and himidity sensor
7. 1x led (heart rate)
8. 1x button
9. 4x 220Ω Resistor

Η βιβλιοθήκες που χρειάζονται για την λειτουργία του σεναρίου είναι η Wire.h, η U8glib.h, η DS1302.h και η dht.h.

Οι λειτουργίες του βραχιολιού περιγράφονται παρακάτω:

Πτώση: Για την πτώση έχει χρησιμοποιηθεί το επιταχυνσιόμετρο- γυροσκόπιο MPU-6050 module το οποίο μπορεί να εντοπίσει μέσω των τριών αξόνων x,y,z, τη θέση που βρίσκεται στον χώρο. Όταν ο αισθητήρας εντοπίσει απότομη αλλαγή κατά κύριο λόγο στον άξονα y και σε συνδυασμό και με τους άλλους άξονες τότε ενεργοποιείται η συνθήκη της πτώσης και στέλνεται ειδοποίηση στα άτομα που είναι υπεύθυνα για την επίβλεψη.

Καρδιακοί παλμοί: Για την ανάγνωση των καρδιακών παλμών έχει χρησιμοποιηθεί ο αισθητήρας Pulsesensor Pulse Heart Rate Sensor Module For Arduino. Ο αισθητήρας τοποθετείται στην κάτω πλευρά του καρπού του ηλικιωμένου και μπορεί να διαβάσει τους παλμούς ανά λεπτό. Οι παλμοί φαίνονται στην οθόνη του βραχιολιού ώστε να μπορεί ο ίδιος ο χρήστης ελέγχει την κατάσταση του και επίσης στέλνονται ασύρματα στα άτομα που είναι υπεύθυνα για την επίβλεψη.

Button έκτακτης ανάγκης: Στο βραχιόλι έχει τοποθετηθεί button το οποίο ο ηλικιωμένος ή το άτομο με ιατρική πάθηση, μπορεί να πατήσει όποτε αυτός κρίνει αναγκαίο. Αυτό είναι πολύ χρήσιμο καθώς μπορεί ο χρήστης να μην είναι σε θέση να πάρει τηλέφωνο οπότε πατώντας το button έκτακτης ανάγκης στέλνεται σήμα στους οικείους του οι οποίοι μπορούν να δράσουν αναλόγως.

Οπτική ένδειξη αυξημένων καρδιακών παλμών: Σε περίπτωση αυξημένων καρδιακών παλμών έχει τοποθετηθεί ενδεικτική λυχνία led Heart rate Alert (κόκκινο) η οποία

αναβοσβήνει σε συγκεκριμένη συχνότητα για να ειδοποιήσει τον χρήστη αν αυτός δεν το αντιληφθεί μέσω της οθόνης.

Ενδειξη ώρας και ημερομηνίας: Με την βοήθεια του αισθητήρα DS 1302 clock module εμφανίζεται στην οθόνη του βραχιολιού η ώρα και η ημερομηνία

Οθόνη: Ως οθόνη για το βραχιολάκι έχει χρησιμοποιηθεί η Geekcreit® 1.3 Inch 4Pin White OLED LCD Display 12864 IIC I2C Interface Module For Arduino.

Για λόγους ευκολίας ανάγνωσης η οθόνη εμφανίζει τις τιμές σε δύο διαφορετικές οθόνες οι οποίες εναλλάσσονται ανά τέσσερα δευτερόλεπτα .

1^η οθόνη: Ημέρα ημερομηνία και ώρα.

2^η οθόνη: Θερμοκρασία, υγρασία και καρδιακοί παλμοί ανά λεπτό.



Εικόνα 79: Πρώτη οθόνη



Εικόνα 80: Δεύτερη οθόνη

Αυτόματες ειδοποιήσεις : Το βραχιολάκι έχει σχεδιαστεί ώστε να στέλνει αυτόματες ειδοποιήσεις στα οικεία άτομα του ηλικιωμένου ή του ατόμου με ιατρική πάθηση. Οι περιπτώσεις ειδοποιήσεων είναι οι παρακάτω:

1. Πτώση
2. Πάτημα button έκτακτης ανάγκης
3. Αυξημένοι καρδιακοί παλμοί

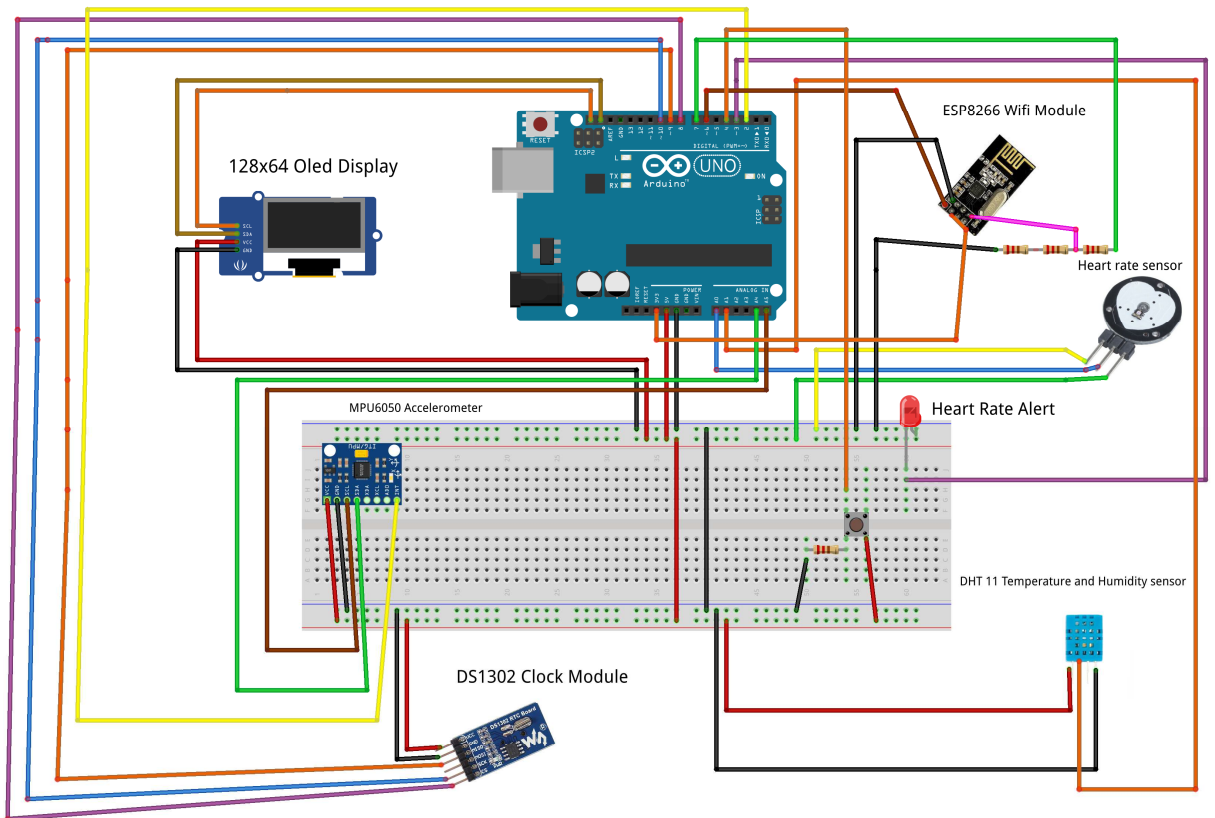
Ο τρόπος με τον οποίο στέλνονται οι ειδοποιήσεις θα εξηγηθεί στο επόμενο κεφάλαιο.

Αισθητήρας θερμοκρασίας: Στο βραχιολάκι έχει τοποθετηθεί και ο αισθητήρας θερμοκρασίας/υγρασίας DHT 11 temperature and himidity sensor, αφενός για να έχει ο χρήστης ένδειξη των τιμών και αφετέρου ώστε σε περίπτωση κάποιου συμβάντος να είναι γνωστές οι συνθήκες του περιβάλλοντος την ώρα του συμβάντος, το οποίο μπορεί να βοηθήσει σε καλύτερη διάγνωση από τον επιβλέπων ιατρό.

Ασύρματη επικοινωνία: Για την ασύρματη επικοινωνία έχει χρησιμοποιηθεί ο αισθητήρας esp8266 wifi module με τον οποίο επιτυγχάνεται σύνδεση με το τοπικό δίκτυο της οικίας και από εκεί αποστέλλονται οι πληροφορίες όπου είναι επιθυμητό. Ο τρόπος που αποστέλλονται οι πληροφορίες αναλύεται στο επόμενο κεφάλαιο.

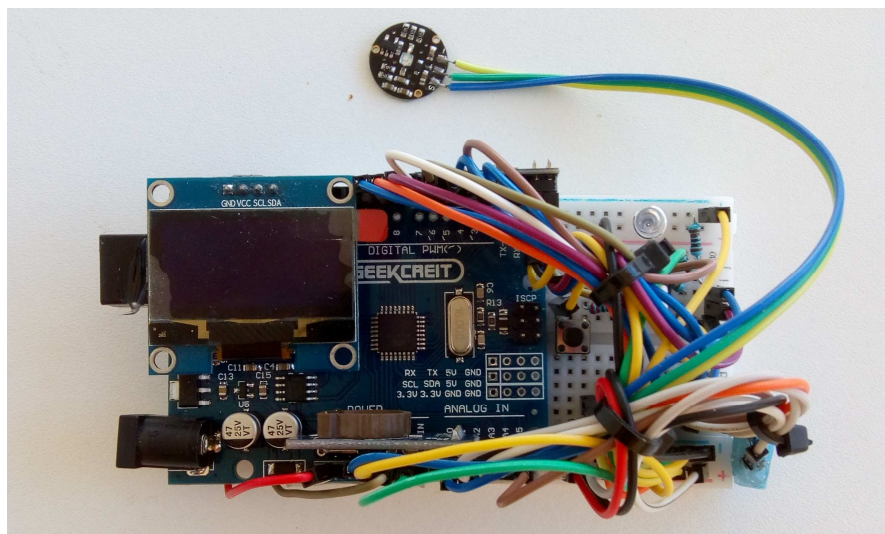
Το βασικότερο στοιχείο (το οποίο δεν πραγματοποιήθηκε λόγω προβλήματος arduino nano και υψηλού κόστους), είναι η χρησιμοποίηση μικρών αισθητήρων, πλακέτας arduino(nano ή μικρότερο) και μπαταρίας, ώστε το βραχιόλι να αποκτήσει φορητότητα και όσο το δυνατόν μικρότερες διαστάσεις.

Το κύκλωμα που υλοποιήθηκε για το βραχιόλι έκτακτης ανάγκης για ηλικιωμένους ή ατόμων με ιατρικές παθήσεις είναι:

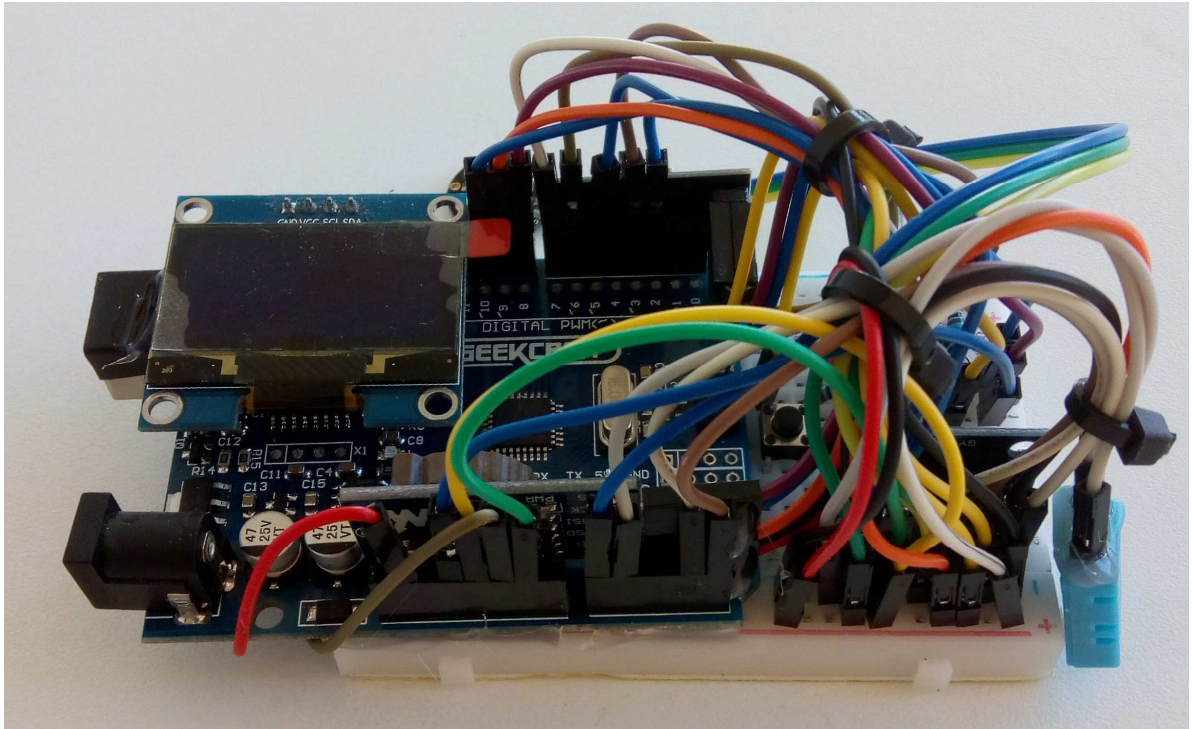


fritzing

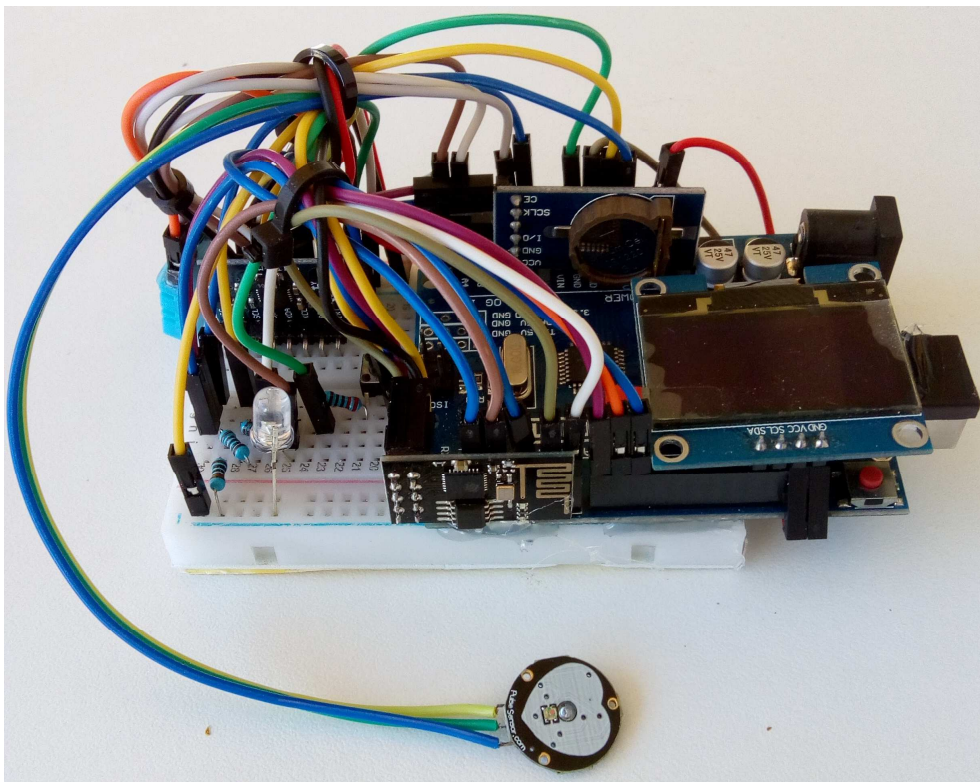
Εικόνα 81: Σχεδίαση τελικού κυκλώματος βραχιολιού έκτακτης ανάγκης για ηλικιωμένους με το fritzing



Εικόνα 82: Βραχιόλι έκτακτης ανάγκης για ηλικιωμένους ή ατόμων με ιατρικές παθήσεις



Εικόνα 83: Πλαϊνή όψη βραχιολιού έκτακτης ανάγκης για ηλικιωμένους ή ατόμων με ιατρικές παθήσεις



Εικόνα 84: Πλαϊνή όψη βραχιολιού έκτακτης ανάγκης για ηλικιωμένους ή ατόμων με ιατρικές παθήσεις

Ο κώδικας για το βραχιόλι έκτακτης ανάγκης για ηλικιωμένους ή ατόμων με ιατρικές παθήσεις. (Τρέχει στο Arduino Uno του βραχιολιού):

```
#include<Wire.h> // βιβλιοθήκη wire
const int emled=3; // pin led έκτακτης ανάγκης
const int MPU_addr=0x68; // I2C address of the MPU-6050
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
float ax=0, ay=0, az=0, gx=0, gy=0, gz=0;
boolean fall = false; //stores if a fall has occurred
boolean trigger1=false; //stores if first trigger (lower threshold)
has occurred
boolean trigger2=false; //stores if second trigger (upper threshold)
has occurred
boolean trigger3=false; //stores if third trigger (orientation
change) has occurred
byte trigger1count=0; //stores the counts past since trigger 1 was
set true
byte trigger2count=0; //stores the counts past since trigger 2 was
set true
byte trigger3count=0; //stores the counts past since trigger 3 was
set true
int angleChange=0;

int sensor = 0; // heart rate sensor pin
double alpha = 0;
double change = 0.0;
double minval = 0.0;
#include "U8glib.h" // βιβλιοθήκη οθόνης oled
U8GLIB_SH1106_128X64 u8g(U8G_I2C_OPT_NONE); // oled

const int buttonPin = 5; // pin em. button
int buttonState = 0;
boolean found = false;
boolean sendFall = false;
boolean emrgButton = false;
int valSensor = 1;
SoftwareSerial esp8266(RX,TX);
unsigned long time_now1 = 0;
unsigned long time_now2 = 0;
#include <DS1302.h> // βιβλιοθήκη ds1302 clock module
DS1302 rtc(8, 9, 10); // pin ds1302 clock module
#include <dht.h> // βιβλιοθήκη αισθητήρα θερμοκρασίας
#define dht_dpín A1 // pin αισθητήρα θερμοκρασίας
dht dht;

void draw(void) { // οθόνη 1

    static double oldValue = 0;
    static double oldChange = 0;
    int rawValue = analogRead (sensor);
    double value = alpha * oldValue + (1 - alpha) *
rawValue*100/1023;
    dht.read11(dht_dpín);
    u8g.setFont(u8g_font_profont12);
    u8g.setPrintPos(0, 10);
    u8g.print(" Temperature");

    u8g.print(dht.temperature);
```

```

u8g.setPrintPos(0, 25);
u8g.print("Humidity=");
u8g.print(dht.humidity);
u8g.setPrintPos(0, 40);
u8g.print("BPM: ");
  u8g.print(value);
}

void draw2(void) { // οθόνη 2
u8g.setFont(u8g_font_profont12);
u8g.setPrintPos(0, 10);
u8g.print(rtc.getDOWStr());
u8g.setPrintPos(0, 25);
u8g.print(rtc.getDateStr());
  u8g.setPrintPos(0, 40);
u8g.print(rtc.getTimeStr());
}

void setup(){

pinMode(buttonPin, INPUT);
Serial.begin(9600);
  Wire.begin();
Wire.beginTransmission(MPU_addr);
Wire.write(0x6B); // PWR_MGMT_1 register
Wire.write(0); // set to zero (wakes up the MPU-6050)
Wire.endTransmission(true);

pinMode(emled, OUTPUT);

  rtc.setDOW(MONDAY); // Set Day-of-Week to FRIDAY
  rtc.setTime(10, 17,00); // Set the time to 12:00:00 (24hr
format)
  rtc.setDate(03, 06, 2019);

void loop(){

dht.read11(dht_dp);

// read the state of the pushbutton value:
buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed. If it is, the buttonState is
HIGH:
if (buttonState == HIGH) {
  Serial.println("Button Pressed");
  emrgButton = true;
}

mpu_read();
//2050, 77, 1947 are values for calibration of accelerometer
// values may be different for you
ax = (AcX-1168)/16384.00;
ay = (AcY-2401)/16384.00;
az = (AcZ+1432)/16384.00;

//270, 351, 136 for gyroscope
gx = (GyX)/131.07;
gy = (GyY)/131.07;
gz = (GyZ)/131.07;

```

```

// calculating Amplitude vector for 3 axis
float Raw_AM = pow(pow(ax,2)+pow(ay,2)+pow(az,2),0.5);
int AM = Raw_AM * 10; // as values are within 0 to 1, I multiplied
// it by for using if else conditions

Serial.println(AM);

if (trigger3==true){
  trigger3count++;
  //Serial.println(trigger3count);
  if (trigger3count>=10){
    angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5);
    //delay(10);
    Serial.println(angleChange);
    if ((angleChange>=0) && (angleChange<=10)){ //if orientation
changes remains between 0-10 degrees
      fall=true; trigger3=false; trigger3count=0;
      Serial.println(angleChange);
    }
    else{ //user regained normal orientation
      trigger3=false; trigger3count=0;
      Serial.println("TRIGGER 3 DEACTIVATED");
    }
  }
}
if (fall==true){ //in event of a fall detection
  Serial.println("FALL DETECTED");

  fall=false;
  sendFall = true;

}
if (trigger2count>=6){ //allow 0.5s for orientation change
  trigger2=false; trigger2count=0;
  Serial.println("TRIGGER 2 DEACTIVATED");
}
if (trigger1count>=6){ //allow 0.5s for AM to break upper threshold
  trigger1=false; trigger1count=0;
  Serial.println("TRIGGER 1 DEACTIVATED");
}
if (trigger2==true){
  trigger2count++;

  angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5);
Serial.println(angleChange);
  if (angleChange>=30 && angleChange<=400){ //if orientation
changes by between 80-100 degrees
    trigger3=true; trigger2=false; trigger2count=0;
    Serial.println(angleChange);
    Serial.println("TRIGGER 3 ACTIVATED");
  }
}
if (trigger1==true){
  trigger1count++;
  if (AM>=12){ //if AM breaks upper threshold (3g)
    trigger2=true;
    Serial.println("TRIGGER 2 ACTIVATED");
    trigger1=false; trigger1count=0;
  }
}
if (AM<=2 && trigger2==false){ //if AM breaks lower threshold
(0.4g)

```

```

    trigger1=true;
    Serial.println("TRIGGER 1 ACTIVATED");
}
//It appears that delay is needed in order not to clog the port
delay(100);

if(millis() > (time_now2 + 4000) && millis() < (time_now2 + 4200))
{
    {
        u8g.firstPage();
        do {
            draw();
        } while (u8g.nextPage() );

    }
}
if(millis() > (time_now2 + 8000))
{
    {
        u8g.firstPage();
        do {
            draw2();
        } while (u8g.nextPage() );
    }
    time_now2 = millis();
}
}

static double oldValue = 0;
static double oldChange = 0;
int rawValue = analogRead (sensor);
double value = alpha * oldValue + (1 - alpha) * rawValue*100/1023;
float value1= value;

if(millis() > (time_now1 + 300)){
digitalWrite(emled,LOW);
time_now1 = millis();}
else if( value1>55) digitalWrite(emled,HIGH);

void mpu_read(){
Wire.beginTransmission(MPU_addr);
Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
Wire.endTransmission(false);
Wire.requestFrom(MPU_addr,14,true); // request a total of 14
registers
AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C
(ACCEL_XOUT_L)
AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E
(ACCEL_YOUT_L)
AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40
(ACCEL_ZOUT_L)
Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42
(TEMP_OUT_L)
GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44
(GYRO_XOUT_L)
GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46
(GYRO_YOUT_L)
GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48
(GYRO_ZOUT_L)

}

```

Κεφάλαιο 5: Λογισμικό μέρος

5.1 Σύνδεση Ethernet και Βάση Δεδομένων

Το υποκεφάλαιο αυτό αποτελείται από τη σύνδεση του Ethernet και τη Βάση Δεδομένων. Θα γίνει αναφορά στις δηλώσεις του Ethernet που χρειάζονται για την σύνδεση του με το διαδίκτυο και στην δημιουργία της Βάσης Δεδομένων για την αποστολή των μετρήσεων από τους αισθητήρες. Στις επόμενες δύο ενότητες του υποκεφαλαίου θα γίνει ανάλυση των παραπάνω.

5.1.1 Σύνδεση Ethernet

Για την σύνδεση του Ethernet δηλώνεται η διεύθυνση IP του server από τον υπολογιστή μας και την IP Address του server που δίνει το Arduino. Επίσης θα χρειαστεί και το byte mac Address του Arduino.

Ο Κώδικας σύνδεσης του Ethernet, ο οποίος τρέχει και στα δύο Arduino του έξυπνου σπιτιού είναι:

```
#include <SPI.h>           //βιβλιοθήκη για την σειριακή επικοινωνία
του Arduino με το Ethernet
#include <Ethernet.h>     //βιβλιοθήκη Ethernet
int Wait_between_dumps = 0;
byte mac[] = { 0x90, 0xA2, 0xDA, 0x10, 0x40, 0x4F };
EthernetClient client;
IPAddress ip(192, 168, 0, 189); // διεύθυνση ip του Arduino
IPAddress serv(192, 168, 0, 104); // διεύθυνση ip του υπολογιστή

void setup()
{
  Ethernet.begin(mac, ip); // initialize Ethernet device

void loop()
{
  db();
}

void db() { // διάβασμα τιμών για αποστολή στην βάση δεδομένων
float temp = dht.temperature;
float hum = dht.humidity;
int fan =digitalRead(FAN);
int flood=digitalRead(Grove_Water_Sensor);
int fire = digitalRead(FLAME);
garage = digitalRead (tiltSwitch) ;
int antlia =digitalRead(ANTLIA);

Wait_between_dumps++;
if (Wait_between_dumps >=10) //αποστολή τιμών ανά 2 sec
{
```



```

    if (client.connect(serv, 80)) { //Connecting at the IP address and
port we saved before
Serial.println("connected");
client.print("GET /ethernet/data.php?"); //Connecting and Sending
values to database

client.print("temperature=");
client.print(temp);
client.print("&&humidity=");
client.print(hum);
client.print("&&FAN=");
client.print( fan );
client.print("&&FIRE=");
client.print( !fire );
client.print("&&FLOOD=");
client.print( flood );
client.print("&&ANTLIA=");
client.print(antlia);
client.print("&&GARAGE=");
client.print( !garage );
Wait_between_dumps = 0;

client.println( " HTTP/1.1");
client.print( "Host: " );
client.println(serv);
client.println( "Connection: close" );
client.println();
client.println();

client.stop(); //Closing the connection
}
else {
// if you didn't get a connection to the server:
Serial.println("connection failed");
}

}
}

```

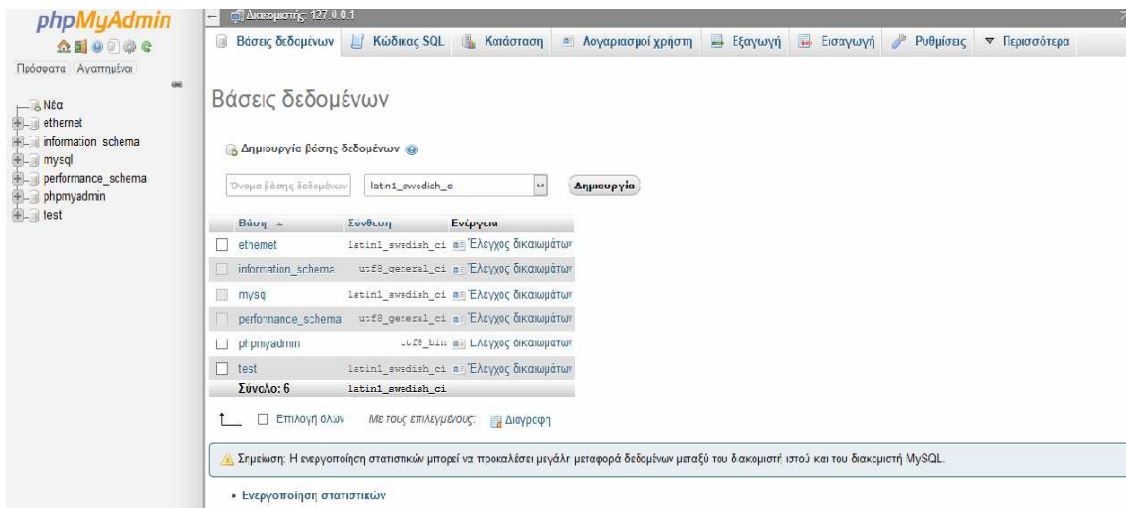
5.1.2 Βάση Δεδομένων

Η βάση δεδομένων δημιουργήθηκε για την καταγραφή των τιμών των αισθητήρων και την αποθήκευση τους για περαιτέρω και αξιοποίηση τους από τον χρήστη. Είναι μία ελεύθερη βάση δεδομένων MySQL και υλοποιήθηκε με την βοήθεια του Apache Xampp και των αρχείων σε μορφή SQL. Στη συνέχεια για να αποτυπωθούν καλύτερα οι τιμές των αισθητήρων, υλοποιήθηκαν γραφήματα ως προς το χρόνο με την βοήθεια των Google Charts τα οποία αντλούν τις τιμές από την βάση δεδομένων. Τα γραφήματα Google Charts θα αναφερθούν σε επόμενο υποκεφάλαιο.

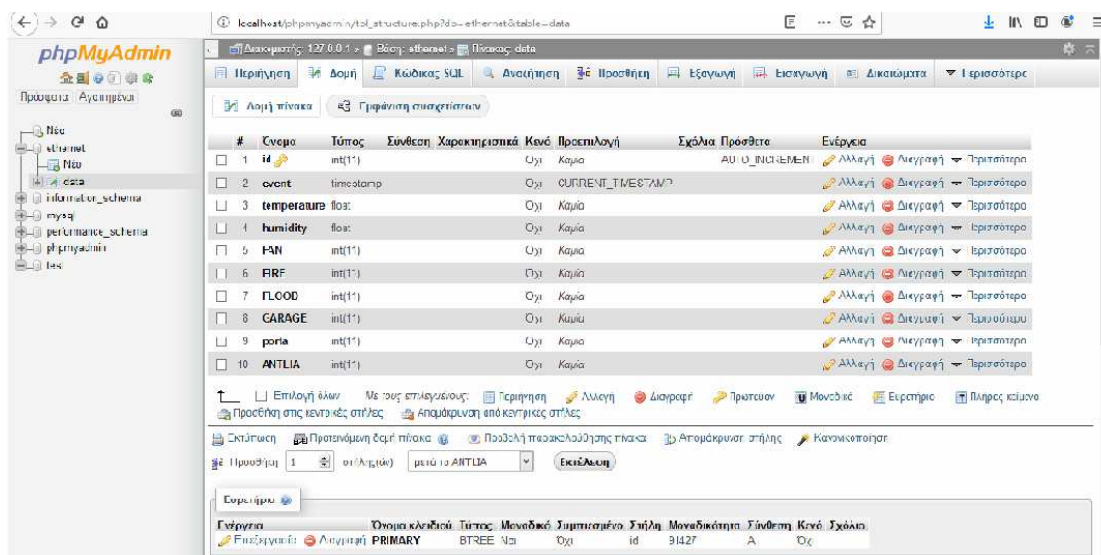
Για να επιτευχθεί η καταγραφή των τιμών θα χρειαστεί να συνδεθεί το Arduino με το Ethernet shield. Επίσης θα χρειαστούν οι βιβλιοθήκες Ethernet.h και SPI.h για την λειτουργία του Ethernet και την σύνδεση του στη βάση δεδομένων.

Όπως φαίνεται στην ενότητα 5.1.1 από τον κώδικα θα χρειαστεί και η δήλωση των αισθητήρων, ώστε με την εντολή client.print να αντλήσει τις τιμές τους και να επικοινωνήσει με την βάση δεδομένων.

Η βάση δεδομένων που υλοποιήθηκε έχει το όνομα “Ethernet” και αποτελείται από τον πίνακα “data”. Για την δημιουργία της βάσης ανοίγουμε την σελίδα phpMyAdmin και φτιάχνουμε μία νέα βάση δεδομένων, δίνοντας της ένα όνομα.



Εικόνα 85: Δημιουργία βάσης δεδομένων



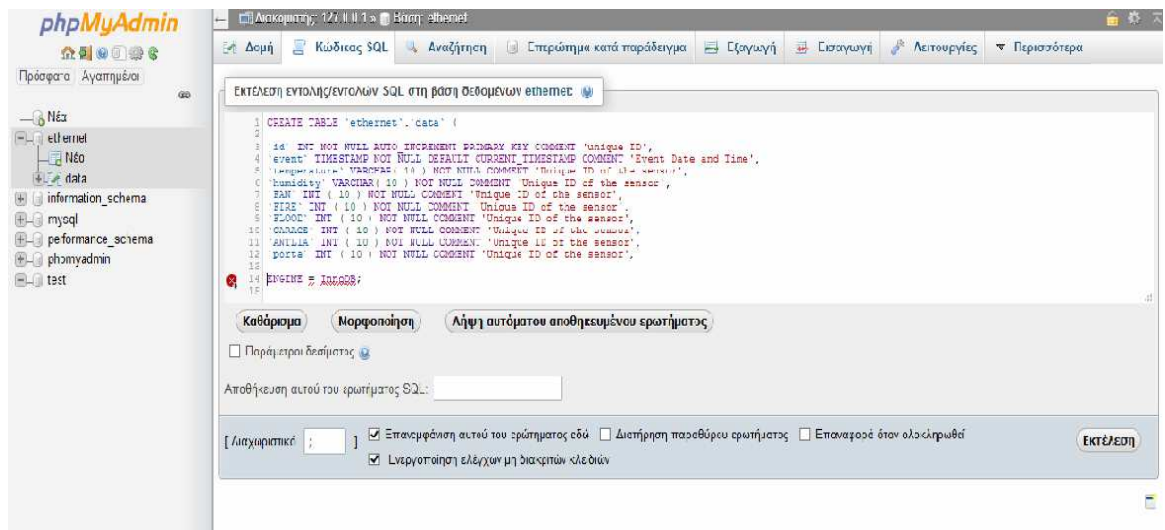
Εικόνα 86: Δομή βάσης δεδομένων

Πίνακας MySQL

Για να δημιουργηθεί ο πίνακας data στην SQL ο οποίος θα περιέχει 10 στήλες για την καταγραφή των τιμών, θα χρειαστούν λίγες γραμμές κώδικα SQL. Ο πίνακας μπορεί επίσης να φτιαχτεί και από το phpMyAdmin όπως φαίνεται παρακάτω.

Ο κώδικας SQL του πίνακα:

```
CREATE TABLE `ethernet`.`data` (  
[97]  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY COMMENT 'unique ID',  
  `event` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT 'Event  
Date and Time',  
  `temperature` VARCHAR( 10 ) NOT NULL COMMENT 'Unique ID of the  
sensor',  
  `humidity` VARCHAR( 10 ) NOT NULL COMMENT 'Unique ID of the sensor',  
  `FAN` INT ( 10 ) NOT NULL COMMENT 'Unique ID of the sensor',  
  `FIRE` INT ( 10 ) NOT NULL COMMENT 'Unique ID of the sensor',  
  `FLOOD` INT ( 10 ) NOT NULL COMMENT 'Unique ID of the sensor',  
  `GARAGE` INT ( 10 ) NOT NULL COMMENT 'Unique ID of the sensor',  
  `ANTLIA` INT ( 10 ) NOT NULL COMMENT 'Unique ID of the sensor',  
  `porta` INT ( 10 ) NOT NULL COMMENT 'Unique ID of the sensor',  
)  
ENGINE = InnoDB;
```



Εικόνα 87: Δημιουργία πίνακα σε SQL

Ο παρακάτω πίνακας χρησιμοποιείται για την καταχώριση των μετρήσεων των αισθητήρων και την καταγραφή της ώρας κατά την οποία έγινε εγγραφή. Ακολουθεί αναλυτική περιγραφή των πεδίων του:

- **id:** Αποτελεί το αναγνωριστικό κάθε εγγραφής. Σε κάθε νέα εγγραφή που πραγματοποιείται έχει προγραμματιστεί να αυξάνεται κατά 1 (Πρόσθετη επιλογή AUTO_INCREMENT). Είναι ακέραιος έως 11 ψηφίων.
- **event:** Γίνεται εγγραφή της ακριβής ημερομηνίας και ώρας (π.χ. 18/07/2019 20:38:12) ώστε ο χρήστης να γνωρίζει ακριβώς πότε έγινε η κάθε καταγραφή δεδομένων.
- **temperature:** Αποθηκεύει την τιμή της θερμοκρασίας περιβάλλοντος.
- **humidity:** Αποθηκεύει την τιμή της υγρασίας περιβάλλοντος.
- **FAN:** Αποθηκεύει την τιμή του εξαερισμού.
- **FIRE:** Αποθηκεύει την τιμή ανίχνευσης της φωτιάς.
- **FLOOD:** Αποθηκεύει την τιμή ανίχνευσης της πλημμύρας.
- **GARAGE:** Αποθηκεύει την τιμή της γκαραζόπορτας όταν αυτή είναι ανοιχτή ή κλειστή.
- **ANTLIA:** Αποθηκεύει την τιμή της αντλίας ανελκυστήρα με ή χωρίς νερό.

Αφού τελείωσε η κατασκευή του πίνακα θα χρειαστεί να δημιουργηθούν τρία αρχεία PHP. Τα αρχεία αυτά θα αποθηκευτούν στο htdocs της Xampp στο φάκελο που δημιουργήσαμε με το όνομα Ethernet.

Εμφάνιση εγγραφών 0 - 24 (91427 συνολικά. Το ερώτημα χρειάστηκε 0.0300 δευτερόλεπτα.)

SELECT * FROM `data`

Δημιουργία προφίλ [Επεξεργασία εισαγωγικά] [Επεξεργασία] [Ανάλυση SQL] [Δημιουργία κώδικα PHP] [Ανανέωση]

1 >>> | Αριθμός εγγραφών: 25 | Φιλτράρισμα εγγραφών: Αναζήτηση σε αυτόν τον | Ταξινόμηση από κλειδί: Καμία

	id	event	temperature	humidity	FAN	FIRE	FLOOD	GARAGE	porta	ANTLIA
<input type="checkbox"/>	1	2019-06-02 13:16:43	20	60	0	0	0	0	0	0
<input type="checkbox"/>	2	2019-05-02 13:27:54	23	75	0	0	0	0	0	0
<input type="checkbox"/>	3	2019-06-02 13:47:48	23	65	0	0	0	0	0	0
<input type="checkbox"/>	4	2019-06-02 13:47:54	23	55	0	0	0	0	0	0
<input type="checkbox"/>	5	2019-06-02 13:47:59	23	55	0	0	0	0	0	0
<input type="checkbox"/>	6	2019-05-02 13:48:04	23	55	0	0	0	0	0	0
<input type="checkbox"/>	7	2019-05-02 13:48:09	23	55	0	0	0	0	0	0
<input type="checkbox"/>	8	2019-05-02 13:48:14	23	55	0	0	0	0	0	0
<input type="checkbox"/>	9	2019-05-02 13:48:19	23	55	0	0	0	0	0	0
<input type="checkbox"/>	10	2019-05-02 13:48:24	23	55	0	0	0	0	0	0
<input type="checkbox"/>	11	2019-05-02 13:48:29	23	55	0	0	0	0	0	0
<input type="checkbox"/>	12	2019-05-02 13:48:34	23	55	0	0	0	0	0	0
<input type="checkbox"/>	13	2019-05-02 13:48:40	23	55	0	0	0	0	0	0
<input type="checkbox"/>	14	2019-05-02 13:48:45	23	55	0	0	0	0	0	0

Πίνακας 5: Βάση δεδομένων

Σύνδεση MySQL με PHP

Το πρώτο αρχείο θα ονομαστεί [connection.php](#) το οποίο θα χρησιμοποιηθεί για την σύνδεση μεταξύ της PHP και της MySQL. Επίσης θα χρησιμοποιηθεί για να στέλνει και να λαμβάνει δεδομένα.

Ο κώδικας του [connection.php](#):

```
<?php
$username = "root";// enter your username for mysql
$password = "";//enter your password for mysql
$host = "localhost"; // this is usually "localhost" unless your
database resides on a different server
$db_name = "ethernet"; // enter your database name
$con = mysqli_connect ($host, $username, $pass);
$db = mysqli_select_db ( $con, $db_name );
?>
```

Φόρτωση δεδομένων

Το δεύτερο αρχείο θα ονομαστεί [data.php](#) το οποίο θα χρησιμοποιηθεί για την αποστολή των τιμών των αισθητήρων από το Arduino προς την βάση δεδομένων.

Ο κώδικας του [data.php](#):

```
<?php
include ('connection.php');
$sql_insert = "INSERT INTO data (temperature,
humidity, FAN, FIRE, FLOOD, GARAGE, ANTLIA, porta) VALUES
('".$_GET["temperature"]."', '".$_GET["humidity"]."',
'".$_GET["FAN"]."', '".$_GET["FIRE"]."', '".$_GET["FLOOD"]."',
'".$_GET["GARAGE"]."', '".$_GET["ANTLIA"]."', '".$_GET["porta"]."'");
if(mysqli_query($con,$sql_insert))
{
echo "Done";
mysqli_close($con);
}
else
{
echo "error is ".mysqli_error($con);
}
?>
```

Εμφάνιση περιεχομένου πίνακα

Το τρίτο αρχείο θα ονομαστεί [display.php](#) το οποίο θα εμφανίζει το περιεχόμενο του πίνακα, αφού πρώτα θα έχουν σταλεί οι τιμές στη βάση δεδομένων σε συνεργασία με τα δύο προηγούμενα αρχεία το [connection.php](#) και το [data.php](#). Ο browser θα ανανεώνεται ανά 5 δευτερόλεπτα και θα φαίνεται άλλη μία γραμμή με δεδομένα να εμφανίζεται στις στήλες που έχουν οριστεί.

Ο κώδικας του [display.php](#):

```
<?php
$url=$_SERVER['REQUEST_URI'];
header("Refresh: 5; URL=$url"); // Refresh the webpage every seconds
?>
<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <title>Smart Home &mdash; Emergency Bracelet &amp;
Footer</title>
        <style type="text/css">

            body {
                margin:0;
                padding:0;
                font-family: Sans-Serif;
                line-height: 1.5em;
                background: #CBFCE4;
            }

            #header {
                background: #B4E8CF;
                height: 100px;
            }

            #header h1 {

                margin: 0;
                padding-top: 15px;

            }

            main {
                padding-bottom: 10010px;
                margin-bottom: -10000px;
                float: left;
                width: 100%;
            }

            #nav {
                padding-bottom: 10010px;
                margin-bottom: -10000px;
                float: left;
                width: 230px;
                margin-left: -100%;
                background: #71E3AD;
```

```

    }

    #footer {
        clear: left;
        width: 100%;
        background:#B4E8CF ;
        text-align: center;
        padding: 4px 0;
    }

    #wrapper {
        overflow: hidden;
    }

    #content {
        margin-left: 230px; /* Same as 'nav' width */
    }

    .innertube {
        margin: 15px; /* Padding for content */
        margin-top: 0;
    }

    p {
        color: #000000;
    }

    nav ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
    }

    nav ul a {
        color: black;
        text-decoration: none;
    }
}

</style>

```

```

<title>Arduino Smart Home</title>
<style type="text/css">
.table_titles {
padding-right: 20px;
padding-left: 20px;
color: #999;
}

.table_titles {
color: #000000;
background-color:#71E3AD;
}

table {
border: 2px solid #555;
}
body { font-family: "Trebuchet MS", Courier;
background-color:#CBFCE4;
}
</style>
</head>

```

```

<body>
    <header id="header">
        <div class="innertube">
            <h1>Smart home and emergency bracelet for
elderly</h1>
        </div>
    </header>

    <div id="wrapper">
        <main>

            <div id="content">

                <div class="content">

<h1>SMART HOME</h1>
<table border="0" cellspacing="0" cellpadding="4">
<tr>
<td class="table_titles">ID</td>
<td class="table_titles">Date and Time</td>
<td class="table_titles">Temperature</td>
<td class="table_titles">Humidity</td>
<td class="table_titles">FAN</td>
<td class="table_titles">FIRE</td>
<td class="table_titles">FLOOD</td>
<td class="table_titles">GARAGE</td>
<td class="table_titles">ANTLIA</td>
<td class="table_titles">porta</td>
<!--<td class="table_titles">Heat_index</td-->
</tr>

<?php
include('connection.php');
$result = mysqli_query($con,'SELECT * FROM data ORDER BY id DESC');
// Process every record
$rowodd = true;
while($row = mysqli_fetch_array($result))
{
if ($rowodd)
{
$css_class=' class="table_cells_odd"';
}
else
{
$css_class=' class="table_cells_even"';
}
$rowodd = !$rowodd;
echo "<tr>";
echo "<td '." . $css_class . "'>" . $row['id'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['event'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['temperature'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['humidity'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['FAN'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['FIRE'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['FLOOD'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['GARAGE'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['ANTLIA'] . "</td>";
echo "<td '." . $css_class . "'>" . $row['porta'] . "</td>";
echo "</tr>";
}
}

```

```

// Close the connection
mysqli_close($con);
?>
</table>
  </div>
</div>
      </div>
</main>

  <nav id="nav">
    <div class="innertube">
      <h3>SMART HOME</h3>
      <ul>
        <li><a href="index.html">First Page</a></li>
      </ul>
      <h3>thingspeak</h3>
      <ul>
        <li><a
href="temperature.php">Temperature</a></li>
        <li><a
href="humidity.php">Humidity</a></li>
        <li><a
href="fan.php">Fan</a></li>
        <li><a
href="fire.php">Fire</a></li>
        <li><a href="flood.php">Flood
</a></li>
        <li><a href="garage.php">Garage
</a></li>
        <li><a
href="antlia.php">Antlia</a></li>
        <li><a
href="porta.php">Porta</a></li>
      </ul>
      <h3>Google charts</h3>
      <ul>
        <li><a
href="statistics1.php">Statistics</a></li>
        <li><a
href="gauge.php">Gauge</a></li>
        <li><a
href="weekly_statistics1.php">Weekly Statistics</a></li>
        <li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
      </ul>
      <h3>Data base</h3>
      <ul>
        <li><a
href="display.php">Display</a></li>
      </ul>
      <h3>Ajax I/O</h3>
      <ul>
        <li><a
href="http://192.168.1.5">Control Led and Relay</a></li>
      </ul>
      <h3>EMERGENCY BRACELET</h3>
      <h3>thingspeak</h3>
      <ul>
        <li><a
href="temperarture1.php">Temperature</a></li>
        <li><a
href="humidity1.php">Humidity</a></li>
        <li><a href="heartrate.php">Heart

```

```




Rate</a></li>
Detection </a></li>
href="button.php">Emergency Button</a></li>
</ul>
</div>
</nav>
</div>
<footer id="footer">
<div class="innertube">
<p>
<span>2019 &copy; </span><a href="#"
>Πανεπιστήμιο Πελοποννήσου</a> | <a href="#" >Πάτρα</a>
</p>
</div>
</footer>
</body>
</html>


```

SMART HOME

ID	Date and Time	Temperature	Humidity	FAN	FIRE	FLOOD	GARAGE	ANTLIA	porta
106958	2019-07-18 12:37:41	28	42	0	1	0	1	1	0
106957	2019-07-18 12:37:37	28	41	0	1	0	1	1	0
106956	2019-07-18 12:37:33	28	42	0	1	0	1	1	0
106955	2019-07-18 12:37:30	28	42	0	1	0	0	0	0
106954	2019-07-18 12:37:28	28	42	0	1	0	0	0	0
106953	2019-07-18 12:37:26	28	42	0	0	0	0	0	0
106952	2019-07-18 12:37:24	28	42	0	0	0	0	0	0
106951	2019-07-18 12:37:22	28	42	0	0	0	0	0	0
106950	2019-07-18 12:37:20	28	42	0	0	0	0	0	0
106949	2019-07-18 12:37:18	28	42	0	0	0	0	0	0
106948	2019-07-18 12:37:17	28	42	0	0	0	0	0	0

Εικόνα 88: Τα δεδομένα της βάσης από τον browser του υπολογιστή

1:45 MM    92

192.168.1.2/ethernet/display.php 2 

Smart home and emergency bracelet for elderly

SMART HOME

ID	Date and Time	Temperature	Humidity	FAN	FIRE	FLOOD	GARAGE
96002	2019-07-09 13:45:04	31	57	0	0	0	0
96001	2019-07-09 13:45:02	31	57	0	0	0	0
96000	2019-07-09 13:45:00	31	57	0	0	0	0
95999	2019-07-09 13:44:58	31	57	0	0	0	0
95998	2019-07-09 13:44:56	31	57	0	0	0	0
95997	2019-07-09 13:44:55	31	57	0	0	0	0
95996	2019-07-09 13:44:53	31	57	0	0	0	0
95995	2019-07-09 13:44:50	31	57	0	0	0	0

Εικόνα 89: Τα δεδομένα της βάσης από τον browser του κινητού

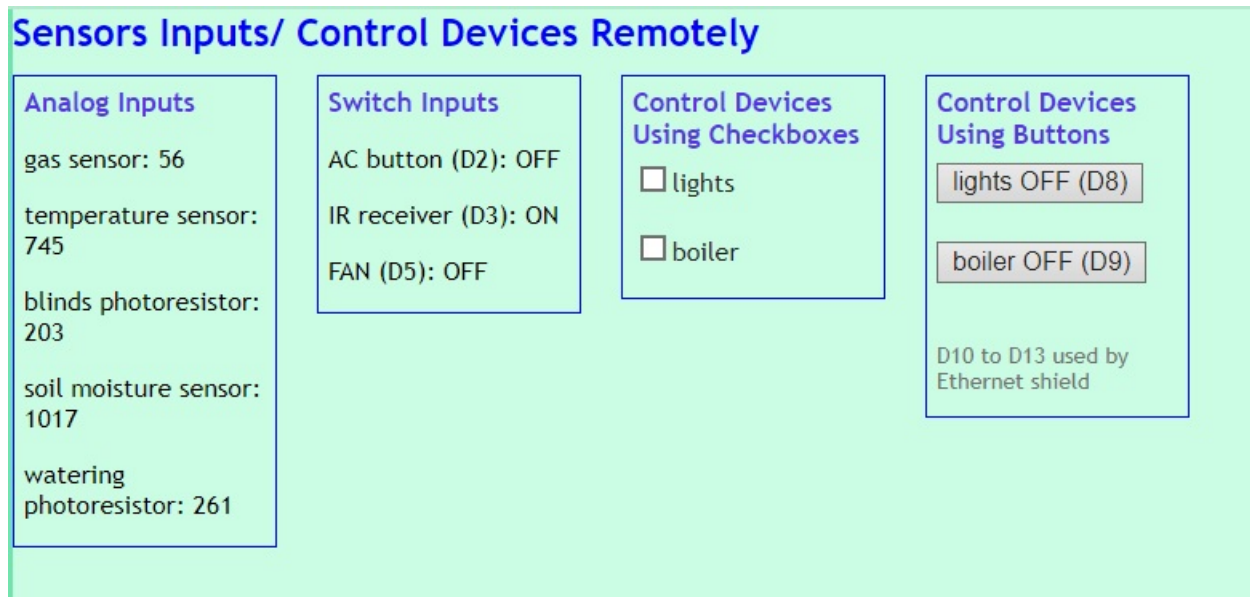
5.2 Ένδειξη Αναλογικών-ψηφιακών εισόδων και έλεγχος ψηφιακών εξόδων με Ajax μέσω κάρτας SD

Για αυτήν την υποενότητα δημιουργήθηκε ιστοσελίδα η οποία διαβάζεται από την κάρτα SD. Αυτή εμφανίζει αναλογικές και ψηφιακές εισόδους του Arduino και δίνει την απομακρυσμένου δυνατότητα ελέγχου σε ψηφιακές εξόδους-συσκευές.

Ο Arduino web server φιλοξενεί μια ιστοσελίδα στην SD κάρτα του Ethernet shield που εμφανίζει τέσσερις τιμές αναλογικών εισόδων και την κατάσταση τριών διακοπών. Η ιστοσελίδα επιτρέπει επίσης τον έλεγχο φωτισμού και θερμοσίφωνα(boiler) με checkboxes ή buttons.

Αφού το πρόγραμμα περιήγησης ιστού ζητήσει και φορτώσει την ιστοσελίδα από τον Arduino web server, το JavaScript στη σελίδα θα στείλει ένα αίτημα Ajax για λήψη δεδομένων από το Arduino κάθε δευτερόλεπτο.

Η ιστοσελίδα που φιλοξενεί ο Arduino web server στην SD κάρτα του Ethernet shield εμφανίζεται εδώ:



Εικόνα 90: Ιστοσελίδα Arduino Web Server

Το Arduino θα ανταποκριθεί σε κάθε αίτημα του Ajax στέλνοντας ένα αρχείο XML πίσω στο πρόγραμμα περιήγησης ιστού. Το αρχείο XML θα περιέχει τις τιμές από τις πέντε αναλογικές εισόδους του Arduino (A0 to A4) την κατάσταση των τριών pins (ψηφιακά pins 2,3,5) και την κατάσταση του φωτισμού και του θερμοσίφωνα.

Όταν ο φωτισμός-θερμοσίφωνασ ανάψει από την ιστοσελίδα, επιλέγοντας ένα checkbox, τότε το JavaScript θα στείλει την κατάσταση του checkbox (αποστολή εντολής για την ενεργοποίηση του φωτισμού- θερμοσίφωνα), με το επόμενο αίτημα του Ajax. Αντίστοιχη είναι η διαδικασία σε περίπτωση απενεργοποίησης του φωτισμού-θερμοσίφωνα, καθώς και αν ο έλεγχος γίνει με τα button.

Για την εκτέλεση του συστήματος αυτού, θα πρέπει το αρχείο index1.htm να τοποθετηθεί στην κάρτα micro SD και αυτή αντίστοιχα στην υποδοχή του Ethernet shield, που επικοινωνεί με το arduino, στο οποίο είναι συνδεδεμένοι οι αισθητήρες και οι συσκευές ελέγχου. Η πρόσβαση στην ιστοσελίδα γίνεται από την ip που έχει δοθεί στο arduino από το δίκτυο. Στην συγκεκριμένη περίπτωση είναι το 192.168.1.5 .

Ο κώδικας για την υλοποίηση ελέγχου inputs- outputs, ο οποίος τρέχει στο Arduino Mega είναι ο εξής:

```
#include <SPI.h> // βιβλιοθήκη σειριακής επικοινωνίας Arduino με Ethernet shield
#include <Ethernet.h> // βιβλιοθήκη ethernet
#include <SD.h> // βιβλιοθήκη αναγνώρισης κάρτας SD
#define REQ_BUF_SZ 60
File webFile; // the web page file on the SD card
```

```

char HTTP_req[REQ_BUF_SZ] = {0}; // buffered HTTP request stored as
null terminated string
char req_index = 0; // index into HTTP_req buffer
EthernetServer server1(80);
boolean LED_state[4] = {0}; // Αποθηκεύει τις καταστάσεις των
συσκευών δράσης
byte mac[] = { 0x90, 0xA2, 0xDA, 0x10, 0x40, 0x4F };
IPAddress ip(192, 168, 1, 5);
IPAddress serv(192, 168, 1, 2)

void setup(){
pinMode(10, OUTPUT); //Απενεργοποίηση Ethernet chip
digitalWrite(10, HIGH); //Ενεργοποίηση Ethernet chip
Serial.begin(9600); //setting the baud rate at 9600
SD.begin(4);

pinMode(30, OUTPUT);
pinMode(RELAY1, OUTPUT); //rele
Ethernet.begin(mac, ip); // initialize Ethernet device
server1.begin(); // start to listen for clients
}

void loop()
{
//Επικοινωνία Arduino με ιστοσελίδα

EthernetClient client = server1.available(); // try to get client
if (client) { // got client?
boolean currentLineIsBlank = true;
while (client.connected()) {
if (client.available()) { // client data available to read
char c = client.read(); // read 1 byte (character) from client
// limit the size of the stored received HTTP request
// buffer first part of HTTP request in HTTP_req array (string)
// leave last element in array as 0 to null terminate string
(REQ_BUF_SZ - 1)
if (req_index < (REQ_BUF_SZ - 1)) {
HTTP_req[req_index] = c; // save HTTP request
character
req_index++;
}
// last line of client request is blank and ends with \n
// respond to client only after last line received
if (c == '\n' && currentLineIsBlank) {
// send a standard http response header
client.println("HTTP/1.1 200 OK");
// remainder of header follows below, depending on if
// web page or XML page is requested
// Ajax request - send XML file
if (StrContains(HTTP_req, "ajax_inputs")) {
// send rest of HTTP header
client.println("Content-Type: text/xml");
client.println("Connection: keep-alive");
client.println();
SetLEDs();
// send XML file containing input states
XML_response(client);
}
else { // web page request
// send rest of HTTP header
client.println("Content-Type: text/html");
client.println("Connection: keep-alive");
client.println();
}
}
}
}

```

```

    // send web page
    webFile = SD.open("index1.htm");    // open web page file
    if (webFile) {
        while(webFile.available()) {
            client.write(webFile.read()); //send web page to client
        }
        webFile.close();
    }
}
// display received HTTP request on serial port
Serial.print(HTTP_req);
// reset buffer index and all buffer elements to 0
req_index = 0;
StrClear(HTTP_req, REQ_BUF_SZ);
break;
}
// every line of text received from the client ends with \r\n
if (c == '\n') {
    // last character on line of received text
    // starting new line with next character read
    currentLineIsBlank = true;
}
else if (c != '\r') {
    // a text character was received from client
    currentLineIsBlank = false;
}
} // end if (client.available())
} // end while (client.connected())
delay(1); // give the web browser time to receive the data
client.stop(); // close the connection
} // end if (client)
}

// checks if received HTTP request is switching on/off LEDs
// also saves the state of the LEDs
void SetLEDs(void)
{
    // LED 1 (pin 6)
    if (StrContains(HTTP_req, "LED1=1")) {
        LED_state[0] = 1; // save LED state
        digitalWrite(30, HIGH);
    }
    else if (StrContains(HTTP_req, "LED1=0")) {
        LED_state[0] = 0; // save LED state
        digitalWrite(30, LOW);
    }
    // LED 2 (pin 7)
    if (StrContains(HTTP_req, "LED2=1")) {
        LED_state[1] = 1; // save LED state
        digitalWrite(47, HIGH);
    }
    else if (StrContains(HTTP_req, "LED2=0")) {
        LED_state[1] = 0; // save LED state
        digitalWrite(47, LOW);
    }
    // LED 3 (pin 8)
    if (StrContains(HTTP_req, "LED3=1")) {
        LED_state[2] = 1; // save LED state
        digitalWrite(30, HIGH);
    }
    else if (StrContains(HTTP_req, "LED3=0")) {
        LED_state[2] = 0; // save LED state
        digitalWrite(30, LOW);
    }
}

```

```

}
// LED 4 (pin 9)
if (StrContains(HTTP_req, "LED4=1")) {
    LED_state[3] = 1; // save LED state
    digitalWrite(47, HIGH);
}
else if (StrContains(HTTP_req, "LED4=0")) {
    LED_state[3] = 0; // save LED state
    digitalWrite(47, LOW);
}
}

// send the XML file with analog values, switch status
// and LED status
void XML_response(EthernetClient cl)
{
    int analog_val; // stores value read from analog inputs
    int count; // used by 'for' loops
    int sw_arr[] = {2, 3, 5}; // pins interfaced to switches

    cl.print("<?xml version = \"1.0\" ?>");
    cl.print("<inputs>");
    // read analog inputs
    for (count = 0; count <= 4; count++) { // A0 to A4
        analog_val = analogRead(count);
        cl.print("<analog>");
        cl.print(analog_val);
        cl.println("</analog>");
    }
    // read switches
    for (count = 0; count < 3; count++) {
        cl.print("<switch>");
        if (digitalRead(sw_arr[count])) {
            cl.print("ON");
        }
        else {
            cl.print("OFF");
        }
        cl.println("</switch>");
    }
    // checkbox LED states
    // LED1
    cl.print("<LED>");
    if (LED_state[0]) {
        cl.print("checked");
    }
    else {
        cl.print("unchecked");
    }
    cl.println("</LED>");
    // LED2
    cl.print("<LED>");
    if (LED_state[1]) {
        cl.print("checked");
    }
    else {
        cl.print("unchecked");
    }
    cl.println("</LED>");
    // button LED states
    // LED3
    cl.print("<LED>");

```

```

    if (LED_state[2]) {
        cl.print("on");
    }
    else {
        cl.print("off");
    }
    cl.println("</LED>");
    // LED4
    cl.print("<LED>");
    if (LED_state[3]) {
        cl.print("on");
    }
    else {
        cl.print("off");
    }
    cl.println("</LED>");

    cl.print("</inputs>");
}

// sets every element of str to 0 (clears array)
void StrClear(char *str, char length)
{
    for (int i = 0; i < length; i++) {
        str[i] = 0;
    }
}

// searches for the string sfind in the string str
// returns 1 if string found
// returns 0 if string not found
char StrContains(char *str, char *sfind)
{
    char found = 0;
    char index = 0;
    char len;

    len = strlen(str);

    if (strlen(sfind) > len) {
        return 0;
    }
    while (index < len) {
        if (str[index] == sfind[found]) {
            found++;
            if (strlen(sfind) == found) {
                return 1;
            }
        }
        else {
            found = 0;
        }
        index++;
    }

    return 0;
}

```

Ο κώδικας της ιστοσελίδας (index1) που φιλοξενεί ο Arduino web server στην SD κάρτα του Ethernet shield :

```
<!DOCTYPE html>
<html>
<head>
<script>
strLED1 = "";
strLED2 = "";
strLED3 = "";
strLED4 = "";
var LED3_state = 0;
var LED4_state = 0;
function GetArduinoIO()
{
nocache = "&nocache=" + Math.random() * 1000000;
var request = new XMLHttpRequest();
request.onreadystatechange = function()
{
if (this.readyState == 4) {
if (this.status == 200) {
if (this.responseXML != null) {
// XML file received - contains analog values, switch values and
LED states
var count;
// get analog inputs
var num_an =
this.responseXML.getElementsByTagName('analog').length;
for (count = 0; count < num_an; count++) {
document.getElementsByClassName("analog")[count].innerHTML =

this.responseXML.getElementsByTagName('analog')[count].childNodes[0].
nodeValue;
}
// get switch inputs
var num_an =
this.responseXML.getElementsByTagName('switch').length;
for (count = 0; count < num_an; count++) {
document.getElementsByClassName("switches")[count].innerHTML =

this.responseXML.getElementsByTagName('switch')[count].childNodes[0].
nodeValue;
}
// LED 1
if
(this.responseXML.getElementsByTagName('LED')[0].childNodes[0].nodeVa
lue == "checked") {
document.LED_form.LED1.checked = true;
}
else {
document.LED_form.LED1.checked = false;
}
// LED 2
if
(this.responseXML.getElementsByTagName('LED')[1].childNodes[0].nodeVa
lue == "checked") {
document.LED_form.LED2.checked = true;
}
else {
document.LED_form.LED2.checked = false;
}
}
}
```

```

// LED 3
if
(this.responseXML.getElementsByTagName('LED')[2].childNodes[0].nodeValue === "on") {
    document.getElementById("LED3").innerHTML = "lights ON (D8)";
    LED3_state = 1;
}
else {
    document.getElementById("LED3").innerHTML = "lights OFF (D8)";
    LED3_state = 0;
}
// LED 4
if
(this.responseXML.getElementsByTagName('LED')[3].childNodes[0].nodeValue === "on") {
    document.getElementById("LED4").innerHTML = "boiler ON (D9)";
    LED4_state = 1;
}
else {
    document.getElementById("LED4").innerHTML = "boiler OFF (D9)";
    LED4_state = 0;
}
}
}
}
}

// send HTTP GET request with LEDs to switch on/off if any
request.open("GET", "ajax_inputs" + strLED1 + strLED2 + strLED3 +
strLED4 + nocache, true);
request.send(null);
setTimeout('GetArduinoIO()', 1000);
strLED1 = "";
strLED2 = "";
strLED3 = "";
strLED4 = "";
}
// service LEDs when checkbox checked/unchecked
function GetCheck()
{
if (LED_form.LED1.checked) {
    strLED1 = "&LED1=1";
}
else {
    strLED1 = "&LED1=0";
}
if (LED_form.LED2.checked) {
    strLED2 = "&LED2=1";
}
else {
    strLED2 = "&LED2=0";
}
}
}

function GetButton1()
{
if (LED3_state === 1) {
    LED3_state = 0;
    strLED3 = "&LED3=0";
}
else {
    LED3_state = 1;
    strLED3 = "&LED3=1";
}
}

```



```

}
}

function GetButton2()
{
if (LED4_state === 1) {
    LED4_state = 0;
    strLED4 = "&LED4=0";
}
else {
    LED4_state = 1;
    strLED4 = "&LED4=1";
}
}
</script>
<style>
body
    .IO_box {
        float: left;
        margin: 0 20px 20px 0;
        border: 1px solid blue;
        padding: 0 5px 0 5px;
        width: 120px;
    }
    h1 {
        font-size: 120%;
        color: blue;
        margin: 0 0 10px 0;
    }
    h2 {
        font-size: 85%;
        color: #5734E6;
        margin: 5px 0 5px 0;
    }
    p, form, button {
        font-size: 80%;
        color: #252525;
    }
    .small_text {
        font-size: 70%;
        color: #737373;
    }
</style>
</head>

<body onload="GetArduinoIO()" >
<h1>Sensors Inputs/ Control Devices Remotely</h1>
<div class="IO_box">
<h2>Analog Inputs</h2>
<p>gas sensor: <span class="analog">...</span></p>
<p>temperature sensor: <span class="analog">...</span></p>
<p>blinds photoresistor: <span class="analog">...</span></p>
<p>soil moisture sensor: <span class="analog">...</span></p>
<p>watering photoresistor: <span class="analog">...</span></p>

</div>
<div class="IO_box">
<h2>Switch Inputs</h2>
<p>AC button (D2): <span class="switches">...</span></p>
<p>IR receiver (D3): <span class="switches">...</span></p>

<p>FAN (D5): <span class="switches">...</span></p>
</div>

```

```

<div class="IO_box">
<h2>Control Devices Using Checkboxes</h2>
<form id="check_LEDs" name="LED_form">
<input type="checkbox" name="LED1" value="0" onclick="GetCheck()"
/>lights<br /><br />
<input type="checkbox" name="LED2" value="0" onclick="GetCheck()"
/>boiler<br /><br />
</form>
</div>
<div class="IO_box">
<h2>Control Devices Using Buttons</h2>
<button type="button" id="LED3" onclick="GetButton1()">lights OFF
(D8)</button><br /><br />
<button type="button" id="LED4" onclick="GetButton2()">boiler OFF
(D9)</button><br /><br />
<p class="small_text">D10 to D13 used by Ethernet shield</p>
</div>
</body>
</html>

```

5.3 ThingSpeak



Εικόνα 91: Λογότυπο Thingpeak

Το βασικό στοιχείο του ThingSpeak είναι το "ThingSpeak Channel". Ένα κανάλι αποθηκεύει τα δεδομένα που στέλνουμε στο ThingSpeak και περιλαμβάνει τα παρακάτω στοιχεία:

- 8 πεδία για την αποθήκευση δεδομένων οποιουδήποτε τύπου - Αυτά μπορούν να χρησιμοποιηθούν για την αποθήκευση δεδομένων από αισθητήρα ή από ενσωματωμένη συσκευή.
- 3 πεδία τοποθεσίας - Μπορούν να χρησιμοποιηθούν για την αποθήκευση του γεωγραφικού πλάτους, του γεωγραφικού μήκους και του υψομέτρου. Αυτά είναι πολύ χρήσιμα για την παρακολούθηση μιας κινούμενης συσκευής.
- 1 πεδίο κατάστασης - Ένα σύντομο μήνυμα για την περιγραφή των δεδομένων που είναι αποθηκευμένα στο κανάλι.

Για να χρησιμοποιήσει κάποιος το ThingSpeak, πρέπει να εγγραφεί και να δημιουργήσει ένα κανάλι. Αφού φτιάξει το κανάλι, μπορεί να στείλει δεδομένα, να τα επεξεργαστεί μέσω του ThingSpeak και επίσης να τα ανακτήσει.

Αρχικά λοιπόν χρειάζεται η εγγραφή στο ThingSpeak με την επιλογή sign up και την εισαγωγή των βασικών στοιχείων (email κ.τ.λ.)

Sign up for ThingSpeak

It is free to sign up for ThingSpeak. Free accounts offer a fully functional experience on ThingSpeak with limits on certain functionality. Commercial users may sign up for a time-limited free evaluation. To send data faster to ThingSpeak or to send more data, consider our [paid license options](#) for commercial, academic, home and student usage. To start using ThingSpeak you must create a new MathWorks account, or, click cancel and log in using an existing MathWorks account.

Create MathWorks Account

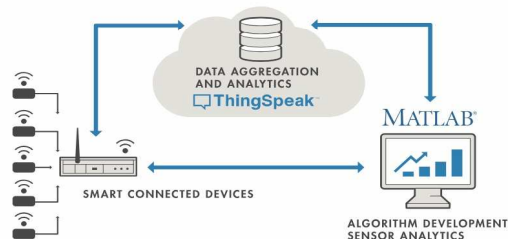
Email Address

To access your organization's MATLAB license, use your school or work email.

Location
 United States

First Name

Last Name



Εικόνα 92: Εγγραφή στο Thingspeak

5.3.1 Έξυπνο σπίτι

Δημιουργία καναλιού(New Channel) :



Εικόνα 93: Δημιουργία καναλιού

Για το έξυπνο σπίτι δημιουργήθηκε κανάλι (smart home) όπου συμπληρώθηκαν τα πεδία με τις τιμές που θα εμφανίζονται σε διαγράμματα:

1. temperature: θερμοκρασία οικίας
2. humidity: υγρασία οικίας
3. fan: εξαερισμός (0 ή 1)
4. fire: φωτιά (0 ή 1)
5. flood: πλυμμήρα (0 ή 1)
6. garage: γκαραζόπορτα (0 ή 1)
7. antlia: αντλία ανελκυστήρα (0 ή 1)
8. porta: εξώπορτα (0 ή 1)

Εικόνα 94: Δημιουργία καναλιού (1)
(2)

Εικόνα 95: Δημιουργία καναλιού

Αφού συμπληρωθούν οι τιμές χρειάζεται αποθήκευση με την επιλογή **Save Channel**.

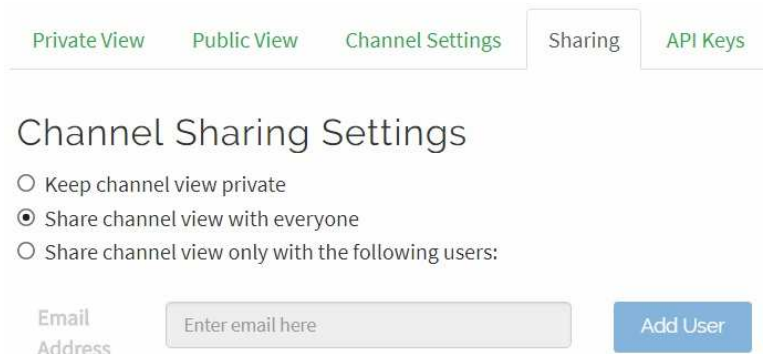
Τα κλειδιά API επιτρέπουν την εγγραφή δεδομένων σε ένα κανάλι ή την ανάγνωση δεδομένων από ένα ιδιωτικό κανάλι. Τα κλειδιά API δημιουργούνται αυτόματα κατά δημιουργία ενός νέου καναλιού.

Write API Key: Αυτό το κλειδί χρησιμοποιείται για την εγγραφή δεδομένων σε ένα κανάλι.

Read API Keys: Αυτό το κλειδί χρησιμοποιείται ώστε κάποιος να μπορεί να προβάλει ροές ή διαγράμματα ενός καναλιού.

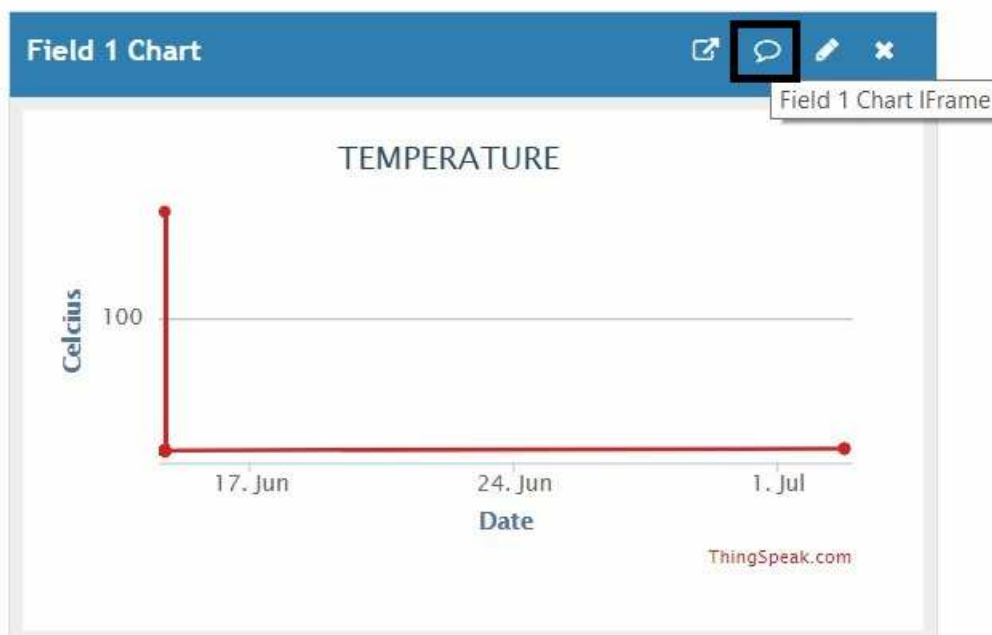
Εικόνα 96: Write/Read API Keys

Στην καρτέλα Sharing υπάρχει η δυνατότητα να επιλεγεί ποιος μπορεί να προβάλει τα δεδομένα του καναλιού. Για την δημιουργία της ιστοσελίδας χρειάστηκε η επιλογή **Share channel view with everyone** ώστε να εμφανίζονται τα διαγράμματα. (στο ιδιωτικό κανάλι δεν εμφανίζονταν τα διαγράμματα)



Εικόνα 97: Channel Sharing Settings

Η επιλογή **Chart IFrame** χρησιμοποιήθηκε για να μπορέσουν να εισαχθούν τα διαγράμματα μέσα στον κώδικα στις ιστοσελίδας



Εικόνα 98: Εξαγωγή IFrame



Εικόνα 99: Field 1 Chart IFrame

Ο κώδικας για την εγγραφή πολλαπλών τιμών στα πεδία ενός καναλιού είναι από τα παραδείγματα της βιβλιοθήκης thingspeak και τρέχει και στα δύο Arduino που είναι για το σπίτι.

```

byte mac[] = { 0x90, 0xA2, 0xDA, 0x10, 0x40, 0x4F };
EthernetClient client;
#include "ThingSpeak.h" // thingspeak
IPAddress ip(192, 168, 1, 5); //ethernet shield ip
IPAddress serv(192, 168, 1, 19); // computer ip

unsigned long myChannelNumber = 774003; // Channel ID
const char * myWriteAPIKey = "WPM6HMY2J5TH6N8Q";
int period = 20000; // send every 20 sec to
thingspeak
unsigned long time_now = 0;

void setup()

Ethernet.init(10); // Most Arduino Ethernet hardware
//thingspeak
//Serial.begin(115200); //Initialize serial

// start the Ethernet connection:
Serial.println("Initialize Ethernet with DHCP");
if (Ethernet.begin(mac) == 0) {
  Serial.println("Failed to configure Ethernet using DHCP");
  // Check for Ethernet hardware present
  if (Ethernet.hardwareStatus() == EthernetNoHardware) {
    Serial.println("Ethernet shield was not found. Sorry, can't
run without hardware. :(");
    while (true) {
      delay(1); // do nothing, no point running without Ethernet
hardware
    }
  }
  if (Ethernet.linkStatus() == LinkOFF) {
    Serial.println("Ethernet cable is not connected.");
  }
  // try to configure using IP address instead of DHCP:
  Ethernet.begin(mac, ip, serv );
} else {

```

```

    Serial.print(" DHCP assigned IP ");
    Serial.println(Ethernet.localIP());
}
// give the Ethernet shield a second to initialize:

delay(1000);

ThingSpeak.begin(client); // Initialize ThingSpeak

void loop()

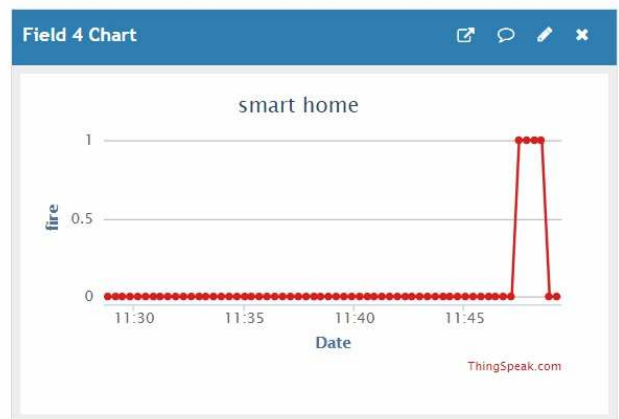
    //thingspeak // update channel every 20
sec
if(millis() > (time_now + period)){
    dht.read11(dht_dp1n);
float temp = dht.temperature;
float hum = dht.humidity;

    ThingSpeak.setField(1, temp); // set fields with values
ThingSpeak.setField(2, hum);
    ThingSpeak.setField(3, fan );
ThingSpeak.setField(4, !fire );
    ThingSpeak.setField(5, flood );
    ThingSpeak.setField(6, !garage );
    ThingSpeak.setField(7, antlia );

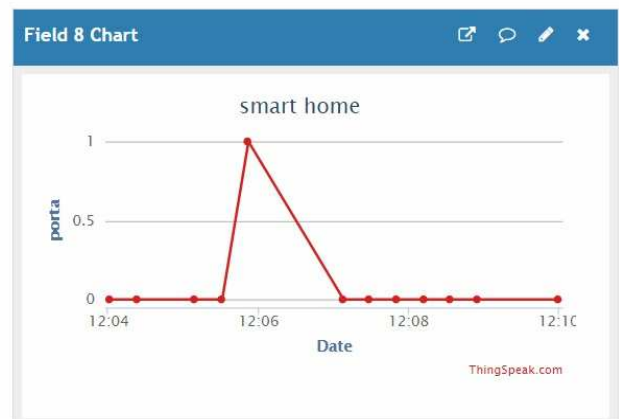
int x =ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    if(x == 200){
        Serial.println("Channel update successful.");
    }
    else{
        Serial.println("Problem updating channel. HTTP error code " +
String(x));
    }

time_now = millis();}

```



Εικόνα 100: Διαγράμματα Thingspeak έξυπνου σπιτιού (1)



Εικόνα 101: Διαγράμματα Thingspeak έξυπνου σπιτιού (2)

5.3.2 Βραχιόλι έκτακτης ανάγκης για ηλικιωμένους ή ατόμων με ιατρικές παθήσεις

5.3.2.1 Διαγράμματα

Για το βραχιόλι έκτακτης ανάγκης δημιουργήθηκε κανάλι (bracelet) όπου συμπληρώθηκαν τα πεδία με τις τιμές που θα εμφανίζονται σε διαγράμματα:

1. heartrate: Ένδειξη καρδιακών παλμών
2. fall: Πτώση (0 ή 1)
3. em. button: Πάτημα emergency button (0 ή 1)
4. temperature: Θερμοκρασία χώρου(0 ή 1)
5. humidity: Υγρασία χώρου

Η διαδικασία δημιουργίας καναλιού, api keys, channel sharing κ.τ.λ. , είναι αντίστοιχη με το έξυπνο στίτι που αναλύεται παραπάνω.

Για την επικοινωνία του βραχιολιού με το διαδίκτυο χρησιμοποιήθηκε το esp8266 wifi module το οποίο επικοινωνεί με το arduino μέσω **AT commands**.

AT COMMAND	DESCRIPTION	PARAMETERS
AT+RST	Reset the module	
AT+CWMODE=<mode>	Client / Access Point mode	1 = Client 2 = AP 3 = Client & AP
AT+CWLAP	List available access points	
AT+CWJAP=<SSID>,<PWD>	Connect to an access point	<SSID> = SSID of access point to join <PWD> = Password of access point
AT+CWQAP	Disconnect from access point	
AT+CWSAP=<SSID>,<PWD>,<CH>,<ENC>	Set parameters as an access point	<SSID> = SSID of access point <PWD> = Password of access point <CH> = 802.11 channel number (1 to 7) <ENC> = Encryption (0 = no encryption, other values not confirmed working)
AT+ CIPSTATUS	Get the current connection status	
AT+CIPMUX=<MUX>	Set single/multiple connection mode	<MUX> Connection mode: 0 = Single 1 = Multiple
AT+CIPSTART=<TYPE>,<URL>,<PORT> OR AT+CIPSTART=<ID>,<TYPE>,<URL>,<PORT>	Enable a UDP or TCP connection	CIPMUX=0: <TYPE> = TCP or UDP <URL> = URL or IP address of server to connect to <PORT> = Port number of server CIPMUX=1: <ID> = Connection ID (0 to 4) <TYPE> = TCP or UDP <URL> = URL or IP address of server to connect to <PORT> = Port number of server
AT+CIPSEND=<LEN> OR AP+CIPSEND=<ID>,<LEN>	Send data to connected server	CIPMUX=0: <LEN> = Number of bytes to send CIPMUX=1: <ID> = Connection ID (0 to 4) <LEN> = Number of bytes to send
AT+CIPCLOSE	Close UDP or TCP connection	
AT+CIFSR	Get the IP address	
AT+CIPSERVER=<MODE>,<PORT>	Set as TCP server	<MODE> = Open/Close server mode: 0 = Close 1 = Open

WWW.HOBBYCOMPONENTS.COM

Εικόνα 102: Βασικές AT commands

Παρακάτω υπάρχει ο κώδικας του βραχιολιού για την σύνδεση με το διαδίκτυο και την αποστολή των δεδομένων στο Thingspeak.

```
#include <SoftwareSerial.h>
#define RX 6
#define TX 7
String AP = "COSMOTE-2AD791"; // CHANGE
String PASS = "xq9y2CxATzH"; // CHANGE
String API = "FE11WY9CASMYAEPX"; // CHANGE
String HOST = "api.thingspeak.com";
String PORT = "80";
String field = "field1";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
boolean sendFall = false;
boolean emrgButton = false;
int valSensor = 1;
SoftwareSerial esp8266(RX,TX);

int period = 20000;
unsigned long time_now = 0;

void setup(){
Serial.begin(9600);
  esp8266.begin(19200);
  sendCommand("AT",5,"OK");
  sendCommand("AT+CWMODE=1",5,"OK");
  sendCommand("AT+CWJAP=\""+ AP +"\", \""+ PASS +"\"",20,"OK");
//connect to wifi
}
  void loop(){

    if(millis() > (time_now + period)){ // update channel
fields
//valSensor = getSensorData();
String getData = "GET /update?api_key="+ API +"&"+ field
+"="+String(value1) + "&field2=" +String(sendFall) + "&field3="
+String(emrgButton)+ "&field4=" +String(dht.temperature)+ "&field5="
+String(dht.humidity);
sendCommand("AT+CIPMUX=1",5,"OK");
sendCommand("AT+CIPSTART=0,\"TCP\", \""+ HOST +"\", "+ PORT,15,"OK");
sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");
esp8266.println(getData);delay(1500);countTrueCommand++;
sendCommand("AT+CIPCLOSE=0",5,"OK");
time_now = millis();
//Epanekinhsh fall kai button katagrafis
sendFall=false;
emrgButton = false;
}
}

//check if command is sent
void sendCommand(String command, int maxTime, char readReplay[]) {
Serial.print(countTrueCommand);
Serial.print(". at command => ");
Serial.print(command);
Serial.print(" ");
while(countTimeCommand < (maxTime*1))
{
  esp8266.println(command);//at+cipsend
  if(esp8266.find(readReplay))//ok
  {
```

```

        found = true;
        break;
    }

    countTimeCommand++;
}

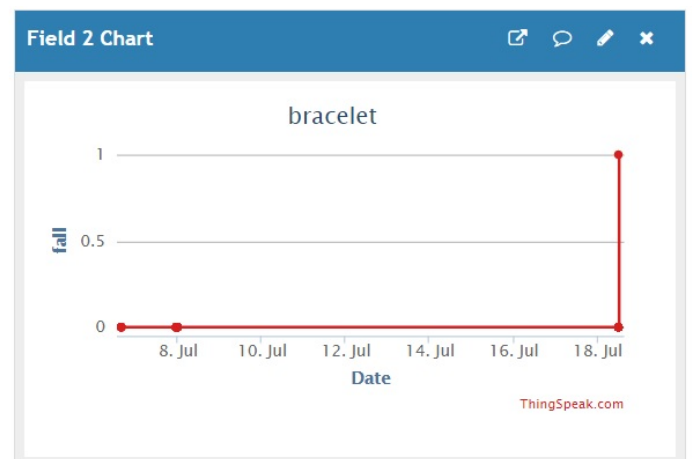
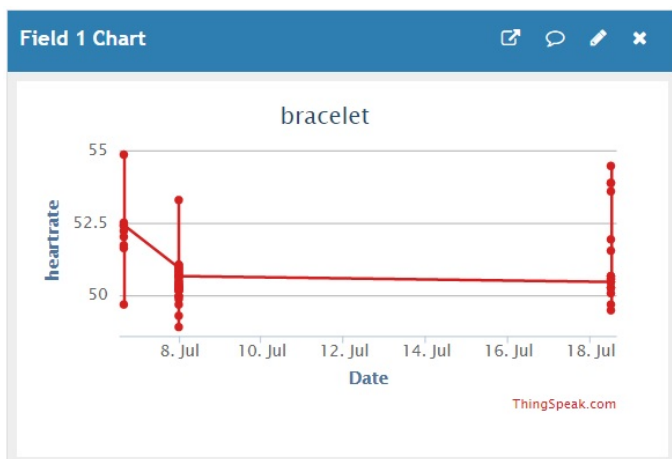
if(found == true)
{
    Serial.println("OYI");
    countTrueCommand++;
    countTimeCommand = 0;
}

if(found == false)
{
    Serial.println("Fail");
    countTrueCommand = 0;
    countTimeCommand = 0;
}

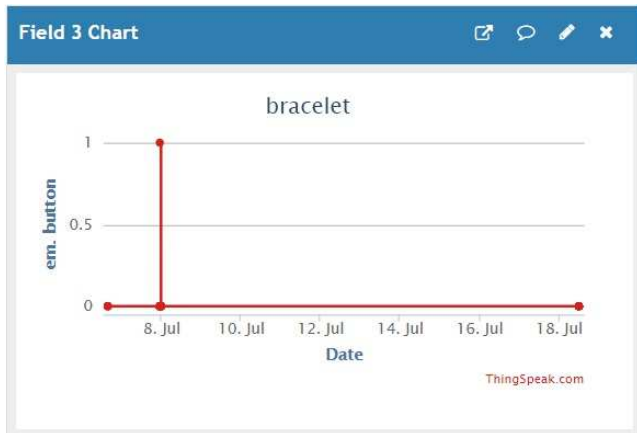
found = false;
}

```

Παρακάτω φαίνονται τα πέντε διαγράμματα τα οποία δημιουργήθηκαν για το βραχιόλι.



Εικόνα 103: Διαγράμματα Thingspeak βραχιολιού (1)



Εικόνα 104: Διαγράμματα Thingspeak βραχιολιού (2)

5.3.2.2 Αυτόματες ειδοποιήσεις

Το βραχιολάκι έχει σχεδιαστεί ώστε να στέλνει αυτόματες ειδοποιήσεις στα οικεία άτομα του ηλικιωμένου ή των ατόμων με ιατρικές παθήσεις. Οι περιπτώσεις ειδοποιήσεων είναι οι παρακάτω:

4. Πτώση
5. Πάτημα button έκτακτης ανάγκης
6. Αυξημένοι καρδιακοί παλμοί

Για να επιτευχθεί αυτό χρησιμοποιήθηκαν οι παρακάτω εφαρμογές- υπηρεσίες:

React app: Το React συνεργάζεται με ThingHTTP και ThingTweet για να πραγματοποιήσει ενέργειες όταν τα δεδομένα καναλιών πληρούν μια συγκεκριμένη προϋπόθεση.

Thing HTTP: Το ThingHTTP επιτρέπει την επικοινωνία μεταξύ συσκευών, ιστοτόπων και υπηρεσιών ιστού χωρίς να χρειάζεται να εφαρμόσει το πρωτόκολλο σε επίπεδο συσκευής. Μπορούν να οριστούν ενέργειες στο ThingHTTP, οι οποίες ενεργοποιούνται χρησιμοποιώντας άλλες εφαρμογές ThingSpeak όπως τα TweetControl, TimeControl και React.

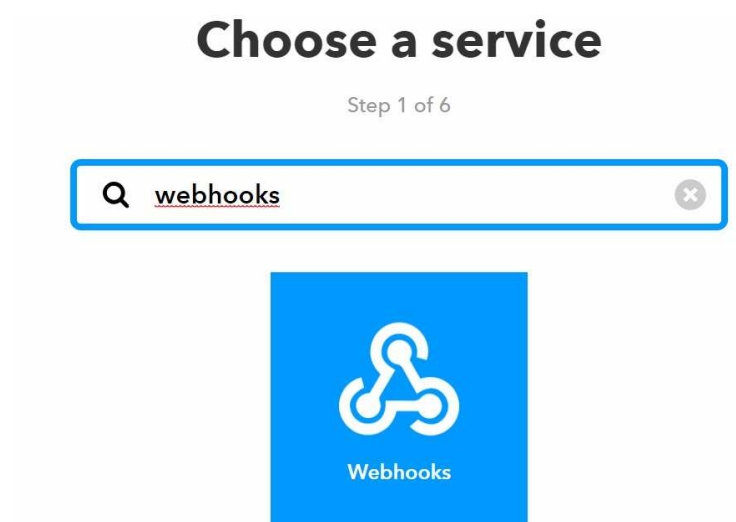
Τα βήματα τα οποία είναι αναγκαία για την δημιουργία της ειδοποίησης πτώσης παρουσιάζονται παρακάτω:

- 1) Δημιουργία λογαριασμού IFTT
- 2) Δημιουργία ενός applet. Επιλογή My Applets και, στη συνέχεια, κλικ στο κουμπί New Applet.
- 3) Επιλογή τη δράση εισαγωγής. κλικ στη λέξη this.



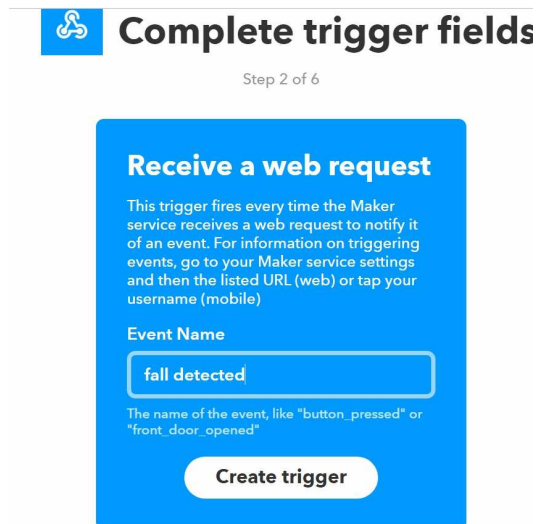
Εικόνα 105: Δημιουργία ειδοποίησης

- 4) Αναζήτηση της υπηρεσίας Webhooks



Εικόνα 106: Επιλογή υπηρεσίας Webhooks

5) Συμπλήρωση του trigger fields
Συμπλήρωση του πεδίου Event Name με fall detected για την ειδοποίηση της πτώσης.



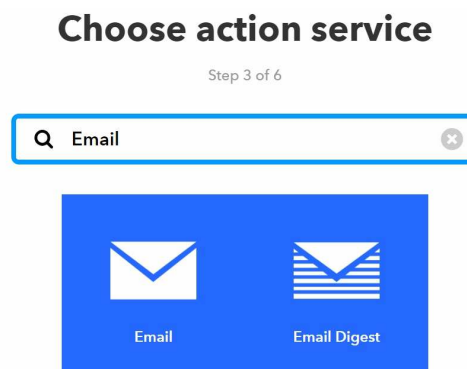
Εικόνα 107: Συμπλήρωση πεδίου event

6) Πάτημα του κουμπιού that



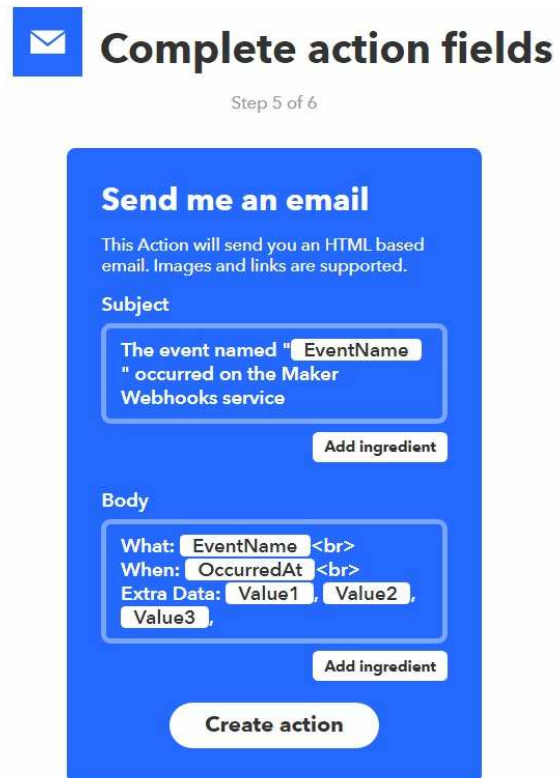
Εικόνα 108: Επιλογή του that

7) Επιλογή του Email ως action service



Εικόνα 109: Επιλογή υπηρεσίας δράσης

8) κλικ στο Create action.



Εικόνα 110: Δημιουργία δράσης

8) Ανάκτηση πληροφοριών Webhooks trigger. Επιλογή My Applets> Υπηρεσίες και αναζήτηση Webhooks. Επιλογή Webhooks και στη συνέχεια κλικ στο κουμπί Documentation Στην επόμενη οθόνη φαίνεται το κλειδί και η μορφή αποστολής αιτήματος.

Your key is: **f_FCnMbhjeBHpnSrm6hUA**

◀ Back to service

To trigger an Event

Make a POST or GET web request to:

https://maker.ifttt.com/trigger/fall_detected/with/key/f_FCnMbhjeBHpnSrm6hUA

Εικόνα 111: Ανάκτηση πληροφοριών Webhooks trigger

Το επόμενο βήμα είναι η δημιουργία ενός thingHTTP στο Thingspeak το οποίο θα ενεργοποιήσει την υπηρεσία webhooks.

Επιλογή Apps → ThingHTTP → New ThingHTTP

Συμπλήρωση πεδίων:

Name: fall

URL: Εισαγωγή του url από το webhooks documentation

Method: Επιλογή GET

Τέλος χρειάζεται η δημιουργία ενός react στο Thingspeak το οποίο θα ενεργοποιεί το αίτημα HTTP ανάλογα με την τιμή της πτώσης δηλαδή 1 στην συγκεκριμένη περίπτωση.

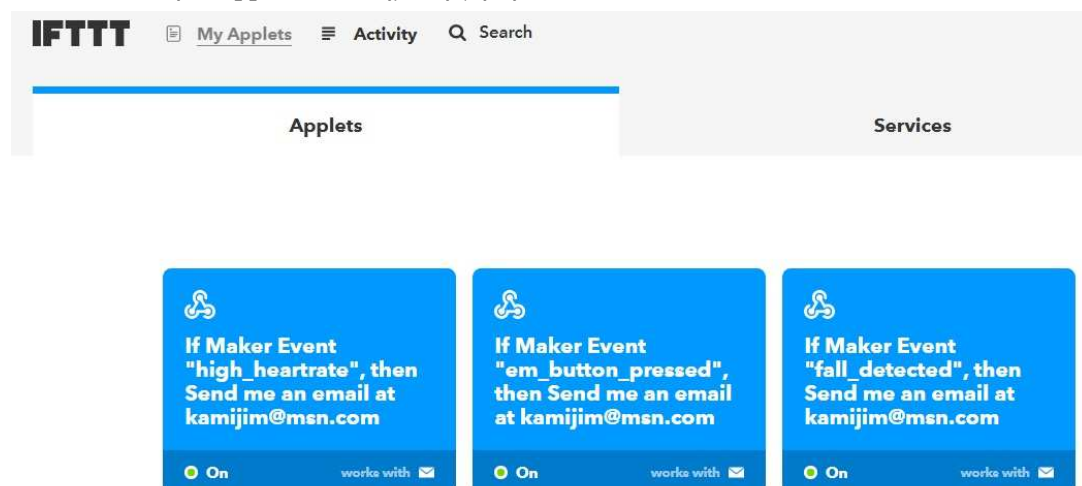
The screenshot shows the configuration form for a new React in Thingspeak. The fields are as follows:

- React Name:** Input field containing "fall1".
- Condition Type:** Dropdown menu set to "Numeric".
- Test Frequency:** Dropdown menu set to "On Data Insertion".
- Condition:** A nested configuration:
 - If channel:** Dropdown menu set to "bracelet (790473)".
 - field:** Dropdown menu set to "2 (fall)".
 - Operator:** Dropdown menu set to "is equal to".
 - Value:** Input field containing "1".
- Action:** Dropdown menu set to "ThingHTTP".
- then perform ThingHTTP:** Dropdown menu set to "fall".
- Options:** Radio buttons for "Run action only the first time the condition is met" (selected) and "Run action each time condition is met".
- Save React:** A green button at the bottom.

Εικόνα 112: Δημιουργία React app

Αντίστοιχα έγινε και η διαδικασία για τις άλλες δύο περιπτώσεις ειδοποιήσεων.

Συνολικά τα τρία applets που δημιουργήθηκαν στο IFTTT.



Εικόνα 113: IFTTT applets

Καθώς και τα τρία thingHTTP και React appsQ:

Apps / ThingHTTP

New ThingHTTP

Name	Created
fall View Edit	2019-05-29
button View Edit	2019-06-03
heartrate View Edit	2019-06-04

Εικόνα 114: ThingHTTP Apps

Apps / React

New React

Name	Created	Last Ran
✓ fall1 View Edit	2019-05-29	2019-06-06 8:36 am
✓ button View Edit	2019-06-03	2019-06-06 8:37 am
✓ heartrate View Edit	2019-06-04	

Εικόνα 115: React Apps


Η ειδοποίηση που στέλνεται στο email σε περίπτωση πατήματος του button έκτακτης ανάγκης:

The event named "em_button_pressed" occurred on the Maker Webhooks service

Getting too much email? [Unsubscribe](#) | [Manage subscriptions](#)

WI Webhooks via IFTTT <action@ifttt.com>
Sun 7/7/2019 11:36 PM
You ▾

What: em_button_pressed
When: July 7, 2019 at 11:36PM
Extra Data: , , ,

 **If Maker Event "em_button_pressed", then Send me an email at kamijim@msn.com** >

[Unsubscribe from these notifications](#) or sign in to manage your [Email Applets](#).

IFTTT

Εικόνα 116: Ειδοποίηση πτώσης με email

5.4 Google Charts

Για την πτυχιακή εργασία υλοποιήθηκαν 4 PHP αρχεία τα [weekly_statistics1.php](#), [monthly_statistics1.php](#), [statistics1.php](#) και [gauge.php](#). Οι τιμές από τους αισθητήρες που βρίσκονται στο Arduino Mega και στο Arduino Uno R3 του έξυπνου σπιτιού, αντλούνται από την βάση δεδομένων και εμφανίζονται σε διαγράμματα.

1. temperature: θερμοκρασία οικίας
2. humidity: υγρασία οικίας
3. fan: εξαερισμός (0 ή 1)
4. fire: φωτιά (0 ή 1)
5. flood: πλυμμήρα (0 ή 1)
6. garage: γκαραζόπορτα (0 ή 1)
7. antlia: αντλία ανελκυστήρα (0 ή 1)
8. porta: εξώπορτα (0 ή 1)

Παρακάτω θα γίνει ανάλυση των στοιχείων που βρίσκονται μέσα στους κώδικες.

Μηνιαία στατιστικά

Ο κώδικας του αρχείου [monthly_statistics1.php](#), που αντιστοιχεί στην εμφάνιση του διαγράμματος της θερμοκρασίας. Με τον ίδιο τρόπο θα υλοποιηθούν και τα υπόλοιπα διαγράμματα.

Το αρχείο PHP `monthly_statistics1.php` θα έχει τον εξής κώδικα:

```
<body>
    // εισαγωγή και φόρτωση του αρχείου Javascript
    <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
    <script type="text/javascript">
        google.charts.load('current', {'packages':['corechart']});
        google.charts.setOnLoadCallback(drawChart);
    // συνάρτηση σχεδίασης πίνακα γραφημάτων των τιμών από τους
    αισθητήρες
    function drawChart() {

        var data = new google.visualization.DataTable();
        data.addColumn('datetime', 'event');
        data.addColumn('number', 'Average temperature');
        data.addColumn('number', 'Max temperature');
        data.addColumn('number', 'Min temperature');
        <?php
    // προετοιμασία μεταβλητών για σύνδεση με την βάση δεδομένων

    $dbusername = "root"; // εισαγωγή όνομα χρήστη για την βάση
    $dbpassword = ""; // εισαγωγή μυστικού κωδικού για την βάση
    $server = "localhost";

    // σύνδεση με την βάση δεδομένων
    $mysqli = new mysqli($server, $dbusername, $dbpassword,
"ethernet");
    if($mysqli->connect_error)
```

```

    die('Connect Error (' . mysqli_connect_errno() . ') ' .
mysqli_connect_error());

    // εκτέση εντολής SQL ομαδοποιούνται ανά μήνα και κρατιούνται οι
τελευταίες 12 τιμές από τις μέγιστες, τις ελάχιστες και τον μέσο όρο
    $result = $mysqli->query("SELECT YEAR(event) AS year,
MONTH(event) AS month, DAY(event) AS day,
AVG(temperature),MAX(temperature),MIN(temperature),
AVG(humidity),MAX(humidity),MIN(humidity),
AVG(FAN),MAX(FAN),MIN(FAN), AVG(FIRE),MAX(FIRE),MIN(FIRE),
AVG(FLOOD),MAX(FLOOD),MIN(FLOOD),
AVG(GARAGE),MAX(GARAGE),MIN(GARAGE),
AVG(porta),MAX(porta),MIN(porta), AVG(ANTLIA),MAX(ANTLIA),MIN(ANTLIA)
from data group by MONTH(event) ORDER BY event DESC limit 12");

    if ($result->num_rows > 0) {

        // εξαγωγή δεδομένων για κάθε τιμή
        while($row = $result->fetch_assoc()) {
            echo "data.addRow([new
Date(".$row["year"].",".$row["month"]."-1",".$row["day"]."),
".$row["AVG(temperature)"].",".$row["MAX(temperature)"].",".$row["MIN
(temperature)"].",".$row["AVG(humidity)"].",".$row["MAX(humidity)"]."
, ".$row["MIN(humidity)"].",".$row["AVG(FAN)"].",".$row["MAX(FAN)"].",
".$row["MIN(FAN)"].",".$row["AVG(FIRE)"].",".$row["MAX(FIRE)"].",".$r
ow["MIN(FIRE)"].",".$row["AVG(FLOOD)"].",".$row["MAX(FLOOD)"].",".$ro
w["MIN(FLOOD)"].",".$row["AVG(GARAGE)"].",".$row["MAX(GARAGE)"].",".$
row["MIN(GARAGE)"].",".$row["AVG(porta)"].",".$row["MAX(porta)"].",".$
row["MIN(porta)"].",".$row["AVG(ANTLIA)"].",".$row["MAX(ANTLIA)"].",
".$row["MIN(ANTLIA)"]."]);\\n";
        }
    }
    else {
    }
?>
// εισαγωγή τίτλου, υπότιτλου και μορφής καθώς το ύψος και το πλάτος
που θα έχουν αυτά
    var temperature_options = {
        title: 'temperature (C) Last Month',
        subtitle: 'measured in C',
        hAxis: { format: ' ' },
        seriesType: 'bars',
        width: 600,
        height: 300
    };

    // αίτημα στην υπηρεσία Google Charts για εμφάνιση του
γραφήματος
    var temperature_chart = new
google.visualization.ComboChart(document.getElementById('temperature_
div'));

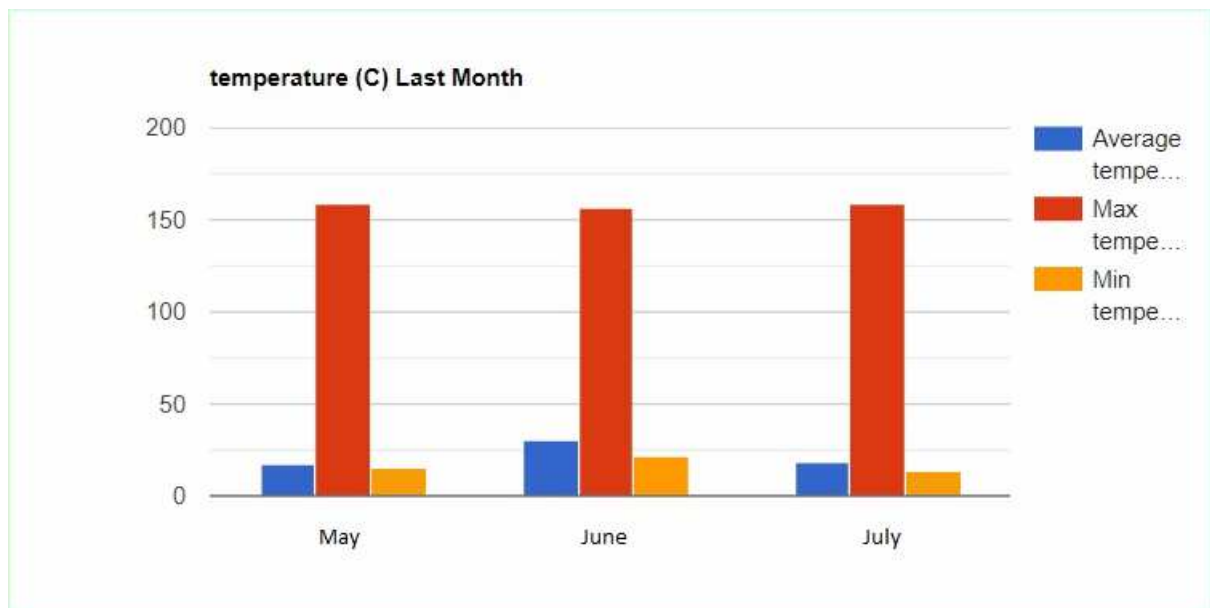
    var temperature_view = new google.visualization.DataView(data);
    temperature_view.setColumns([0,1,2,3]);
    temperature_chart.draw(temperature_view, temperature_options);
}
</script>
</head>
<body>
    <div style="width:100%">
        //ρύθμιση των χαρακτηριστικων του γραφήματος όπως ύψος, πλάτος
και προσδιορισμός για το πώς θα φαίνεται στην οθόνη
    </div>

```

```

<table style="border-spacing: 15px">
  <tr>
    <td style="border: 1px solid green"
      <div id="temperature_div" style="width: 600px;
height: 300px"></div>
    </td>
  <tr>
</table>
</body>

```



Εικόνα 117: Διάγραμμα μηνιαίων στατιστικών

Εβδομαδιαία στατιστικά

Ο κώδικας του αρχείου [weekly_statistics1.php](#), που αντιστοιχεί στην εμφάνιση του διαγράμματος της θερμοκρασίας. Με τον ίδιο τρόπο θα υλοποιηθούν και τα υπόλοιπα διαγράμματα.

Το αρχείο PHP `weekly_statistics1.php` θα έχει τον εξής κώδικα:

```

<body>
// εισαγωγή και φόρτωση του αρχείου Javascript
<script type="text/javascript"

```

```

src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
    google.charts.load('current', {'packages':['corechart']});
    google.charts.setOnLoadCallback(drawChart);
// συνάρτηση σχεδίασης πίνακα γραφημάτων των τιμών από τους
αισθητήρες
function drawChart() {

    var data = new google.visualization.DataTable();
    data.addColumn('datetime', 'event');
data.addColumn('number', 'Average temperature');
    data.addColumn('number', 'Max temperature');
    data.addColumn('number', 'Min temperature');
    <?php
// προετοιμασία μεταβλητών για σύνδεση με την βάση δεδομένων

$dbusername = "root"; // εισαγωγή όνομα χρήστη για την βάση
$dbpassword = ""; // εισαγωγή μυστικού κωδικού για την βάση
$server = "localhost";

// σύνδεση με την βάση δεδομένων
    $mysqli = new mysqli($server, $dbusername, $dbpassword,
"ethernet");
    if($mysqli->connect_error)
        die('Connect Error (' . mysqli_connect_errno() . ') ' .
mysqli_connect_error());

    // εκτέση εντολής SQL, ομαδοποιούνται ανά ημέρα και για κάθε ημέρα
οι μέγιστες, οι ελάχιστες και ο μέσος όρος. Κρατούνται οι τιμές των
τελευταίων 7 ημέρων
    $result = $mysqli->query("SELECT YEAR(event) AS year,
MONTH(event) AS month, DAY(event) AS day,
AVG(temperature),MAX(temperature),MIN(temperature),
AVG(humidity),MAX(humidity),MIN(humidity),
AVG(FAN),MAX(FAN),MIN(FAN), AVG(FIRE),MAX(FIRE),MIN(FIRE),
AVG(FLOOD),MAX(FLOOD),MIN(FLOOD),
AVG(GARAGE),MAX(GARAGE),MIN(GARAGE),
AVG(porta),MAX(porta),MIN(porta), AVG(ANTLIA),MAX(ANTLIA),MIN(ANTLIA)
from data group by DAY(event) ORDER BY event DESC limit 7");

    if ($result->num_rows > 0) {

        // εξαγωγή δεδομένων για κάθε τιμή
        while($row = $result->fetch_assoc()) {
            echo "data.addRow([new
Date(".$row["year"].",".$row["month"]."-
1,".$row["day"]."),".$row["AVG(temperature)"].",".$row["MAX(temperatu
re)"].",".$row["MIN(temperature)"].",".$row["AVG(humidity)"].",".$row
["MAX(humidity)"].",".$row["MIN(humidity)"].",".$row["AVG(FAN)"].",".
$row["MAX(FAN)"].",".$row["MIN(FAN)"].",".$row["AVG(FIRE)"].",".$row["
MAX(FIRE)"].",".$row["MIN(FIRE)"].",".$row["AVG(FLOOD)"].",".$row["M
AX(FLOOD)"].",".$row["MIN(FLOOD)"].",".$row["AVG(GARAGE)"].",".$row["
MAX(GARAGE)"].",".$row["MIN(GARAGE)"].",".$row["AVG(porta)"].",".$row
["MAX(porta)"].",".$row["MIN(porta)"].",".$row["AVG(ANTLIA)"].",".$ro
w["MAX(ANTLIA)"].",".$row["MIN(ANTLIA)"]."]); \n";
        }
    }
    else {
    }
?>

// εισαγωγή τίτλου, υπότιτλου και μορφής καθώς το ύψος και το
πλάτος που θα έχουν αυτά

```

```

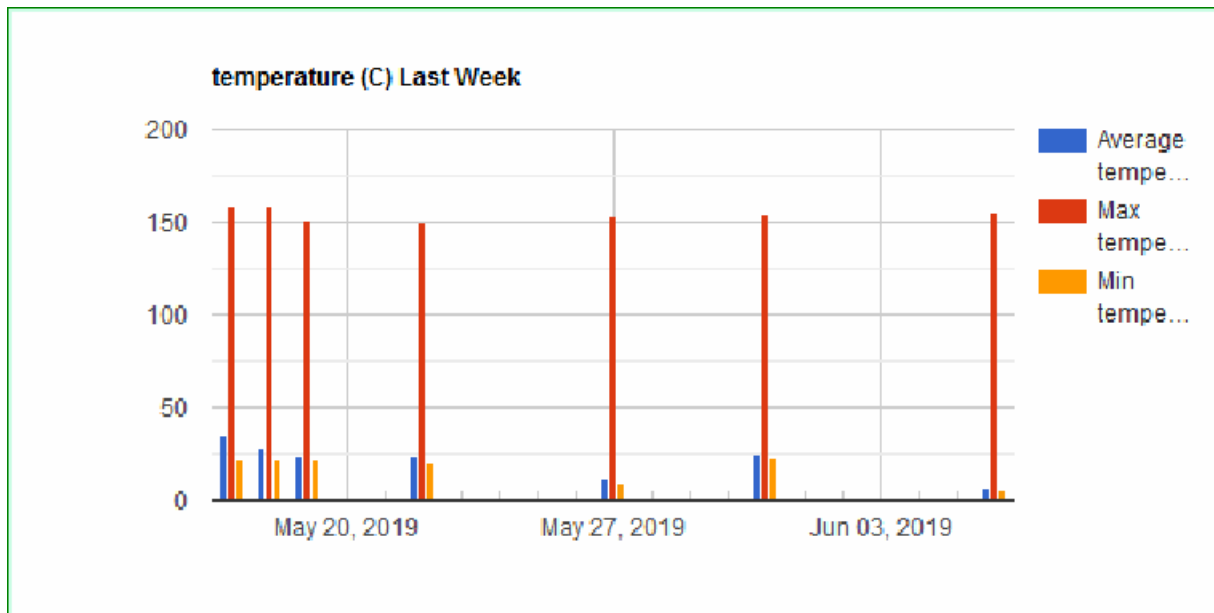
var temperature_options = {
  title: 'temperature (C) Last Week',
  subtitle: 'measured in C',
  hAxis: { format: 'MMM dd, yyyy' },
  seriesType: 'bars',
  width: 600,
  height: 300
};

// αίτημα στην υπηρεσία Google Charts για εμφάνιση του
γραφήματος
var temperature_chart = new
google.visualization.ComboChart(document.getElementById('temperature_
div'));

var temperature_view = new google.visualization.DataView(data);
temperature_view.setColumns([0,1,2,3]);
temperature_chart.draw(temperature_view, temperature_options);
}
</script>
</head>
<body>
  <div style="width:100%">
    //ρύθμιση των χαρακτηριστικων του γραφήματος όπως ύψος,
    πλάτος και προσδιορισμός για το πώς θα φαίνεται στην οθόνη
    </div>

    <table style="border-spacing: 15px">
      <tr>
        <td style="border: 1px solid green">
          <div id="temperature_div" style="width: 600px;
height: 300px"></div>
        </td>
      <tr>
      </table>
    </body>

```



Εικόνα 118: Διάγραμμα εβδομαδιαίων στατιστικών

Στατιστικά

Ο κώδικας του αρχείου [statistics1.php](#), που αντιστοιχεί στην εμφάνιση του διαγράμματος της θερμοκρασίας. Με τον ίδιο τρόπο θα υλοποιηθούν και τα υπόλοιπα διαγράμματα.

Το αρχείο PHP statistics1.php θα έχει τον εξής κώδικα:

```
<body>
  //εισαγωγή και φόρτωση του αρχείου Javascript
  <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
  <table align="center">
    <script type="text/javascript">
      google.charts.load('current', {'packages':['line']});
      google.charts.setOnLoadCallback(drawChart);
    // συνάρτηση σχεδίασης πίνακα γραφημάτων των τιμών από τους
    αισθητήρες
    function drawChart() {

      var data = new google.visualization.DataTable();
      data.addColumn('datetime', 'event');
      data.addColumn('number', 'temperature');
      <?php
    // προετοιμασία μεταβλητών για σύνδεση με την βάση δεδομένων

    $dbusername = "root"; // εισαγωγή όνομα χρήστη για την βάση
    $dbpassword = ""; // εισαγωγή μυστικού κωδικού για την βάση
    $server = "localhost";

    // σύνδεση με την βάση δεδομένων
```

```

    $mysqli = new mysqli($server, $dbusername, $dbpassword,
"ethernet");
    if($mysqli->connect_error)
        die('Connect Error (' . mysqli_connect_errno() . ') ' .
mysqli_connect_error());

    // εκτέση εντολής SQL, τραβάω από την βάση το time stamp και το
σπάω σε χρόνο,μήνα,μέρα,ώρα,λεπτό και δευτερόλεπτο για να βάλω στον
πίνακα την timestamp όπως την θέλει το Javascript για όλες τις
μετρήσεις. (τελευταίες 40)
    $result = $mysqli->query("SELECT YEAR(event) AS year,
MONTH(event) AS month, DAY(event) AS day, HOUR(event) AS hour,
MINUTE(event) AS minute, SECOND(event) AS sec, temperature,
humidity,FAN,FIRE,FLOOD,GARAGE,porta,ANTLIA FROM data ORDER BY event
DESC LIMIT 5");

    if ($result->num_rows > 0) {

        // εξαγωγή δεδομένων για κάθε τιμή
        while($row = $result->fetch_assoc()) {
            echo "data.addRow([new
Date(".$row["year"].",".$row["month"]."-
1, ".$row["day"].",".$row["hour"].",".$row["minute"].",".$row["sec"]."
),
".$row["temperature"].",".$row["humidity"].",".$row["FAN"].",".$row["
FIRE"].",".$row["FLOOD"].",".$row["GARAGE"].",".$row["porta"].",".$ro
w["ANTLIA"]."]); \n";
        }
    }
    else {
    }
?>
// εισαγωγή τίτλου, υπότιτλου και μορφής καθώς το ύψος και το πλάτος
που θα έχουν αυτά
var temperature_options = {
    chart: {

        title: 'temperature vs Time',
        subtitle: 'measured in C'
    },
    hAxis: { format: 'HH:mm:ss a' },
    width: 600,
    height: 300
};

// αίτημα στην υπηρεσία Google Charts για εμφάνιση του
γραφήματος
var temperature_chart = new
google.charts.Line(document.getElementById('temperature_div'));

var temperature_view = new google.visualization.DataView(data);
temperature_view.setColumns([0,1]);
temperature_chart.draw(temperature_view,
google.charts.Line.convertOptions(temperature_options));
</script>
</head>
<body>

        <div style="width:100%">
            // ρύθμιση των χαρακτηριστικων του γραφήματος όπως ύψος, πλάτος
και προσδιορισμός για το πώς θα φαίνεται στην οθόνη
            </div>

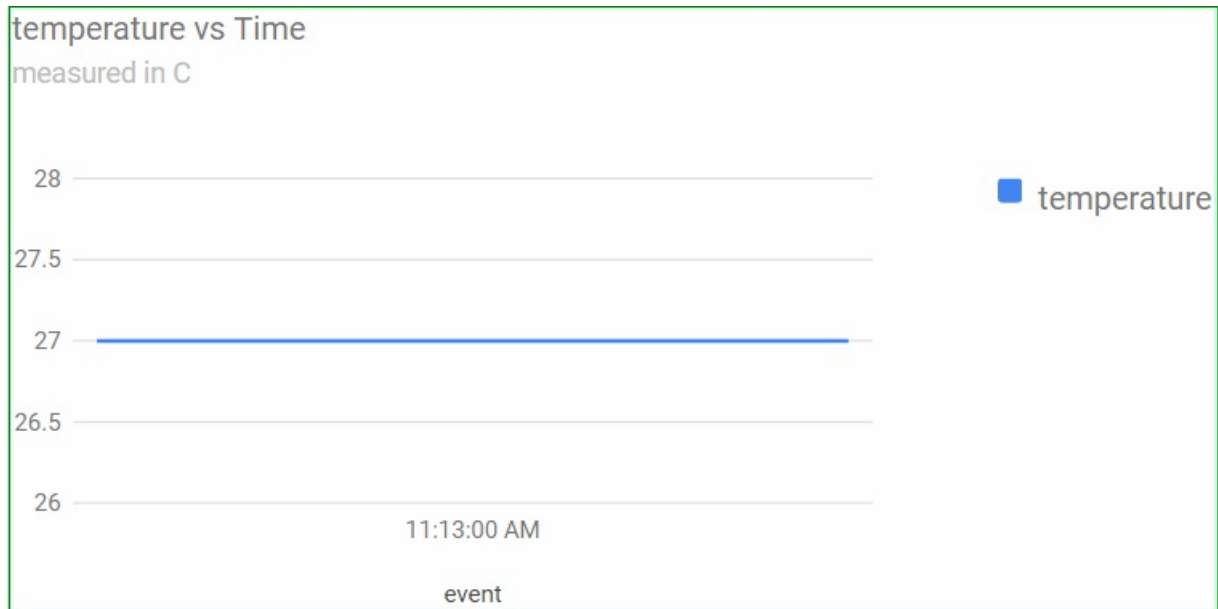
```



```

<table style="border-spacing: 15px">
  <tr>
    <td style="border: 1px solid green">
      <div id="temperature_div" style="width: 600px;
height: 300px"></div>
    </td>
  </tr>
</table>
</body>

```



Εικόνα 119: Διάγραμμα στατιστικών

Gauge

Ο κώδικας του αρχείου [gauge.php](#), που αντιστοιχεί στην εμφάνιση του διαγράμματος της θερμοκρασίας. Με τον ίδιο τρόπο θα υλοποιηθούν και τα υπόλοιπα διαγράμματα. Επίσης για την εμφάνιση των διαγραμμάτων χρειάζονται κάποια αρχεία Javascript να αποθηκευτούν στο htdocs στην Xampp στο φάκελο Ethernet με ονόματα gauge και gauge.min.

Το αρχείο PHP gauge.php θα έχει τον εξής κώδικα:

```

<?php
  // προετοιμασία μεταβλητών για σύνδεση με την βάση δεδομένων

  $dbusername = "root"; // εισαγωγή όνομα χρήστη για την βάση
  $dbpassword = ""; // εισαγωγή μυστικού κωδικού για την βάση
  $server = "localhost";

  // σύνδεση με την βάση δεδομένων
  $mysqli = new mysqli($server, $dbusername, $dbpassword,

```

```

"ethernet");
    if($mysqli->connect_error)
        die('Connect Error (' . mysqli_connect_errno() . ') ' .
mysqli_connect_error());

    // εκτέλεση εντολής SQL, παίρνω την τελευταία μέτρηση από την βάση με
το limit1
    $result = $mysqli->query("SELECT temperature,
humidity,FAN,FIRE,FLOOD,GARAGE,porta,ANTLIA FROM data ORDER BY id
DESC LIMIT 1");

    // εξαγωγή δεδομένων για κάθε τιμή
    if ($result->num_rows > 0) {

        while($row = $result->fetch_assoc()) {
            $TEValue = $row["temperature"];
            $HValue = $row["humidity"];
            $FAValue = $row["FAN"];
            $FIValue = $row["FIRE"];
            $FLValue = $row["FLOOD"];
            $GAValue = $row["GARAGE"];
            $pValue = $row["porta"];
            $AValue = $row["ANTLIA"];
        }
    }
    else {
        $TEValue = 0;
        $HValue = 0;
        $FAValue = 0;
        $FIValue = 0;
        $FLValue = 0;
        $GAValue = 0;
        $pValue = 0;
        $AValue = 0;
    }

?>
// εισαγωγή και φόρτωση του αρχείου Javascript
<script src="gauge.js"></script>
<meta http-equiv="refresh" content="180" />
<script>
// συναρτήσεις σχεδίασης γραφημάτων το μέγεθος και την μορφή που θα
έχει το διάγραμμα gauge
if (!Array.prototype.forEach) {
    Array.prototype.forEach = function(cb) {
        var i = 0, s = this.length;
        for (; i < s; i++) {
            cb && cb(this[i], i, this);
        }
    }
}

document.fonts && document.fonts.forEach(function(font) {
    font.loaded.then(function() {
        if (font.family.match(/Led/)) {
            document.gauges.forEach(function(gauge) {
                gauge.update();
                gauge.options.renderTo.style.visibility = 'visible';
            });
        }
    });
});

```

```

    });
});

var timers = [];

function animateGauges() {
    document.gauges.forEach(function(gauge) {
        timers.push(setInterval(function() {
            gauge.value = Math.random() *
                (gauge.options.maxValue - gauge.options.minValue) +
                gauge.options.minValue;
        }, gauge.animation.duration + 50));
    });
}

function stopGaugesAnimation() {
    timers.forEach(function(timer) {
        clearInterval(timer);
    });
}

function resize() {
    var size = parseFloat(document.getElementById('gauge-
size').value) || 400;

    document.gauges.forEach(function (gauge) {
        gauge.update({ width: size, height: size });
    });
}

function setText() {
    var text = document.getElementById('gauge-text').value;

    document.gauges.forEach(function (gauge) {
        gauge.update({ valueText: text });
    });
}
</script>

</head>
<body>

```

```

//τεχνικά χαρακτηριστικά για το γράφημα gauge

<canvas data-type="radial-gauge" // εδω δηλώνεται ο τύπος του gauge
    data-width="250" // δήλωση πλάτους του gauge
    data-height="250" // δήλωση ύψους του gauge
    data-units="&deg;C" // δήλωση μονάδας μέτρησης της
τιμής που προβάλλει το gauge
    data-title="temperature" // δήλωση τίτλου για το gauge
    data-value=<?php echo "\".$.TEValue.""\n"?> //άντληση της τιμής από
τον πίνακα και εμφάνιση της στο gauge
    data-animate-on-init="false"
    data-animated-value="false"
    data-min-value="0" // δήλωση ελάχιστης τιμής που θα πάρει
το gauge

```

```

data-max-value="100" // δήλωση μέγιστης τιμής που θα πάρει το
gauge
data-major-ticks="0,10,20,30,40,50,60,70,80,90,100" //
δήλωση κλίμακας με βήμα ανά 10 που θα πάρει το gauge
data-minor-ticks="0.1" // δήλωση βήματος που θα πάρει το
gauge
data-stroke-ticks="false"
// εδώ μπορούμε να αλλάξουμε τα χρώματα της κλίμακας για
να τονίσουμε αυτό που θέλουμε. Όπως φαίνεται παρακάτω από 0 έως 30
είναι το χρώμα πράσινο από 30 έως 65 είναι το χρώμα κίτρινο και από
65 έως 100 είναι το χρώμα κόκκινο τα οποία ορίζονται με την μορφή RGB.
data-highlights='[
  { "from": 0, "to": 30, "color": "rgba(0,255,0)" },
  { "from": 30, "to": 65, "color": "rgba(255, 255, 0)" },
  { "from": 65, "to": 100, "color": "rgba(255, 0, 0)" }
]'
```

κλίμακας με βήμα ανα 10 που θα πάρει το gauge

```

data-color-plate="#001433"
data-color-major-ticks="#f5f5f5" // δήλωση χρώματος της
βήματος που θα πάρει το gauge
data-color-minor-ticks="#ddd" // δήλωση χρώματος τού
data-color-title="#fff" // δήλωση χρώματος τίτλου στο gauge
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
</canvas>
</body>

```



Εικόνα 120: Διαγράμματα Gauge

5.5 Δημιουργία ιστοσελίδας

Στα πλαίσια της πτυχιακής εργασίας δημιουργήθηκε μία ιστοσελίδα η οποία τρέχει σε web server ενός υπολογιστή, με σκοπό να μπορεί ο χρήστης να μπαίνει από τον υπολογιστή του ή το κινητό του τηλέφωνο και να παρακολουθεί τις ενδείξεις του έξυπνου σπιτιού και του έξυπνου βραχιολιού έκτακτης ανάγκης για ηλικιωμένους ή άτομα με ιατρικές παθήσεις. Επίσης θα μπορεί να ελέγχει κάποιες λειτουργίες για το έξυπνο σπίτι. Για την προβολή της ιστοσελίδας δημιουργήθηκε το αρχείο index.html, το οποίο περιέχεται στο htdocks του Xampp μαζί με τα υπόλοιπα php αρχεία, εικόνες και Javascripts που απαρτίζουν τα περιεχόμενα της ιστοσελίδας. Για την δημιουργία της ιστοσελίδας χρησιμοποιήθηκε έτοιμη διαδικτυακή φόρμα από το quackit.com. Η συγκεκριμένη ιστοσελίδα διαφέρει από την ιστοσελίδα εποπτείας και ελέγχου της υποενοτήτας 5.2 η οποία τρέχει στην SD κάρτα του Ethernet shield.

Η ιστοσελίδα html με το αρχείο index.html θα έχει τον εξής κώδικα:

```
<!DOCTYPE html>
<!-- Template by quackit.com -->
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Smart Home &mdash; Emergency Bracelet &amp;
Footer</title>
    <style type="text/css">

        body {                                     // εισγωγή χρώματος
φόντιου της ιστοσελίδας
            margin:0;
            padding:0;
            font-family: Sans-Serif;
            line-height: 1.5em;
            background: #CBFCE4;
        }

        #header {                                 // εισγωγή χρώματος
τίτλου της ιστοσελίδας
            background: #B4E8CF;
            height: 100px;
        }

        #header h1 {                              // γραμματοσειρά και
θέση του τίτλου στην ιστοσελίδα
            margin: 0;
            padding-top: 15px;
        }

        main {
            padding-bottom: 10010px;
            margin-bottom: -10000px;
            float: left;
            width: 100%;
        }

        #nav {                                     // εισαγωγή χρώματος
```

```

ορισμός πλάτους και θέσης πλαϊνής στήλης ιστοσελίδας
padding-bottom: 10010px;
margin-bottom: -10000px;
float: left;
width: 230px;
margin-left: -100%;
background: #71E3AD;
}

#footer { // εισαγωγή χρώματος
ορισμός πλάτους και θέσης στο τέλος της ιστοσελίδας
clear: left;
width: 100%;
background:#B4E8CF ;
text-align: center;
padding: 4px 0;
}

#wrapper {
overflow: hidden;
}

#content {
margin-left: 230px;
}

.innertube {
margin: 15px;
margin-top: 0;
}

p {
color: #000000;
}

nav ul {
list-style-type: none;
margin: 0;
padding: 0;
}

nav ul a { // εισαγωγή χρώματος γραμματοσειράς
color: black;
text-decoration: none;
}

</style>

</head>

<body>

<header id="header">
<div class="innertube">
<h1>Smart home and emergency bracelet for
elderly</h1> // εισαγωγή τίτλου ιστοσελίδας
</div>
</header>

<div id="wrapper">

```

```

<main>
  <div id="content">

    <div class="content">
      // εισαγωγή εικόνων ιστοσελίδας
      <p><br> <br></p>

      <p>

        <br><br>Η υλοποίηση της μακέτας και η σύνδεση των αισθητήρων
έγινε στο πλαίσιο της πτυχιακής εργασίας των Σωτηρίου Δριμή και
Δημητρίου Καμινέλη με εισηγητή τον Δρ. Τοπάλη Ευάγγελο.
        <br>
        <br>Το σύστημα που παρουσιάζεται παρακάτω αφορά τη
δημιουργία ενός συστήματος εποπτείας και ελέγχου ενός έξυπνου σπιτιού
και ενός βραχιολιού έκτακτης ανάγκης για ηλικιωμένους, μέσω
κατάλληλου υλικού και λογισμικού μέρους .Θα έχει την δυνατότητα να
συνδεθεί στο διαδίκτυο και μέσω αυτού να γίνεται έλεγχος και
παρακολούθηση τόσο του βραχιολιού έκτακτης ανάγκης όσο και των
αυτοματισμών στο έξυπνο σπίτι.Επίσης για την καλύτερη προσέγγιση των
πραγματικών συνθηκών και αναγκών τόσο του έξυπνου σπιτιού όσο και του
βραχιολιού έκτακτης ανάγκης για ηλικιωμένους έχει κατασκευαστεί η
παρακάτω μακέτα και το βραχιόλι έκτακτης ανάγκης . </p>
        <br>
        <br>Η μακέτα του έξυπνου σπιτιού και του
βραχιολιού έκτακτης ανάγκης για ηλικιωμένους<br>

<p><br>
  </p>

  <p>
  </p>

    </div>
  </div>
</main>
// πλαϊνή στήλη ιστοσελίδας
<nav id="nav">
  <div class="innertube">
    <h3>SMART HOME</h3> // εισαγωγή τίτλου έξυπνο
σπίτι γιατί θα ακολουθήσουν οι ενδείξεις του έξυπνου σπιτιού
    <ul>
      <li><a href="index.html">First Page</a></li>
// εισαγωγή αρχείου index.html για επιστροφή στην αρχική σελίδα
    </ul>
    <h3>thingspeak</h3> // εισαγωγή τίτλου
thingspeak καθώς ακολουθούν τα γραφήματα thingspeak του έξυπνου
σπιτιού
    <ul>
      <li><a
href="temperature.php">Temperature</a></li> // εισαγωγή αρχείου
temperature.php και εμφάνιση γραφήματος thingspeak. Με τον ίδιο τρόπο
εισάγονται και εμφανίζονται τα διαγράμματα και των υπόλοιπων αρχείων.
      <li><a
href="humidity.php">Humidity</a></li>
      <li><a

```

```

href="fan.php">Fan</a></li>
</li></ul>
</a></li>
</a></li>
href="antlia.php">Antlia</a></li>
href="porta.php">Porta</a></li>
</ul>
<h3>Google charts</h3> // εισαγωγή
τίτλου Google charts καθώς ακολουθούν τα γραφήματα google charts του
έξυπνου σπιτιού
<ul>
<li><a
href="statistics1.php">Statistics</a></li> // εισαγωγή αρχείου
statistics1.php και εμφάνιση γραφήματος google charts. Με τον ίδιο
τρόπο εισάγονται και εμφανίζονται τα διαγράμματα και των υπόλοιπων
αρχείων.
<li><a
href="gauge.php">Gauge</a></li>
<li><a
href="weekly_statistics1.php">Weekly Statistics</a></li>
<li><a
href="monthly_statistics1.php">Monthly Statistics</a></li>
</ul>
<h3>Data base</h3> // εισαγωγή τίτλου
Data base καθώς ακολουθεί ο πίνακας της βάσης δεδομένων του έξυπνου
σπιτιού
<ul>
<li><a
href="display.php">Display</a></li> // εισαγωγή αρχείου display.php
και εμφάνιση του πίνακα
</ul>
<h3>Ajax I/O</h3> // εισαγωγή τίτλου
Ajax I/O καθώς ακολουθεί η σελίδα ελέγχου κάποιων λειτουργιών του
έξυπνου σπιτιού
<ul>
<li><a
href="http://192.168.0.189">Control Led and Relay</a></li> //
εισαγωγή IP που έχει πάρει το Arduino για εμφάνιση της σελίδας
ελέγχου
</ul>
<h3>EMERGENCY BRACELET</h3> // εισαγωγή
τίτλου βραχιόλι έκτακτης ανάγκης γιατί θα ακολουθήσουν οι ενδείξεις
του βραχιολιού έκτακτης ανάγκης για ηλικιωμένους
<h3>thingspeak</h3> // εισαγωγή τίτλου
thingspeak καθώς ακολουθούν τα γραφήματα thingspeak του βραχιολιού
έκτακτης ανάγκης για ηλικιωμένους
<ul>
<li><a
href="temperature1.php">Temperature</a></li> // εισαγωγή αρχείου
temperature1.php και εμφάνιση γραφήματος thingspeak. Με τον ίδιο
τρόπο εισάγονται και εμφανίζονται τα διαγράμματα και των υπόλοιπων
αρχείων.
<li><a
href="humidity1.php">Humidity</a></li>
<li><a href="heartrate.php">Heart
Rate</a></li>
<li><a href="fall.php">Fall
Detection </a></li>

```



```

</li><a href="button.php">Emergency Button</a></li>
</ul>
</div>
</nav>
</div>
<div id="footer">
<div class="innertube">
<p>
<span>2019 &copy; </span><a href="#">
>Πανεπιστήμιο Πελοποννήσου</a> | <a href="#">Πάτρα</a>
</p>
</div>
</div>
</body>
</html>

```

Smart home and emergency bracelet for elderly

SMART HOME

First Page

thingspeak

Temperature
Humidity
Fan
Fire
Flood
Garage
Antlia
Porta

Google charts

Statistics
Gauge
Weekly Statistics
Monthly Statistics

Data base

Display



Ajax I/O

Control Led and Relay

EMERGENCY BRACELET

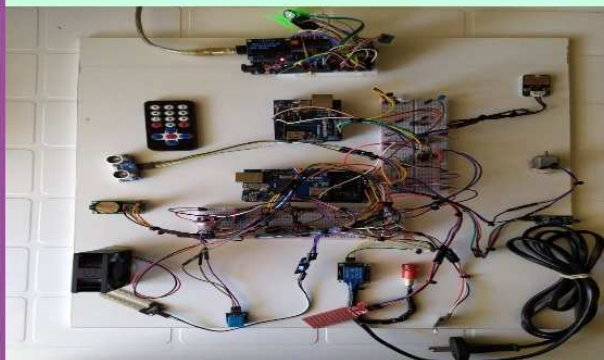
thingspeak

Temperature
Humidity
Heart Rate
Fall Detection
Emergency Button

Η υλοποίηση της μακέτας και η σύνδεση των αισθητήρων έγινε στο πλαίσιο της πτυχιακής εργασίας των Σωτηρίου Δριμή και Δημητρίου Καμινέλη με εισηγητή τον Δρ. Τοπάλη Ευάγγελο.

Το σύστημα που παρουσιάζεται παρακάτω αφορά τη δημιουργία ενός συστήματος εποπτείας και ελέγχου ενός έξυπνου σπιτιού και ενός βραχιολιού έκτακτης ανάγκης για ηλικιωμένους. Αυτό να γίνεται έλεγχος και παρακολούθηση τόσο του βραχιολιού έκτακτης ανάγκης όσο και των αυτοματισμών στο έξυπνο σπίτι. Επίσης για την καλύτερη προσέγγιση των πραγματικών ηλικιωμένους έχει κατασκευαστεί η παρακάτω μακέτα και το βραχιόλι έκτακτης ανάγκης.



Η μακέτα του έξυπνου σπιτιού και του βραχιολιού έκτακτης ανάγκης για ηλικιωμένους

2019 © Πανεπιστήμιο Πελοποννήσου | Πάτρα

Εικόνα 121: Η ιστοσελίδα όπως φαίνεται από το browser του υπολογιστή

Στην παραπάνω εικόνα φαίνεται με μπλε χρώμα ο τίτλος (επικεφαλίδα) της κεντρικής ιστοσελίδας, με κόκκινο χρώμα η πλαϊνή στήλη, (που περιέχει όλες τις υποσελίδες που μπορούμε να καλέσουμε), η οποία χωρίζεται στην αρχική σελίδα, στο έξυπνο σπίτι και στο βραχιόλι έκτακτης ανάγκης για ηλικιωμένους. Το έξυπνο σπίτι, περιέχει τα διαγράμματα Thingspeak, Google Charts, της μετρήσεις της βάσεις δεδομένων και την ιστοσελίδα που βρίσκεται στην κάρτα SD. Στη συνέχεια το βραχιόλι έκτακτης ανάγκης περιέχει τα διαγράμματα Thingspeak. Με το μωβ χρώμα είναι η κεντρική σελίδα όπου εμφανίζονται τα περιεχόμενα της πλαϊνής στήλης και η αρχική σελίδα με τις φωτογραφίες και κάποια λόγια για την πτυχιακή. Τέλος με καφέ χρώμα φαίνεται το κάτω μέρος (υποσέλιδο) της ιστοσελίδας όπου αναγράφονται τα πνευματικά δικαιώματα.



Εικόνα 122: Το πρώτο μισό της ιστοσελίδας όπως φαίνεται από τον browser του κινητού

1:44 MM ☑

Humidity
Fan
Fire
Flood
Garage
Antlia
Porta

Google charts
Statistics
Gauge
Weekly Statistics
Monthly Statistics

Data base
Display

Ajax I/O
Control Led and Relay

EMERGENCY BRACELET
thingspeak
Temperature
Humidity
Heart Rate
Fall Detection
Emergency Button



Η υλοποίηση της μακέτας και η σύνδεση των αισθητήρων έγινε στο πλαίσιο της πτυχιακής εργασίας των Σωτηρίου Δριμή και Δημητρίου Καμινέλη με εισηγητή τον Δρ. Τοπάλη Ευάγγελο.

Το σύστημα που παρουσιάζεται παρακάτω αφορά τη δημιουργία ενός συστήματος εποπτείας και ελέγχου ενός έξυπνου σπιτιού και ενός βραχιολιού έκτακτης ανάγκης για ηλικιωμένους, μέσω κατάλληλου υλικού και λογισμικού μέρους. Θα έχει την δυνατότητα να συνδεθεί στο διαδίκτυο και μέσω αυτού να γίνεται έλεγχος και παρακολούθηση τόσο του βραχιολιού έκτακτης ανάγκης όσο και των αυτοματισμών στο έξυπνο σπίτι. Επίσης για την καλύτερη προσέγγιση των πραγματικών συνθηκών και αναγκών τόσο του έξυπνου σπιτιού όσο και του βραχιολιού έκτακτης ανάγκης για ηλικιωμένους έχει κατασκευαστεί η παρακάτω μακέτα και το βραχιόλι έκτακτης ανάγκης .



Η μακέτα του έξυπνου σπιτιού και του βραχιολιού έκτακτης ανάγκης για ηλικιωμένους

2019 © Πανεπιστήμιο Πελοποννήσου | Πάτρα

Εικόνα 123: Το δεύτερο μισό της ιστοσελίδας όπως φαίνεται από τον browser του κινητού

Διαγράμματα ιστοσελίδας βραχιολιού έκτακτης ανάγκης:

Smart home and emergency bracelet for elderly

SMART HOME

First Page

thingspeak

Temperature

Humidity

Fan

Fire

Flood

Garage

Antlia

Porta

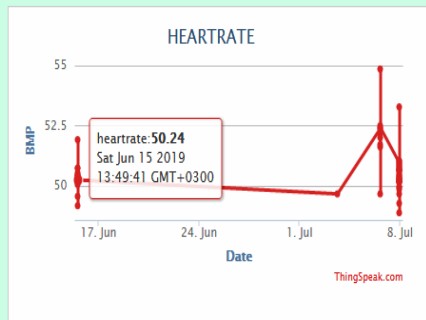
Google charts

Statistics

Gauge

Διάγραμμα καρδιακών παλμών συναρτήσει χρόνου δυναμικής ανανέωσης ανα 20 δευτερόλεπτα μέσω του Thingspeak.

Στο παρακάτω διάγραμμα απεικονίζονται οι τιμές των καρδιακών παλμών στο βραχιόλι έκτακτης ανάγκης για ηλικιωμένους σε συνάρτηση με το χρόνο.



Εικόνα 124: Διάγραμμα Thingspeak βραχιολιού καρδιακών παλμών όπως φαίνεται στην ιστοσελίδα

Smart home and emergency bracelet for elderly

SMART HOME

First Page

thingspeak

Temperature

Humidity

Fan

Fire

Flood

Garage

Antlia

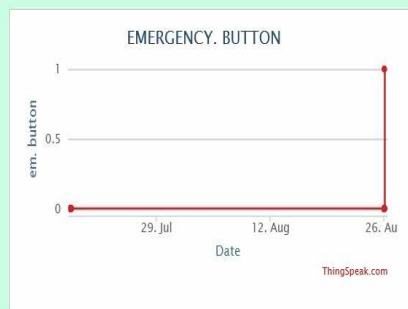
Porta

Google charts

Statistics

Διάγραμμα κουμπιού έκτακτης ανάγκης συναρτήσει χρόνου δυναμικής ανανέωσης ανά 20 δευτερόλεπτα μέσω του Thingspeak.

Στο παρακάτω διάγραμμα απεικονίζονται οι τιμές του κουμπιού έκτακτης ανάγκης στο βραχιόλι έκτακτης ανάγκης για ηλικιωμένους σε συνάρτηση με το χρόνο.



Εικόνα 125: Διάγραμμα Thingspeak βραχιολιού κουμπιού έκτακτης ανάγκης όπως φαίνεται στην ιστοσελίδα

Smart home and emergency bracelet for elderly

HOME PAGE

SMART HOME

First Page

thingspeak

Temperature

Humidity

Fan

Fire

Flood

Garage

Antlia

Porta

Google charts

Statistics

Gauge

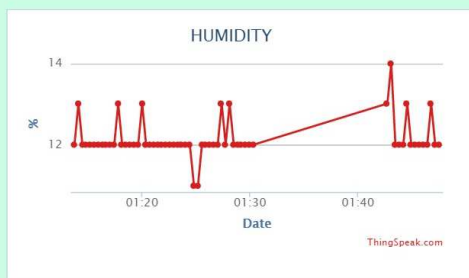
Weekly Statistics

Monthly Statistics

Data base

Διάγραμμα υγρασίας συναρτήσει χρόνου δυναμικής ανανέωσης ανα 20 δευτερόλεπτα μέσω του Thingspeak.

Στο παρακάτω διάγραμμα απεικονίζονται τιμές της υγρασίας στο βραχιόλι έκτακτης ανάγκης για ηλικιωμένους σε συνάρτηση με το χρόνο.



Εικόνα 126: Διάγραμμα Thingspeak βραχιολιού ένδειξης υγρασίας όπως φαίνεται στην ιστοσελίδα

Διαγράμματα-εποπτεία ιστοσελίδας έξυπνου σπιτιού:

Smart home and emergency bracelet for elderly

SMART HOME

First Page

thingspeak

Temperature

Humidity

Fan

Fire

Flood

Garage

Antlia

Porta

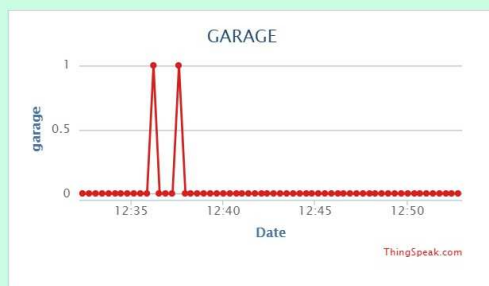
Google charts

Statistics

Gauge

Διάγραμμα γκαραζόπορτας συναρτήσει χρόνου δυναμικής ανανέωσης ανα 20 δευτερόλεπτα μέσω του Thingspeak.

Στο παρακάτω διάγραμμα απεικονίζονται οι τιμές της γκαραζόπορτας στο έξυπνο σπίτι σε συνάρτηση με το χρόνο.



Εικόνα 127: Διάγραμμα Thingspeak γκαραζόπορτας όπως φαίνεται στην ιστοσελίδα

Smart home and emergency bracelet for elderly

SMART HOME

First Page

thingspeak

Temperature

Humidity

Fan

Fire

Flood

Garage

Antlia

Porta

Google charts

Statistics

Gauge

Weekly Statistics

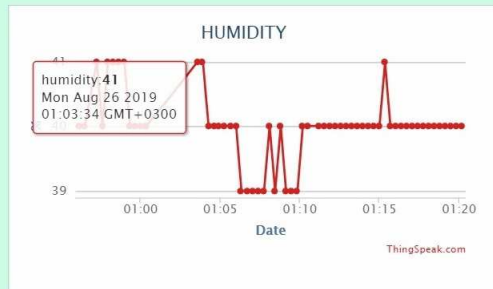
Monthly Statistics

Data base

Display

Διάγραμμα υγρασίας συναρτήσει χρόνου δυναμικής ανανέωσης ανα 20 δευτερόλεπτα μέσω του Thingspeak.

Στο παρακάτω διάγραμμα απεικονίζονται οι τιμές της υγρασίας στο έξυπνο σπίτι σε συνάρτηση με το χρόνο.



Εικόνα 128: Διάγραμμα Thingspeak υγρασίας όπως φαίνεται στην ιστοσελίδα

Smart home and emergency bracelet for elderly

SMART HOME

First Page

thingspeak

Temperature

Humidity

Fan

Fire

Flood

Garage

Antlia

Porta

Google charts

Statistics

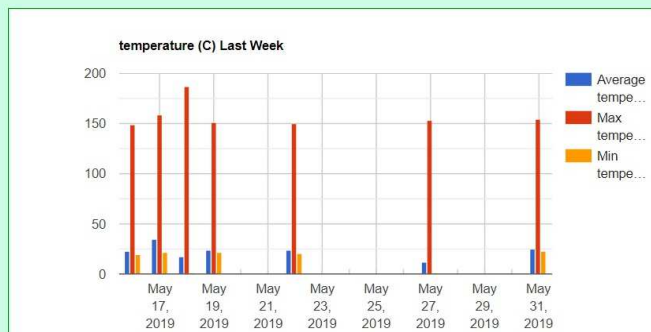
Gauge

Weekly Statistics

Monthly Statistics

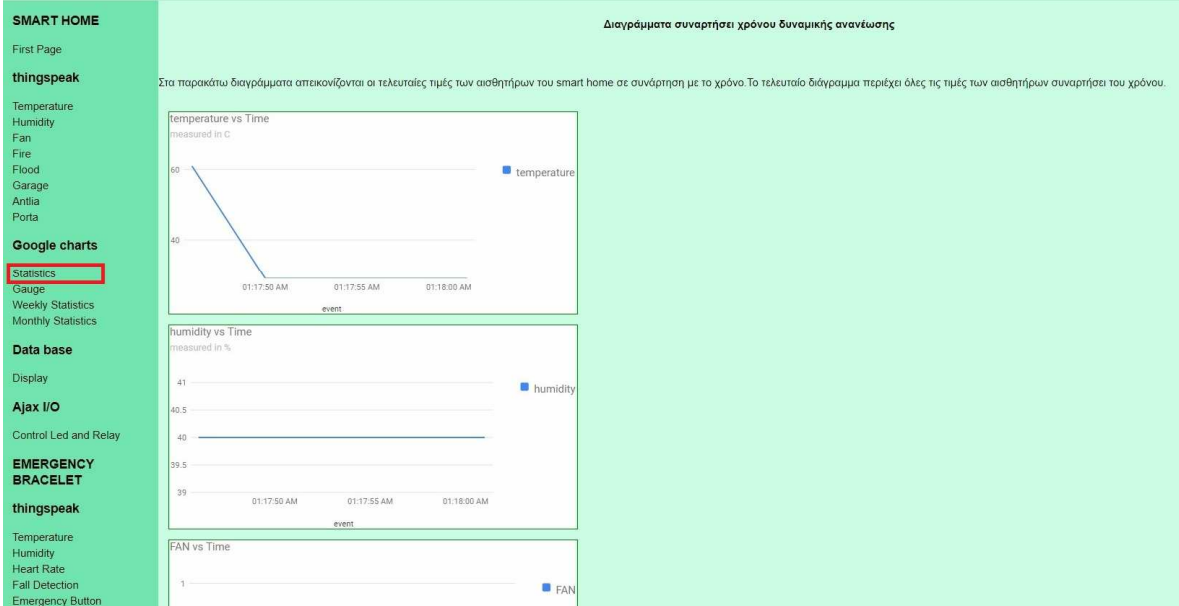
Εβδομαδιαία Στατιστικά

Στα παρακάτω διαγράμματα απεικονίζονται η ελάχιστη, μέγιστη και μέση τιμή των αισθητήρων (θερμοκρασίας, υγρασίας, εξαερισμού, φωτιάς, πλημμύρας, γκαράζ, πόρτας, αντλίας) για την κάθε μέρα της τελευταίας εβδομάδας.

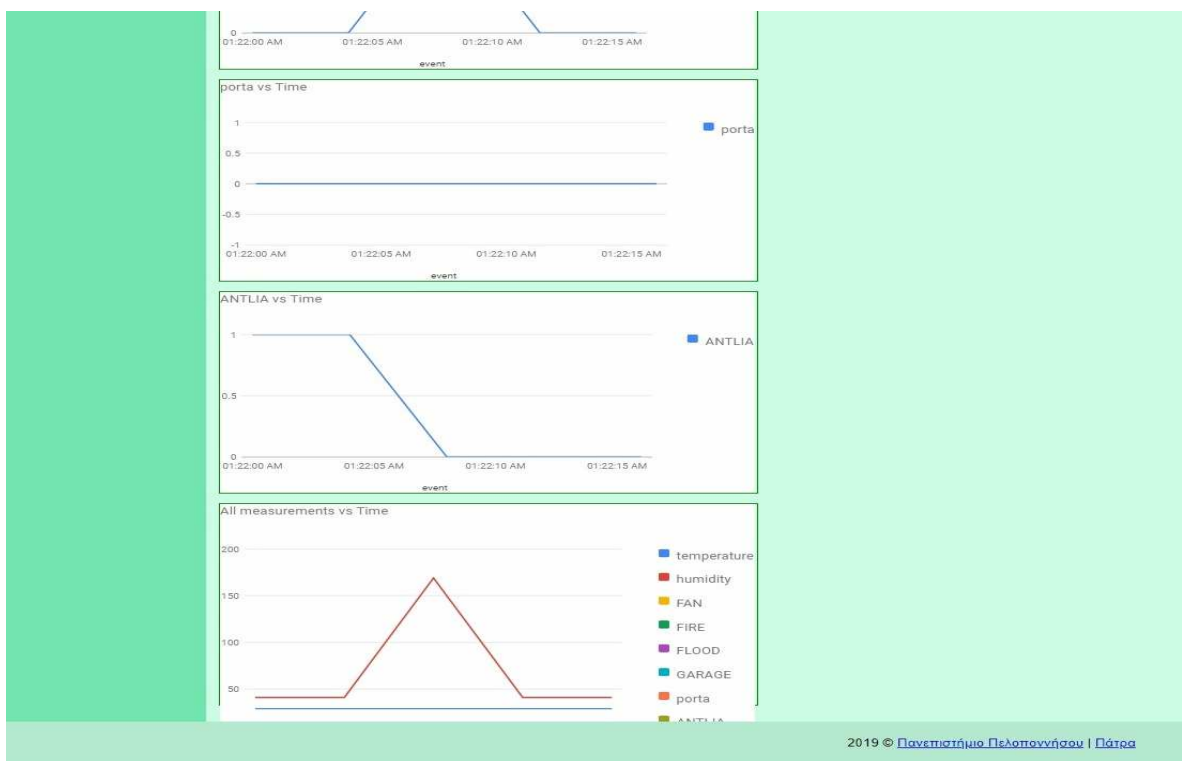


Εικόνα 129: Διάγραμμα Google charts εβδομαδιαίων στατιστικών όπως φαίνεται στην ιστοσελίδα

Smart home and emergency bracelet for elderly

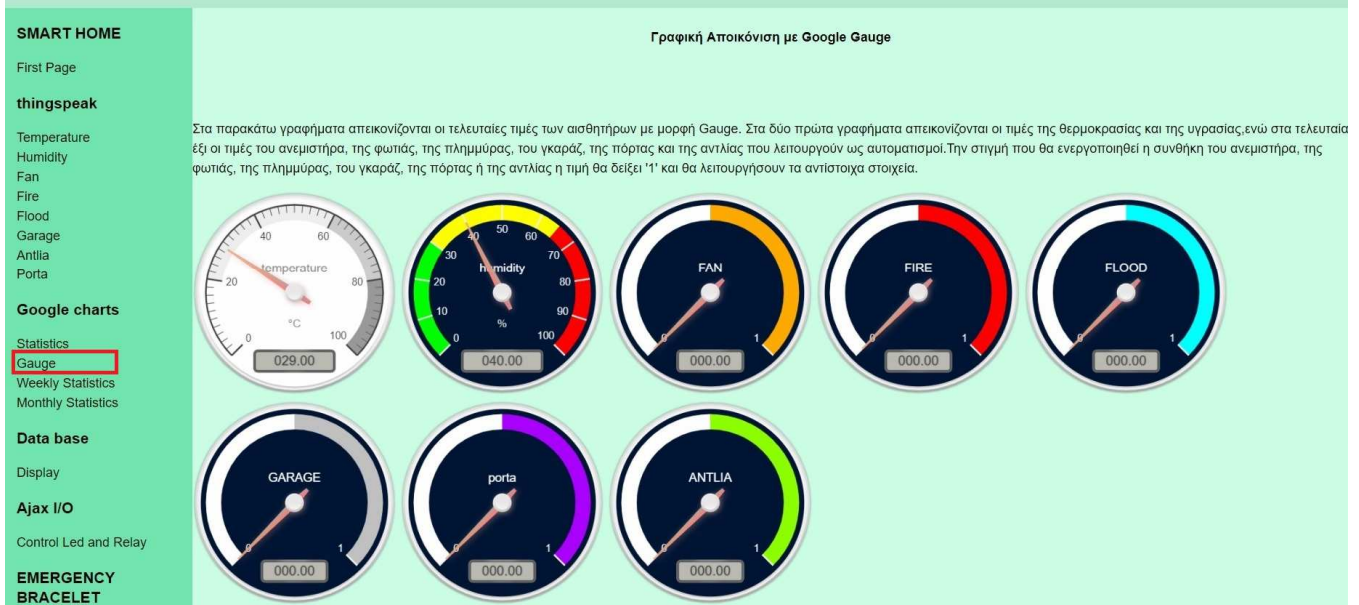


Εικόνα 130: Διάγραμμα Google charts στατιστικών όπως φαίνεται στην ιστοσελίδα(1)



Εικόνα 131: Διάγραμμα Google charts στατιστικών όπως φαίνεται στην ιστοσελίδα(2)

Smart home and emergency bracelet for elderly



Εικόνα 132: Διάγραμμα Google charts gauge όπως φαίνεται στην ιστοσελίδα

Smart home and emergency bracelet for elderly

SMART HOME

ID	Date and Time	Temperature	Humidity	FAN	FIRE	FLOOD	GARAGE	ANTLIA	porta
109704	2019-08-26 01:26:03	29	41	0	0	0	0	0	0
109703	2019-08-26 01:26:00	29	41	0	0	0	0	0	0
109702	2019-08-26 01:25:55	29	41	0	0	0	0	0	0
109701	2019-08-26 01:25:52	29	41	0	0	0	0	0	0
109700	2019-08-26 01:25:48	29	41	0	0	0	0	0	0
109699	2019-08-26 01:25:45	29	41	0	0	0	0	0	0
109698	2019-08-26 01:25:41	29	41	0	0	0	0	0	0
109697	2019-08-26 01:25:37	29	41	0	0	0	0	0	0
109696	2019-08-26 01:25:33	29	41	0	0	0	0	0	0
109695	2019-08-26 01:25:30	29	41	0	0	0	0	0	0
109694	2019-08-26 01:25:26	29	41	0	0	0	0	0	0
109693	2019-08-26 01:25:22	61	41	0	0	0	0	0	0
109692	2019-08-26 01:25:19	29	41	0	0	0	0	0	0
109691	2019-08-26 01:25:14	29	41	0	0	0	0	0	0
109690	2019-08-26 01:25:11	29	41	0	0	0	0	0	0
109689	2019-08-26 01:25:07	29	41	0	0	0	0	0	0
109688	2019-08-26 01:25:04	29	41	0	0	0	0	0	0
109687	2019-08-26 01:25:00	29	41	0	0	0	0	0	0
109686	2019-08-26 01:24:56	29	41	0	0	0	0	0	0
109685	2019-08-26 01:24:52	29	41	0	0	0	0	0	0
109684	2019-08-26 01:24:49	29	42	0	0	0	0	0	0
109683	2019-08-26 01:24:45	29	42	0	0	0	0	0	0
109682	2019-08-26 01:24:41	29	42	0	0	0	0	0	0

Εικόνα 133: Πίνακας βάσης δεδομένων όπως φαίνεται στην ιστοσελίδα

Smart home and emergency bracelet for elderly

SMART HOME

First Page

thingspeak

Temperature
Humidity
Fan
Fire
Flood
Garage
Antlia
Porta

Google charts

Statistics
Gauge
Weekly Statistics
Monthly Statistics

Data base

Display

Ajax I/O

Control Led and Relay

Sensors Inputs/ Control Devices Remotely

Analog Inputs

gas sensor: 82

temperature sensor: 755

blinds photoresistor: 104

soil moisture sensor: 1021

watering photoresistor: 64

Switch Inputs

AC button (D2): OFF

IR receiver (D3): ON

FAN (D5): OFF

Control Devices Using Checkboxes

lights

boiler

Control Devices Using Buttons

lights ON (D8)

boiler OFF (D9)

D10 to D13 used by Ethernet shield

Εικόνα 134: Ιστοσελίδα εποπτείας και ελέγχου στην κάρτα SD όπως φαίνεται στην ιστοσελίδα

Κεφάλαιο 6: Συμπεράσματα- Μελλοντικές Επεκτάσεις

Με την συγκεκριμένη πτυχιακή εργασία δόθηκε η δυνατότητα εκμάθησης της πλατφόρμας Arduino, ενός ισχυρού μικροελεγκτή με τον οποίο μπορούν να κατασκευαστούν πληθώρα εφαρμογών, σε συνεργασία με διάφορους αισθητήρες και συσκευές δράσης.

Το θέμα της πτυχιακής εργασίας ήταν η δημιουργία κάποιων αυτοματισμών έξυπνου σπιτιού, σκοπός των οποίων είναι να προσφέρουν ευκολία, άνεση, ασφάλεια και ευχαρίστηση στους κατοίκους του. Αυτό επιτεύχθηκε με οχτώ αυτόματα σενάρια με τα οποία δύναται ο έλεγχος:

- του αυτόματου ποτίσματος
- της αντλίας ανελκυστήρα
- της γκαραζόπορτας
- του κλιματισμού και ποιότητας αέρα
- των παντζουριών και εξωτερικού φωτισμού
- και του φωτισμού της αίθουσας ψυχαγωγίας.

Προσφέρεται επίσης επιπλέον ασφάλεια και άνεση με την πρόσβαση στο σπίτι χωρίς την χρήση κλειδιών καθώς και ανίχνευση φωτιάς και πλημμύρας.

Επίσης βασικός στόχος ήταν η παρακολούθηση της υγείας ηλικιωμένων ατόμων και η έγκαιρη ενημέρωση των οικείων τους σε περιπτώσεις κινδύνου ώστε να αποφευχθούν δυσμενέστερες καταστάσεις. Για το σκοπό αυτό κατασκευάστηκε το βραχιόλι έκτακτης ανάγκης ηλικιωμένων ή ατόμων με ιατρικές παθήσεις το οποίο:

- Παρέχει ζωτικές ενδείξεις (όπως καρδιακοί παλμοί) και χρήσιμες πληροφορίες (όπως ώρα, θερμοκρασία και υγρασία) στα ίδια τα άτομα και σε οικείους του.
- Στέλνει αυτόματες ειδοποιήσεις σε περιπτώσεις κινδύνου όπως πτώση, αυξημένοι καρδιακοί παλμοί και σε οποιαδήποτε άλλη περίπτωση στην οποία ο ηλικιωμένος θα κρίνει ότι είναι αναγκαίο πατώντας το κουμπί έκτακτης ανάγκης (SOS).

Τέλος κρίνεται πολύ χρήσιμη η δυνατότητα ελέγχου και εποπτείας όλων των αυτοματισμών από απόσταση. Για αυτόν τον λόγο δημιουργήθηκε η ιστοσελίδα του έξυπνου σπιτιού και του βραχιολιού στην οποία συγκεντρώθηκαν όλα τα διαγράμματα καταστάσεων των αυτοματισμών καθώς δόθηκε και η δυνατότητα ελέγχου κάποιων συσκευών από απόσταση.

Μελλοντικές επεκτάσεις:

Έξυπνο σπίτι:

- Ένας επιπλέον αυτοματισμός ο οποίος θα ήταν χρήσιμος για την λειτουργία του έξυπνου σπιτιού είναι ο έλεγχος των παραθύρων, σύμφωνα με την διαφορά εσωτερικής και εξωτερικής θερμοκρασίας, για ελαχιστοποίηση των απωλειών θερμότητας στον χώρο.
- Μπορούν να χρησιμοποιηθούν περισσότεροι αισθητήρες ανά αυτόματο σενάριο ώστε να αυξηθούν οι παράμετροι ενεργοποίησης κάποιας συνθήκης. Για παράδειγμα, στο σενάριο ποιότητας αέρα, να τοποθετηθούν αισθητήρες που θα μετράνε και άλλα στοιχεία του αέρα για την ενεργοποίηση του εξαερισμού. Επίσης στο σενάριο φωτιάς να τοποθετηθεί επιπλέον και αισθητήρας καπνού για καλύτερη πρόληψη του φαινομένου.

Βραχιόλι έκτακτης ανάγκης για ηλικιωμένους ή ατόμων με ιατρικές παθήσεις

- Μία επιπλέον δυνατότητα που θα βελτιώνει το βραχιόλι είναι η προσθήκη ενός pedometer. Αυτό μπορεί να αξιοποιηθεί είτε για την παρακολούθηση από απόσταση της κίνησης του ηλικιωμένου ή του ατόμου με ιατρικές παθήσεις μέσα στο σπίτι, είτε σε περίπτωση που ο ηλικιωμένος ή το άτομο με ιατρικές παθήσεις, επιθυμεί να επιβλέπει την απόσταση που διανύει σε έναν περίπατο.
- Τέλος το βασικότερο στοιχείο (το οποίο δεν πραγματοποιήθηκε λόγω προβλήματος arduino nano και υψηλού κόστους), είναι η χρησιμοποίηση μικρών αισθητήρων, πλακέτας arduino (nano ή μικρότερο) και μπαταρίας, ώστε το βραχιόλι να αποκτήσει φορητότητα και όσο το δυνατόν μικρότερες διαστάσεις.

Βιβλιογραφία

Διαδίκτυο:

<https://www.wikipedia.org>

<http://forum.hobbycomponents.com/viewtopic.php?f=110&t=2114>

<https://www.codeproject.com/Articles/845538/An-Introduction-to-ThingSpeak>

<https://thingspeak.com>

<https://grobotronics.com/esp8266-wifi-module.html>

<https://www.pinterest.com/pin/371054456794131758/>

<https://www.bcrpainting.com.au/bcrpainterblogs/2017/9/1/why-should-you-pay-attention-to-colour-psychology>

https://www.tutorialspoint.com/arduino/arduino_dc_motor.htm

<https://arduinomodules.info>

https://www.mouser.com/catalog/specsheets/A000056_DATASHEET.pdf

<https://www.instructables.com/id/Arduino-Ethernet-Shield-Tutorial/>

<https://www.elprocus.com/different-types-of-arduino-boards/>

<https://learn.sparkfun.com>

<https://www.mathworks.com/help/thingspeak/use-ifttt-to-send-text-message-notification.html>

<https://startingelectronics.org>

https://www.quackit.com/html/templates/layout_templates.cfm

<https://www.banggood.com>

<https://ifttt.com>

<https://www.arduino.cc>

<https://jeffwhelpley.com/category/css/>

https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics

<https://dwh-uk.com/tutorials-what-is-php/>

<https://www.meow.bz/what-is-an-ajax-advantages-and-disadvantages-of-using-jacks-technology/>

https://www.ntu.edu.sg/home/ehchua/programming/sql/MySQL_HowTo.html

<https://torquemag.io/2015/11/install-wordpress-locally-xampp-windows-mac/>

<https://www.ybierling.com/en/blog-officeproductivity-comparetextfileswithnotepadplusplus>

<https://www.euro2day.gr/specials/opinions/article/1555381/h-tehnologikh-proodos-h-ypokatastash-toy-anthropoy.html>

<http://www.smart-systems.gr/%CE%AD%CE%BE%CF%85%CF%80%CE%BD%CE%BF-%CF%83%CF%80%CE%AF%CF%84%CE%B9>

<http://www.opencourses.gr/opencourse.xhtml;jsessionid=0CACC42524FED014112962091BEBBA5E?id=14104&ln=el>

<http://comparecamp.com/google-chart-tools-review-pricing-pros-cons-features/>

<https://blog.weekdone.com/freebie-google-chart-styles/>

<https://www.pcmag.com/review/293236/ifttt>

<https://www.pocket-lint.com/smart-home/news/130082-what-is-ifttt-and-how-does-it-work>

https://indico.cern.ch/event/654636/contributions/2834242/attachments/1636045/2610243/MATLAB_IoT_public.pdf

<https://medium.com/acellere/is-javascript-a-programming-language-b068b6eda749>

<https://aeroarduino.com/2018/06/23/connecting-arduino-uno-with-esp8266-and-to-thingspeak>

<https://electronics hobbyists.com/logging-data-to-database-using-arduino-ethernet-shield>

<https://www.instructables.com/id/Emergency-Fall-Notifier-Cum-Panic-Button>

Παράρτημα Α

Κώδικας έξυπνου σπιτιού που τρέχει στο Arduino Mega

```
#include <SPI.h>
#include <Ethernet.h>      //ethernet
#include <SD.h>
#define REQ_BUF_SZ  60
File webFile;           // the web page file on the SD card
char HTTP_req[REQ_BUF_SZ] = {0}; // buffered HTTP request stored as
null terminated string
char req_index = 0;      // index into HTTP_req buffer
EthernetServer server1(80);
boolean LED_state[4] = {0}; // Αποθηκεύει τις καταστάσεις των
συσκευών δράσης

int Wait_between_dumps = 0;

byte mac[] = { 0x90, 0xA2, 0xDA, 0x10, 0x40, 0x4F };

EthernetClient client;
#include "ThingSpeak.h"           // thingspeak
IPAddress ip(192, 168, 1, 5);
IPAddress serv(192, 168, 1, 2);
unsigned long myChannelNumber = 774003;
const char * myWriteAPIKey = "WPM6HMY2J5TH6N8Q";
int period = 20000;
unsigned long time_now = 0;

#define RELAY1  47      //rele

int ANTLIA=34;          //asanser
#define trigPin 35
#define echoPin 36
int flag=0;

int sensorPin2 = A4;    //potisma
int sensorPin1 = A3;
int potisma=33;
int water_value ;
int sun_Value1 = 0; // variable to store the value coming from the
sensor

#include <IRremote.h>    //RGB
int RECV_PIN = 3;
int redPin = 7;
int greenPin = 8;
int bluePin = 9;
IRrecv irrecv(RECV_PIN);
decode_results results;

int tiltSwitch = 32; // Tilt Switch Input           //garage
int garage; // variable to store tilt input
int buzzer=6;
int greenled1=49;
int redled1=42;

int Grove_Water_Sensor=28; // WATER-FLAME
int FLAME=22;
```

```

int ALARM=27;
int FLOOD=48;
int FIRE=46;

#include <DS1302.h> //eks.fwta- patzouria
int led=45;
int sensorPin = A2;
int sun_Value = 0;
int led1=44;
DS1302 rtc(24,25,26);
Time t;
const int OnHour = 00;
const int OnMin = 04;
const int OffHour = 00;
const int OffMin = 06;

int sensorValue; //atmosfaira
int fan1=0;
int pinButton = 2;
int stateLED = HIGH;
int stateButton;
int previous = LOW;
long time = 0;
long debounce = 200;
int FAN=5;
int AC=29;
#include <dht.h>
#define dht_dpın A1
dht dht;

void setup()
{
  pinMode(10, OUTPUT); //Ανεργοποίηση Ethernet chip
  digitalWrite(10, HIGH); //Ενεργοποίηση Ethernet chip
  Serial.begin(9600); //setting the baud rate at 9600
  SD.begin(4);

  Ethernet.init(10); // Most Arduino Ethernet hardware
  //thingspeak

  // start the Ethernet connection:
  Serial.println("Initialize Ethernet with DHCP:");
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // Check for Ethernet hardware present
    if (Ethernet.hardwareStatus() == EthernetNoHardware) {
      Serial.println("Ethernet shield was not found. Sorry, can't
run without hardware. :(");
      while (true) {
        delay(1); // do nothing, no point running without Ethernet
hardware
      }
    }
    if (Ethernet.linkStatus() == LinkOFF) {
      Serial.println("Ethernet cable is not connected.");
    }
    // try to congifure using IP address instead of DHCP:
    Ethernet.begin(mac, ip,serv );

```

```

} else {
  Serial.print(" DHCP assigned IP ");
  Serial.println(Ethernet.localIP());
}
// give the Ethernet shield a second to initialize:

delay(1000);

ThingSpeak.begin(client); // Initialize ThingSpeak

//Ethernet.begin(mac, ip); //ethernet

pinMode(30, OUTPUT);

pinMode(RELAY1, OUTPUT); //rele

pinMode(ANTLIA,OUTPUT); //asanser
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);

pinMode(potisma,OUTPUT); //potisma

pinMode(redPin,OUTPUT); //RGB
pinMode(greenPin,OUTPUT);
pinMode(bluePin,OUTPUT);
irrecv.enableIRIn(); // Start the receiver

pinMode(buzzer,OUTPUT); //garage
pinMode(greenled1,OUTPUT);
pinMode(redled1,OUTPUT);
pinMode (tiltSwitch, INPUT) ;// define the mercury tilt switch
sensor output interface

Serial.println("Fire and Flood test"); // WATER-FLAME
pinMode(FLAME, INPUT); //define FLAME input pin
pinMode(ALARM, OUTPUT); //define ALARM output pin
// Flame sensor code for Robojax.com
pinMode (FLOOD,OUTPUT);
pinMode(Grove_Water_Sensor, INPUT); // The Water Sensor is an
Input
pinMode(FIRE,OUTPUT ); // The LED is an Output

pinMode(pinButton, INPUT); //atmosfaira
pinMode(FAN, OUTPUT);
Serial.println("Air Pollution Level\n\n");
pinMode(AC,OUTPUT);
Serial.println("Humidity and temperature\n\n");

rtc.halt(false); //eks.fwta- patzouria
rtc.writeProtect(false);
pinMode (led,OUTPUT);
pinMode (led1,OUTPUT);

Ethernet.begin(mac, ip); // initialize Ethernet device
server1.begin(); // start to listen for clients
}

```

```

void loop()
{
  db();

  int antlia =digitalRead(ANTLIA);           //asanser
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;

  if (distance <=9) flag=1;
  if (distance >=16) flag=0;
  if (flag==1)
  {
    do{
      digitalWrite(ANTLIA,HIGH);
      Serial.println("antlia on"); }
    while(distance>=16);
  }
  else
  {
    Serial.println("antlia off");
    digitalWrite(ANTLIA,LOW);
  }

  water_value= analogRead(sensorPin1);       // potisma
  sun_Valuel = analogRead(sensorPin2);
  water_value = map(water_value,350,1023,120,0);
  sun_Valuel = map(sun_Valuel,15,800,100,0);

  if(sun_Valuel<60 && (water_value<40 ||water_value<80 ))

    {digitalWrite(potisma,HIGH); }

    else
    {digitalWrite(potisma,LOW); }
    Serial.println(water_value);
    Serial.println(sun_Valuel);

  if (irrecv.decode(&results))             //RGB
  {
    Serial.println(results.value, HEX);
    translateIR();
    for (int z=0; z<2; z++) // ignore 2nd and 3rd signal repeat
    {
      irrecv.resume(); // receive the next value
    }
  }

  garage = digitalRead (tiltSwitch) ;// read the switch value
  //garage
  if (garage == LOW) // Means we've tilted
  {
    digitalWrite(buzzer,HIGH);
  }
}

```



```

delay(100);
digitalWrite(buzzer,LOW);
delay(100);
digitalWrite(redled1,HIGH);
digitalWrite(greenled1,LOW);
Serial.println("*garage open");
}

else
{
digitalWrite(greenled1,HIGH);
digitalWrite(redled1,LOW);
digitalWrite(buzzer,LOW);
Serial.println("garage closed");
}

stateButton = digitalRead(pinButton); //atmosfaira
sensorValue = analogRead(0); // read analog input pin 0
int fan =digitalRead(FAN);
if(sensorValue>400)
{
digitalWrite(FAN,HIGH );
}
else
digitalWrite(FAN,LOW );
Serial.print("air pollution = ");
Serial.println(sensorValue, DEC); // prints the value read

if ((dht.temperature>28 || dht.temperature>24)|| (dht.temperature<15
|| dht.temperature<22))
{digitalWrite(AC,HIGH );
}
else digitalWrite(AC,LOW );

if(dht.temperature>=21 && dht.temperature<=24){

if(stateButton == HIGH && previous == LOW && millis() - time >
debounce) {
if(stateLED == HIGH){
stateLED = LOW;
} else {
stateLED = HIGH;
}
time = millis();
}

digitalWrite(AC, stateLED);
previous = stateButton;}

dht.read11(dht_dp);

Serial.print("Current humidity = ");
Serial.print(dht.humidity);
Serial.print("% ");
Serial.print("temperature = ");
Serial.print(dht.temperature);
Serial.println("C ");

t = rtc.getTime(); //eks.fwta- patzouria
Serial.print(t.hour);
Serial.print(" hour(s), ");

```

```

Serial.print(t.min);
Serial.print(" minute(s)");
Serial.println(" ");

sun_Value = analogRead(sensorPin);
sun_Value = map(sun_Value,100,800,100,0);
if(sun_Value<50){
  digitalWrite(led,HIGH);
  Serial.println("LIGHT ON");
  if (t.hour == OnHour && t.min == OnMin)
  {digitalWrite(led1,HIGH);
  Serial.println("pantzouri kleisto");}
}

else if(sun_Value>40){
  digitalWrite(led,LOW);
  Serial.println("LIGHT OFF");
  if (t.hour == OffHour && t.min == OffMin)
  {digitalWrite(led1,LOW);
  Serial.println("pantzouri anoixto");}
}

int flood=digitalRead(Grove_Water_Sensor);
int fire = digitalRead(FLAME);// read FLAME sensor //
WATER-FLAME
if (flood==HIGH && fire == LOW ) {
  digitalWrite(ALARM,HIGH);
  digitalWrite(FIRE,HIGH);
  digitalWrite(FLOOD,HIGH);
  Serial.println("Flood fire !"); }

else if (digitalRead(Grove_Water_Sensor)==HIGH){
  digitalWrite(ALARM,HIGH);
  delay(200);
  digitalWrite(ALARM,LOW);
  delay(200);
  Serial.println("Flood!");
  digitalWrite(FLOOD,HIGH);
}
else if( fire == LOW)
{
  Serial.println("Fire! Fire!");
  digitalWrite(ALARM,HIGH);// set the buzzer ON
  delay(50);
  digitalWrite(ALARM,LOW);
  delay(50);
  digitalWrite(FIRE,HIGH);
}
else {
  digitalWrite(ALARM,LOW); // Set the buzzer OFF
  digitalWrite(FLOOD,LOW);
  Serial.println("Peace");
  Serial.println("Not Flood");
  digitalWrite(FIRE,LOW); }

//thingspeak
if(millis() > (time_now + period)){
  dht.read11(dht_dpIn);
float temp = dht.temperature;
float hum = dht.humidity;

```

```

    ThingSpeak.setField(1, temp);
    ThingSpeak.setField(2, hum);
    ThingSpeak.setField(3, fan );
    ThingSpeak.setField(4, !fire );
    ThingSpeak.setField(5, flood );
    ThingSpeak.setField(6, !garage );
    ThingSpeak.setField(7, antlia );

int x =ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    if(x == 200){
        Serial.println("Channel update successful.");
    }
    else{
        Serial.println("Problem updating channel. HTTP error code " +
String(x));
    }
time_now = millis();
}

//Επικοινωνία Arduino με ιστοσελίδα

EthernetClient client = server1.available(); // try to get client
if (client) { // got client?
    boolean currentLineIsBlank = true;
    while (client.connected()) {
        if (client.available()) { // client data available to
read
            char c = client.read(); // read 1 byte (character)
from client
            // limit the size of the stored received HTTP request
            // buffer first part of HTTP request in HTTP_req
array (string)
            // leave last element in array as 0 to null terminate
string (REQ_BUF_SZ - 1)
            if (req_index < (REQ_BUF_SZ - 1)) {
                HTTP_req[req_index] = c; // save HTTP
request character
                req_index++;
            }
            // last line of client request is blank and ends with
\n
            // respond to client only after last line received
            if (c == '\n' && currentLineIsBlank) {
                // send a standard http response header
                client.println("HTTP/1.1 200 OK");
                // remainder of header follows below, depending
on if
                // web page or XML page is requested
                // Ajax request - send XML file
                if (StrContains(HTTP_req, "ajax_inputs")) {
                    // send rest of HTTP header
                    client.println("Content-Type: text/xml");
                    client.println("Connection: keep-alive");
                    client.println();
                    SetLEDs();
                    // send XML file containing input states
                    XML_response(client);
                }
                else { // web page request
                    // send rest of HTTP header
                    client.println("Content-Type: text/html");
                    client.println("Connection: keep-alive");

```

```

        client.println();
        // send web page
        webFile = SD.open("index1.htm");           //
open web page file
        if (webFile) {
            while(webFile.available()) {
web page to client
                client.write(webFile.read()); // send
            }
            webFile.close();
        }
        // display received HTTP request on serial port
        Serial.print(HTTP_req);
        // reset buffer index and all buffer elements to
0
        req_index = 0;
        StrClear(HTTP_req, REQ_BUF_SZ);
        break;
    }
    // every line of text received from the client ends
with \r\n
    if (c == '\n') {
        // last character on line of received text
        // starting new line with next character read
        currentLineIsBlank = true;
    }
    else if (c != '\r') {
        // a text character was received from client
        currentLineIsBlank = false;
    }
    } // end if (client.available())
} // end while (client.connected())
delay(1); // give the web browser time to receive the
data
    client.stop(); // close the connection
} // end if (client)

}

```

```

void db() {
    float temp = dht.temperature;
    float hum = dht.humidity;
    int fan =digitalRead(FAN);
    int flood=digitalRead(Grove_Water_Sensor);
int fire = digitalRead(FLAME); // read FLAME sensor
garage = digitalRead (tiltSwitch) ;
int antlia =digitalRead(ANTLIA);

    Wait_between_dumps++;
    if (Wait_between_dumps >=10)
    {
        if (client.connect(serv, 80)) { //Connecting at the IP address and
port we saved before
            Serial.println("connected");
            client.print("GET /ethernet/data.php?"); //Connecting and Sending
values to database

```

```

client.print("temperature=");
client.print(temp);
client.print("&&humidity=");
client.print(hum);
client.print("&&FAN=");
client.print( fan );
client.print("&&FIRE=");
client.print( !fire );
client.print("&&FLOOD=");
client.print( flood );
client.print("&&ANTLIA=");
client.print(antlia);
client.print("&&GARAGE=");
client.print( !garage );
Wait_between_dumps = 0;

client.println( " HTTP/1.1");
client.print( "Host: " );
client.println(serv);
client.println( "Connection: close" );
client.println();
client.println();

client.stop(); //Closing the connection
}
else {
// if you didn't get a connection to the server:
Serial.println("connection failed");
}

}

}

// send the XML file with analog values, switch status
// and LED status
void XML_response(EthernetClient cl)
{
    int analog_val;           // stores value read from analog
    inputs
    int count;                // used by 'for' loops
    int sw_arr[] = {2, 3, 5}; // pins interfaced to switches

    cl.print("<?xml version = \"1.0\" ?>");
    cl.print("<inputs>");
    // read analog inputs
    for (count = 0; count <= 4; count++) { // A0 to A4
        analog_val = analogRead(count);
        cl.print("<analog>");
        cl.print(analog_val);
        cl.println("</analog>");
    }
    // read switches
    for (count = 0; count < 3; count++) {
        cl.print("<switch>");
        if (digitalRead(sw_arr[count])) {
            cl.print("ON");
        }
        else {
            cl.print("OFF");
        }
        cl.println("</switch>");
    }
}

```

```

    }
    // checkbox LED states
    // LED1
    cl.print("<LED>");
    if (LED_state[0]) {
        cl.print("checked");
    }
    else {
        cl.print("unchecked");
    }
    cl.println("</LED>");
    // LED2
    cl.print("<LED>");
    if (LED_state[1]) {
        cl.print("checked");
    }
    else {
        cl.print("unchecked");
    }
    cl.println("</LED>");
    // button LED states
    // LED3
    cl.print("<LED>");
    if (LED_state[2]) {
        cl.print("on");
    }
    else {
        cl.print("off");
    }
    cl.println("</LED>");
    // LED4
    cl.print("<LED>");
    if (LED_state[3]) {
        cl.print("on");
    }
    else {
        cl.print("off");
    }
    cl.println("</LED>");

    cl.print("</inputs>");
}

// sets every element of str to 0 (clears array)
void StrClear(char *str, char length)
{
    for (int i = 0; i < length; i++) {
        str[i] = 0;
    }
}

// searches for the string sfind in the string str
// returns 1 if string found
// returns 0 if string not found
char StrContains(char *str, char *sfind)
{
    char found = 0;
    char index = 0;
    char len;

    len = strlen(str);

```

```

    if (strlen(sfind) > len) {
        return 0;
    }
    while (index < len) {
        if (str[index] == sfind[found]) {
            found++;
            if (strlen(sfind) == found) {
                return 1;
            }
        }
        else {
            found = 0;
        }
        index++;
    }

    return 0;
}

// checks if received HTTP request is
switching on/off LEDs
// also saves the state of the LEDs
void SetLEDs(void)
{
    // LED 1 (pin 6)
    if (StrContains(HTTP_req, "LED1=1")) {
        LED_state[0] = 1; // save LED state
        digitalWrite(30, HIGH);
    }
    else if (StrContains(HTTP_req, "LED1=0")) {
        LED_state[0] = 0; // save LED state
        digitalWrite(30, LOW);
    }
    // LED 2 (pin 7)
    if (StrContains(HTTP_req, "LED2=1")) {
        LED_state[1] = 1; // save LED state
        digitalWrite(47, HIGH);
    }
    else if (StrContains(HTTP_req, "LED2=0")) {
        LED_state[1] = 0; // save LED state
        digitalWrite(47, LOW);
    }
    // LED 3 (pin 8)
    if (StrContains(HTTP_req, "LED3=1")) {
        LED_state[2] = 1; // save LED state
        digitalWrite(30, HIGH);
    }
    else if (StrContains(HTTP_req, "LED3=0")) {
        LED_state[2] = 0; // save LED state
        digitalWrite(30, LOW);
    }
    // LED 4 (pin 9)
    if (StrContains(HTTP_req, "LED4=1")) {
        LED_state[3] = 1; // save LED state
        digitalWrite(47, HIGH);
    }
    else if (StrContains(HTTP_req, "LED4=0")) {
        LED_state[3] = 0; // save LED state
        digitalWrite(47, LOW);
    }
}

```

```

void translateIR() // takes action based on IR code received
// describing Sony IR codes on LCD module
{
  switch(results.value)
  {
    case 0xFF22DD:
    setColor(255, 0, 0); //Green Color
    break;
    case 0xFF02FD:
    setColor(0, 255, 0); // Red Color
    break;
    case 0xFFC23D:
    setColor(0, 0, 255); // Blue Color
    break;
    case 0xFFE01F:
    setColor(255, 255, 255); // White Color
    break;
    case 0xFFA857:
    setColor(50, 255, 200); // Orange Color
    break;
    case 0xFF906F:
    setColor(0, 0, 0); // off
    break;

    default:
    Serial.println("OTHER");
  }
  //delay(200);
}

void setColor(int redValue, int greenValue, int blueValue) {
  analogWrite(redPin, redValue);
  analogWrite(greenPin, greenValue);
  analogWrite(bluePin, blueValue);}

```

Κώδικας έξυπνου σπιτιού που τρέχει στο Arduino Uno

```

#include <SPI.h> //ethernet
#include <Ethernet.h>
byte mac[] = { 0xB0, 0xAC, 0xD2, 0x2A, 0xD7, 0x91 };
byte ip[] = {192, 168, 1, 9 }; //Enter the IP of ethernet shield
byte serv[] = {192, 168, 1, 2} ; //Enter the IPv4 address
EthernetClient client;
int Wait_between_dumps = 0;

#include "ThingSpeak.h" //thingspeak
unsigned long myChannelNumber = 774003;
const char * myWriteAPIKey = "WPM6HMY2J5TH6N8Q";
//byte mac[] = { 0x90, 0xA2, 0xDA, 0x10, 0x40, 0x4F };
int period1 = 15000;
unsigned long time_now1 = 0;

int led1 = 6; // the pin that the LED is attached to
//asfaleia
int sensor = 2; // the pin that the sensor is attached
to

```



```

int state = LOW;           // by default, no motion detected
int val = 0;              // variable to store the sensor status
                           (value)
#define ctsPin 9
int flag1=0;
int flag=0;
int counter=0;
int led2=7;
int led3=8;
int period = 10000;
unsigned long time_now = 0;
int led=5;
int a=0;
int b=0;
int c=0;
const int buttonPin = 4;
int buttonState = 0;
#include <IRremote.h>
int RECV_PIN = 3;

IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{

Ethernet.init(10); // Most Arduino Ethernet hardware
//thingspeak
  //Serial.begin(115200); //Initialize serial

  // start the Ethernet connection:
  Serial.println("Initialize Ethernet with DHCP:");
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // Check for Ethernet hardware present
    if (Ethernet.hardwareStatus() == EthernetNoHardware) {
      Serial.println("Ethernet shield was not found. Sorry, can't
run without hardware. :(");
      while (true) {
        delay(1); // do nothing, no point running without Ethernet
hardware
      }
    }
    if (Ethernet.linkStatus() == LinkOFF) {
      Serial.println("Ethernet cable is not connected.");
    }
    // try to congifure using IP address instead of DHCP:
    Ethernet.begin(mac, ip, serv );
  } else {
    Serial.print(" DHCP assigned IP ");
    Serial.println(Ethernet.localIP());
  }
  // give the Ethernet shield a second to initialize:

  delay(1000);

  ThingSpeak.begin(client); // Initialize ThingSpeak

Ethernet.begin(mac, ip);           //ethernet

```

```

    pinMode(led1, OUTPUT);      // initialize LED as an output
    //asfaleia
    pinMode(sensor, INPUT);    // initialize sensor as an input

pinMode(led2,OUTPUT);
pinMode(led3,OUTPUT);
pinMode(ctsPin, INPUT);

    pinMode(led,OUTPUT);

Serial.begin(9600);
irrecv.enableIRIn(); // Start the receiver

}

void loop()
{
    db();
    if (irrecv.decode(&results))           //asfaleia
    {
        Serial.println(results.value, HEX);
        translateIR();
        for (int z=0; z<2; z++) // ignore 2nd and 3rd signal repeat
        {
            irrecv.resume(); // receive the next value
        }
    }
    buttonState = digitalRead(buttonPin);
    if (buttonState == HIGH) {
        if(millis() > (time_now + period)){
            if (flag+flag1==2) digitalWrite(led3,HIGH);
            else digitalWrite(led3,LOW);
            counter=0;

            Serial.println("-----");
            time_now = millis();
            a=0; b=0; c=0;
            flag1=0;
            digitalWrite(led,LOW);
        }
        else{
            if (digitalRead(ctsPin)==HIGH) counter=counter+1;

            if (counter<5 &&counter>1 ){
                digitalWrite(led2,HIGH);
                flag=1;
            }
            else {
                flag=0;
                digitalWrite(led2,LOW);}

            if (a+b+c==3){
                digitalWrite(led,HIGH);
                flag1=1;
            }
        }
    }

    Serial.println(counter);

```

```

val = digitalRead(sensor); // read sensor value
if (val == HIGH) { // check if the sensor is HIGH
    digitalWrite(led1, HIGH); // turn LED ON
    delay(100); // delay 100 milliseconds

    if (state == LOW) {
        Serial.println("Motion detected!");
        state = HIGH; // update variable state to HIGH
    }
}
else {
    digitalWrite(led1, LOW); // turn LED OFF
    delay(200); // delay 200 milliseconds

    if (state == HIGH){
        Serial.println("Motion stopped!");
        state = LOW; // update variable state to LOW
    }
}

//thingspeak
if(millis() > (time_now + period)){

    ThingSpeak.setField(8, digitalRead(buttonPin));

int x =ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
if(x == 200){
    Serial.println("Channel update successful.");
}
else{
    Serial.println("Problem updating channel. HTTP error code " +
String(x));
}
time_now = millis();
}

}
void db(){
    Wait_between_dumps++;
if (Wait_between_dumps >= 10){
//ethernet
    if (client.connect(serv, 80)) { //Connecting at the IP address and
port we saved before
Serial.println("connected");
client.print("GET /ethernet/data.php?"); //Connecting and Sending
values to database

client.print("porta=");
client.println(digitalRead(buttonPin));

client.stop(); //Closing the connection
}
else {
// if you didn't get a connection to the server:
Serial.println("connection failed");
}
}
}

```

```

}
void translateIR() // takes action based on IR code received
// describing Sony IR codes on LCD module
{
switch(results.value)
{
case 0xFFA25D:

a=1;
break;
case 0xFF629D:

b=1;
break;
case 0xFFE21D:

c=1;
break;

default:
Serial.println("OTHER");
}
delay(200);
}

```

Παράρτημα Β

Κώδικας βραχιολιού έκτακτης ανάγκης για ηλικιωμένους ή άτομων με ιατρικές παθήσεις

```

#include<Wire.h>
const int emled=3;
const int MPU_addr=0x68; // I2C address of the MPU-6050
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
float ax=0, ay=0, az=0, gx=0, gy=0, gz=0;

//int data[STORE_SIZE][5]; //array for saving past data
//byte currentIndex=0; //stores current data array index (0-255)
boolean fall = false; //stores if a fall has occurred
boolean trigger1=false; //stores if first trigger (lower threshold)
has occurred
boolean trigger2=false; //stores if second trigger (upper threshold)
has occurred
boolean trigger3=false; //stores if third trigger (orientation
change) has occurred

byte trigger1count=0; //stores the counts past since trigger 1 was
set true
byte trigger2count=0; //stores the counts past since trigger 2 was
set true
byte trigger3count=0; //stores the counts past since trigger 3 was
set true
int angleChange=0;

```

```

int sensor = 0;          // heart rate
double alpha = 0;
double change = 0.0;
double minval = 0.0;
#include "U8glib.h"
U8GLIB_SH1106_128X64 u8g(U8G_I2C_OPT_NONE); // oled

const int buttonPin = 5;
int buttonState = 0;

#include <SoftwareSerial.h>
#define RX 6
#define TX 7
String AP = "COSMOTE-2AD791"; // CHANGE
String PASS = "xq9y2CxAYNNEATzH"; // CHANGE
String API = "FE11WY9CASMYAEPX"; // CHANGE
String HOST = "api.thingspeak.com";
String PORT = "80";
String field = "field1";
String field1 = "field2";
String field3 = "field3";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
boolean sendFall = false;
boolean emrgButton = false;
int valSensor = 1;
SoftwareSerial esp8266(RX,TX);

int period = 20000;
unsigned long time_now = 0;
unsigned long time_now1 = 0;
unsigned long time_now2 = 0;

#include <DS1302.h>
DS1302 rtc(8, 9, 10);

#include <dht.h>
#define dht_dpin A1
dht dht;

void draw(void) {

    static double oldValue = 0;
    static double oldChange = 0;
    int rawValue = analogRead (sensor);
    double value = alpha * oldValue + (1 - alpha) *
rawValue*100/1023;
    dht.read11(dht_dpin);
    u8g.setFont(u8g_font_profont12);
    u8g.setPrintPos(0, 10);
    u8g.print(" Temperature");

    u8g.print(dht.temperature);
    u8g.setPrintPos(0, 25);
    u8g.print("Humidity=");
    u8g.print(dht.humidity);
    u8g.setPrintPos(0, 40);

```

```

u8g.print("BPM: ");
u8g.print(value);
}

void draw2(void) {
u8g.setFont(u8g_font_profont12);
u8g.setPrintPos(0, 10);
u8g.print(rtc.getDOWStr());
u8g.setPrintPos(0, 25);
u8g.print(rtc.getDateStr());
u8g.setPrintPos(0, 40);
u8g.print(rtc.getTimeStr());
}

void setup(){

pinMode(buttonPin, INPUT);

Serial.begin(9600);
esp8266.begin(19200);
sendCommand("AT",5,"OK");
sendCommand("AT+CWMODE=1",5,"OK");
sendCommand("AT+CWJAP=\"" + AP +"\", \"" + PASS +"\"",20,"OK");

Wire.begin();
Wire.beginTransmission(MPU_addr);
Wire.write(0x6B); // PWR_MGMT_1 register
Wire.write(0); // set to zero (wakes up the MPU-6050)
Wire.endTransmission(true);

pinMode(emled, OUTPUT);

rtc.setDOW(MONDAY); // Set Day-of-Week to FRIDAY
rtc.setTime(10, 17,00); // Set the time to 12:00:00 (24hr
format)
rtc.setDate(03, 06, 2019);
}
void loop(){

dht.read11(dht_dpin);

// read the state of the pushbutton value:
buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed. If it is, the buttonState is
HIGH:
if (buttonState == HIGH) {
Serial.println("Button Pressed");
emrgButton = true;
}

mpu_read();
//2050, 77, 1947 are values for calibration of accelerometer
// values may be different for you
ax = (AcX-1168)/16384.00;
ay = (AcY-2401)/16384.00;
az = (AcZ+1432)/16384.00;

//270, 351, 136 for gyroscope
gx = (GyX)/131.07;

```

```

gy = (GyY)/131.07;
gz = (GyZ)/131.07;

// calculating Amplitude vector for 3 axis
float Raw_AM = pow(pow(ax,2)+pow(ay,2)+pow(az,2),0.5);
int AM = Raw_AM * 10; // as values are within 0 to 1, I multiplied
                       // it by for using if else conditions

Serial.println(AM);
//Serial.println(PM);
//delay(500);

if (trigger3==true){
  trigger3count++;
  //Serial.println(trigger3count);
  if (trigger3count>=10){
    angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5);
    //delay(10);
    Serial.println(angleChange);
    if ((angleChange>=0) && (angleChange<=10)){ //if orientation
changes remains between 0-10 degrees
      fall=true; trigger3=false; trigger3count=0;
      Serial.println(angleChange);
    }
    else{ //user regained normal orientation
      trigger3=false; trigger3count=0;
      Serial.println("TRIGGER 3 DEACTIVATED");
    }
  }
}

if (fall==true){ //in event of a fall detection
  Serial.println("FALL DETECTED");
  fall=false;
  sendFall = true;
}

if (trigger2count>=6){ //allow 0.5s for orientation change
  trigger2=false; trigger2count=0;
  Serial.println("TRIGGER 2 DEACTIVATED");
}

if (trigger1count>=6){ //allow 0.5s for AM to break upper threshold
  trigger1=false; trigger1count=0;
  Serial.println("TRIGGER 1 DEACTIVATED");
}

if (trigger2==true){
  trigger2count++;
  angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5);
Serial.println(angleChange);
  if (angleChange>=30 && angleChange<=400){ //if orientation
changes by between 80-100 degrees
    trigger3=true; trigger2=false; trigger2count=0;
    Serial.println(angleChange);
    Serial.println("TRIGGER 3 ACTIVATED");
  }
}

if (trigger1==true){
  trigger1count++;
  if (AM>=12){ //if AM breaks upper threshold (3g)
    trigger2=true;
    Serial.println("TRIGGER 2 ACTIVATED");
    trigger1=false; trigger1count=0;
  }
}
}

```

```

    if (AM<=2 && trigger2==false){ //if AM breaks lower threshold
(0.4g)
    trigger1=true;
    Serial.println("TRIGGER 1 ACTIVATED");
    }
//It appears that delay is needed in order not to clog the port
delay(100);

if(millis() > (time_now2 + 4000) && millis() < (time_now2 + 4200))
{
    {
    u8g.firstPage();
    do {
    draw();
    } while (u8g.nextPage() );
    // delay(1000);
    }
}
if(millis() > (time_now2 + 8000))
{
    {
    u8g.firstPage();
    do {
    draw2();
    } while (u8g.nextPage() );
    }
    time_now2 = millis();
}

static double oldValue = 0;
static double oldChange = 0;
int rawValue = analogRead (sensor);
double value = alpha * oldValue + (1 - alpha) * rawValue*100/1023;
float value1= value;

if(millis() > (time_now1 + 300)){
digitalWrite(emled,LOW);
time_now1 = millis();}
else if( value1>55) digitalWrite(emled,HIGH);

if(millis() > (time_now + period)){
String getData = "GET /update?api_key="+ API +"&"+ field
+"="+String(value1) + "&field2=" +String(sendFall) + "&field3="
+String(emrgButton)+ "&field4=" +String(dht.temperature)+ "&field5="
+String(dht.humidity);
sendCommand("AT+CIPMUX=1",5,"OK");
sendCommand("AT+CIPSTART=0,\"TCP\", \""+ HOST +"\", "+ PORT,15,"OK");
sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");
esp8266.println(getData);delay(1500);countTrueCommand++;
sendCommand("AT+CIPCLOSE=0",5,"OK");
time_now = millis();
//Epanekinhsh fall kai button katagrafis
sendFall=false;
emrgButton = false;
}
}

void sendCommand(String command, int maxTime, char readReplay[]) {
Serial.print(countTrueCommand);
Serial.print(". at command => ");

```



```

Serial.print(command);
Serial.print(" ");
while(countTimeCommand < (maxTime*1))
{
    esp8266.println(command);//at+cipsend
    if(esp8266.find(readReplay))//ok
    {
        found = true;
        break;
    }

    countTimeCommand++;
}

if(found == true)
{
    Serial.println("OYI");
    countTrueCommand++;
    countTimeCommand = 0;
}

if(found == false)
{
    Serial.println("Fail");
    countTrueCommand = 0;
    countTimeCommand = 0;
}

found = false;
}

void mpu_read(){
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr,14,true); // request a total of 14
registers
    AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C
(ACCEL_XOUT_L)
    AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E
(ACCEL_YOUT_L)
    AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40
(ACCEL_ZOUT_L)
    Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42
(TEMP_OUT_L)
    GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44
(GYRO_XOUT_L)
    GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46
(GYRO_YOUT_L)
    GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48
(GYRO_ZOUT_L)

}

```