

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός και κατασκευή ενσωματωμένου συστήματος συναγερμού στην αρχιτεκτονική Arduino με υποστήριξη δυναμικού ιστοχώρου διαχείρισης

Μερτ Σ. Σοζούμπεκ

Επιβλέποντες καθηγητές:

Μηνάς Δασυγένης

Παντελής Αγγελίδης

Κοζάνη, Ιούλιος 2012

ΠΡΟΛΟΓΟΣ

Στις μέρες μας η ασφάλεια του ιδιωτικού χώρου είναι ένα πολύ σημαντικό ζήτημα. Η αύξηση της εγκληματικότητας και κυρίως στον τομέα των διαρρήξεων, έχει δώσει σημαντική έμφαση στο συνεχή έλεγχο/παρακολούθηση των ιδιωτικών μας χώρων.

Αντικείμενο της εργασίας είναι η σχεδίαση και κατασκευή ενσωματωμένου συστήματος συναγερμού στην αρχιτεκτονική Arduino με υποστήριξη δυναμικού ιστοχώρου διαχείρισης που αλληλεπιδρά με δίκτυα αισθητήρων με σκοπό τη διατήρηση εσωτερικών χώρων σε ασφαλή κατάσταση.

Βασική επιδίωξη του συστήματος είναι η ενσωμάτωση των νέων τεχνολογιών και η επικοινωνιακή χρήση του διαδικτύου στην ασφαλιστική διαδικασία.

Υπεύθυνος κατά την εκπόνηση της διπλωματικής εργασίας ήταν ο Καθηγητής κ. Μηνάς Δασυγένης.

ΠΕΡΙΛΗΨΗ

Τα δίκτυα αισθητήρων χρησιμοποιούνται σήμερα σε ευρεία κλίμακα για την παρακολούθηση χώρων, τον έλεγχο της κυκλοφοριακής κίνησης, τη συλλογή δεδομένων και την αντιμετώπιση εκτάκτων αναγκών, τη παρακολούθηση κλιματικών μεταβολών και αντιμετώπιση των φυσικών κινδύνων κ.α.

Πρόκειται λοιπόν για ένα σύστημα για εσωτερικούς χώρους, κυρίως σπίτια, αποθήκες υλικού, σχολεία, επιχειρήσεις κλπ, που αποστολή έχει τη συνεχή επιτήρηση του χώρου και διατήρηση του σε ασφαλή κατάσταση.

Σε αυτή τη διπλωματική εργασία έχει σχεδιαστεί ένα σύστημα συναγερμού βασισμένο στην αρχιτεκτονική Arduino το οποίο διαχειρίζεται από έναν απομακρυσμένο υπολογιστή μέσω διαδικτύου. Συγκεκριμένα, δημιουργήθηκε ένα σύστημα συναγερμού που αποτελείται από έναν Arduino (Microcontroller), τους απαραίτητους αισθητήρες (αναλόγως τις απαιτήσεις του ιδιοκτήτη) καθώς επίσης, χρησιμοποιήθηκε το Ethernet Shield του Arduino για την πραγματοποίηση του συστήματος σε μία διασύνδεση LAN.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Ενσωματωμένα συστήματα, Arduino, Δίκτυα αισθητήρων, Προγραμματισμός διαδικτύου, Php, MySql, Γλώσσα C, Παθητικό αισθητήριο PIR, Αισθητήρας φαινόμενου Hall

ABSTRACT

Design and construction of an embedded alarm system based on Arduino's architecture which is supported by site management

At the present day, sensor networks are used widely in alarm systems, surveillance systems, vehicular traffic control systems, data collection, detection of climate changes and natural disasters and in similar areas. It is therefore a system that mainly aims to ensure a constant surveillance of the environment and to maintain a safe condition for interior areas especially for houses, warehouses, schools, workplaces etc.

In this thesis, the sensor system has been designed as an alarm system which is based on Arduino architecture and is commanded by a remote computer through internet. In particular, it is a system including an Arduino Ethernet Shield which enables the system to work on a local area network (LAN) and which is consisted of an Arduino microcontroller and the relevant sensors required for the owner's needs.

KEY WORDS

Embedded systems, Arduino, Sensor networks, Web programming, Php, MySql, C language, Hall effect sensor, PIR sensor

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ευκαιρία αυτής της διπλωματικής εργασίας θέλω να εκφράσω ένα μεγάλο ευχαριστώ στον κ. Δημήτρη Δεμιρτζή που είναι καθηγητής του Πανεπιστημίου Άγκυρας που με βοήθησε και με ενέπνευσε να έρθω στην Ελλάδα για τις σπουδές μου. Επίσης, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Μηνά Δασυγένη για την υποστήριξη, την καθοδήγηση και τις συμβουλές που μου προσέφερε καθ' όλη τη διάρκεια αυτής της Διπλωματικής Εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	3
ΠΕΡΙΛΗΨΗ	4
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ	5
ΕΥΧΑΡΙΣΤΙΕΣ	8
ΠΕΡΙΕΧΟΜΕΝΑ	9
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	11
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	13
ΚΕΦΑΛΑΙΟ 1^ο - ΕΙΣΑΓΩΓΗ	14
1.1. Περιγραφή του προβλήματος	15
1.2. Το αντικείμενο της διπλωματικής εργασίας.....	15
1.3. Οργάνωση του κειμένου	16
ΚΕΦΑΛΑΙΟ 2^ο - ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	17
2.1. Συστήματα συναερμού.....	18
2.2. Ενσωματωμένα συστήματα	19
2.3. Δίκτυα αισθητήρων	19
2.4. Προγραμματισμός διαδικτύου	19
2.5. Προγραμματισμός μικροελεγκτή	20
2.6. Τα εργαλεία που χρησιμοποιήθηκαν	21
ΚΕΦΑΛΑΙΟ 3^ο - ΤΟ ΥΛΙΚΟ ΜΕΡΟΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	24
3.1. Γενική αρχιτεκτονική του συστήματος.....	25
3.2. Κόστος υλοποίησης μονάδας	26
3.3. Arduino	26
3.3.1. Μοντέλα Arduino	27
3.3.2. Χαρακτηριστικά Arduino Uno	28
3.3.3. Arduino Shields / Ethernet Shield	34
3.3.4. Η δομή ενός προγράμματος και η γλώσσα	36
3.4. Αισθητήρες	39

3.4.1.	Αισθητήρας φαινόμενου Hall (Hall Effect)	39
3.4.2.	Παθητικό αισθητήριο PIR	41
3.5.	Το κύκλωμα του συστήματος.....	44
ΚΕΦΑΛΑΙΟ 4^ο - ΤΟ ΛΟΓΙΣΜΙΚΟ ΜΕΡΟΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....		46
4.1.	Διαδικτυακό τμήμα του συστήματος	47
4.1.1.	Διαχείριση συστήματος	48
4.1.3.	CSS	60
4.1.4.	Βάση δεδομένων.....	61
4.1.5.	Σύνδεση με τον Arduino	65
4.2.	Το τμήμα μικροελεγκτή	68
4.2.1.	Δομή προγράμματος.....	69
4.2.2.	Βιβλιοθήκες που χρησιμοποιήθηκαν	72
4.3.	Ασφάλεια συστήματος	73
4.3.1.	Ασφάλεια της ιστοσελίδας	73
4.3.2.	Ασφάλεια των συσκευών.	75
ΚΕΦΑΛΑΙΟ 5^ο - ΣΥΜΠΕΡΑΣΜΑ & ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΤΙΩΣΕΙΣ		76
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		79

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

<i>Εικόνα 3.1</i> Διάγραμμα αρχιτεκτονικής του συστήματος.....	25
<i>Εικόνα 3.2</i> Arduino Uno.....	28
<i>Εικόνα 3.3</i> Είσοδοι/Εξοδοι Τροφοδοσίας.....	29
<i>Εικόνα 3.4</i> Ψηφιακοί ακροδέκτες.....	31
<i>Εικόνα 3.5</i> Αναλογικοί ακροδέκτες.....	32
<i>Εικόνα 3.6</i> Διεπαφή χρήστη του Arduino IDE.....	33
<i>Εικόνα 3.7</i> Μερικά από τα Shields.....	34
<i>Εικόνα 3.8</i> Ethernet Shield.....	36
<i>Εικόνα 3.9</i> Βασική δομή ενός προγράμματος.....	37
<i>Εικόνα 3.10</i> Αισθητήρας φαινομένου Hall.....	39
<i>Εικόνα 3.11</i> Παθητικό αισθητήριο PIR.....	41
<i>Εικόνα 3.12</i> Λειτουργία του αισθητήρα PIR.....	42
<i>Εικόνα 3.13</i> Σχηματικό διάγραμμα του κυκλώματος.....	44
<i>Εικόνα 3.14</i> Διάγραμμα Breadboard του κυκλώματος.....	45
<i>Εικόνα 4.1</i> Ρυθμίσεις της ιστοσελίδας του συστήματος.....	47
<i>Εικόνα 4.2</i> Ενέργειες χρηστών/διαχειριστών.....	48
<i>Εικόνα 4.3</i> Αρχική σελίδα/φόρμα εισαγωγής.....	49
<i>Εικόνα 4.4</i> Μήνυμα Log Out.....	50
<i>Εικόνα 4.5</i> Ιστορικά του συστήματος.....	51
<i>Εικόνα 4.6</i> Προσθήκη νέου χρήστη.....	52
<i>Εικόνα 4.7</i> Διαχείριση χρηστών.....	53
<i>Εικόνα 4.8</i> Διαχείριση αισθητήρων.....	54
<i>Εικόνα 4.9</i> Διαχείριση ζωνών.....	55

Εικόνα 4.10 Ρυθμίσεις στοιχείων χρήστη.....	56
Εικόνα 4.11 Διαχείριση αισθητήρων.....	57
Εικόνα 4.12 Διαχείριση ζωνών.....	58
Εικόνα 4.13 Η χρήση της Ajax στο σύστημα.....	59
Εικόνα 4.11 Ο κώδικας της Ajax για την ενημέρωση του ιστορικού.....	60
Εικόνα 4.15 Συσχετίσεις πινάκων.....	64
Εικόνα 4.16 Παράδειγμα του string που λαμβάνεται από τον Arduino.....	65
Εικόνα 4.17 Εμφάνιση μηνύματος στο Serial Monitor.....	66
Εικόνα 4.18 Παράδειγμα του string που αποστέλλεται στην ιστοσελίδα.....	67
Εικόνα 4.19 Παράδειγμα μηνύματος προειδοποίησης στα ιστορικά.....	67
Εικόνα 4.20 Κώδικας του αρχείου post_data.php που λαμβάνει το string.....	68
Εικόνα 4.21 Το γενικό διάγραμμα ροής του προγράμματος.....	69
Εικόνα 4.22 Γενικές ρυθμίσεις του προγράμματος στο Arduino.....	70
Εικόνα 4.23 Κώδικας της ρουτίνας Setup.....	71
Εικόνα 4.24 Η χρήση της εντολής mysql_real_escape_string() στο σύστημα.....	74
Εικόνα 4.25 Συνθηματικό συσκευής που ορίζεται μέσα στις ρυθμίσεις.....	75

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

<i>Πίνακας 3.1</i> Κόστος υλοποίησης μονάδας.....	26
<i>Πίνακας 3.2</i> Χαρακτηριστικά του <i>Arduino Uno</i>	28
<i>Πίνακας 3.3</i> Γενικές εντολές της γλώσσας του <i>Arduino</i>	38
<i>Πίνακας 4.1</i> Πίνακας αισθητήρων.....	63
<i>Πίνακας 4.2</i> Πίνακας τύπων αισθητήρα.....	63
<i>Πίνακας 4.3</i> Πίνακας ζωνών.....	63
<i>Πίνακας 4.4</i> Πίνακας των <i>Events</i>	63
<i>Πίνακας 4.5</i> Πίνακας ιστορικού.....	63
<i>Πίνακας 4.6</i> Πίνακας χρηστών.....	64

ΚΕΦΑΛΑΙΟ 1^ο

ΕΙΣΑΓΩΓΗ

1. Εισαγωγή

1.1. Περιγραφή του προβλήματος

Τα συστήματα συναγερμού είναι πλέον κομμάτι της καθημερινότητας μας. Μέσω αισθητήρων, ειδοποιούν για γεγονότα τα οποία μπορεί να είναι επικίνδυνα για τον χώρο και για την ζωή. Οι συναγερμοί έχουν τη δυνατότητα να ανιχνεύσουν κίνηση, αλλαγή θέσης αντικειμένων, άνοιγμα θυρών, ανίχνευση φωτιάς και ότι άλλο είναι εφικτό και επιθυμητό.

Τα συστήματα συναγερμού έχουν εξελιχθεί σε πολύ μεγάλο βαθμό, καλύπτοντας τις ανάγκες φύλαξης που έχουν προκύψει σε δημόσιους και ιδιωτικούς χώρους. Είναι πλέον απαραίτητη η ενεργοποίηση δικλείδων ασφαλείας που θα προστατεύσουν τις εγκαταστάσεις από πράξεις δολιοφθοράς-διάρρηξης-κλοπής-πυρκαγιάς κ.α. Κάθε σύστημα συναγερμού αποτελείται από διάφορες συσκευές, όπου η κάθε μια εκτελεί ένα συγκεκριμένο σκοπό και συγκεκριμένη εργασία, ανάλογα με τις ανάγκες που απαιτούνται. Κάθε χρήστης του συστήματος είναι ξεχωριστός και έχει ξεχωριστές απαιτήσεις. Αυτό πρέπει να λαμβάνεται υπόψη για να ικανοποιούνται πάντα οι απαιτήσεις του χρήστη όποιες και να είναι αυτές. Στην παρούσα εργασία κατασκευάστηκε ένα σύστημα συναγερμού το οποίο διαχειρίζεται μέσω μιας ιστοσελίδας με τη χρήση των νέων τεχνολογιών και του διαδικτύου.

1.2. Το αντικείμενο της διπλωματικής εργασίας

Το αντικείμενο της εργασίας είναι η σχεδίαση και κατασκευή ενσωματωμένου συστήματος συναγερμού στην αρχιτεκτονική Arduino με υποστήριξη δυναμικού ιστοχώρου διαχείρισης, που αλληλεπιδρά με δίκτυα αισθητήρων, με σκοπό τη διατήρηση εσωτερικών χώρων σε ασφαλή κατάσταση.

Οι χρήστες και οι διαχειριστές έχουν τη δυνατότητα να σχεδιάσουν το σύστημα ανάλογα με τις απαιτήσεις τους και να επεξεργάζονται τα δεδομένα που συλλέγονται από τους αισθητήρες και αποστέλλονται στη βάση, ώστε να αντιμετωπίζουν εγκαίρως τις έκτακτες ανάγκες που στην συγκεκριμένη περίπτωση μας είναι η πιθανότητα πυρκαγιάς,

πλημμύρας καθώς επίσης, και η μη εξουσιοδοτημένη πρόσβαση στο χώρο στον οποίο αναφερόμαστε.

1.3. Οργάνωση του κειμένου

Το κείμενο αρχίζει με την εισαγωγή στις έννοιες των ενσωματωμένων συστημάτων, συστημάτων συναγερού, δικτύων αισθητήρων και του προγραμματισμού διαδικτύου, συνεχίζει με την παρουσίαση των εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.

Στη συνέχεια στο 3^ο κεφάλαιο παρουσιάζεται το υλικό μέρος του συστήματος. Σ' αυτό το κεφάλαιο αναφέρεται το κάθε υλικό που χρησιμοποιήθηκαν. Εξηγείται αναλυτικά η κατασκευή, η λειτουργία και η χρήση τους.

Στο 4^ο κεφάλαιο γίνεται παρουσίαση του λογισμικού μέρους του συστήματος.

Στο 5^ο κεφάλαιο παρουσιάζονται τα συμπεράσματα καθώς και μελλοντικές βελτιώσεις του συστήματος.

ΚΕΦΑΛΑΙΟ 2^ο

ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2. Θεωρητικό υπόβαθρο

2.1. Συστήματα συναγερμού

Τα συστήματα συναγερμού [11] αποτελούν διεθνώς την πιο συμφέρουσα και αποτελεσματική λύση για την αύξηση της ασφάλειας ενός χώρου. Σκοπός ενός συστήματος συναγερμού, είναι η προστασία κτιρίων ή άλλων χώρων από παράνομες εισβολές. Η σωστή κατασκευή, αλλά και ο σχεδιασμός από πλευράς του ιδιώτη ή της εταιρίας, που θα εγκαταστήσει και θα αναλάβει την παρακολούθηση του συστήματος, είναι οι βασικές παράμετροι αποτελεσματικότητάς του.

Τόσο στα μεγάλα αστικά κέντρα, όσο και σε μικρότερους οικισμούς, στην επαρχία ή και στα περίχωρα των πόλεων, έχει αρχίσει να διεισδύει ανάμεσα στους κατοίκους, το αίσθημα της ανασφάλειας, τόσο για τη σωματική ακεραιότητά τους όσο και για τη διαφύλαξη της υλικής περιουσίας τους. Η προστασία ενός χώρου απέναντι σε προσπάθειες διάρρηξης, δεν είναι απλή υπόθεση. Ο βαθμός αποτελεσματικότητας αυξάνεται όταν υπάρχει μια καλά σχεδιασμένη μελέτη, που θα πρέπει να συνυπολογίζει όσο το δυνατό μεγαλύτερο αριθμό ενδεχομένων και να διαθέτει ένα συνδυασμό μέτρων ασφάλειας, ούτως ώστε να υπάρχει μια δικλείδα προστασίας, στην περίπτωση που υπάρξει αστοχία ενός εκ των συστημάτων. Η ύπαρξη συστημάτων ασφαλείας, καταρχήν, λειτουργεί αποτρεπτικά. Δηλαδή, είναι σίγουρο ότι ένας εντελώς απροστάτευτος χώρος αποτελεί μαγνήτη για ένα διαρρήκτη, σε αντίθεση με εκείνον, που διαθέτει ένα σύστημα συναγερμού και το οποίο μόνο με την ύπαρξή του θα προβληματίσει τον υποψήφιο διαρρήκτη.

Συνοπτικά τα τρία κριτήρια που χρειάζεται να ελέγχει ο ενδιαφερόμενος πρίν την αγορά και εγκατάσταση ενός συστήματος συναγερμού είναι η αξιοπιστία των συσκευών από τις οποίες απαρτίζεται, η συνεχής τεχνική υποστήριξη, δηλαδή η συντήρηση και ο έλεγχος της καλής λειτουργίας των συσκευών και τέλος η δυνατότητα παροχής 24ωρης παρακολούθησης από κάποιο κέντρο λήψης των σημάτων του συναγερμού.

2.2. Ενσωματωμένα συστήματα

Ενσωματωμένα Συστήματα [5] είναι αυτά στα οποία κάποιος επεξεργαστής λειτουργεί σαν μέρος μίας ολότητας, επιτελώντας συγκεκριμένο έργο, και στον οποίο εν γένει ο χρήστης δεν έχει πρόσβαση για να αλλάξει το πρόγραμμα ή τη λειτουργικότητα του συστήματος. Για όλο το μάθημα, με τον όρο «ενσωματωμένα συστήματα» θα εννοούμε την ενσωμάτωση κάποιου μικροεπεξεργαστή στην λειτουργία ενός ολόκληρου συστήματος με ηλεκτρονικά, μηχανολογικά, και άλλα μέρη, αλλά θα μας αφορούν μόνο οι λειτουργίες του μικροεπεξεργαστή και η διεπαφή αυτού με το εξωτερικό περιβάλλον. Στην πράξη προκύπτει ότι κάποια από τα παραπάνω χαρακτηριστικά κατά περίπτωση και κατ' εξαίρεση μπορεί να μην ισχύουν, ο ορισμός όμως παραμένει χρήσιμος για να διαφοροποιήσει τα ενσωματωμένα συστήματα από τα γενικής χρήσης.

2.3. Δίκτυα αισθητήρων

Ένα δίκτυο αισθητήρων [3], [8] αποτελείται από γεωγραφικά κατανεμημένους αυτόνομους αισθητήρες οι οποίοι μπορούν να μετρήσουν φυσικά μεγέθη όπως θερμοκρασία, ήχος, πίεση, κίνηση, κλπ. Η αυτονομία ενός αισθητήρα βασίζεται στο κύκλωμα στο οποίο είναι ενσωματωμένος ή απλά συνδεδεμένος. Το σύστημα αυτό περιλαμβάνει ένα μικροελεγκτή ή ένα μικροεπεξεργαστή καθώς και άλλα συμπληρωματικά κυκλώματα που υποστηρίζουν τη διαχείριση ενέργειας και τις επικοινωνίες.

Τα δίκτυα αισθητήρων χρησιμοποιούνται σήμερα σε ευρεία κλίμακα για την παρακολούθηση χώρων, τον έλεγχο της κυκλοφοριακής κίνησης, τη συλλογή δεδομένων και την αντιμετώπιση εκτάκτων αναγκών, τη παρακολούθηση κλιματικών μεταβολών και αντιμετώπιση των φυσικών κινδύνων, στο σύστημα υγείας με καινοτόμες διαγνωστικές, απεικονιστικές και θεραπευτικές προσεγγίσεις κ.α.

2.4. Προγραμματισμός διαδικτύου

Η PHP (Hypertext PreProcessor) [15], είναι μια γλώσσα προγραμματισμού σεναρίων που χρησιμοποιείται ευρέως και η οποία μπορεί να ενσωματωθεί στον κώδικα της HTML. Ενσωματώνοντας μικρά προγράμματα σεναρίων σε έναν HTML κώδικα,

αποκτούμε τη δυνατότητα να ελέγχουμε τι θα εμφανίζεται στην οθόνη του φυλλομετρητή με τρόπο πολύ πιο ευέλικτο από ότι με τη χρήση της HTML. Η γλώσσα αυτή εκτελείται στην πλευρά του εξυπηρετητή, κάτι που την κάνει να ξεχωρίζει από άλλες γλώσσες όπως η JavaScript, και κάτι που σημαίνει ότι ο εξυπηρετητής Ιστού τη διερμηνεύει προτού ακόμα τη στείλει στο φυλλομετρητή. Σχεδιάστηκε με σκοπό την ανάπτυξη διαδικτυακών εφαρμογών και τη δημιουργία δυναμικού περιεχομένου ιστοσελίδων.

Το πιο σημαντικό κομμάτι της, όμως είναι το γεγονός ότι υποστηρίζει ένα ευρύ σύνολο από βάσεις δεδομένων, κάτι που κάνει πολύ πιο εύκολη και απλή τη δημιουργία μιας ιστοσελίδας η οποία επικοινωνεί με βάσεις δεδομένων.

Στο παρόν σύστημα, η PHP χρησιμοποιήθηκε για την επικοινωνία μεταξύ της βάσης δεδομένων και της ιστοσελίδας καθώς επίσης, για την υλοποίηση των περισσότερων λειτουργιών της ιστοσελίδας.

2.5. Προγραμματισμός μικροελεγκτή

Ο μικροελεγκτής είναι ένα αυτόνομο υπολογιστικό σύστημα, με πολύ μικρό μέγεθος, σε ένα και μοναδικό ολοκληρωμένο κύκλωμα (computer on a chip). Όπως και όλα τα VLSI κυκλώματα, αποτελείται από μέρη που κατασκευάζονται με διάφορες λιθογραφικές μεθόδους πάνω σε πλάκες πυριτίου, τα λεγόμενα Silicon Wafers. Πάνω σε αυτά σχηματίζονται χιλιάδες έως εκατομμύρια τρανζίστορ και κατ' επέκταση δημιουργούνται τα λεγόμενα ολοκληρωμένα κυκλώματα που είναι συνδυασμός λογικών πυλών. Συνδυάζοντας τις λογικές πύλες, δημιουργούνται υπομονάδες που επιτελούν ορισμένες πιο εξειδικευμένες λειτουργίες στο μικροελεγκτή. Μια κύρια διαφορά μεταξύ ενός μικροελεγκτή και ενός μικροεπεξεργαστή είναι πως στο μικροελεγκτή υπάρχει ενσωματωμένη μνήμη και μονάδες ελέγχου περιφερειακών συσκευών. Οι μικροελεγκτές έχουν ως κύριο σκοπό τους να επικοινωνούν με άλλες ηλεκτρονικές διατάξεις και όχι με τον άνθρωπο.

Ο προγραμματισμός [5] αυτών των συσκευών γίνεται με ένα σετ εντολών που έχουν συγκεκριμένη σύνταξη και χρησιμοποιούν δύο καταστάσεις το 1 και το 0. Οι μικροελεγκτές αρχικά, προγραμματίζονταν μόνο σε γλώσσα assembly, αλλά πλέον διάφορες γλώσσες υψηλού επιπέδου χρησιμοποιούνται για αυτό το σκοπό. Αυτές οι

γλώσσες προγραμματισμού είναι είτε αποκλειστικά στοχευμένες για τον προγραμματισμό μικροελεγκτών, ή απλά εκδόσεις τους που μας εξυπηρετούν, όπως η γλώσσα C. Οι εντολές της γλώσσας μηχανής επιτελούν περιορισμένες διεργασίες και πολλές φορές χρειάζεται εκατοντάδες εντολών για να εκτελέσουν πράξεις και λειτουργίες πιο σύνθετες από τις συνηθισμένες. Υπάρχει, όμως η δυνατότητα να γράψουμε το πρόγραμμα σε γλώσσα προγραμματισμού υψηλού επιπέδου. Οι γλώσσες υψηλού επιπέδου μπορούν να μεταφραστούν σε γλώσσα χαμηλού επιπέδου με τη βοήθεια μεταφραστικών εργαλείων, όπως είναι ο compiler (μεταγλωττιστής) και ο interpreter (διερμηνέας). Το πρώτο είναι εργαλείο λογισμικού, το οποίο δέχεται τις εντολές υψηλού επιπέδου που δίνει ο χρήστης και τις μετατρέπει σε γλώσσα μηχανής.

2.6. Τα εργαλεία που χρησιμοποιήθηκαν

Όλα τα προγράμματα που χρησιμοποιήθηκαν για τους σκοπούς της παρούσας εργασίας είναι ανοιχτού ή ελεύθερου λογισμικού. Τα κύρια εργαλεία που χρησιμοποιήθηκαν κατά την εκπόνηση της διπλωματικής εργασίας είναι τα εξής:

- **Xampp**

Το **XAMPP** [14] είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξάρτητου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache, την βάση ιστοσελίδων MySQL και ένα διερμηνέα για σενάρια γραμμένα σε γλώσσες προγραμματισμού PHP και Perl.

Το XAMPP είναι ακρωνύμιο και αναφέρεται στα παρακάτω αρχικά:

- X (αναφέρεται στο "cross-platform" που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας)
- Apache HTTP εξυπηρετητής
- MySQL
- PHP
- Perl

Το XAMPP είναι ένα 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.

- **Apache Web Server**

Είναι ένας εξυπηρετητής Ιστού. Ο Apache [14], αποτελεί ένα ανοικτού κώδικα λογισμικό, το οποίο υποστηρίζει ένα σύνολο από διάφορα λειτουργικά συστήματα, όπως Linux, Unix, Windows κ.α. Είναι γραμμένος στη γλώσσα C και αποτελεί το πιο διάσημο HTTP λογισμικό εξυπηρετητή. Το λογισμικό αυτό υποστηρίζει διάφορες εφαρμογές, όπως είναι γλώσσες προγραμματισμού της πλευράς του εξυπηρετητή, όπως η PHP. Υποστηρίζει Perl, Python, Tcl, καθώς επίσης, και SSL και TLS. Χρησιμοποιείται για να τρέχουν σε αυτόν τόσο στατικές, όσο και δυναμικές ιστοσελίδες. Ο APACHE, χρησιμοποιείται πολλές φορές σε περιπτώσεις που είναι αναγκαίο διάφορες πληροφορίες να γίνουν προσβάσιμες με ένα ασφαλή και αξιόπιστο τρόπο, όπως για παράδειγμα το διαμοίρασμα αρχείων από έναν ηλεκτρονικό υπολογιστή στο διαδίκτυο.

Επιπρόσθετα, χρησιμοποιείται από πολλούς προγραμματιστές διαδικτυακών εφαρμογών για να ελέγχουν, τοπικά στον ηλεκτρονικό τους υπολογιστή, τον κώδικά τους. Αυτός είναι ένας από τους λόγους που χρησιμοποιήθηκε το λογισμικό αυτό στην παρούσα διπλωματική εργασία. Ένας ακόμη λόγος, είναι γιατί υποστηρίζει τη γλώσσα προγραμματισμού σεναρίων PHP, η οποία χρησιμοποιήθηκε για την υλοποίηση των λειτουργιών της ιστοσελίδας.

- **MySql Server**

Η MySql [17] είναι ένα ελεύθερο σύστημα διαχείρισεως βάσεων δεδομένων που είναι ευρύτατα διαδεδομένη στα UNIX συστήματα, αν και τα τελευταία χρόνια δείχνει να κατακτά και τον κόσμο των Win32.

Επίσης, η MySql είναι πολυνηματική και πολυχρηστική και υποστηρίζει τα τελευταία standards της SQL.

Η ανάγκη για διασύνδεση του συστήματος με μια βάση δεδομένων ώθησε στην επιλογή της εφαρμογής MySql για τη δημιουργία της βάσης δεδομένων που χρειάστηκε κατά την ανάπτυξη του διαδικτυακού τμήματος του συστήματος.

- **Arduino IDE**

Για τη διαχείριση του Arduino από τον υπολογιστή χρησιμοποιήθηκε το Arduino IDE [21]. Η λειτουργία του αναφέρονται πιο αναλυτικά στο Κεφάλαιο 3.

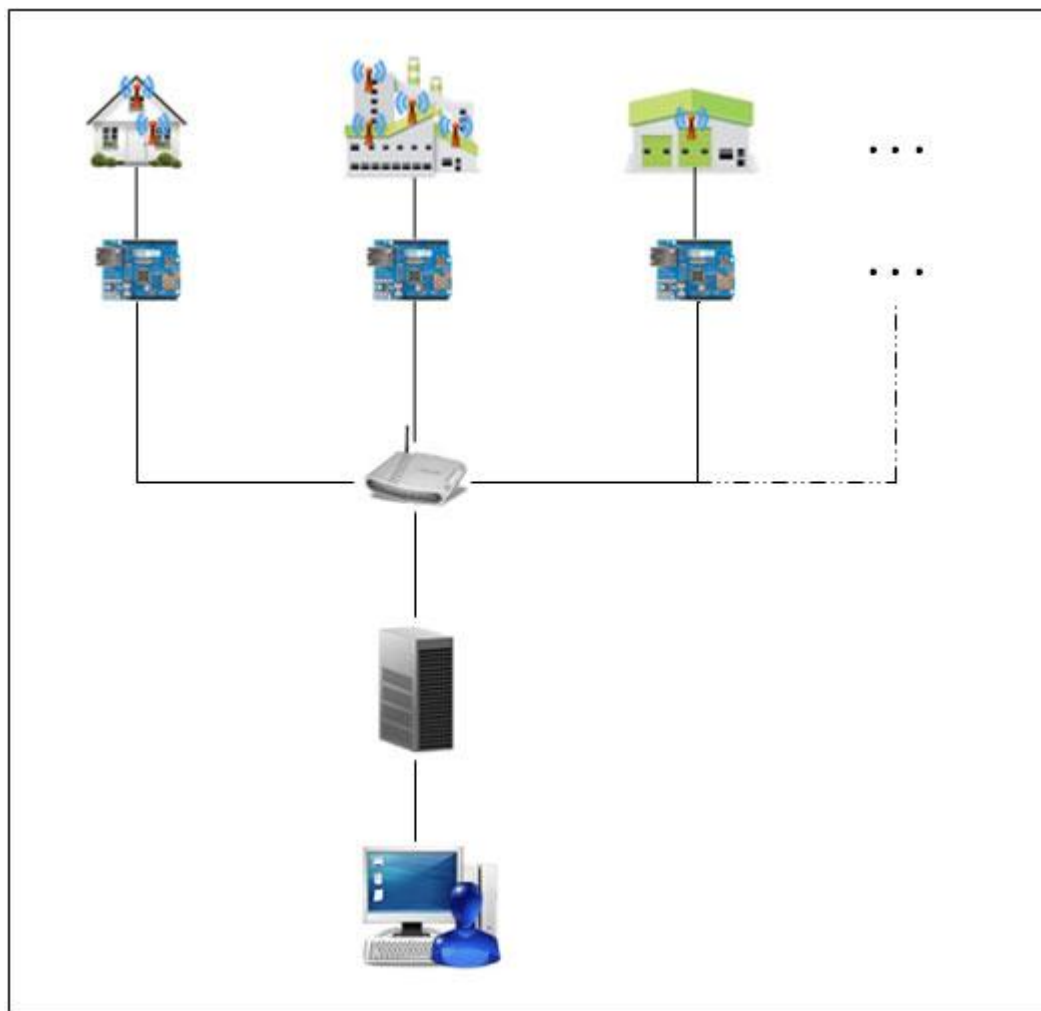
ΚΕΦΑΛΑΙΟ 3^ο

ΤΟ ΥΛΙΚΟ ΜΕΡΟΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

3. Το υλικό μέρος του συστήματος

3.1. Γενική αρχιτεκτονική του συστήματος

Το σύστημα συναγερμού αποτελείται από 1 έως 9 μικροελεγκτές Arduino. Αυτό σημαίνει ότι μπορούμε να τοποθετήσουμε από έναν Arduino μέχρι και σε 9 διαφορετικές τοποθεσίες. Ο κάθε μικροελεγκτής μας δίνει τη δυνατότητα να συνδέσουμε μέχρι και 8 αισθητήρες (αναλόγως με τις απαιτήσεις του ιδιοκτήτη του χώρου). Οι αισθητήρες αυτοί μόλις τοποθετηθούν στο σύστημα πρέπει να οριστούν και στην ιστοσελίδα, για να λειτουργήσουν. Όλο το σύστημα διαχειρίζεται/ελέγχεται μέσω της ιστοσελίδας διαχείρισης.



Εικόνα 3.1 Διάγραμμα αρχιτεκτονικής του συστήματος

3.2. Κόστος υλοποίησης μονάδας

Το κόστος υλοποίησης του συγκεκριμένου συστήματος συναγερμού είναι σχετικά χαμηλό σε σχέση με τα άλλα συστήματα ασφάλειας. Το προτέρημα του οφείλεται κυρίως στο ότι όλο το σύστημα είναι βασισμένο στο διαδίκτυο χωρίς κάποια απαραίτητη κεντρική μονάδα. Αυτό μας προσφέρει ένα βιώσιμο και εναλλακτικό μέσο υλοποίησης. Παρακάτω παρουσιάζονται οι προσωρινές τιμές των υλικών:

ΥΛΙΚΟ	ΚΟΣΤΟΣ
Arduino Uno	25 €
Ethernet Shield	50€
Αισθητήρας φαινόμενου Hall	2€
Παθητικό αισθητήριο PIR	7€
2xLed	0,20€
Σύνολο:	84,20€

Πίνακας 3.2 Κόστος υλοποίησης μονάδας

3.3. Arduino

Η πλατφόρμα στην οποία είναι βασισμένο το όλο σύστημα ονομάζεται Arduino [20] και πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στο μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για τη λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα. Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει το μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου.

Το Arduino μπορεί να αισθανθεί το περιβάλλον προσλαμβάνοντας δεδομένα από μεγάλη ποικιλία αισθητήρων και μπορεί να επηρεάσει το χώρο με τον έλεγχο των φώτων, των μηχανών και άλλων ηλεκτρονικών επεξεργαστών. Ο μικροεπεξεργαστής στον πίνακα της πλατφόρμας είναι προγραμματισμένος στη χρήση της γλώσσας προγραμματισμού του Arduino και στο περιβάλλον ανάπτυξης Arduino. Τα σχέδια Arduino μπορούν να είναι αυτόνομα ή μπορούν να επικοινωνήσουν με οποιοδήποτε λογισμικό ενός υπολογιστή.

3.3.1. Μοντέλα Arduino

Στην αγορά διατίθενται πολλά μοντέλα της πλατφόρμας και στην παρούσα εργασία χρησιμοποιήθηκε το Arduino Uno. Τα μερικά από τα μοντέλα Arduino αναφέρονται παρακάτω:

- Arduino Diecimila
- Arduino Nuova Generazione
- Arduino Extreme
- Arduino Usb
- Arduino Lilypad
- Arduino Nano
- Arduino Mini
- Arduino Duemilanove
- Arduino Uno
- Arduino Mega

3.3.2. Χαρακτηριστικά Arduino Uno



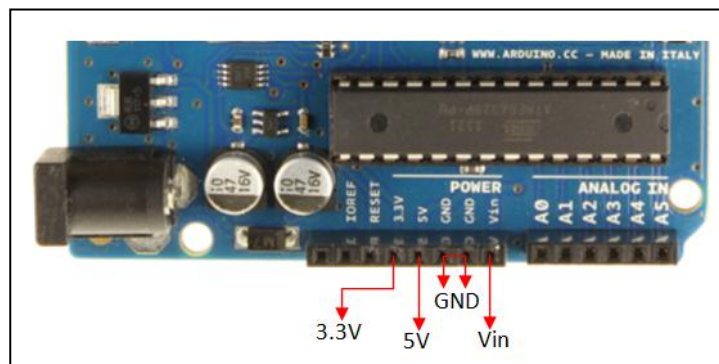
Εικόνα 3.2 Arduino Uno

Μικροελεγκτής	ATMEGA328
Τάση λειτουργίας	5V
Τάση εισόδου(συνιστάται)	7-12V
Όρια τάσης εισόδου	6-20V
Ψηφιακοί ακροδέκτες I/O	14(6 εκ των οποίων PWM έξοδο)
Αναλογικοί ακροδέκτες εισόδου	6
Ισχύς συνεχόμενου ρεύματος ανά ακροδέκτη	40mA
Ισχύς συνεχόμενου ρεύματος για ακροδέκτη τάσης 3.3V	50mA
Μνήμη flash	32KB(ATMEGA328)
Μνήμη SRAM	2KB(ATMEGA328)
Μνήμη EEPROM	1KB(ATMEGA328)
Ταχύτητα ρολογιού	16MHz

Πίνακας 3.2 Χαρακτηριστικά του Arduino Uno

- **Τροφοδοσία**

Το Arduino Uno μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φισ των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω- αριστερή γωνία του Arduino. Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC.



Εικόνα 3.3 Είσοδοι/Εξοδοι Τροφοδοσίας

Οι ακροδέκτες τροφοδοσίας είναι οι ακόλουθοι:

V_{in} : Η τάση εισόδου της πλακέτας όταν χρησιμοποιεί εξωτερική πηγή ενέργειας. Η τροφοδοσία τάσης γίνεται μέσω αυτού του ακροδέκτη. Αν τροφοδοτηθεί με τάση από τον ακροδέκτη τροφοδοσίας (jack), έχει πρόσβαση σε αυτή μέσω του ακροδέκτη V_{in} .

5V: Η τάση που χρησιμοποιείται από τα διάφορα μέρη της πλακέτας και το μικροελεγκτή είναι 5V. Η τάση αυτή, την οποία δίνει αυτός ο ακροδέκτης, είναι είτε η τάση 5V που δίνει η σύνδεση με USB, είτε η ρυθμισμένη τάση που δίνεται μέσω του V_{in} .

3.3V: Η τάση αυτή παράγεται από το ολοκληρωμένο FTDI. Το όριο άντλησης ρεύματος είναι 50mA.

GND: Είσοδοι εδάφους.

- **Μνήμη**

Ο μικροεπεξεργαστής ATmega328 έχει τρεις ομάδες μνήμης. Διαθέτει flash memory, στην οποία αποθηκεύονται τα Arduino sketch, SRAM (static random access memory), στην οποία δημιουργείται το sketch και χρησιμοποιεί τις μεταβλητές όταν τρέχει, και EEPROM, η οποία χρησιμοποιείται από τους προγραμματιστές για την αποθήκευση μακροχρόνιων πληροφοριών.

- **2KB μνήμης SRAM** που είναι η ωφέλιμη μνήμη που μπορούν να Χρησιμοποιήσουν τα προγράμματά για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.

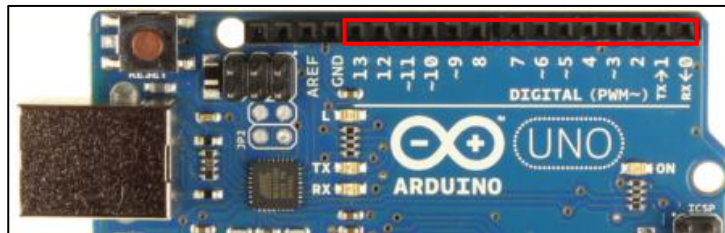
- **1KB μνήμης EEPROM** η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματα κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δε χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου.

- **32KB μνήμης Flash**, από τα οποία τα 2KB χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση προγραμμάτων στο μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 30KB της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή. Η μνήμη Flash, όπως και η EEPROM δε χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για χρήση runtime μέσα από τα προγράμματά, λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2KB SRAM + 1KB EEPROM), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει τη χρήση όσου χώρου περισσεύει (30KB μείον το μέγεθος του προγράμματός σας σε μεταγλωττισμένη μορφή).

- **Είσοδοι/Εξοδοι**

Το Arduino διαθέτει σειριακό interface. Ο μικροελεγκτής ATmega υποστηρίζει σειριακή επικοινωνία, την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για τη μεταφορά των προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο Arduino, αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται. Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 ψηφιακοί ακροδέκτες, αριθμημένα από 0 ως 13. Κάθε ένας από τους 14 ψηφιακούς ακροδέκτες στο Arduino Uno μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος.

Όλοι αυτοί οι ακροδέκτες λειτουργούν σε 5 βολτ. Κάθε είσοδος μπορεί να παρέχει ή να λάβει ένα μέγιστο 40 mA και έχει έναν εσωτερικό pull-up αντίσταση των 20-50 kΩ. Ως ψηφιακή έξοδος, ένα από αυτούς του ακροδέκτες μπορεί να τεθεί από το πρόγραμμά σας σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο ακροδέκτη. Με αυτόν τον τρόπο ανάβει και να σβήνει ένα LED που είναι συνδεδεμένο στο συγκεκριμένο ακροδέκτη. Αν πάλι ρυθμιστεί ένα από αυτούς του ακροδέκτες ως ψηφιακή είσοδος μέσα από το πρόγραμμά, μπορεί με την κατάλληλη εντολή να διαβάσει την κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχει συνδεθεί σε αυτόν τον ακροδέκτη διοχετεύει ή όχι ρεύμα στον ακροδέκτη. Μερικά από αυτούς τους 14 ακροδέκτες, εκτός από ψηφιακές είσοδοι/έξοδοι έχουν και δεύτερη λειτουργία. Συγκεκριμένα:

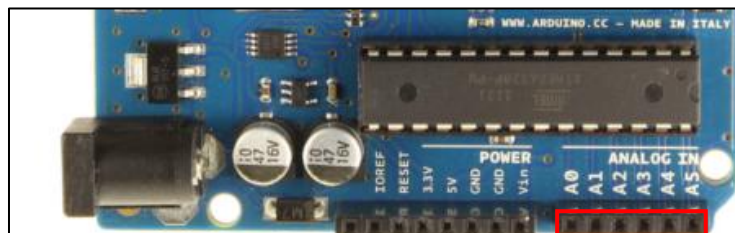


Εικόνα 3.4 Ψηφιακοί ακροδέκτες

- **Οι ακροδέκτες 0 και 1** λειτουργούν ως RX και TX της σειριακής θύρας όταν το πρόγραμμα ενεργοποιεί τη σειριακή θύρα. Έτσι, όταν το πρόγραμμα στέλνει δεδομένα στη σειριακή θύρα, αυτά προωθούνται και στη θύρα USB μέσω του ελεγκτή Serial-Over-USB, αλλά και στον ακροδέκτη 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή. Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμα ενεργοποιήσει το σειριακό interface, χάνει 2 ψηφιακές εισόδους/εξόδους η πλατφόρμα.

- **Οι ακροδέκτες 2 και 3** λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Ρυθμίζονται μέσα από το πρόγραμμα ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει άμεσα και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

- **Οι ακροδέκτες 3, 5, 6, 9, 10 και 11** μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation), το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων.



Εικόνα 3.5 Αναλογικοί ακροδέκτες

- Στην κάτω πλευρά του Arduino, με τη σήμανση **ANALOG IN**, υπάρχει μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με τη σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτηθεί ο ακροδέκτης AREF με 3.3V και στη συνέχεια διαβάσει

κάποιον ακροδέκτη αναλογικής εισόδου στο οποίο εφαρμόζεται τάση 1.65V, το Arduino θα επιστρέψει την τιμή 512.

- **Διασύνδεση**



Εικόνα 3.6 Διεπαφή χρήστη του *Arduino IDE*

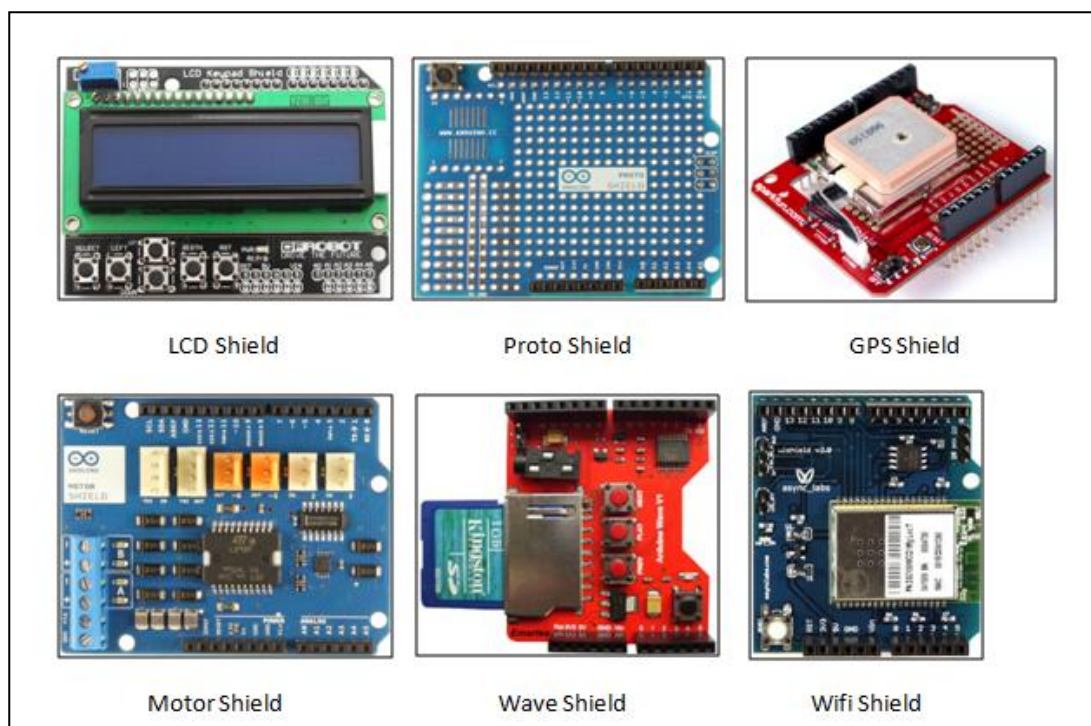
Το Arduino Uno έχει τη δυνατότητα επικοινωνίας με υπολογιστή, άλλες ίδιες πλακέτες ή άλλους μικροελεγκτές. Παρέχεται η δυνατότητα σειριακής επικοινωνίας μέσω USB θύρας και τους FTDI οδηγούς (drivers). Το λογισμικό που παρέχεται μας επιτρέπει να στέλνουμε εντολές ή δεδομένα γενικά από και προς την πλακέτα και να φορτώσουμε το πρόγραμμα στον bootloader, το οποίο θα εκτελεί αυτόματα η πλακέτα. Για τη διαχείριση του Arduino από τον υπολογιστή χρησιμοποιείται το Arduino IDE. Το Arduino IDE είναι βασισμένο σε Java και συγκεκριμένα παρέχει:

- ένα πρακτικό περιβάλλον για τη συγγραφή των προγραμμάτων (τα οποία ονομάζονται sketch στην ορολογία του Arduino) με συντακτική χρωματική σήμανση,

- μερικές έτοιμες βιβλιοθήκες για προέκταση της,
- τον compiler για τη μεταγλώττιση των sketch,
- ένα serial monitor που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για το debugging των sketch
- και την επιλογή για ανέβασμα των μεταγλωττισμένων sketch στο Arduino.

3.3.3. Arduino Shields / Ethernet Shield

Τα Shield είναι ολοκληρωμένες πλακέτες που είναι σχεδιασμένες ώστε να κουμπώνουν πάνω στο Arduino προεκτείνοντας τη λειτουργικότητά του. Είναι η hardware αντίστοιχη έννοια των plugin, addon και extension που υπάρχουν στο software. Το Arduino έχει πάρα πολλά Shields και ανάλογα με τη δουλειά που θέλουμε να κάνουμε μπορούμε να φτιάξουμε και το δικό μας.

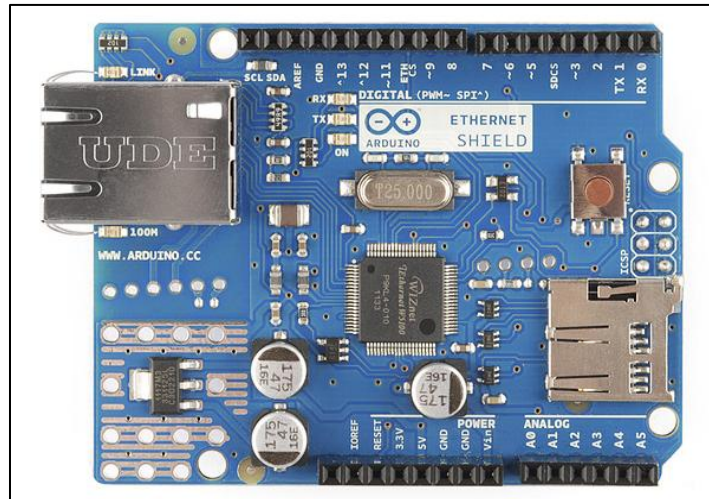


Εικόνα 3.7 Μερικά από τα Shields

Τα Shield είναι σχεδιασμένα ώστε αφού κουμπωθούν πάνω στο Arduino να προωθούν τις υποδοχές του, ώστε να μπορείτε να συνδέσετε επιπλέον τα δικά σας εξαρτήματα ή να κουμπώσετε και επόμενο Shield. Φυσικά, το κάθε Shield χρησιμοποιεί

ορισμένους από τους πόρους συνδεσιμότητας του Arduino και έτσι δεν μπορείτε να συνδέσετε απεριόριστα Shield. Μάλιστα κάποια Shield μπορεί να μην είναι συμβατά μεταξύ τους γιατί χρησιμοποιούν τα ίδια pin του Arduino για επικοινωνία με αυτό. Επίσης, επειδή κάποια Shield δεν προωθούν τις συνδέσεις του Arduino (όπως π.χ. οι οθόνες οι οποίες δεν έχουν νόημα αν τις καλύψετε από πάνω με ένα επόμενο Shield), υπάρχουν ειδικά extender Shield που κουμπώνουν στο Arduino και δίνουν τη δυνατότητα σε δύο άλλα Shield να κουμπώσουν πάνω τους, λειτουργώντας σαν πολύπριζα. Όπως και για το ίδιο το Arduino, το βασικό πλεονέκτημα των Shield δεν είναι τόσο το προφανές πλεονέκτημα του έτοιμου hardware όσο ότι συνοδεύονται συνήθως από έτοιμες βιβλιοθήκες που σας επιτρέπουν να προγραμματίζετε τα sketch σας σε high level. Μερικά από τα πιο δημοφιλή Shield που κυκλοφορούν στο εμπόριο για το Arduino είναι:

- Ethernet Shield: Δίνει στο Arduino τη δυνατότητα να δικτυωθεί σε ένα LAN ή στο internet μέσω ενός τυπικού καλωδίου Ethernet.
- WiFi Shield: Όμοιο με το Ethernet Shield, χωρίς φυσικά το καλώδιο.
- Διάφορα Shield οθόνης: Προσθέτουν οθόνη στο Arduino. Κυκλοφορούν από απλές οθόνες τύπου calculator μέχρι OLED touchscreen υψηλής ανάλυσης τύπου iPhone.
- Wave Shield: Δίνει στο Arduino τη δυνατότητα να παίζει ήχους/μουσική από κάρτες SD.
- GPS Shield: Προσθέτει GPS δυνατότητες στο Arduino (εντοπισμό στίγματος).
- Διάφορα Motor Shields: Σας επιτρέπουν να οδηγήσετε εύκολα μοτέρ διάφορων τύπων (απλά DC, servo, stepper κ.λπ.) από το Arduino.
- ProtoShield: Μια προσχεδιασμένη πλακέτα πρωτοτυποποίησης, συμβατή στις διαστάσεις του Arduino και χωρίς εξαρτήματα για να φτιάξετε το δικό σας Shield.



Εικόνα 3.8 Ethernet Shield

Το Ethernet Shield, όμως είναι ίσως από τα πιο χρήσιμα αφού προσφέρει έναν επιπλέον τρόπο για επικοινωνία με το Arduino και αυτός είναι μέσω του δικτύου. Το Arduino Ethernet Shield επιτρέπει σε μια πλακέτα Arduino να συνδεθεί στο διαδίκτυο χρησιμοποιώντας τη βιβλιοθήκη Ethernet.

3.3.4. Η δομή ενός προγράμματος και η γλώσσα

- **Sketch**

Τα προγράμματα που γράφονται στο IDE του Arduino ονομάζονται sketch και τους δίνεται αυτόματα όνομα βάση της ημερομηνίας που ξεκίνησαν. Τα αρχεία αυτά έχουν επέκταση INO.

- **Βασική δομή**

Τα προγράμματα του Arduino διαιρούνται σε τρία μέρη: δομή (structure), τιμές (values) και συναρτήσεις (functions).

```
<δήλωση μεταβλητών>
void setup()
{
  <δήλωση λειτουργιών που ισχύουν για όλο το πρόγραμμα>
}
void loop()
{
  <δήλωση λειτουργιών που επαναλαμβάνονται κατά τη λειτουργία
του προγράμματος>
}
```

Εικόνα 3.9 Βασική δομή ενός προγράμματος

- **Σχόλια**

Τα σχόλια μιας γραμμής γράφονται μετά από διπλές καθέτους '//'. (//σχόλιο)

Σχόλια περισσότερων γραμμών γράφονται εντός '/'* και '*/'. (/* σχόλιο */)

Το IDE αναγνωρίζει τα σχόλια και τα χρωματίζει γκριζα, για να δείξει ότι δεν είναι ενεργά στο πρόγραμμα ή στο debugging.

- **Γλώσσα**

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring [19] μια παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Λόγω της καταγωγής της από τη C [6], στη γλώσσα του Arduino [9], [4] μπορούν να χρησιμοποιηθούν ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στη C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για τη διαχείριση του ειδικού hardware του Arduino. Τα προγράμματα του Arduino διαιρούνται σε τρία μέρη: δομή (structure), τιμές (values) και συναρτήσεις (functions).

Οι πιο σημαντικές εντολές ανά μέρη επεξηγούνται στον πίνακα που ακολουθεί:

ΔΟΜΗ	ΤΙΜΕΣ	ΣΥΝΑΡΤΗΣΕΙΣ
setup()	Σταθερές	Ψηφιακές I/O
loop()	HIGH – LOW	pinMode()
Έλεγχος	INPUT – OUTPUT	digitalWire()
if	true - false	digitalRead()
if..else	integer constants	Αναλογικές I/O
for	floating point constants	analogReference()
switch case	Τύποι δεδομένων	analogRead()
while	void	analogWrite()
do...while	boolean	Ειδικές I/O
break	(unsigned)char	tone(), noTone()
continue	byte	shiftOut()
return	(unsigned)int	pulseIn()
goto	word	Χρόνος
Σύνταξη	(unsigned)long	millis(), micros()
;	float	delay(), delayMicroseconds()
{ }	double	Μαθηματικές
// ή /* */	string (πίνακας char)	min(), max(),abs(), constraint(), map()
#define	String (αντικείμενο)	pow(), sqrt(), sin(), cos(), tan()
#include	array	randomSeed(), random()
Αριθμητικοί τελεστές	Μετατροπή	Bits/Bytes
=	char(), byte(), int(), word()	lowByte(), highByte(), bitRate(), bitWrite
+, -, *, /, %	long(), float()	bitSet(), bitClear(), bit()
Σύγκριση	Πεδίο μεταβλητών	Interrupts
==, !=	variable scope, static	attachInterrupt(), detachInterrupt()
<, >, <=, >=	volatile, const	interrupts(), noInterrupts()
Λογικοί τελεστές	Εργαλεία	Επικοινωνία
&&, , !	sizeof()	Serial

Πίνακας 3.3 Γενικές εντολές της γλώσσας του Arduino

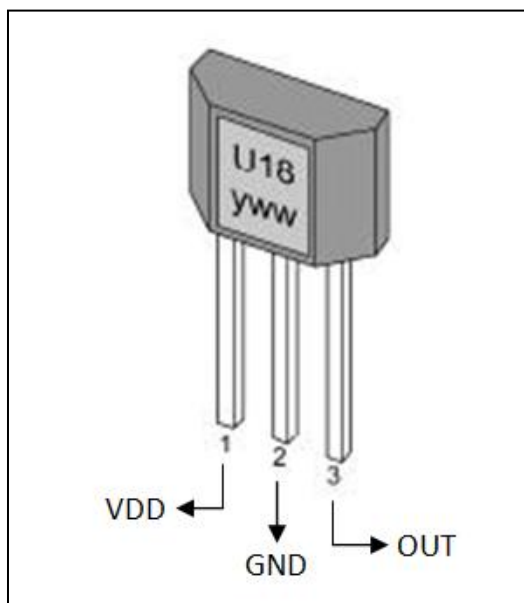
3.4. Αισθητήρες

Στην παρούσα εργασία χρησιμοποιήθηκαν οι εξής τύποι αισθητήρων:

- Αισθητήρας φαινόμενου Hall (Hall Effect) [12], [18]
- Παθητικό αισθητήριο PIR [13], [18]

3.4.1. Αισθητήρας φαινόμενου Hall (Hall Effect)

- Αρχή λειτουργίας



Εικόνα 3.10 Αισθητήρας φαινόμενου Hall

Οι αισθητήρες φαινόμενου Hall [1], [10] χρησιμοποιούνται στους διανομείς πολλών συστημάτων ανάφλεξης για να ενεργοποιήσουν το πρωτεύον κύκλωμα (έναυση σπινθήρα) και για να μετρήσουν τις στροφές/λεπτό του κινητήρα.

Αλλά χρησιμοποιούνται επίσης, σε πολλά συστήματα ανάφλεξης χωρίς διανομέα (DIS) για να καθορίσουν τη θέση του στροφαλοφόρου και του εκκεντροφόρου άξονα. Αυτό συμβαίνει επειδή ο εγκέφαλος του κινητήρα πρέπει να ξέρει που ο αριθμός ένα κύλινδρος είναι μόλις η μηχανή αρχίσει να στροφάρει. Μόλις ο πολύ σημαντικός "συγχρονισμένος παλμός" ανιχνεύεται, η ακολουθία ανάφλεξης μπορεί να αρχίσει να δίνει τάση στους σπινθηριστές στη σωστή ακολουθία. Αυτό επιτρέπει επίσης, το χρόνο

έγχυσης να αντιστοιχηθεί με το χρόνο που θα δοθεί ο σπινθήρας στις μηχανές με διαδοχική έγχυση καυσίμων. Οι αισθητήρες Hall αναφέρονται μερικές φορές ως "διακόπτες" λόγω του on-off "ψηφιακού" σήματος τάσης που παράγουν. Αντίθετα από τους μαγνητικούς αισθητήρες που παράγουν ένα σήμα εναλλασσόμενου ρεύματος (AC) που ποικίλλει στην τάση με την ταχύτητα, οι αισθητήρες Hall παράγουν ένα σταθερό σήμα τάσης που μπορεί να αλλάξει απότομα από τη μέγιστη τάση σε σχεδόν μηδέν και πάλι πίσω ανεξάρτητα από τις στροφές της μηχανής. Αυτό παράγει σήμα κυματομορφής σχεδόν τετραγωνικής που μπορεί να χρησιμοποιηθεί εύκολα από τον υπολογιστή για λόγους συγχρονισμού.

Ένας χαρακτηριστικός αισθητήρας Hall έχει τρία καλώδια ή τερματικά: ένα για τη γείωση, ένα για την τάση μπαταρίας ή αναφοράς και ένα για το σήμα εξόδου.

Χαρακτηριστικά:

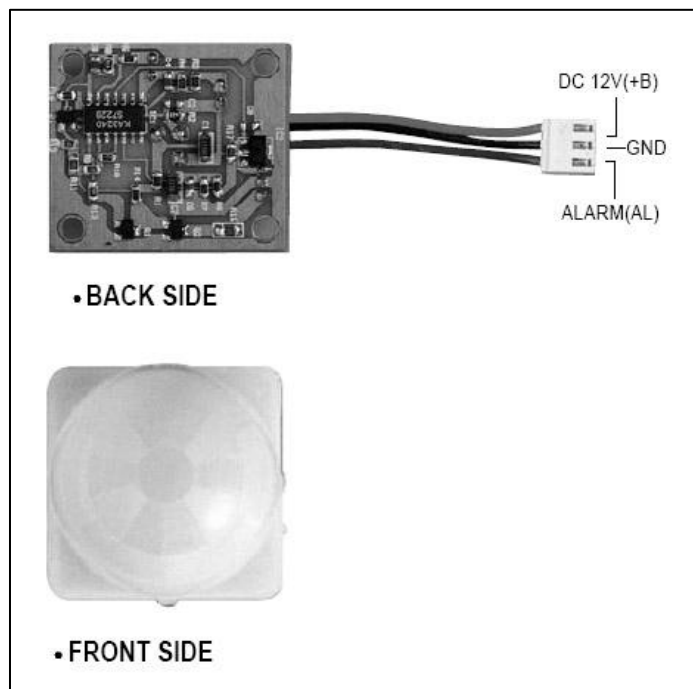
- Έξοδος ενός και μόνο bit από το αισθητήριο (1 ανίχνευση κίνησης, 0 κατάσταση αδράνειας περιβάλλοντος χώρου)
- Το μικρό μέγεθος του αισθητηρίου, το καθιστά εύκολο στο να τοποθετηθεί χωρίς να είναι εμφανές.
- Συμβατό με όλους τους μικροελεγκτές της Parallax και το μικροελεγκτή Arduino
- Λειτουργία σε 3.3Volt ή 5 Volt με ρεύμα μικρότερο των 100μΑ

• Ο ρόλος του στο σύστημα

Όταν ένα μεταλλικό έλασμα περνά μέσω του κενού αέρα μεταξύ του μαγνητικού πεδίου και του τσιπ πυριτίου, διακόπτει το μαγνητικό πεδίο και αναγκάζει την τάση παραγωγής του τσιπ να μειωθεί ξαφνικά στο μηδέν. Στο σύστημα μας αυτοί οι αισθητήρες τοποθετούνται στις πόρτες και στα παράθυρα μαζί με τους μαγνήτες για να ανιχνεύσουν την κίνηση μόλις ανοίξουν οι αντίστοιχες πόρτες/ παράθυρα.

3.4.2. Παθητικό αισθητήριο PIR

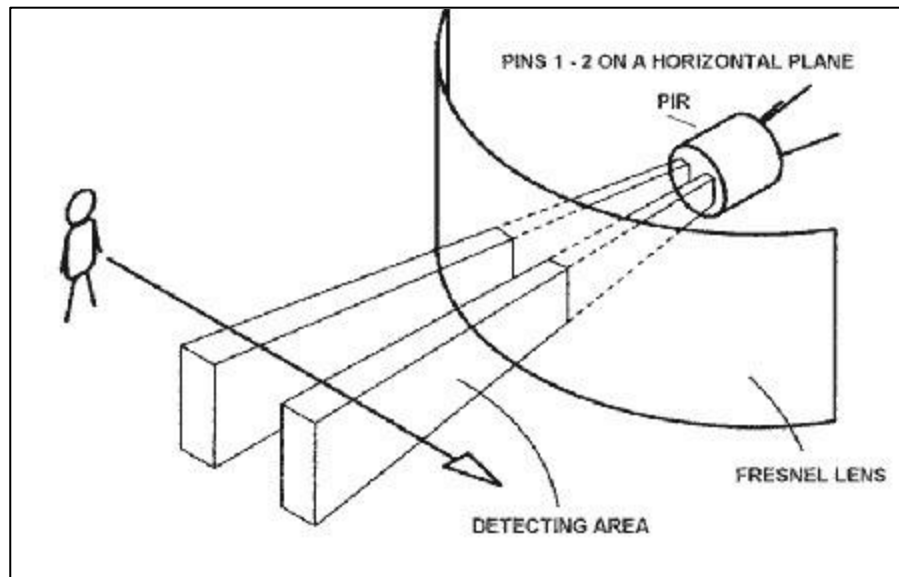
- Αρχή λειτουργίας



Εικόνα 3.11 Παθητικό αισθητήριο PIR

Με τον όρο παθητικό αισθητήριο εννοούμε παθητικό αισθητήριο υπέρυθρων [1] (Passive Infrared Sensor). Όπως υποδηλώνει το «παθητικό» στην ονομασία του, απαντά μόνο στην υπέρυθρη ενέργεια που ακτινοβολείται από το αντικείμενο που ανιχνεύτηκε. Το πιο συνηθισμένο αντικείμενο του οποίου ανιχνεύει την ακτινοβολία που εκπέμπεται ένα PIR είναι το ανθρώπινο σώμα, έτσι οι αισθητήρες αυτοί βρίσκουν χρήση σε αυτόματους διακόπτες φωτισμού, συστήματα συναγερμού, και ελεγκτές ανοίγματος πόρτας. Κάθε αντικείμενο με θερμοκρασία πάνω από το απόλυτο μηδέν εκπέμπει υπέρυθρη ενέργεια με μορφή ακτινοβολίας. Αυτή η ενέργεια είναι αόρατη στο ανθρώπινο μάτι, αλλά όχι στο πυροηλεκτρικό υλικό που βρίσκεται στον πυρήνα του αισθητήρα PIR. Όταν υποβάλλονται σε υπέρυθρη ακτινοβολία, τα πυροηλεκτρικά υλικά δημιουργούν ένα μικρό ηλεκτρικό φορτίο παρόμοιο με το ηλεκτρικό φορτίο που δημιουργείται όταν το ορατό φως χτυπάει ένα ηλιακό κύτταρο. Τα Πυροηλεκτρικά υλικά που χρησιμοποιούνται σε αυτούς τους αισθητήρες περιλαμβάνουν galliumnitride, νιτρικό

καΐσιο και Τανταλικό λίθιο. Η θερμοκρασία του δέρματος είναι περίπου 34°C , κατά κανόνα υψηλότερη από τη θερμοκρασία που υπάρχει στον περιβάλλοντα χώρο.



Εικόνα 3.12 Λειτουργία του αισθητήρα PIR

Καθώς ένα άτομο περνά μπροστά από το πεδίο του αισθητήρα, η υψηλότερη θερμοκρασία του δέρματός του δημιουργεί μια μεγαλύτερη φόρτιση στο πυροηλεκτρικό υλικό. Ένα κύκλωμα ενίσχυσης ενισχύει το μικρό σήμα που παράγεται από την προστιθέμενη ενέργεια των υπέρυθρων και τροφοδοτεί ένα διαφορετικό κύκλωμα σύγκρισης. Το κύκλωμα σύγκρισης αναζητά μια διαφορά στο σήμα από την αρχική ανάγνωση του, για να προκαλέσει μια έξοδο. Ωστόσο, αυτή η απλή λειτουργία μπορεί επίσης, να ενεργοποιείται από οποιαδήποτε πηγή ταχέως μεταβαλλόμενης φωτεινότητας ή θερμότητας, όπως είναι η λάμψη από έντονα φώτα ή ανακλάσεις από αντικείμενα κατά τη διάρκεια ζέστης, σε ηλιόλουστες ημέρες. Υπάρχουν αρκετές τεχνικές για να μειωθούν αυτές οι ψευδείς ενεργοποιήσεις. Πρώτον, το ανθρώπινο σώμα εκπέμπει υπέρυθρη ακτινοβολία σε μήκος κύματος από 9 έως 10 μm . Έτσι, ένα υπέρυθρο φίλτρο που περνά μήκη κύματος από τις 8 έως 14 μm είναι τοποθετημένο μπροστά από τον αισθητήρα για την ενίσχυση της ευαισθησίας στην υπέρυθρη ενέργεια που εκπέμπεται από τους ανθρώπους. Δεύτερον, ένας φακός Fresnel τοποθετείται μπροστά από τον αισθητήρα και εκτελεί δύο λειτουργίες. Επικεντρώνει την ενέργεια των

υπερύθρων που εκπέμπονται σε μια ευρύτερη περιοχή πάνω από τον αισθητήρα και χωρίζει την περιοχή σε θερμές και ψυχρές ζώνες ευαισθησίας. Καθώς λοιπόν ένα άτομο περπατά διασχίζοντας τις ζώνες, ο αισθητήρας βλέπει μια μεταβαλλόμενη αξία IR που παράγει ένα διαφορετικό σήμα εξόδου από τον αισθητήρα που ανιχνεύει την κίνηση. Το κύκλωμα σύγκρισης αναζητά και ανταποκρίνεται σε αυτή την αλλαγή σήματος. Θερμά αντικείμενα που δε μετακινούνται όπως θερμάστρες και φώτα, δεν παράγουν διακυμάνσεις στην εκπομπή υπερύθρων. Το κύκλωμα σύγκρισης αγνοεί αυτές τις σταθερές πηγές υπερύθρων ακτινοβολιών. Το αισθητήριο PIR είναι συμβατό με το μικροελεγκτή Arduino.

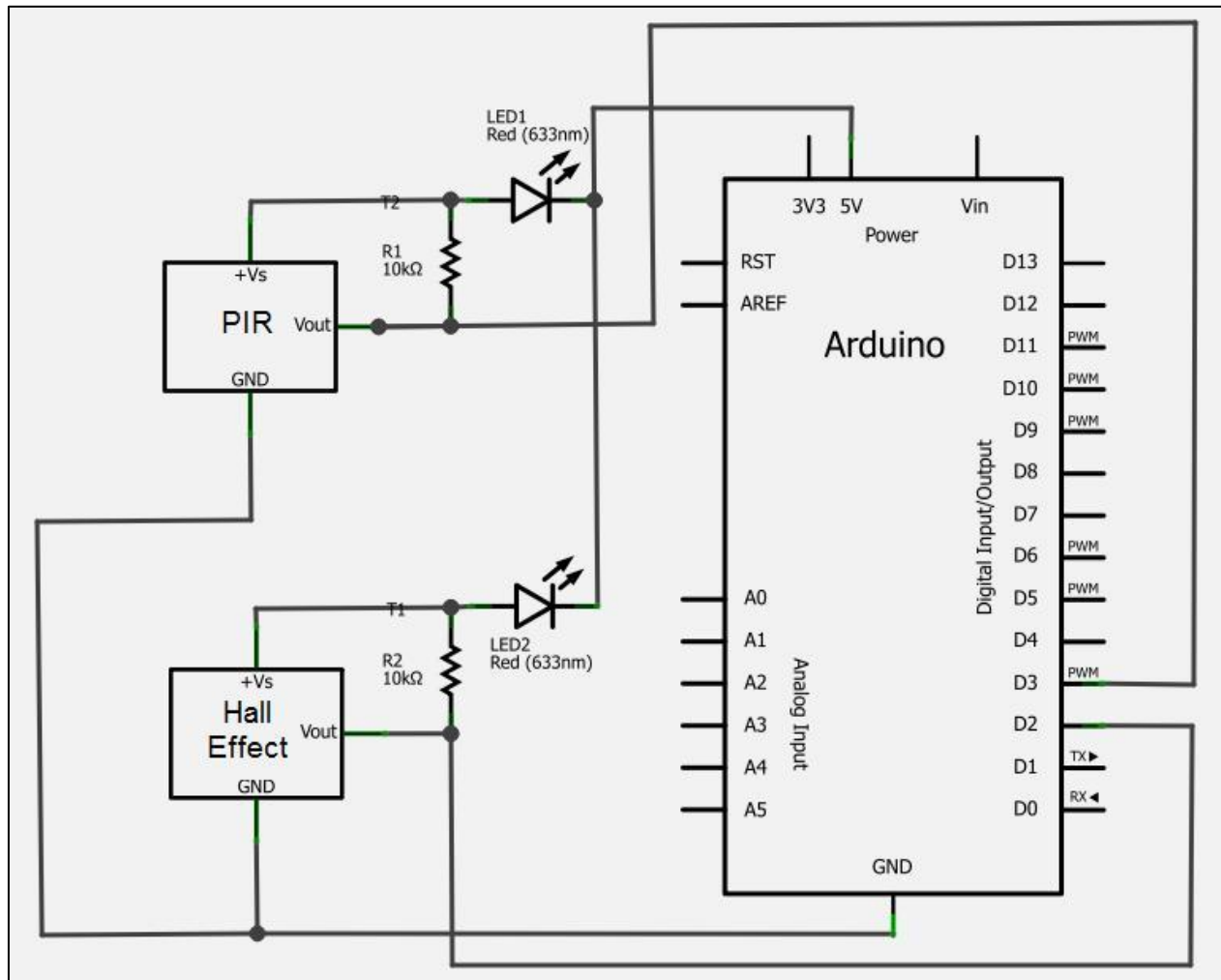
Χαρακτηριστικά:

- Έξοδος ενός και μόνο bit από το αισθητήριο (1 ανίχνευση κίνησης, 0 κατάσταση αδράνειας περιβάλλοντος χώρου)
- Το μικρό μέγεθος του αισθητηρίου, το καθιστά εύκολο στο να τοποθετηθεί χωρίς να είναι εμφανές.
- Συμβατό με όλους τους μικροελεγκτές της Parallax και το μικροελεγκτή Arduino
- Λειτουργία σε 3.3Volt ή 5 Volt με ρεύμα μικρότερο των 100μΑ

• Ο ρόλος του αισθητηρίου στο σύστημα

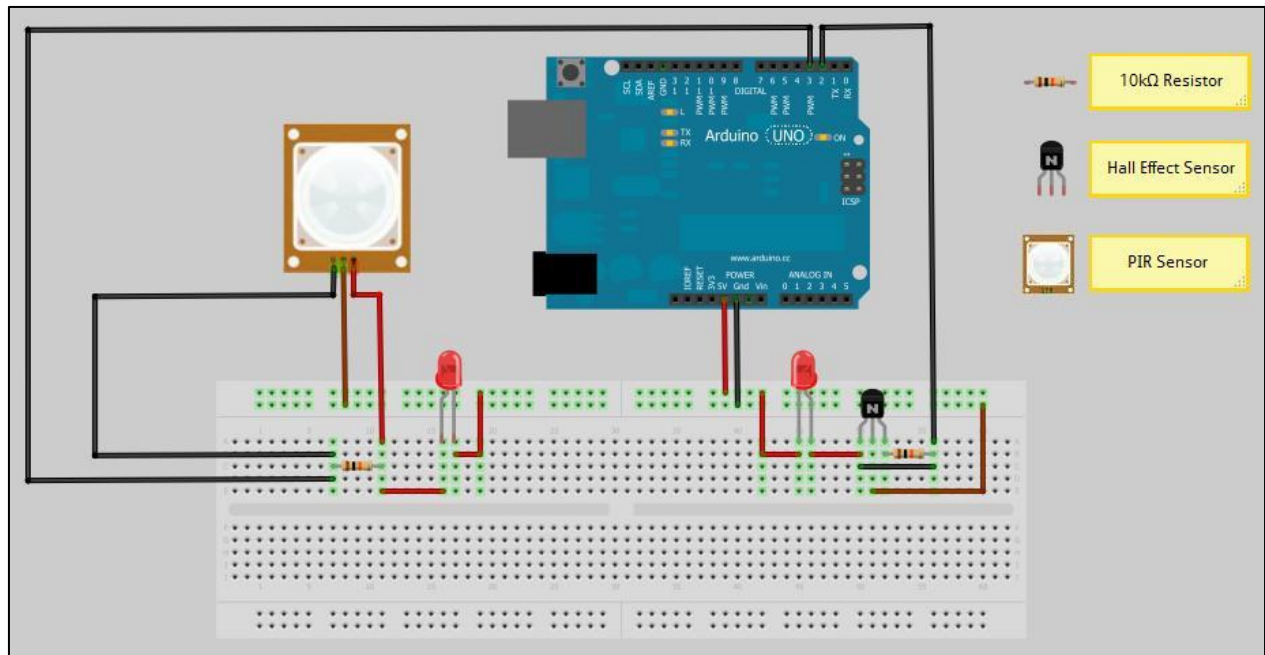
Η κίνηση αντικειμένων στο χώρο (και επομένως η αλλαγή της εκπεμπόμενης ποσότητας υπερύθρων) ανιχνεύεται από ένα μικροελεγκτή Arduino με τον έλεγχο του Pin Εισόδου που έχουμε ορίσει σε αυτόν. Αν ο ακροδέκτης εισόδου έχει στάθμη υψηλή, τότε σημαίνει ότι ανιχνεύθηκε κίνηση από το αισθητήριο.

3.5. Το κύκλωμα του συστήματος



Εικόνα 3.13 Σχηματικό διάγραμμα του κυκλώματος

Κάθε αισθητήρας είτε τύπου φαινομένου Hall είτε τύπου PIR προστέθηκε στο σύστημα από μία αντίσταση pull-up των 10 kΩ σύμφωνα με τα Data Sheets [12], [13] των αισθητήρων και ένα λαμπτήρα τύπου Led για να ανάβει όταν παρατηρηθεί κίνηση από τους αισθητήρες (όταν η κατάσταση είναι HIGH). Η εμφάνιση της προτυποποιημένη πλακέτα συνδέσεων (Breadboard) παρουσιάζεται στην *Εικόνα 3.13*.



Εικόνα 3.14 Διάγραμμα Breadboard του κυκλώματος

Στην Εικόνα 3.14 παρουσιάζονται οι συνδέσεις ανάμεσα στα υλικά που αποτελούν το σύστημα επάνω στην Breadboard .

ΚΕΦΑΛΑΙΟ 4^ο

ΤΟ ΛΟΓΙΣΜΙΚΟ ΜΕΡΟΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

4. Το λογισμικό μέρος του συστήματος

Το λογισμικό μέρος του συστήματος χωρίζεται σε δυο βασικά τμήματα. Το πρώτο τμήμα το οποίο ονομάζεται ως «διαδικτυακό τμήμα» δίνει τις δυνατότητες διαχείρισης του συστήματος στους χρήστες. Το δεύτερο τμήμα που ονομάζεται ως «τμήμα μικροελεγκτή» είναι η τεχνική πλευρά του συστήματος. Η αναγνώριση συσκευών, η σύνδεση μεταξύ των συσκευών και η σύνδεση του κυκλώματος με τον μικροελεγκτή πραγματοποιούνται σε αυτό το τμήμα του κώδικα. Αυτά τα δυο τμήματα παρουσιάζονται αναλυτικά σε αυτό το κεφάλαιο.

4.1. Διαδικτυακό τμήμα του συστήματος

Το διαδικτυακό τμήμα του συστήματος έχει αναπτυχθεί με χρήση Php, MySql και Ajax [15], [16], [17]. Όλες οι ρυθμίσεις που χρειάζονται για την εγκατάσταση του συστήματος σε ένα Web Server βρίσκονται στο αρχείο settings.php.

```
<? php
//database settings

$dbhost = "localhost";
$dbuser = "root";
$dbpass = "";
$dbname = "security2";
$homepage ="/";
$connect = mysql_connect($dbhost, $dbuser, $dbpass);

if(!$connect) die("Couldn't connect to MySQL host!");
mysql_select_db($dbname, $connect) or die ("Couldn't
connect to database!");

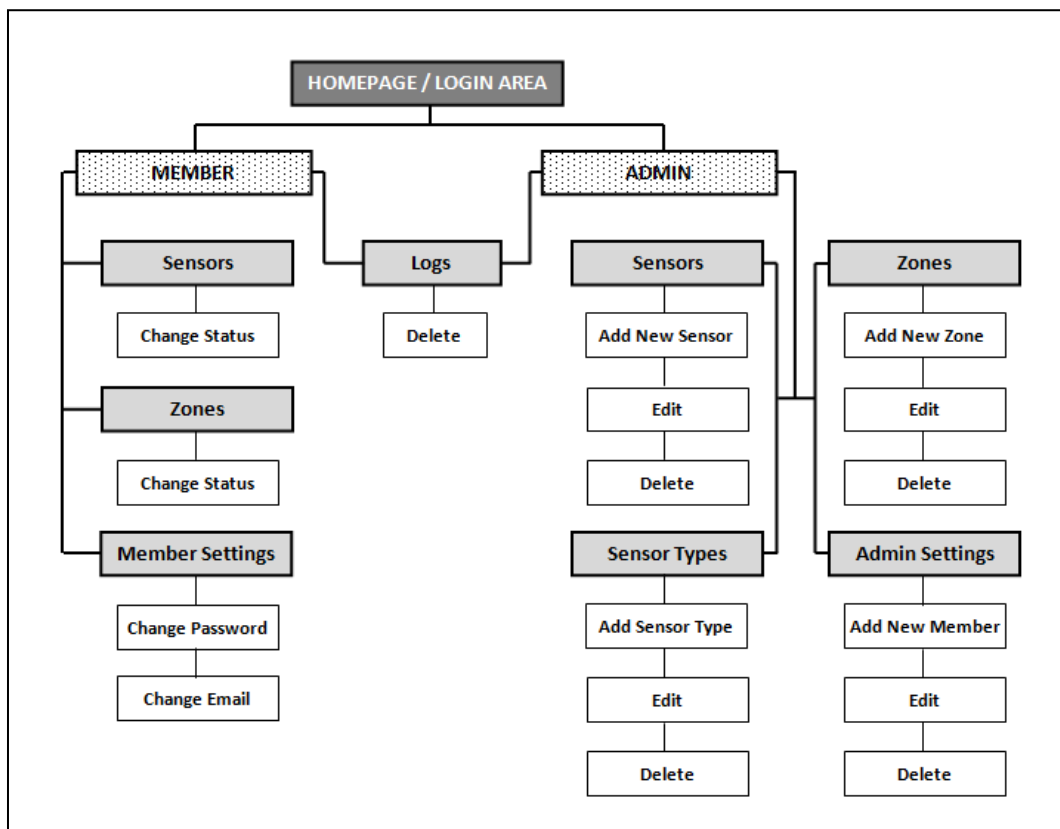
?>
```

Εικόνα 4.1 Ρυθμίσεις της ιστοσελίδας του συστήματος

Στις ρυθμίσεις της ιστοσελίδας ορίζονται και ελέγχεται τα στοιχεία της βάσης δεδομένων και έπειτα εάν το Hostname, το όνομα της βάσης δεδομένων, το όνομα χρήστη και το συνθηματικό είναι σωστά τότε γίνεται η σύνδεση στη βάση, αλλιώς εμφανίζεται το κατάλληλο μήνυμα λάθους.

4.1.1. Διαχείριση συστήματος

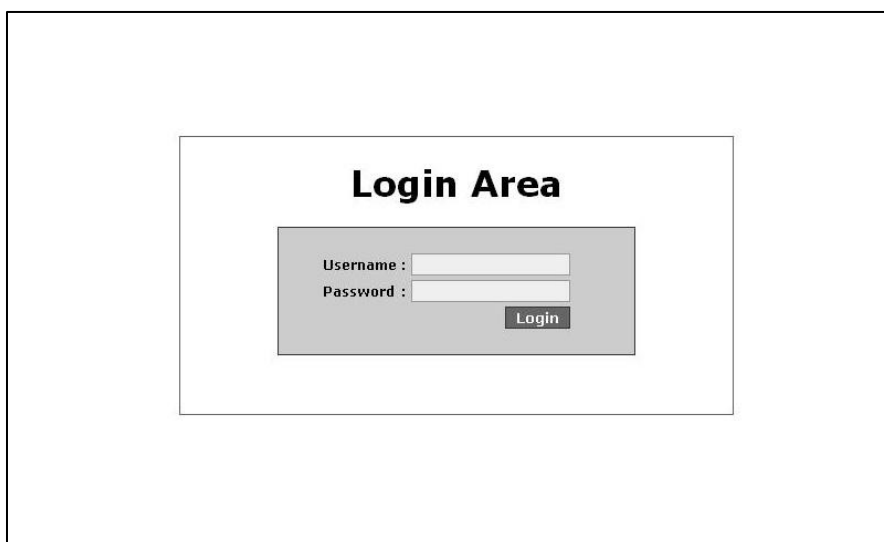
Οι λειτουργίες του συστήματος συναγερμού εξαρτώνται άμεσα από την κατηγορία στην οποία ανήκει ο χρήστης. Ο απλός χρήστης έχει διαφορετικό μενού από το διαχειριστή με περιορισμένη πρόσβαση στο σύστημα. Ο διαχειριστής έχει τη δυνατότητα να προσθέσει/επεξεργαστεί/διαγράψει τις ζώνες, τους αισθητήρες, τους τύπους αισθητήρα και τους χρήστες, ενώ ο απλός χρήστης έχει μόνο τη δυνατότητα να αλλάξει τις καταστάσεις των αισθητήρων και των ζωνών. Από την άλλη πλευρά ο διαχειριστής και οι απλοί χρήστες σαν κοινή λειτουργία έχουν τη δυνατότητα να δουν τα ιστορικά του συστήματος και να διαγράψουν οποία καταγραφή επιθυμούν. Η ενέργειες των χρηστών/διαχειριστών που μπορούν να κάνουν παρουσιάζονται στην *Εικόνα 4.2*.



Εικόνα 4.2 Ενέργειες χρηστών/διαχειριστών

- **Κοινές λειτουργίες για χρήστες και διαχειριστές**

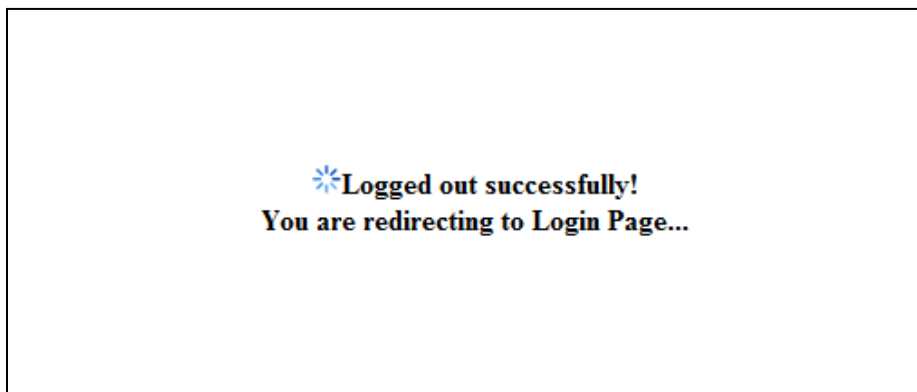
Σύνδεση: Κάθε χρήστης πρέπει να εισάγει το όνομα χρήστη και τον κωδικό του, προκειμένου να εισέλθει στο σύστημα. Αν δε συμπληρώσει τα πεδία ή εισάγει λάθος στοιχεία, τότε το σύστημα θα τον επαναφέρει στη σελίδα σύνδεσης με ένα μήνυμα λάθους. Μόλις ο χρήστης συνδεθεί στο σύστημα αμέσως δημιουργούνται κάποια Session, έτσι ώστε να μπορεί να έχει άμεση πρόσβαση σε βασικές πληροφορίες κατά την πλοήγησή του. Μόλις συνδεθεί ο χρήστης στο σύστημα δημιουργείται Session για να μεταφέρονται τα στοιχεία του από την μία ιστοσελίδα στην επόμενη.



The image shows a login form titled "Login Area". It contains two input fields: "Username :" and "Password :". Below the password field is a "Login" button. The form is centered within a larger rectangular frame.

Εικόνα 4.3 Αρχική σελίδα/φόρμα εισαγωγής

Αποσύνδεση: Κάθε χρήστης μπορεί να αποσυνδεθεί από τον αντίστοιχο σύνδεσμο στο μενού, έχοντας ως αποτέλεσμα τον τερματισμό της συνεδρίας του. Το σύστημα τον επαναφέρει αυτόματα στη σελίδα σύνδεσης:



Εικόνα 4.4 Μήνυμα Log Out

Μόλις αποσυνδεθεί ο χρήστης από το σύστημα καταστρέφεται το Session που έχει δημιουργηθεί πριν ώστε να μη μπορέσει κάποιος άγνωστος να συνδεθεί στο σύστημα.

Διαχείριση ιστορικού: Εδώ εμφανίζονται τα εξής ιστορικά στοιχεία του συστήματος ανά ημερομηνία:

- Αλλαγές κατάστασης ζωνών
- Αλλαγές κατάστασης αισθητήρων
- Μήνυμα προειδοποίησης που λαμβάνεται από τα Arduino μαζί με τα σχετικά στοιχεία των αισθητήρων και των ζωνών.

Event	Date	
➔ Zone: <i>Room</i> is turned on .	2012-05-07 02:41:04	✘
➔ Zone: <i>Room</i> is turned off .	2012-05-07 02:40:58	✘
➔ Sensor: <i>Sensor 1</i> is turned off .	2012-05-07 02:40:14	✘
➔ Sensor: <i>Sensor 2</i> is turned on .	2012-05-05 17:09:28	✘
➔ Zone: <i>Salloon</i> is turned on .	2012-05-05 16:55:01	✘
➔ Zone: <i>Salloon</i> is turned off .	2012-05-05 16:28:48	✘
⚠ Alarm Activated in Zone: <i>Room</i> by Sensor: <i>Sensor 2</i> !!!	2012-05-05 15:58:50	✘
⚠ Alarm Activated in Zone: <i>Salloon</i> by Sensor: <i>Sensor 1</i> !!!	2012-05-05 15:54:30	✘
⚠ Alarm Activated in Zone: <i>Room</i> by Sensor: <i>Sensor 2</i> !!!	2012-05-05 15:54:22	✘
⚠ Alarm Activated in Zone: <i>Salloon</i> by Sensor: <i>Sensor 1</i> !!!	2012-05-05 15:53:57	✘

Εικόνα 4.5 Ιστορικά του συστήματος

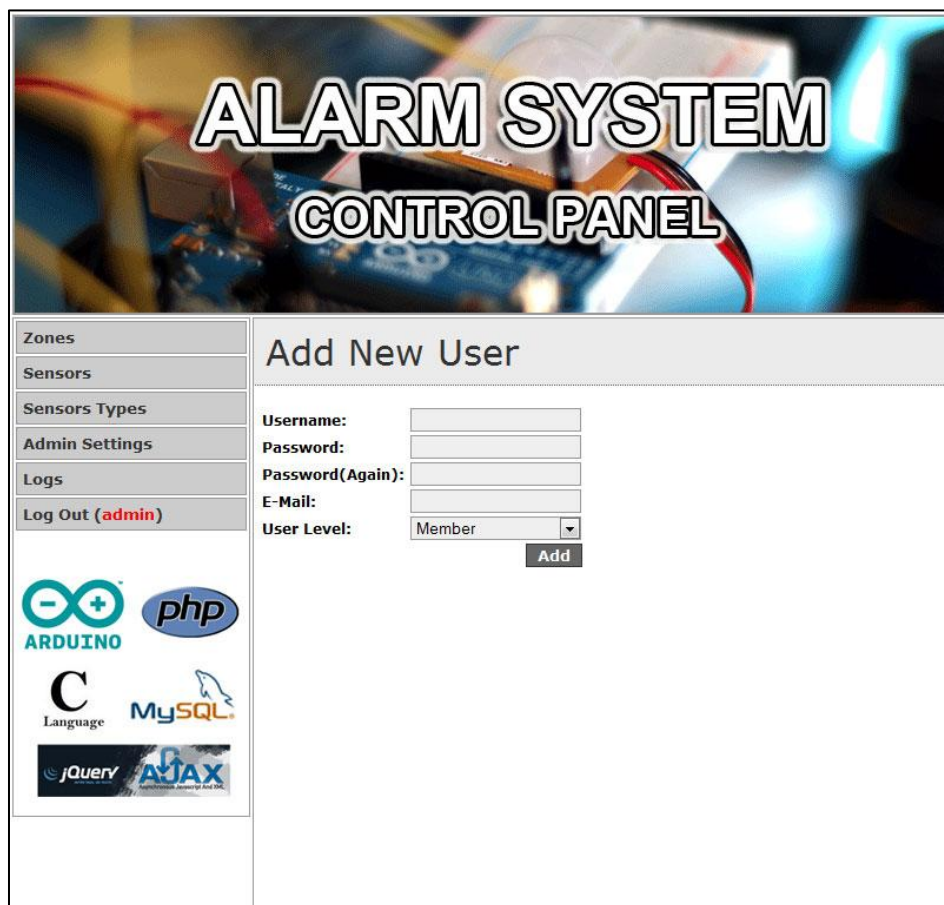
Επίσης, κάθε χρήστης έχει τη δυνατότητα να εμφανίσει τα αποτελέσματα ανά ζώνη ή ανά αισθητήρας και να διαγράψει όποια καταγραφή (log) επιθυμεί.

- **Λειτουργίες διαχειριστή**

Ο διαχειριστής έχει πρόσβαση στις εξής λειτουργίες:

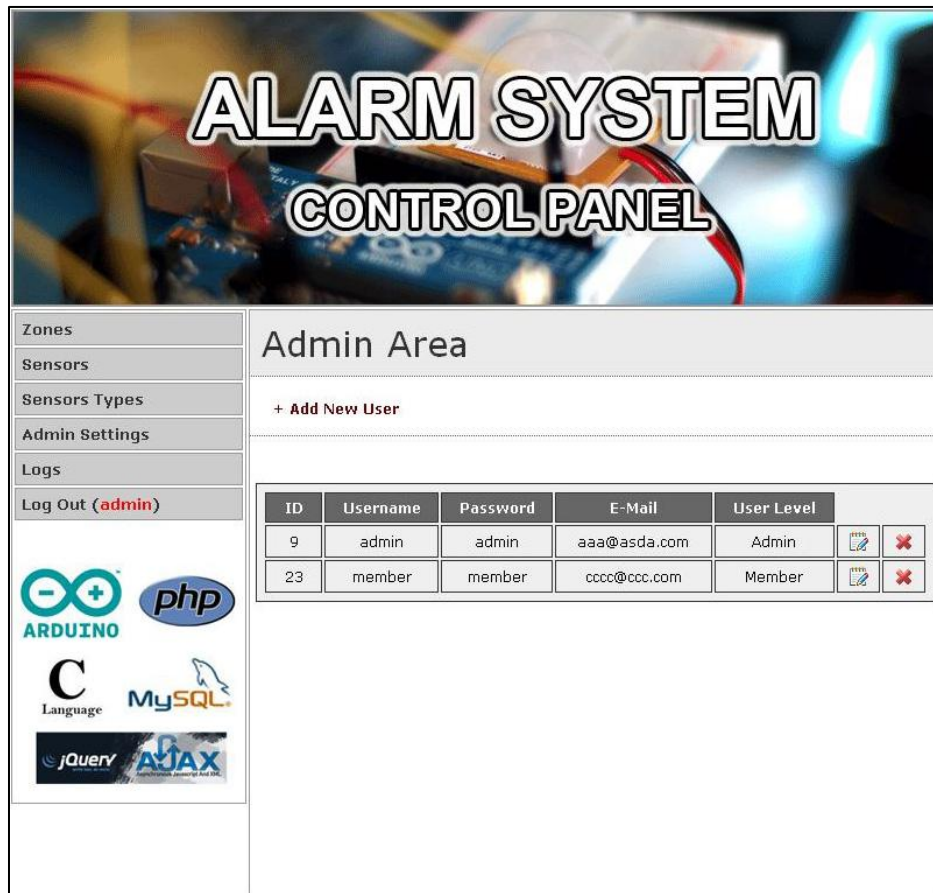
- **Διαχείριση χρηστών**

Ο διαχειριστής μπορεί να προσθέσει ένα νέο χρήστη, συμπληρώνοντας ένα μοναδικό όνομα χρήστη (υποχρεωτικό πεδίο), έναν κωδικό (υποχρεωτικό πεδίο), όνομα χρήστη και την κατηγορία στην οποία ανήκει. Αν δε συμπληρωθούν τα υποχρεωτικά πεδία, εμφανίζεται κατάλληλο μήνυμα λάθους. Μετά τη συμπλήρωση όλων των πεδίων χρήστη, το σύστημα οδηγεί το διαχειριστή στο σύνδεσμο «Admin Settings»:



Εικόνα 4.6 Προσθήκη νέου χρήστη

Εδώ ο διαχειριστής μπορεί να δει όλους τους χρήστες και είτε να ενημερώσει τα πεδία τους είτε να διαγράψει ένα συγκεκριμένο χρήστη:



Εικόνα 4.7 Διαχείριση χρηστών

- **Διαχείριση αισθητήρων**

Εδώ ο διαχειριστής μπορεί να δει όλους τους αισθητήρες και είτε να ενημερώσει τα πεδία τους είτε να διαγράψει ένα συγκεκριμένο αισθητήρα. Ο διαχειριστής μπορεί να προσθέσει ένα νέο αισθητήρα, συμπληρώνοντας ένα όνομα αισθητηρίου, ένα Arduino ID στο οποίο ανήκει ο αισθητήρας, ένα Pin ID το οποίο ανήκει στο συγκεκριμένο μικροελεγκτή Arduino, τον τύπο αισθητηρίου και τη ζώνη που ανήκει. Αν δε συμπληρωθούν όλα τα πεδία, εμφανίζεται το μήνυμα λάθους.

ALARM SYSTEM CONTROL PANEL

Sensors

Arduino No: 1
 Pin No: 2
 New Sensor Name:
 Sensor Type: Magnet
 Zone Name: Salloon
 Sensor Status: Off

Add

Ard.No	Pin No	Name	Type	Zone	Status
1	2	Sensor 1	Magnet	Salloon	ON
1	3	Sensor 2	Motion	Room	ON

Εικόνα 4.8 Διαχείριση αισθητήρων

- Διαχείριση ζωνών

Ο διαχειριστής μπορεί να δει όλες τις ζώνες και είτε να ενημερώσει τα πεδία τους είτε να διαγράψει μία συγκεκριμένη ζώνη. Ο διαχειριστής μπορεί να προσθέσει μια νέα ζώνη, συμπληρώνοντας απλά ένα όνομα ζώνης.



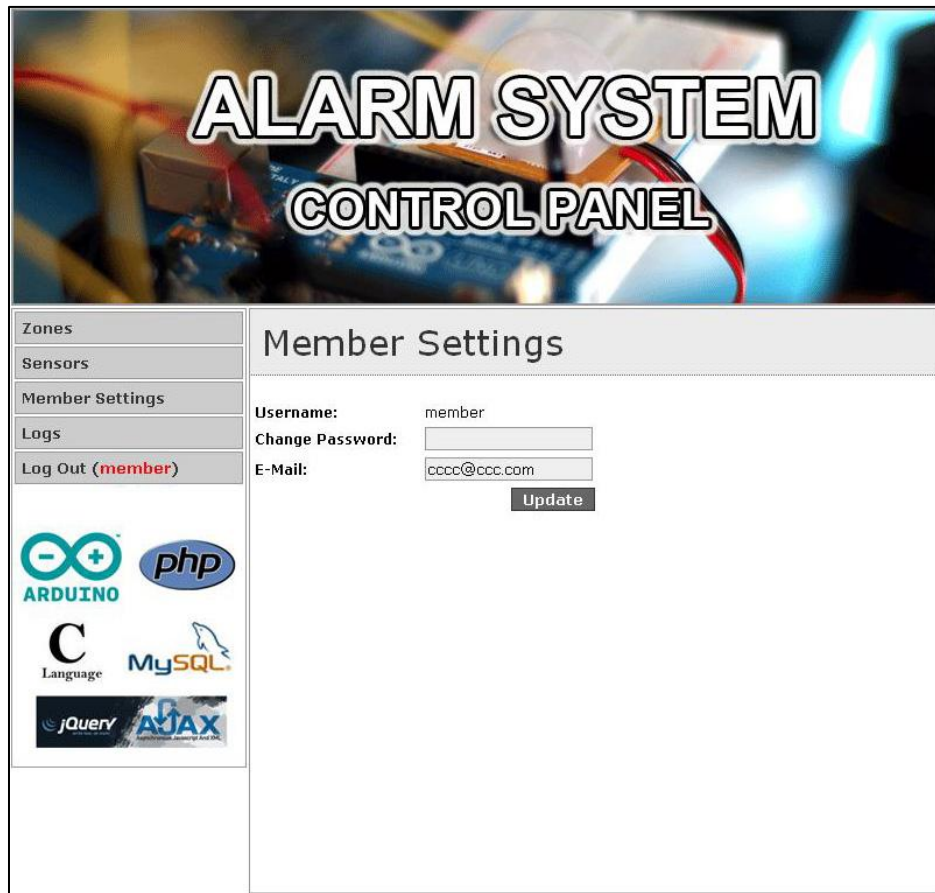
Εικόνα 4.9 Διαχείριση ζωνών

- **Λειτουργίες χρήστη**

Οι απλοί χρήστες έχουν πρόσβαση στις εξής λειτουργίες:

- **Διαχείριση στοιχείων χρήστη**

Οι χρήστες μέσα από το σύνδεσμο «Member Settings» μπορεί να δουν τα προσωπικά του στοιχεία καθώς και να αλλάξει κάποια από αυτά. Αυτά που μπορεί να ενημερώσει είναι ο κωδικός του ή/και την e-mail του. Το όνομα χρήστη χαρακτηρίζει μοναδικά το συγκεκριμένο χρήστη και έτσι δε μπορεί να αλλάξει:



Εικόνα 4.10 Ρυθμίσεις στοιχείων χρήστη

- **Διαχείριση αισθητήρων**

Οι χρήστες μέσα από το σύνδεσμο «Sensors» μπορεί να δουν τη λίστα αισθητήρων με τα εξής στοιχεία, καθώς και να αλλάξουν τη κατάσταση του κάθε αισθητηρίου.

- Όνομα αισθητηρίου
- Arduino ID το οποίο ανήκει
- Ζώνη που ανήκει
- Τύπος αισθητηρίου
- Κατάσταση αισθητηρίου

Ard. No	Sensor Name	Sensor Type	Zone Name	Status
1	Sensor 1	Magnet	Salloon	On Off
1	Sensor 2	Motion	Room	On Off

Εικόνα 4.11 Διαχείριση αισθητήρων

- **Διαχείριση ζωνών**

Μέσα από το σύνδεσμο «Sensors» μπορεί να δουν τη λίστα ζωνών και μαζί με τα ID τους, καθώς και να αλλάξουν τη κατάσταση της κάθε ζώνης.

ID	Zone Name	Status
213	Salloon	<input checked="" type="button" value="On"/> <input type="button" value="Off"/>
214	Room	<input type="button" value="On"/> <input checked="" type="button" value="Off"/>

Εικόνα 4.12 Διαχείριση ζωνών

4.1.2. Ajax

Η τεχνολογία AJAX (Asynchronous Javascript and XML) στην πραγματικότητα είναι μία ομάδα από αλληλένδετες τεχνικές ανάπτυξης λογισμικού για τον ιστοχώρο ή για τη δημιουργία rich Internet εφαρμογών. Με τη χρήση AJAX, οι web εφαρμογές μπορούν να ανακτήσουν δεδομένα από το server ασύγχρονα στο background χωρίς να παρεμβαίνουν στον τρόπο που παρουσιάζονται και συμπεριφέρονται οι σελίδες. Σε πολλές περιπτώσεις, υπάρχουν σελίδες σ'ένα website το περιεχόμενο των οποίων είναι κοινό. Χρησιμοποιώντας παραδοσιακές μεθόδους, το περιεχόμενο αυτό θα έπρεπε να φορτώνεται στον browser ξανά και ξανά με κάθε αίτημα. Από την άλλη, χρησιμοποιώντας AJAX, μία web εφαρμογή μπορεί να αιτηθεί μόνο το περιεχόμενο που

χρειάζεται και επομένως να μειωθεί το απαιτούμενο bandwidth και ο χρόνος για να φορτώσει.

Η χρήση ασύγχρονων αιτημάτων επιτρέπει στον interface του browser του client να είναι περισσότερο διαδραστικό, να ανταποκρίνεται πιο γρήγορα και οι σελίδες να ανανεώνονται ανεξάρτητα. Από την άποψη του χρήστη, η εφαρμογή θα φαίνεται πιο γρήγορη ή ότι ανταποκρίνεται καλύτερα παρόλο που στην μεριά του server αυτή δεν έχει αλλάξει. Χρησιμοποιώντας AJAX, οι συνδέσεις στο server μειώνονται, αφού τα κομμάτια κώδικα (scripts) και τα style sheets πρέπει να μεταφερθούν μόνο μία φορά ενώ, τέλος, οι JavaScript μεταβλητές θα διαρκούν, αφού η κυρίως σελίδα δε χρειάζεται να ανανεώνεται. Όπως βλέπουμε και στη παρακάτω Εικόνα 4.13, μόνο η συγκεκριμένη περιοχή που είναι πλαισιωμένη με κόκκινο χρώμα ανανεώνεται ανά 7 δευτερόλεπτα.

Log Out (member)

[-][+] php ARDUINO C Language MySQL jQuery AJAX

[1][2][3]...[7] Next >>

Event	Date	
➔ Zone: <i>Room</i> is turned on.	2012-05-07 02:41:04	✘
➔ Zone: <i>Room</i> is turned off.	2012-05-07 02:40:58	✘
➔ Sensor: <i>Sensor 1</i> is turned off.	2012-05-07 02:40:14	✘
➔ Sensor: <i>Sensor 2</i> is turned on.	2012-05-05 17:09:28	✘
➔ Zone: <i>Salloon</i> is turned on.	2012-05-05 16:55:01	✘
➔ Zone: <i>Salloon</i> is turned off.	2012-05-05 16:28:48	✘
⚠ Alarm Activated in Zone: <i>Room</i> by Sensor: <i>Sensor 2</i> !!!	2012-05-05 15:58:50	✘
⚠ Alarm Activated in Zone: <i>Salloon</i> by Sensor: <i>Sensor 1</i> !!!	2012-05-05 15:54:30	✘
⚠ Alarm Activated in Zone: <i>Room</i> by Sensor: <i>Sensor 2</i> !!!	2012-05-05 15:54:22	✘
⚠ Alarm Activated in Zone: <i>Salloon</i> by Sensor: <i>Sensor 1</i> !!!	2012-05-05 15:53:57	✘

Εικόνα 4.13 Η χρήση της Ajax στο σύστημα

```
<script type="text/javascript">
  $(document).ready(function() {
    var j = jQuery.noConflict();
    j(document).ready(function()
    {
      j(".refreshMe").everyTime(7000, function(i) {
        j.ajax({
          url: "log_table.php",
          cache: false,
          success: function(html) {
            j(".refreshMe").html(html);
          }
        });
      });
    });
  });
  j('.refreshMe');
});
</script>
```

Εικόνα 4.12 Ο κώδικας της Ajax για την ενημέρωση του ιστορικού

Σύμφωνα με τον παραπάνω κώδικα (Εικόνα 4.14) Ajax ενημερώνουμε σύγχρονα την πλαισιωμένη περιοχή (Εικόνα 4.13) που περιέχει τα δεδομένα του ιστορικού συστήματος.

4.1.3. CSS

Η **CSS** (*Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ*) ή (αλληλουχία φύλλων στυλ) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ 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.

Παρακάτω παρουσιάζεται το πως συντάσσονται τα στοιχεία στα φύλλα στυλ με ένα παράδειγμα από το σύστημα:

```
h1{  
  color: red;  
  font-size: 12px;  
}
```

Εδώ οι επικεφαλίδες h1, δηλαδή τα στοιχεία html στα οποία έχουμε δώσει το tag h1, είναι ο selector και του δίνουμε την τιμή κόκκινο για την ιδιότητα του χρώματος των γραμμάτων και μέγεθος γραμματοσειράς 12 pixel.

4.1.4. Βάση δεδομένων

Η MySQL είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (RDBMS) που τρέχει ως εξυπηρετητής, παρέχοντας πρόσβαση σε πολλούς χρήστες σε μια σειρά από βάσεις δεδομένων. Ουσιαστικά, είναι ένα πακέτο λογισμικού, το οποίο βοηθά πολύ στην οργάνωση και τη διαχείριση μεγάλων ποσοτήτων πληροφοριών. Η χρήση της διευκολύνει την πρόσβαση σε όλες τις πληροφορίες που βρίσκονται αποθηκευμένες στη βάση δεδομένων του συστήματος μας. Είναι ίσως το πιο διαδεδομένο σύστημα διαχείρισης βάσεων δεδομένων και χρησιμοποιείται ευρέως.

Για τη χρήση της MySQL, χρησιμοποιήθηκε η εφαρμογή ανοικτού κώδικα phpMyAdmin, η οποία είναι γραμμένη σε PHP και έχει ως σκοπό τη διαχείριση της MySQL στο διαδίκτυο. Μέσω αυτής της εφαρμογής, μπορεί κάποιος να δημιουργήσει τη βάση του στη MySQL, να δημιουργήσει τους πίνακες της βάσης του, να εκτελέσει κάποια επερωτήματα SQL και να διαγράψει ή να προσθέσει πίνακες, δεδομένα ή βάσεις.

Για να είναι επιτυχής μια βάση δεδομένων [7], θα πρέπει να είναι προσαρμοσμένη στις ανάγκες, απαιτήσεις, αλλά και προσδοκίες του τελικού χρήστη. Αυτό σημαίνει ότι το ζητούμενο είναι, τι πραγματικά επιθυμεί ο χρήστης, τι ακριβώς περιμένει από το σύστημα και πόσο φιλικό είναι αυτό σ' αυτόν και κατά πόσο ικανοποιεί τους σκοπούς για τους οποίους υλοποιήθηκε.

Η δημιουργία της βάσης δεδομένων του συστήματος συναγερμού έχει ως σκοπό να δώσει τις απαραίτητες πληροφορίες για το σύστημα , είτε σε χρήστες που θέλουν να διαχειριστούν τις καταστάσεις των αισθητήρων και των ζωνών, είτε σε διαχειριστές που θέλουν να κατασκευάσουν και να διαχειριστούν όλο το σύστημα (τους χρήστες, τις ζώνες, τους αισθητήρες κ.α.), είτε σε μικροελεγκτή που λειτουργεί με τα δεδομένα αυτά. Έτσι οι απαιτήσεις που σε αρχικό στάδιο θέλουμε να περιέχει η βάση είναι οι παρακάτω:

- **Υπαρξη αισθητήρων**

Κάθε αισθητήρας πρέπει να συμπεριλάβει: ένα μοναδικό ID, ένα Arduino Device ID και ένα αντίστοιχο Pin ID το οποίο ανήκει, ένα όνομα, ένα Type ID που εκφράζει τον τύπο αισθητήρα, μία ζώνη στην οποία ανήκει και ένα δείκτη κατάστασης.

- **Υπαρξη ζωνών**

Κάθε ζώνη πρέπει να συμπεριλάβει: ένα μοναδικό ID και ένα όνομα καθώς επίσης, και ένα δείκτη κατάστασης.

- **Υπαρξη χρηστών**

Κάθε χρήστης πρέπει να συμπεριλάβει: ένα μοναδικό ID, ένα όνομα, ένα συνθηματικό, μια ηλεκτρονική διεύθυνση καθώς και μια κατηγορία η οποία θα περιορίζει τα δικαιώματά του στο σύστημα.

- **Υπαρξη τύπων αισθητηρίου**

Κάθε τύπος αισθητηρίου πρέπει να συμπεριλάβει: ένα μοναδικό ID και ένα όνομα.

- **Υπαρξη δεδομένων ιστορικού**

Στα ιστορικά του συστήματος χρειάζονται τα εξής στοιχεία:

Ένα μοναδικό ID (History ID), ημερομηνία ιστορικού, Event ID, Device ID το οποίο ανήκει ο συγκεκριμένος αισθητήρας, το σχετικό Pin ID, το Zone ID και το Sensor ID.

- **Υπαρξη των Events**

Στο συγκεκριμένο σύστημα υπάρχουν 5 διαφορετικά Events για το ιστορικό του συστήματος. Ενεργοποίηση αισθητήρα, απενεργοποίηση αισθητήρα, ενεργοποίηση

ζώνης, απενεργοποίηση ζώνης και το μήνυμα προειδοποίησης. Κάθε Event πρέπει να συμπεριλάβει ένα μοναδικό ID και ένα όνομα.

Συνοψίζοντας, προκύπτουν οι εξής πίνακες στη Βάση Δεδομένων σύμφωνα με τις απαιτήσεις των χρηστών και των συσκευών:

Sensors

<u>sensor_id</u>	device_id	pin_id	sensor_name	<u>type_id</u>	<u>zone_id</u>	sensor_status
------------------	-----------	--------	-------------	----------------	----------------	---------------

Πίνακας 4.1 Πίνακας αισθητήρων

Sensor types

<u>type_id</u>	type_name
----------------	-----------

Πίνακας 4.2 Πίνακας τύπων αισθητήρα

Zones

<u>zone_id</u>	zone_name	zone_status
----------------	-----------	-------------

Πίνακας 4.3 Πίνακας ζωνών

Events

<u>event_id</u>	event_name
-----------------	------------

Πίνακας 4.4 Πίνακας των Events

History

<u>history_id</u>	history_date	<u>event_id</u>	device_id	pin_id	<u>zone_id</u>	<u>sensor_id</u>
-------------------	--------------	-----------------	-----------	--------	----------------	------------------

Πίνακας 4.5 Πίνακας ιστορικού

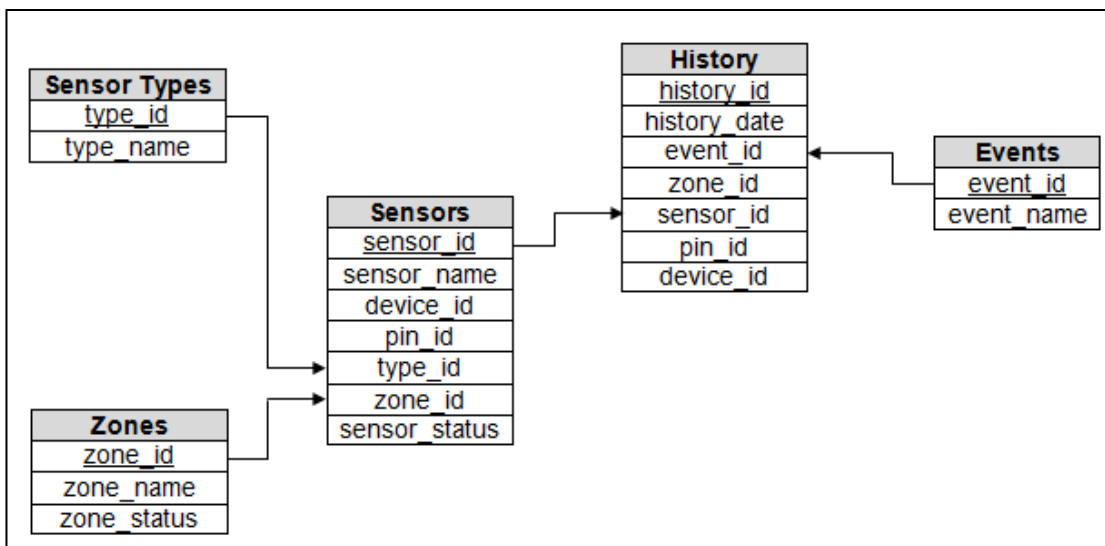
Users

<u>user_id</u>	username	password	email	date	user_level
----------------	----------	----------	-------	------	------------

Πίνακας 4.6 Πίνακας χρηστών

Τα πρωτεύοντα κλειδιά είναι υπογραμμισμένα με γραμμή, ενώ τα δευτερεύοντα κλειδιά (που είναι πρωτεύοντα κλειδιά σε άλλους πίνακες) είναι υπογραμμισμένα με διακεκομμένη γραμμή.

Το παρακάτω διάγραμμα (Εικόνα 4.15) περιγράφει τις σχέσεις [2] που προκύπτουν μεταξύ των πινάκων:



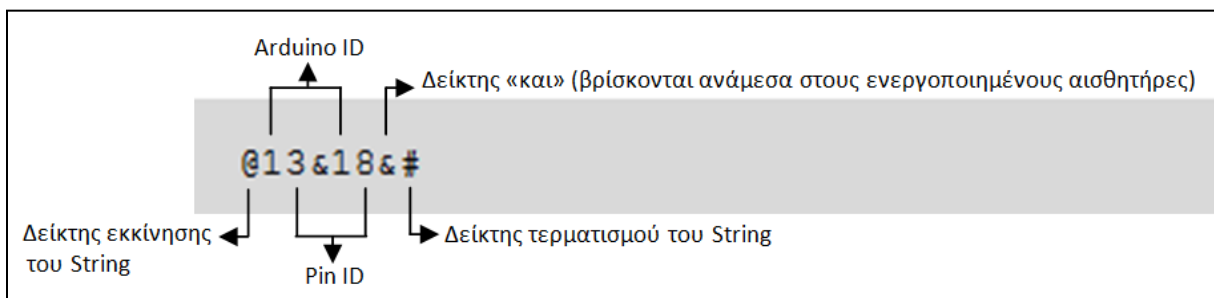
Εικόνα 4.15 Συσχετίσεις πινάκων

4.1.5. Σύνδεση με τον Arduino

Για τη σύνδεση της ιστοσελίδας με το μικροελεγκτή Arduino δημιουργήθηκαν τα εξής αρχεία Php:

- **arduino.php**

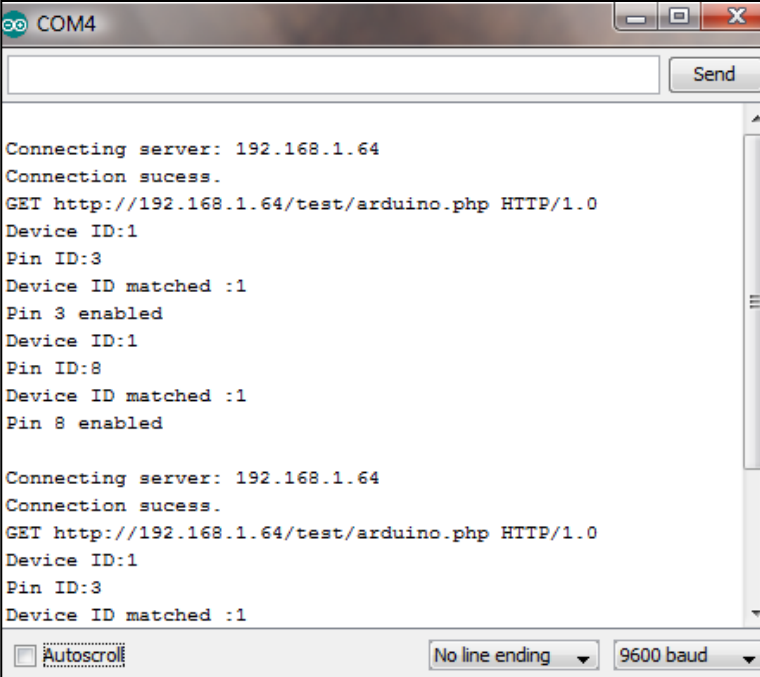
Εδώ η επικοινωνία γίνεται αμφίδρομα μεταξύ της ιστοσελίδας και των συσκευών με ένα ειδικά διαμορφωμένο String το οποίο παρουσιάζεται στην Εικόνα 4.16. Όλες η συσκευές κάνουν GET το ίδιο string που περιλαμβάνεται στο arduino.php.



Εικόνα 4.16 Παράδειγμα του string που λαμβάνεται από τον Arduino

Μόλις αρχίσει να λειτουργεί ο μικροελεγκτής Arduino, συνδέεται στο DHCP Server και παίρνει μια αυτόματη διεύθυνση IP. Στη συνέχεια χρησιμοποιώντας το πρωτόκολλο HTTP ελέγχει το string που παράγει το συγκεκριμένο αρχείο να λαμβάνει τις ορισμένες ρυθμίσεις. Αυτό το string περιέχει τα Pin ID των ενεργοποιημένων αισθητήρων και Arduino Device ID που ανήκουν. Το string αυτό ξεκινά με το σύμβολο «@» και τερματίζεται με το σύμβολο «#». Ανάμεσα στους ενεργοποιημένους αισθητήρες βρίσκεται το σύμβολο «&». Επίσης, το πρώτο ψηφίο πριν το σύμβολο «&» χαρακτηρίζει το Pin ID του συγκεκριμένου αισθητηρίου και το δεύτερο ψηφίο χαρακτηρίζει το Arduino Device ID που ανήκει. Δηλαδή όλοι οι άλλοι αισθητήρες και μικροελεγκτές παραμένουν απενεργοποιημένοι που δεν ορίζονται μέσα σε αυτό το String. Στο παράδειγμα που παρουσιάζεται στην Εικόνα 4.16 το string περιγράφεται ως εξής: ενεργοποίησε τον αισθητήρα με ID 3 και 8 που ανήκουν στο Arduino με ID 1. Αυτό το string λαμβάνεται από τον Arduino ανά 5 δευτερόλεπτα και ο Arduino συγκρίνει το σχετικό ID με το δικό

του και έπειτα δημιουργεί το κατάλληλο μήνυμα στο Serial Monitor το οποίο παρουσιάζεται στην Εικόνα 4.17.



```
COM4  
Connecting server: 192.168.1.64  
Connection success.  
GET http://192.168.1.64/test/arduino.php HTTP/1.0  
Device ID:1  
Pin ID:3  
Device ID matched :1  
Pin 3 enabled  
Device ID:1  
Pin ID:8  
Device ID matched :1  
Pin 8 enabled  
  
Connecting server: 192.168.1.64  
Connection success.  
GET http://192.168.1.64/test/arduino.php HTTP/1.0  
Device ID:1  
Pin ID:3  
Device ID matched :1
```

Εικόνα 4.17 Εμφάνιση μηνύματος στο *Serial Monitor*

- **post_data.php**

Εδώ η επικοινωνία γίνεται από συσκευή προς την ιστοσελίδα. Όλες οι συσκευές δημιουργούν από ένα διαμορφωμένο string σύμφωνα με τα στοιχεία τους και το στέλνουν στο `post_data.php` μόλις παρατηρηθεί κίνηση στο σύστημα.

Όταν παρατηρηθεί κίνηση στις περιοχές που τοποθετούνται οι αισθητήρες, ο μικροελεγκτής στέλνει στο συγκεκριμένο αρχείο `php` ένα string μέσω HTTP το οποίο περιλαμβάνει το συνθηματικό, το Pin ID, το σχετικό Device ID και τη συγκεκριμένη Event ID (που είναι το 5). Στο παράδειγμα που παρουσιάζεται στην Εικόνα 4.18 το string που στέλνει ο Arduino στην ιστοσελίδα, περιγράφεται ως εξής: έχει παρατηρηθεί κίνηση από τον αισθητήρα που τοποθετείται στο Pin 3 του μικροελεγκτή Arduino με ID 1.




```

COM4
Connecting server: 192.168.1.64
Connection success.
GET http://192.168.1.64/test/arduino.php HTTP/1.0
Device ID:1
Pin ID:8
Device ID matched :1
Pin 8 enabled
Device ID:1
Pin ID:3
Device ID matched :1
Pin 3 enabled
posting data to web
Server connected.
Sending web request:
GET http://192.168.1.64/test/post_data.php?password=123456&event_id=5&pin_id=3&device_id=1 HTTP/1.0
posting data to web completed

Connecting server: 192.168.1.64
Connection success.
GET http://192.168.1.64/test/arduino.php HTTP/1.0
Device ID:1
    
```

Εικόνα 4.18 Παράδειγμα του *string* που αποστέλλεται στην ιστοσελίδα

Μόλις σταλεί το *string* στην ιστοσελίδα (στο *post_data.php*) ελέγχεται το συνθηματικό αν είναι σωστό ή λάθος (Εικόνα 4.20), εάν είναι σωστό τότε τα δεδομένα αυτά καταγράφονται στη βάση δεδομένων και εμφανίζεται το κατάλληλο μήνυμα προειδοποίησης στο ιστορικό της ιστοσελίδας όπως φαίνεται στην εικόνα 4.18.

Event	Date	
 Alarm Activated in Zone: <u>Salloon</u> by Sensor: <u>Sensor 1</u>!!!	2012-06-19 13:44:13	

Εικόνα 4.19 Παράδειγμα μηνύματος προειδοποίησης στα ιστορικά

Το μήνυμα που εμφανίζεται στα ιστορικά (4.19) περιλαμβάνει η ημερομηνία του μηνύματος, το όνομα του αισθητήρα και το όνομα της ζώνης που ανήκει ο σχετικός αισθητήρας.

```
<?php
$password = $_REQUEST['password'];
$event_id = $_REQUEST['event_id'];
$pin_id = $_REQUEST['pin_id'];
$device_id = $_REQUEST['device_id'];

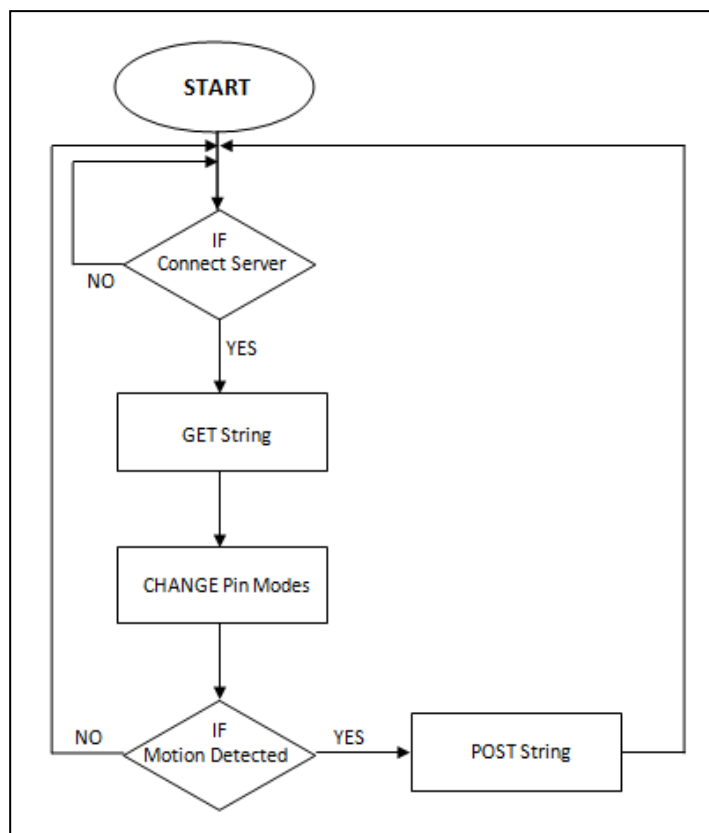
if($password == '123456')
{
    $sql="INSERT INTO history
(history_id,event_id,history_date,pin_id,device_id) VALUES
(NULL,'$event_id',NOW(),'$pin_id','$device_id')";
    $result = mysql_query($sql) or die('error executing
command');
}
?>
```

Εικόνα 4.20 Κώδικας του αρχείου *post_data.php* που λαμβάνει το string

Μετά τη λήψη του συγκεκριμένου String το αρχείο *post_data.php* ελέγχει το συνθηματικό αν είναι σωστό ή λάθος. Εάν το συνθηματικό σωστό τότε καταγράφει τα δεδομένα (*event_id*, *pin_id*, *device_id* καθώς και την προσωρινή ημερομηνία) στη βάση δεδομένων τα οποία αποστέλλονται από τον Arduino.

4.2. Το τμήμα μικροελεγκτή

Σε αυτό το υποκεφάλαιο παρουσιάζεται η δομή του προγράμματος, οι βιβλιοθήκες που χρησιμοποιήθηκαν κατά τη διάρκεια του προγραμματισμού. Όπως βλέπουμε και στο διάγραμμα ροής του προγράμματος (Εικόνα) παρατηρούμε ότι ο αλγόριθμος που σχεδιάστηκε τρέχει συνέχεια σε μία ρουτίνα Loop. Αρχικά δημιουργείται σύνδεση μεταξύ του μικροελεγκτή και της ιστοσελίδας. Εάν δεν πραγματοποιηθεί η σύνδεση τότε ο αλγόριθμος δεν προχωράει μέχρι να γίνει σύνδεση. Μόλις πραγματοποιηθεί η σύνδεση ο μικροελεγκτής κάνει Request για να λάβει το String και σύμφωνα με το String που λαμβάνεται καθορίζονται οι καταστάσεις των ψηφιακών ακροδεκτών. Εάν παρατηρηθεί κίνηση από κάποιο αισθητήρα τότε αποστέλλεται το σχετικό String (που αναφέραμε αναλυτικά στο Κεφάλαιο 4.1.5), έπειτα ο αλγόριθμος επιστρέφει στην αρχή. Εάν δεν παρατηρηθεί κίνηση τότε ο αλγόριθμος επιστρέφει στην αρχή.



Εικόνα 4.21 Το γενικό διάγραμμα ροής του προγράμματος

4.2.1. Δομή προγράμματος

Στην αρχή δηλώνονται τις βιβλιοθήκες (SPI,Ethernet,TimerOne) [21], [4] που απαιτούνται για την λειτουργία του συστήματος και στη συνέχεια τις συναρτήσεις και τις μεταβλητές.

Στη περιοχή των ρυθμίσεων (Εικόνα 4.22) ορίζονται οι ρυθμίσεις του συστήματος όπως η διεύθυνση MAC του Ethernet Shield, η διεύθυνση του εξυπηρετητή (αν τρέχει σε localhost τότε η IP), η θύρα που γίνεται η σύνδεση με τον υπολογιστή, ο αριθμός του Arduino (Device ID) και το συνθηματικό.

```
//-----Settings-----  
byte mac[] = {0x90,0xA2,0xDA,0x0D,0x0E,0xC5}; //mac address of  
Ethernet Shield  
//char serverName[] = "demomert.herobo.com";  
char serverName[] = "192.168.1.64";  
//char GetHeader[]="demomert.herobo.com";  
char GetHeader[]="192.168.1.64/security";  
int PortNumber = 80;  
char arduinoId='1'; // arduino device id  
String password = "123456"; //password  
//-----
```

Εικόνα 4.22 Γενικές ρυθμίσεις του προγράμματος στο Arduino

Έπειτα ακολουθεί η ρουτίνα setup (Εικόνα 4.23) που τρέχει μόνο μια φορά στην αρχή του προγράμματος. Μετά από τη ρουτίνα setup που είναι για να ρυθμίσει τις αρχικές παραμέτρους ακολουθεί η ρουτίνα loop όπου εκεί γράφεται ο κυρίως κώδικας. Η ρουτίνα αυτή τρέχει συνέχεια (σε loop), εκτός αν στη ρουτίνα setup γραφτεί η εντολή noLoop();. Μόνο τότε η κύρια ρουτίνα θα εκτελεστεί μια φορά.

```

void setup()
{
  Timer1.initialize(5000000);
  Timer1.attachInterrupt( timerIsr );
  pinMode(13, OUTPUT);
  Serial.begin(9600);

  if (Ethernet.begin(mac) == 0)
  {
    Serial.println("Failed to configure Ethernet using DHCP");
    Serial.println("Failed to obtain IP address from DHCP server. Now halting..");
    while(true);
  }
  delay(1000);
  Serial.println("IP address obtained from DHCP server");

  pinMode(2, INPUT);
  pinMode(3, INPUT);
  pinMode(4, INPUT);
  pinMode(5, INPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT);
  pinMode(8, INPUT);
  pinMode(9, INPUT);

  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, HIGH);
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(8, HIGH);
  digitalWrite(9, HIGH);

  Serial.println("Connecting Server:");
  if (client.connect(serverName, PortNumber))
  {
    Serial.println("Server connected.");
    header="GET http://";
    header+=GetHeader;
    header+="/arduino.php HTTP/1.0";
    Serial.println("Sending header request:");
    Serial.println(header);
    client.println(header);
    client.println();
  }
  else
  {
    Serial.println("Server connection failed. Host name invalid or server down.");
  }
}

```

Εικόνα 4.23 Κώδικας της ρουτίνας Setup

Στη ρουτίνα Setup αρχικά ελέγχεται η διεύθυνση MAC του Ethernet Shield και εάν είναι σωστή τότε αποστέλλεται Request για να πάρει μια IP διεύθυνση από το DHCP Server. Μετά τη λήψη της IP διεύθυνσης ορίζονται οι ψηφιακές ακροδέκτες 2,3,4,5,6,7,8,9 ως είσοδοι (INPUT). Έπειτα συνδέεται στην ιστοσελίδα στο αρχείο arduino.php και αποστέλλεται Request για τη λήψη του String από την ιστοσελίδα. Και όλη αυτή η διαδικασία επαναλαμβάνεται κάθε 5 δευτερόλεπτα με τη χρήση της εντολής Timer1 την οποία περιλαμβάνει η βιβλιοθήκη TimerOne που παρουσιάζεται στο Κεφάλαιο 4.2.2.

4.2.2. Βιβλιοθήκες που χρησιμοποιήθηκαν

Για τη χρήση ειδικών συσκευών ή μεθόδων λειτουργίας μπορεί να γίνει εισαγωγή ειδικών βιβλιοθηκών για χρήση περαιτέρω ειδικών εντολών, οι οποίες δε φορτώνονται στην εκκίνηση του προγράμματος. Υπάρχουν ήδη κάποιες βιβλιοθήκες εγκατεστημένες οι οποίες μπορούν να φορτωθούν για να γίνουν προσβάσιμες οι εντολές τους, στην αρχή του προγράμματος.

- **SPI**

Σειριακή περιφερική διεπαφή (SPI) είναι ένα σύγχρονο σειριακό πρωτόκολλο δεδομένων που χρησιμοποιείται από μικροελεγκτές για την επικοινωνία με μία ή περισσότερες περιφερειακές συσκευές σε μικρές αποστάσεις. Μπορεί επίσης, να χρησιμοποιηθεί για την επικοινωνία μεταξύ των δύο μικροεπεξεργαστών.

Αυτή η βιβλιοθήκη επιτρέπει να δημιουργηθεί επικοινωνία μεταξύ των SPI συσκευών, με το Arduino ως κύρια συσκευή (master device).

- **Ethernet**

Αυτή η βιβλιοθήκη επιτρέπει σε μία πλακέτα Arduino για να συνδεθεί στο διαδίκτυο. Μπορεί να χρησιμεύσει είτε ως διακομιστής που αποδέχεται εισερχόμενες συνδέσεις είτε ως client που δημιουργεί εξερχόμενες συνδέσεις. Αυτή η βιβλιοθήκη υποστηρίζει έως και τέσσερις ταυτόχρονες συνδέσεις (εισερχόμενες και εξερχόμενες ή ένα συνδυασμό). Ο μικροελεγκτής Arduino Uno επικοινωνεί με το Ethernet Shield με τη θύρα SPI. Επίσης, τα digital pins 10,11,12 και 13 χρησιμοποιούνται από το Ethernet Shield.

- **TimerOne**

Η βιβλιοθήκη TimerOne είναι μια συλλογή από ρουτίνες για τη ρύθμιση της 16-bit χρονιστής υλικού που ονομάζεται χρονιστή Timer1 στο ATmega168/328.

4.3. Ασφάλεια συστήματος

Η ασφάλεια στο σύστημα συναγερμού είναι ένα πολύ σημαντικό στοιχείο. Αυτό οφείλεται διότι όλος ο μηχανισμός βασίζεται στους υπολογιστές και στα δίκτυα που γίνεται η σύνδεση. Η ασφάλεια του συστήματος βασίζεται σε δυο βασικά μέρη ώστε να αποτρέπει τη μη εξουσιοδοτημένη πρόσβαση ή χρήση.

4.3.1. Ασφάλεια της ιστοσελίδας

Για την ασφάλεια της ιστοσελίδας του συστήματος προστέθηκαν οι εξής ενέργειες:

- **Χρήση Session:**

Για να αποτρέψουμε τον μη έγκυρο χρήστη από το να τυπώσει κατευθείαν στο URL την διεύθυνση των αρχείων (sensors.php, zones.php, settings.php, κ.α.) να αποκτήσει έτσι πρόσβαση στην βάση δεδομένων χρησιμοποιήσαμε τα Sessions. Με το που αποδεικνύεται η εγκυρότητα των στοιχείων του χρήστη, εκτελείται την εντολή `session_start()`; και δημιουργείται μια session μεταβλητή την `$_SESSION['entrance']` στην οποία δίνουμε μια τιμή στο `$_SESSION['entrance']` ανάλογα με τα δικαιώματα πρόσβασης του χρήστη και μετά κάνουμε `redirect` στην ιστοσελίδα `admin_settings.php/user_settings.php`.

Στο αρχείο `admin_settings.php/member_settings.php` καλούμε ξανά την `session_start()`; και μετά ελέγχω την τιμή της μεταβλητής `$_SESSION['entrance']`. Αν η τιμή είναι η αναμενόμενη (χρήστης/διαχειριστής), τότε προχωράμε στην εμφάνιση των περιεχομένων της ιστοσελίδας. Μόλις κάνει ο χρήστης Logout από το σύστημα καταστρέφεται το Session που έχει δημιουργηθεί με τη χρήση της εντολής `session_destroy()`; .

- **Χρήση της εντολής `mysql_real_escape_string`:**

Η SQL Injection είναι μια επίθεση που χρησιμοποιείται σήμερα από πολλούς «hackers». Με αυτή την επίθεση δίνεται η δυνατότητα σε κάποιον τρίτο να τρέξει sql εντολές και να πάρει ΠΟΛΥ σημαντικές πληροφορίες από την βάση δεδομένων στην οποία επιτίθεται. Η δυνατότητα αυτή οφείλεται στον λανθασμένο προγραμματισμένο κώδικα πολλών ιστοσελίδων. (Bug). Πληροφορίες όπως κωδικοί πρόσβασης, ονόματα χρηστών, emails είναι ο κύριος στόχος των «Hackers» μέσω αυτής. Αν καταφέρουν και πάρουν αυτά τα στοιχεία τότε θα είναι πλέον σίγουρο πως την ιστοσελίδα την έχει βαμμένη.

Υπάρχουν διάφοροι τρόποι να προστατευτεί κανείς και η PHP μας δίνει αρκετές συναρτήσεις να μας βοηθήσει.

Ουσιαστικά, αυτό που πρέπει να γίνει, είναι να γίνουν escaped όλοι οι ειδικοί χαρακτήρες έτσι ώστε ο SQL server που θα εκτελέσει το ερώτημα να μη τους λάβει υπ' όψιν του ως τμήμα του ερωτήματος, αλλά ως περιεχόμενο.

Παλιότερα υπήρχε η συνάρτηση `addslashes()` η οποία έπαιρνε ένα string, και, όπως λέει και το όνομά της, τοποθετούσε καθέτους πριν από τους ειδικούς χαρακτήρες.

Επίσης για να διευκολύνουν τους προγραμματιστές, έβαλαν και μια directive ονόματι `magic_quotes_gpc`, η οποία ουσιαστικά όταν ήταν σε λειτουργία εφάρμοζε αυτόματα την `addslashes()` σε όλες τις gpc (`$_GET`, `$_POST`) μεταβλητές.

Στις πιο καινούριες εκδόσεις προστέθηκε η πιο αποτελεσματική `mysql_real_escape_string()`. Με τη χρήση αυτής της εντολής διακόπτονται οι ειδικοί χαρακτήρες σε ένα String σε μια πρόταση SQL, λαμβάνοντας υπόψη την τρέχουσα charset της σύνδεσης.

```
$username = htmlentities(mysql_real_escape_string($_POST["username"]));  
$password = htmlentities(mysql_real_escape_string($_POST["password"]));
```

Εικόνα 4.24 Η χρήση της εντολής `mysql_real_escape_string()` στο σύστημα

Στην παρούσα εργασία χρησιμοποιήθηκε η εντολή `mysql_real_escape_string()` (Εικόνα 4.24) για την ασφάλεια της σύνδεσης.

4.3.2. Ασφάλεια των συσκευών

Για την ασφάλεια των συσκευών προστέθηκε ένα συνθηματικό στο πρόγραμμα για κάθε συσκευή. Όταν παρατηρηθεί κίνηση στο σύστημα μαζί με το string που περιλαμβάνει τα στοιχεία των αισθητήρων αποστέλλεται και αυτό το συνθηματικό έτσι ώστε εξασφαλίζεται ασφαλώς η πρόσβαση του string στο σύστημα.

```
//-----Settings-----  
byte mac[] = {0x90,0xA2,0xDA,0x0D,0x0E,0xC5}; //mac address of  
Ethernet Shield  
//char serverName[] = "demomert.herobo.com";  
char serverName[] = "192.168.1.64";  
//char GetHeader[] = "demomert.herobo.com";  
char GetHeader[] = "192.168.1.64/security";  
int PortNumber = 80;  
char arduinoId = '1'; // arduino device id  
String password = "123456"; //password  
//-----
```

Εικόνα 4.25 Συνθηματικό συσκευής που ορίζεται μέσα στις ρυθμίσεις

ΚΕΦΑΛΑΙΟ 5^ο

ΣΥΜΠΕΡΑΣΜΑ & ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΤΙΩΣΕΙΣ

1. Συμπέρασμα και μελλοντικές βελτιώσεις

Στη διπλωματική αυτή εργασία ύστερα από διερεύνηση και επιλογή των εργαλείων ανάπτυξης καθώς επίσης, από την ανάλυση των απαιτήσεων και το σχεδιασμό περιγράφηκε ο σχεδιασμός, η κατασκευή, η υλοποίηση και η λειτουργία ενός συστήματος συναγερμού που βασίζεται στην αρχιτεκτονική Arduino.

Η μονάδα που σχεδιάστηκε απαιτεί πολύ χαμηλό κόστος σε σχέση με τα άλλα συστήματα συναγερμού. Το χαμηλό κόστος σε συνδυασμό με τη διαδικτυακή υποστήριξη και σύνδεση σε Server καθιστούν μια καλή επιλογή μιας συγκεκριμένης κατάστασης.

Με την παρούσα διπλωματική εργασία προσθέτουμε ένα λιθαράκι που μας πλησιάζει πιο κοντά στην επίτευξη της καθολικής διασύνδεσης όλων των συσκευών με το Internet of things [21].

Το IOT (Internet of things), το οποίο στα ελληνικά μεταφράζεται ως διαδίκτυο των αντικειμένων, μπορεί κάλλιστα να χαρακτηριστεί ως το διαδίκτυο του μέλλοντος (Future Internet). Από την πρώιμη εποχή της πρώτης επιτυχημένης μεταφοράς δεδομένων μεταξύ δύο υπολογιστών σε διαφορετικά δωμάτια, η άμεση εξέλιξη των πραγμάτων όριζε, την ιδεατή για την τότε εποχή, δικτύωση όλων των υπολογιστών (τερματικών) πάνε σε ένα κοινό δίκτυο. Η ιδέα αυτή όσον αφορά τη σύνδεση (δικτύωση) των προσωπικών υπολογιστών του χρήστη και των υπολογιστικών συστημάτων (mainframes, servers) υλοποιείται εδώ και χρόνια με το ευρέως πλέον γνωστό διαδίκτυο (Internet), το οποίο δεν είναι τίποτα άλλο από τη σύνδεση πολλών δικτύων υπολογιστών μεταξύ τους.

Σίγουρα μία εφαρμογή σαν την παρούσα πάντοτε δέχεται βελτιώσεις και επεκτάσεις. Το σύστημα θα μπορούσε να περιλαμβάνει και άλλες δυνατότητες, μερικές από αυτές παρουσιάζονται παρακάτω

- Να παρέχει τη δυνατότητα διασύνδεσης με άλλους αισθητήρες (π.χ. αισθητήρας θερμοκρασίας, αισθητήρας αερίου κ.α.)

- Να έχει αποδοτικότερο προγραμματισμό για να χρησιμοποιεί λιγότερους πόρους του συστήματος,
- Θα μπορούσε να ακολουθεί τη φιλοσοφία του υλικού ασύρματης τεχνολογίας Zigbee, η οποία μας προσφέρει μια γρήγορη και έγκυρη μετάδοση δεδομένων χωρίς καλώδια, με ελάχιστη ενέργεια και κόστος
- Αντί για τη σύνδεση LAN θα μπορούσε να χρησιμοποιηθεί GPRS Modem ώστε να δουλεύει το σύστημα χωρίς Ιντερνέτ.
- Να έχει υποστήριξη ιστορικού κάμερας (διατήρηση φωτογραφιών στο αρχείο όταν υπάρξει event).
- Να έχει αντίστοιχες εφαρμογές για τα κινητά τηλέφωνα Android ή iPhone.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] I. N. Αβαριτσιώτης, *Τεχνολογία αισθητήρων και μικροσυστημάτων*, 2003.
- [2] Batini C., Cesi S. and Navathe S., *Conceptual Database Design: An Entity-Relationship Approach*, 1992.
- [3] Mahalik, N., *Sensor Networks and Configuration: Fundamentals, Standards, Platforms and Applications*, 2007,
- [4] Noble, J., *Programming interactivity: A Designer's Guide to Processing, Arduino and openFramework*, O' Reilly Media, 2009.
- [5] Barr, M., *Programming Embedded Systems in C and C ++*, 1999
- [6] Tan H., D' orazio T., *C για μηχανικούς*, Εκδόσεις Τζιόλα, 2000.
- [7] Ramakrishnan R. and J. Gehrke, *Συστήματα διαχείρισης βάσεων δεδομένων*, Εκδόσεις Τζιόλα, 2002.
- [8] Ilyas M., Mahgoub I., *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, CRC Press, 2005.
- [9] Evans, B., *Arduino Programming Notebook*, 2007.
- [10] Lenz JE. *A review of magnetic sensors*, IEEE, 1990.
- [11] Γαρύφαλλος Α. Γιώργος, *Ηλεκτρονικά συστήματα ασφάλειας*, Εκδόσεις ΙΩΝ, 2009.
- [12] Data Sheet of Hall Effect Sensor, US1881, 2006
- [13] Data Sheet of PIR Sensor, SE-10
- [14] Apache friends – XAMPP: <http://www.apachefriends.org>, Μάιος 2012
- [15] PHP - Hypertext Preprocessor: <http://www.php.net>, Μάιος 2012
- [16] W3C *Cascading Style Sheets*: <http://www.w3.org>, Μάιος 2012
- [17] MySql - Open source database: <http://www.MySql.com/>, Μάιος 2012
- [18] Sensirion AG: <http://www.sensirion.com>, Ιούνιος 2012
- [19] Wiring: <http://www.wiring.org.co/>, Ιούνιος 2012
- [20] Arduino – Homepage: <http://www.arduino.cc>, Ιούνιος 2012
- [21] The internet of things | Are you ready for the Internet of Things?: <http://www.theinternetofthings.eu/>, Ιούνιος 2012