



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Σύστημα Ασφαλείας  
&  
Απομακρυσμένος Έλεγχος Οθόνης**

**Σάλλας Παντελής  
Λαμπρινός Κωνσταντίνος**

**Εισηγήτρια: Αναστασία Ν. Βελώνη, Λέκτορας Εφαρμογών**

**ΑΙΓΑΛΕΩ**  
**2019**



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Σύστημα Ασφαλείας & Απομακρυσμένος Έλεγχος Οθόνης**

**Σάλλας Παντελής  
Α.Μ. 38193  
Λαμπρινός Κωνσταντίνος  
Α.Μ. 38470**

**Εισηγήτρια:**

**Αναστασία Ν. Βελώνη, Λέκτορας Εφαρμογών**

**Εξεταστική Επιτροπή:**

\_\_\_\_\_, Καθηγητής  
\_\_\_\_\_, Καθηγητής

**Ημερομηνία εξέτασης:     \_\_/\_\_/2019**



## ΕΥΧΑΡΙΣΤΙΕΣ

Ολοκληρώνοντας τον κύκλο σπουδών μας με την παρούσα πτυχιακή εργασία, θα θέλαμε να εκφράσουμε τις ευχαριστίες μας σε συγκεκριμένα άτομα που με τον δικό τους τρόπο μας βοήθησαν και μας στήριξαν.

Πρώτα και κυριότερα θα θέλαμε να ευχαριστήσουμε τις οικογένειές μας, και οι δύο ξεχωριστά, που μας στήριξαν και μας παρότρυναν να επιμείνουμε στο τελευταίο και δυσκολότερο κομμάτι, που ήταν αυτό της πτυχιακής μας εργασίας και εν τέλει τα καταφέραμε.

Τον φίλο μας Μιχάλη Βασιλάκη για την πολύτιμη βοήθεια που μας προσέφερε στην εκπλήρωση του προγραμματιστικού μέρους και για τις γνώσεις του πάνω στο αντικείμενο.

Την καθηγήτρια μας, Δρ. Αναστασία Βελώνη, για την άριστη συνεργασία μας και την υπομονή του καθ'όλη την διάρκεια εκπόνησης της εργασίας.

Τέλος, θα θέλαμε να ευχαριστήσουμε όλους εκείνους τους καθηγητές του τμήματος μας που μας προσέφεραν και μας μετέδωσαν τις γνώσεις τους πάνω στον τομέα της πληροφορικής και της μηχανικής υπολογιστών.



## ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με την ανάπτυξη ενός συστήματος ασφαλείας εργαστηριακού χώρου καθώς και τον απομακρυσμένο έλεγχο μια οθόνης που λειτουργεί ως πίνακας ανακοινώσεων. Τα συστήματα ασφαλείας έχουν εξελιχθεί σε πολύ μεγάλο βαθμό, καλύπτοντας τις ανάγκες φύλαξης που έχουν προκύψει σε δημόσιους και ιδιωτικούς χώρους.

Είναι πλέον απαραίτητη η ενεργοποίηση δικλίδων ασφαλείας που θα προστατεύσουν τις εγκαταστάσεις από πράξεις δολιοφθοράς-διάρρηξης-κλοπής-πυρκαγιάς κ.τ.λ.. Κάθε σύστημα ασφαλείας αποτελείται από διάφορες συσκευές, όπου η κάθε μια εκτελεί ένα συγκεκριμένο σκοπό και συγκεκριμένη εργασία, ανάλογα με τις ανάγκες που απαιτούνται.

Το σύστημα ασφαλείας της εργασίας μας αποτελείται από μαγνητικούς αισθητήρες για την κύρια πόρτα εισόδου και τα παράθυρα, καθώς κι από έναν αισθητήρα υπερήχων για την ανίχνευση της κίνησης στον χώρο. Όλα αυτά συνδέονται πάνω στο «κουτί» διαχείρισης όπου είναι και ο εγκέφαλος της εργασίας μας καθώς εκεί φυλάσσεται το Arduino Mega. Επιπλέον υπάρχει η οθόνη προβολής των ανακοινώσεων που εισάγει ο χρήστης, η οποία κι αυτή με τη σειρά της συνδέεται πάνω στο Arduino.

Στην εργασία μας γίνεται μία εισαγωγή στη γενική δομή των συστημάτων ασφαλείας και στη συνέχεια έχουμε μια περιγραφή των διάφορων τύπων αισθητήρων. Έπειτα παρουσιάζονται τα βασικά στοιχεία τόσο για την πλατφόρμα του Arduino όσο και για τις εντολές του. Τέλος μπορούμε να δούμε τη λίστα με τα υλικά τα οποία χρησιμοποιήθηκαν για την υλοποίηση της εργασίας.

**Επιστημονική Περιοχή:** Ασύρματο Δίκτυο

**Λέξεις Κλειδιά:** Θερμόμετρο, Ασύρματο Δίκτυο, Αισθητήρας, Arduino, ATmega328

## ABSTRACT

This diploma thesis deals with the development of a laboratory security system as well as the remote control of a screen that is used as a board for announcements. Security systems have evolved to a great extent, covering the needs of care that have arisen in public and private spaces.

It is now necessary to activate safety valves that will protect the installations from sabotage-breaking, theft-fire, etc. Each security system is consisted of various devices, each of whom executing a specific purpose, depending on user needs each time.

Our thesis security system is consisted of magnetic sensors for the main entrance door and windows, as well as an ultrasonic sensor that is used for motion control when the room is empty. All sensors are connected to the main box, which is kind of the central unit of our security system, because Arduino Mega is enclosed. In addition, there is a display screen for the announcements entered by the user, which in turn is connected to the Arduino as well.

At the beginning of our thesis, we are introducing the general structure of security systems and then we have a description of the different types of sensors. Then we present the key elements for both the Arduino platform and its commands. Last but not least, we can see the list of materials used to implement the whole thesis structure.

**Science Field:** Security System

**Key Words:** Sensor, Security System, Arduino, ATmega328, Remote control



## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ.....	7
ΚΕΦΑΛΑΙΟ 1.....	9
ΕΙΣΑΓΩΓΗ .....	9
ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΔΟΜΗ ΤΩΝ ΣΥΣΤΗΜΑΤΩΝ ΑΣΦΑΛΕΙΑΣ .....	10
ΚΕΦΑΛΑΙΟ 2.....	13
ΑΙΣΘΗΤΗΡΕΣ ΣΥΣΤΗΜΑΤΩΝ ΑΣΦΑΛΕΙΑΣ .....	13
ΚΕΦΑΛΑΙΟ 3.....	34
ARDUINO .....	34
ΚΕΦΑΛΑΙΟ 4.....	42
ΤΑ ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ ΓΙΑ ΤΗΝ ΕΡΓΑΣΙΑ.....	42
ΚΕΦΑΛΑΙΟ 5.....	46
ΤΟ ΛΟΓΙΣΜΙΚΟ ΜΕΡΟΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ .....	46
ΚΕΦΑΛΑΙΟ 6.....	53
ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΕΞΑΡΤΗΜΑΤΑ ΕΡΓΑΣΙΑΣ.....	53
ΚΕΦΑΛΑΙΟ 7.....	58
ΠΑΡΑΡΤΗΜΑ Α' – ΚΩΔΙΚΑΣ ΣΥΣΤΗΜΑΤΟΣ ΑΣΦΑΛΕΙΑΣ.....	58
ΚΕΦΑΛΑΙΟ 8.....	87
ΠΑΡΑΡΤΗΜΑ Β' – ΠΙΝΑΚΑΣ ΑΝΑΚΟΙΝΩΣΕΩΝ.....	87
ΚΕΦΑΛΑΙΟ 9.....	90
ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ .....	90
ΚΕΦΑΛΑΙΟ 10.....	92
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	92



## ΚΕΦΑΛΑΙΟ 1

### ΕΙΣΑΓΩΓΗ

Η ανάγκη εγκατάστασης συστημάτων ασφαλείας γίνεται όλο και μεγαλύτερη λόγω της αυξανόμενης εγκληματικότητας. Οι χώροι εγκατάστασης των συστημάτων ασφαλείας μπορεί να είναι σπίτια, δημόσια κτίρια, σχολεία, νοσοκομεία, επαγγελματικοί χώροι, γραφεία, καταστήματα, αποθήκες, εργοστάσια, αλλά και μεγάλες εγκαταστάσεις όπως αεροδρόμια, στρατόπεδα και γενικά κάθε είδους εγκαταστάσεις αλλά και εξωτερικοί χώροι όπου υπάρχει η ανάγκη προστασίας από διαρρήκτες ή εισβολείς όπου έχουν σκοπό για κακοβουλίες ενέργειες.

Ένα σύστημα ασφαλείας αποτελείται συνήθως από την κεντρική μονάδα επεξεργασίας στην οποία συνδέονται οι περιφερειακές συσκευές και αισθητήρες. Η κεντρική μονάδα δέχεται σήματα από τους αισθητήρες τα οποία τα επεξεργάζεται και καταλήγει για την ενεργοποίηση συναγερμού ή μη. Επίσης, στην κεντρική μονάδα συνδέονται ένα ή παραπάνω πληκτρολόγια για τον έλεγχο του συστήματος από το χρήστη, σειρήνες για την ηχητική ειδοποίηση σε περίπτωση συναγερμού, φάροι σηματοδότησης για την οπτική ειδοποίηση, καθώς και συσκευές για απομακρυσμένη ειδοποίηση μέσω τηλεφωνικού δικτύου, GSM, GPRS κτλ.

Οι αισθητήρες ενός συστήματος ασφαλείας είναι ό,τι πιο σημαντικό για την ορθή λειτουργία του συστήματος, λόγω του ότι είναι υπεύθυνοι για την ανίχνευση εισβολών στην προστατευόμενη περιοχή ή γενικά την παρουσία συνθηκών που θα προκαλούσαν την ενεργοποίηση συναγερμού. Για τους παραπάνω λόγους, ανάλογα τον χώρο όπου χρειάζεται επιτήρηση από το σύστημα ασφαλείας, θα πρέπει να υπάρξει και η σωστή επιλογή αισθητήρων με τα κατάλληλα χαρακτηριστικά για να υπάρχει σωστή κάλυψη του προστατευόμενου χώρου. Τέλος, πολύ μεγάλη σημασία έχει η σωστή τοποθέτηση και ρύθμιση των αισθητήρων.

## ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΔΟΜΗ ΤΩΝ ΣΥΣΤΗΜΑΤΩΝ ΑΣΦΑΛΕΙΑΣ

### 1.1 ΚΕΝΤΡΙΚΗ ΜΟΝΑΔΑ ΕΛΕΓΧΟΥ



Η κεντρική μονάδα ελέγχου είναι η καρδιά ενός συστήματος ασφαλείας, λόγω του ότι λαμβάνει και εκτελεί όλες τις εντολές που του δίνει ο χρήστης του συστήματος ασφαλείας μέσω ηλεκτρολογίου. Επίσης, όταν το σύστημα είναι ενεργοποιημένο επιτηρεί τις ζώνες ανίχνευσης συναγερμού όπου πάνω τους είναι συνδεδεμένοι οι αισθητήρες. Βέβαια, ανάλογα τον τύπο και τις ανάγκες του κάθε χώρου μπορεί να είναι συνδεδεμένος ένας αισθητήρας ή περισσότεροι ανά ζώνη. Οι αισθητήρες είναι συνδεδεμένοι στην κεντρική μονάδα συνήθως με ένα ζεύγος καλωδίων όπου μεταφέρει το σήμα της ενεργοποίησης ή μη συναγερμού σχετικά με την επιτήρηση της ζώνης στην οποία βρίσκονται. Επιπροσθέτως, αν είναι ηλεκτρονικές συσκευές τροφοδοτούνται και από τις εξόδους βοηθητικής τροφοδοσίας της μονάδας με ένα ζεύγος καλωδίων τροφοδοσίας. Όταν κάποιος ή κάποιοι αισθητήρες διεγερθούν λόγω εισβολής ή γενικά διαταραχής της κανονικής τους κατάστασης τότε στέλνεται ανάλογο σήμα στην κεντρική μονάδα, όπου το επεξεργάζεται και μεταβαίνει σε κατάσταση συναγερμού δίνοντας εντολή ενεργοποίησης σε συσκευές ηχητικής και οπτικής σήμανσης καθώς επίσης εφόσον υπάρχουν, ενεργοποιεί συσκευές τηλεειδοποίησης όπως κωδικοποιητής κλήσεων μέσω τηλεφωνικού δικτύου, εφεδρική συσκευή GSM, GPRS κ.τ.λ.

Η κεντρική μονάδα είναι μια ηλεκτρονική πλακέτα όπου είναι υπεύθυνη για όλες τις παραπάνω λειτουργίες και βρίσκεται συνήθως εντός ενός μεταλλικού ή πλαστικού πλαισίου όπου μέσα σε αυτό βρίσκεται και ο κατάλληλος μετασχηματιστής για τον υποβιβασμό της τάσης δικτύου για την τροφοδότηση της μονάδας, καθώς επίσης και μπαταρία που με απώτερο σκοπό την εφεδρική τροφοδότηση της μονάδας σε περίπτωση διακοπής ρεύματος.

## 1.2 ΠΛΗΚΤΡΟΛΟΓΙΑ



Το πληκτρολόγιο σε ένα σύστημα ασφαλείας είναι η συσκευή με την οποία ο χρήστης του συστήματος μπορεί να χειρίζεται το σύστημα, να το ενεργοποιεί ή να το απενεργοποιεί και γενικά να ρυθμίζει τις παραμέτρους λειτουργίας του. Τα πληκτρολόγια χωρίζονται ανάλογα την διάταξη που έχουν για να παρουσιάζουν τις ενδείξεις της κατάστασης του συστήματος. Έτσι ,χωρίζονται σε πληκτρολόγια με οθόνη LCD και σε πληκτρολόγια με LED. Στην πρώτη περίπτωση, ο χρήστης μπορεί να διαβάσει σε ποια κατάσταση είναι το σύστημα και την ορθή ή μη χρήση του καθώς επίσης και άλλες πληροφορίες που αφορούν τη λειτουργία του συστήματος. Στη δεύτερη περίπτωση των πληκτρολογίων με δείκτες LED, ο χρήστης μπορεί να κατανοήσει από το άναμμα ή μη των LED την κατάσταση και της λειτουργίες τους συστήματος. Γενικά, στα πληκτρολόγια με LED είναι δυσκολότερη η κατανόηση της λειτουργίας του συστήματος και γενικότερα οι πληροφορίες που παρέχονται είναι πολύ περιορισμένες. Μια άλλη ενδιαφέρουσα λειτουργία που έχουν τα πληκτρολόγια συγκεκριμένων κατασκευαστών είναι η δυνατότητα ενεργοποίησης και απενεργοποίησης του συστήματος ασφαλείας με την βοήθεια συσκευών proximity. Αυτή η λειτουργία έχει ως σκοπό να βοηθήσει άτομα που δυσκολεύονται να χειριστούν το πληκτρολόγιο (ηλικιωμένοι, άτομα με αναπηρία κ.ά.) να μπορούν να ενεργοποιούν και να απενεργοποιούν το σύστημα με την βοήθεια του proximity. Στην περίπτωση αυτή δεν θα πρέπει η συσκευή proximity να πέσει στα χέρια λάθος ατόμων διότι θα μπορούν να απενεργοποιήσουν το σύστημα κατά βούληση και χωρίς καθολου εμπόδια.

### 1.3 ΣΕΙΡΗΝΕΣ ΚΑΙ ΦΑΡΟΙ



Οι σειρήνες είναι συσκευές που ενεργοποιούνται με την κατάσταση συναγερμού στο σύστημα και σκοπό έχουν την ηχητική ειδοποίηση των ιδιοκτητών των προστατευόμενων χώρων και τη δημιουργία πανικού στους εισβολείς. Χωρίζονται σε εξωτερικές σειρήνες όπου τοποθετούνται στην εξωτερική πλευρά των κτιρίων και στόχο έχουν την ειδοποίηση όσων βρίσκονται έξω από το κτίριο και συνήθως έχουν και ένα φλας όπου εκπέμπει έντονο φώς για να εντοπίζεται και οπτικά το κτίριο που προέρχεται ο συναγερμός. Οι εσωτερικές σειρήνες τοποθετούνται εντός των κτιρίων και συνήθως ηχούν πιο δυνατά από τις εξωτερικές, με σκοπό την δημιουργία πανικού στους εισβολείς. Τον ίδιο σκοπό με το φλας των εξωτερικών σειρήνων έχουν και οι φάροι λόγο του ότι εκπέμπουν ένα έντονο φώς για να παρατηρείται για όσους δεν καταλαβαίνουν από πού προέρχεται ο ήχος της σειρήνας ή δεν τον ακούν. Για τον τελευταίο λόγο, τοποθετείται και σε εσωτερικούς χώρους και ιδιαίτερα σε βιομηχανικές μονάδες που είναι επιβαρυνμένες λόγω θορύβων και ιδιαίτερα για την ειδοποίηση ύπαρξης φωτιάς.

## ΚΕΦΑΛΑΙΟ 2

### ΑΙΣΘΗΤΗΡΕΣ ΣΥΣΤΗΜΑΤΩΝ ΑΣΦΑΛΕΙΑΣ

#### 2.1 ΜΑΓΝΗΤΙΚΟΣ ΔΙΑΚΟΠΤΗΣ



Οι μαγνητικοί διακόπτες χρησιμοποιούνται για να ανιχνεύουν το άνοιγμα μιας προστατευόμενης πόρτας ή παραθύρου. Αυτοί οι διακόπτες είναι διακόπτες επαφής και εξαρτώνται από μια άμεση φυσική ενέργεια στον αισθητήρα για να μπει σε κατάσταση συναγερμού.

Οι μαγνητικοί διακόπτες αποτελούνται από δύο μέρη: ένα δύο θέσεων μαγνητικό διακόπτη που τοποθετείται στο εσωτερικό μιας πόρτας ή ενός παραθύρου και ένα δύο θέσεων διακόπτη που λειτουργεί

μαγνητικά. Ο διακόπτης είναι σχεδιασμένος έτσι ώστε να είναι κανονικά κλειστός (normal closed) ή κανονικά ανοικτός (normal open), αυτό εξαρτάται από τον σχεδιασμό του.

Ο μαγνητικός διακόπτης αποτελείται από δύο μεταλλικά ελάσματα όπου το ένα έλκεται από το μαγνητικό πεδίο του άλλου. Όταν βρίσκεται μέσα σε αυτό, το ένα έλασμα μετακινείται και ακουμπάει με το άλλο έλασμα που παραμένει ακίνητο με αποτέλεσμα να κλείνει ο διακόπτης (για την περίπτωση normal closed). Ως συνέπεια, όταν η πόρτα ή το παράθυρο είναι κλειστά ο μαγνήτης έλκει τον διακόπτη στην φυσιολογική του κατάσταση όπου δεν έχουμε κατάσταση συναγερμού. Όταν η πόρτα ή το παράθυρο ανοίξει ο μαγνήτης αφήνει τον διακόπτη, ανοίγει η επαφή και ενεργοποιείται ο συναγερμός.

### **Εφαρμογές και παράμετροι ορθής λειτουργίας:**

Οι μαγνητικοί διακόπτες τοποθετούνται πάνω στις πόρτες ή στα παράθυρα για να ανιχνεύουν το άνοιγμα τους. Σε ιδιαίτερες περιπτώσεις θα πρέπει να χρησιμοποιούνται από κοινού με μια συσκευή ανίχνευσης κίνησης, που θα βρίσκεται μέσα στον προστατευόμενο χώρο για να ανιχνεύσει μια παρείσφρηση που γίνεται με παράκαμψη της πόρτας ή του παραθύρου που είναι επιτηρούμενο από τον συναγερμό.

Συνθήκες για μη αξιόπιστη ανίχνευση θα μπορούσε να είναι το υπερβολικό κούνημα της πόρτας, του παραθύρου που μπορούν να προκαλέσουν ψευδή ανίχνευση και θα πρέπει η πόρτα ή το παράθυρο να διορθωθεί πριν την τοποθέτηση των μαγνητικών διακοπών.

Κύριες αιτίες για ψευδείς συναγερμούς θα μπορούσαν να είναι η κακή προσαρμογή των πορτών ή των παραθύρων (που προκαλείται είτε από την παλαίωση από ακατάλληλη εγκατάσταση) και επιδεινώνεται από καιρικές συνθήκες ικανές να προκαλούν υπερβολική κίνηση των πορτών ή των παραθύρων και είναι η κύρια αιτία πρόκλησης ψευδών συναγερμών.

Συνήθη μέτρα ματαίωσης είναι η διείσδυση από την πόρτα χωρίς να μετακινηθεί ο μαγνητικός διακόπτης που θα παρακάμψει αυτόν τον αισθητήρα του συναγερμού. Επίσης ένας φορητός και ισχυρός μαγνήτης μπορεί να χρησιμοποιηθεί για να μιμηθεί τον υπάρχον μαγνήτη αφού θα επηρεάζει τον διακόπτη με το δικό του μαγνητικό πεδίο και έτσι θα μπορεί να ανοιχτεί η πόρτα ή το παράθυρο χωρίς να προκύψει συναγερμός. Η τοποθέτηση του μαγνητικού διακόπτη θα πρέπει να μην είναι εμφανής έτσι ώστε να μειώνει την πιθανότητα σε έναν διαρρήκτη να παρακάμψει το σύστημα.



## 2.2 ΙΣΟΡΡΟΠΟΣ ΜΑΓΝΗΤΙΚΟΣ ΔΙΑΚΟΠΤΗΣ



Οι ισόρροποι μαγνητικοί διακόπτες αποτελούνται από ένα συναρμολογημένο διακόπτη με ένα εσωτερικό μαγνήτη που τοποθετείται συνήθως πάνω στο σταθερό πλαίσιο της πόρτας ή του παραθύρου και ένα μαγνήτη ισορροπίας που τοποθετείται στο κινούμενο μέρος της πόρτας ή του παραθύρου.

Ο διακόπτης στις περισσότερες περιπτώσεις ισορροπεί στην ανοικτή θέση ανάμεσα στο μαγνητικό πεδίο των δύο μαγνητών. Αν το μαγνητικό πεδίο διαταραχθεί από την κίνηση του εξωτερικού μαγνήτη, ο διακόπτης θα μετακινηθεί στην κλειστή θέση. Όταν η πόρτα είναι στην κλειστή θέση, το μαγνητικό πεδίο που δημιουργείται από την πόλωση του μαγνήτη του κινούμενου φύλλου, αλληλεπιδρά με το πεδίο που δημιουργείται από το μαγνήτη του διακόπτη, και έτσι η συνολική επίδραση στον διακόπτη είναι σταθερή. Όταν η πόρτα ανοίξει, ο διακόπτης αλλάζει θέση, γίνεται ασταθής και έτσι δημιουργείται συναγερμός.

### Εφαρμογές και παράμετροι ορθής λειτουργίας:

Οι ισόρροποι μαγνητικοί διακόπτες παρέχουν ένα υψηλότερο επίπεδο ασφάλειας για παράθυρα και πόρτες από τους απλούς μαγνητικούς ή μηχανικούς διακόπτες. Οι ισόρροποι μαγνητικοί διακόπτες είναι διαθέσιμοι σε περίβλημα που μπορεί να αποτρέψει την εναλλαγή κατάστασης που προκαλείται ηλεκτρικά από μια έκρηξη σε μια περιοχή με επικινδυνότητα. Αυτοί οι διακόπτες συστήνονται για τοποθέτηση σε εύφλεκτες και επικίνδυνες περιοχές ή χώρους.

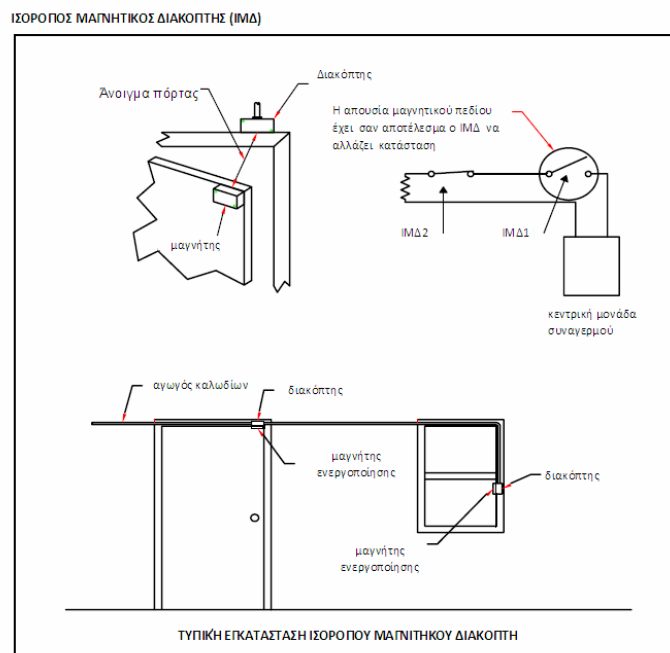
Ο ισόρροπος μαγνητικός διακόπτης θα πρέπει να τοποθετείται στο πλαίσιο της πόρτας και ο μαγνήτης ισορροπίας στο φύλλο της πόρτας. Ο διακόπτης πρέπει να ρυθμίζεται έτσι ώστε να ξεκινά ο συναγερμός μεταξύ μισής με 1 ίντσας κατά το άνοιγμα της πόρτας. Για να ενισχύσουμε την ασφάλεια, ο ισόρροπος μαγνητικός

διακόπτης (όπως ο μαγνητικός και ο μηχανικός διακόπτης) θα πρέπει να χρησιμοποιείται σε συνδυασμό με έναν ανιχνευτή κινήσεων που βρίσκεται μέσα στον φυλασσόμενο χώρο για την περίπτωση που γίνει παράκαμψη του διακόπτη.

Μια ψευδής ανίχνευση θα μπορούσε να προκληθεί από την υπερβολική κίνηση της πόρτας ή του παραθύρου που θα δημιουργήσουν συνθήκες αναξιόπιστης ανίχνευσης και θα πρέπει να εξαιρεθεί προτού να εγκατασταθούν οι διακόπτες του συστήματος συναγερμού.

Κάποιες από τις κύριες αιτίες για ψευδείς συναγερμούς θα μπορούσε να είναι η κακή προσαρμογή των πορτών και των παραθύρων, καθώς και ακατάλληλη εγκατάσταση στο φυλασσόμενο χώρο. Επίσης, ακραίες καιρικές συνθήκες οι οποίες προκαλούν υπερβολική κίνηση των πορτών και των παραθύρων προκαλούν επίσης ψευδείς συναγερμούς.

Ένα ξεχωριστό πλεονέκτημα της χρήσης του ισόρροπου μαγνητικού διακόπτη, είναι η εργοστασιακή του ικανότητά του να αντιμετωπίζει ένα κοινό μέτρο ματαίωσης των μαγνητικών αισθητήρων. Αυτό το μέτρο ματαίωσης περιλαμβάνει την τοποθέτηση ενός εξωτερικού μαγνήτη στο σημείο που βρίσκεται ο μαγνητικός διακόπτης, ούτως ώστε όταν ανοίγει η πόρτα ή το παράθυρο να μην ανοίγει ή επαφή του διακόπτη. Ο σχεδιασμός του ισόρροπου μαγνητικού διακόπτη αποκλείει σε αυτόν το μηχανισμό ματαίωσης να γίνει αποτελεσματικός.



### 2.3 ΑΝΙΧΝΕΥΤΗΣ ΘΡΑΥΣΗΣ ΚΡΥΣΤΑΛΛΩΝ(GLASSBREAK)



Ο αισθητήρας ανίχνευσης θραύσης κρυστάλλων ελέγχει το πιθανό σπάσιμο τζαμαρίας ή παραθύρων κατά την διάρκεια μιας εισβολής στον προστατευμένο χώρο. Οι αισθητήρες τοποθετούνται ως ένα μόνιμο εσωτερικό στοιχείο (σε τοίχους ή ταβάνια) με προσανατολισμό προς την επιφάνεια του προστατευόμενου τζαμιού.

Τρεις είναι οι τύποι αισθητήρων που χρησιμοποιούνται : οι ακουστικοί , οι παλμού και οι διπλής τεχνολογίας(και ακουστικός και παλμού). Ανεξάρτητα από τον αισθητήρα που χρησιμοποιείται, κανονικά η κάλυψη που παρέχει δεν ξεπερνά τα 100 τετραγωνικά πόδια της επιφάνειας του τζαμιού.

Ο αισθητήρας ανίχνευσης θραύσης κρυστάλλων ανιχνεύει για συχνότητες που ταυτίζονται με το σπάσιμο του γυαλιού. Ένας επεξεργαστής είναι υπεύθυνος για να φιλτράρει όλες της ανεπιθύμητες συχνότητες και επιτρέπει να αναλυθούν μόνο κάποιες συχνότητες συγκεκριμένων περιοχών του ακουστικού φάσματος. Ο επεξεργαστής συγκρίνει τις συχνότητες που έλαβε με συγκεκριμένες τιμές συχνοτήτων που είναι καταγεγραμμένες ως συχνότητες που συνδέονται με το σπάσιμο του γυαλιού και αν οι λαμβανόμενες συχνότητες ταιριάζουν με τα χαρακτηριστικά αυτών του σπασμένου γυαλιού, τότε έχουμε την δημιουργία κατάστασης συναγερμού.

Όπως αναφέραμε και πριν, υπάρχουν 3 βασικοί τύποι αισθητήρων ανίχνευσης θραύσης κρυστάλλων: ο ακουστικός , ο παλμού και ο αισθητήρας που είναι συνδυασμός αυτών των δύο και έχει ως αποτέλεσμα έναν αισθητήρα διπλής τεχνολογίας ( ακουστικό και παλμού).

Οι ακουστικοί αισθητήρες αφουγκράζονται και ανιχνεύουν , την υψηλή συχνότητα που συνήθως δημιουργείται ως αποτέλεσμα μιας θραύσης τζαμιού. Εξαιτίας της θραύσης, δημιουργούνται υψηλές συχνότητες που προκαλούνται από το σπάσιμο

του γυαλιού και απομακρύνονται από το σημείο της κρούσης προς της εξωτερικές άκρες της επιφάνεια του γυαλιού. Αυτές οι κρούσεις διεγείρουν τον επεξεργαστή του ακουστικού αισθητήρα ο οποίος περνάει της συχνότητες μέσα από ένα φίλτρο , συγκρίνει τις συχνότητες αν ταιριάζουν με τα σήματα της θραύσης των κρυστάλλων και αν αυτό ισχύει ενεργοποιείται ο συναγερμός.

Γενικά, δύο τύποι αισθητήρων παλμού χρησιμοποιούνται και είναι ο πιεζοηλεκτρικός-ηλεκτρικός και ο πιεζοηλεκτρικός-μη ηλεκτρικός. Κυρίως βέβαια χρησιμοποιούνται οι πιεζοηλεκτρικοί-ηλεκτρικοί ώστε να αντιλαμβάνονται την συχνότητα των 5 khz. Απ' την άλλη, ένας μη ηλεκτρικός πιεζοηλεκτρικός αισθητήρας δεν έχει καμία παρουσία ρεύματος μέχρι τη στιγμή που το πιεζοηλεκτρικό στοιχείο κάμπτεται από τη διέγερση του σήματος των 5 khz. Ο μη ηλεκτρικός πιεζοηλεκτρικός τύπος αισθητήρα όμως, έχει πολύ λιγότερους ψευδούς συναγερμούς.

Στους διπλής τεχνολογίας αισθητήρες, μια ακουστική συσκευή συνδέεται με μια συσκευή παλμού. Αυτός ο συνδυασμός χρησιμοποιεί της συμπληρωματικές δυνατότητες των δύο αυτών συσκευών και παρουσιάζει ένα χαμηλό ρυθμό εμφάνισης εφαιμένων συναγερμών. Τα δυο αισθητήρια στοιχεία βρίσκονται εντός της ίδιας συσκευασίας και συνδέονται ηλεκτρονικά δια μέσου λογικής συνάρτησης «AND».

Το ακουστικό μέρος του αισθητήρα χρησιμοποιεί ένα μικρόφωνο για να ανιχνεύει συχνότητες που σχετίζονται με τη θραύση των τζαμιών. Ένας επεξεργαστής φιλτράρει και απορρίπτει όλες τις ανεπιθύμητες συχνότητες και επιτρέπει μόνο σε ορισμένες περιοχές συχνοτήτων να προχωρήσουν και να αναλυθούν. Μόλις ο επεξεργαστής λάβει αυτές τις συχνότητες, τις συγκρίνει για να διαπιστωθεί αν το σήμα ταιριάζει με τις χαρακτηριστικές συχνότητες του σπασίματος του τζαμιού και τότε ένα σήμα στέλνεται στην πύλη.

Το μέρος του αισθητήρα παλμού αισθάνεται τη συχνότητα των 5Khz του παλμικού κύματος που δημιουργείται όταν ένα τζάμι σπάσει. Όταν ο επεξεργαστής ανιχνεύσει αυτόν τον παλμό, τότε στέλνεται ένα σήμα στην πύλη AND. Τη στιγμή που η πύλη AND λάβει και τα δύο σήματα και είναι θετικά στην ανίχνευση συναγερμού, εκείνη δημιουργείται συναγερμός.

Ένα πολύ μεγάλο και πολύ σημαντικό πλεονέκτημα αυτού του αισθητήρα είναι η ενσωμάτωση δύο τεχνολογιών ανίχνευσης θραύσης τζαμιών στον αισθητήρα. Αυτό μειώνει σημαντικά τους ψευδείς συναγερμούς από θορύβους όπως ο θόρυβος από Σάλλας Παντελής – Λαμπρινός Κωνσταντίνος

παρεμβολές ραδιοκυμάτων ή το θόρυβο που δημιουργείται από διάφορες ηλεκτρικές συσκευές που μπορεί να βρίσκονται στον χώρο όλο το 24ωρο.

### **Εφαρμογές και παράμετροι ορθής λειτουργίας:**

Οι εφαρμογές των ακουστικών αισθητήρων είναι άμεσα συνδεδεμένες με τις προδιαγραφές του κατασκευαστή των ακουστικών αισθητήρων και ορίζουν σε ποιο σημείο μπορούν να τοποθετηθούν σε παράθυρα, πλαίσια παραθύρων, τοίχους ή ταβάνια. Αν τοποθετηθεί στο τζάμι, ο αισθητήρας θα πρέπει να τοποθετηθεί στην γωνία, περίπου 2 ίντσες από την άκρη του πλαισίου. Αν τοποθετηθεί σε τοίχο ή σε ταβάνι, ο αισθητήρας θα πρέπει τοποθετηθεί απέναντι από το παράθυρο.

Ο αισθητήρας ανίχνευσης θραύσης τζαμιών θα πρέπει να χρησιμοποιείται σε συνδυασμό με διακόπτες (μαγνητικούς διακόπτες, ισόρροπους μαγνητικούς διακόπτες) για την περίπτωση παραβίασης ανοίγοντας το παράθυρο χωρίς να το σπάσουν.

Ένας ογκομετρικός ανιχνευτής κίνησης θα πρέπει επίσης να ενσωματωθεί στην προστατευόμενη εσωτερική ζώνη, για να μπορεί να ανιχνεύει την παρείσφρηση από μια δίοδο εκτός του παραθύρου. Ο ανιχνευτής κίνησης θα πρέπει να βρίσκεται σε κατάλληλο σημείο και γωνία, το οποίο να του επιτρέπει να βλέπει προς το παράθυρο, έτσι ώστε να καλύπτει τη μέγιστη ικανότητα ανίχνευσης.

Παρόλοφυσικά που δε συστήνεται, ο αισθητήρας μπορεί να τοποθετηθεί πάνω στο παράθυρο. Σε αυτή την περίπτωση, η κόλλα στήριξης θα πρέπει να αντέχει την έκθεση σε υψηλές θερμοκρασίες το καλοκαίρι, χαμηλές θερμοκρασίες που προκαλεί το κρύο το χειμώνα, καθώς και την υγρασία που μπορεί να μαζεύει το παράθυρο από τις βροχές. Αξιοσημείωτο είναι ότι η θερμοκρασία στα τζάμια των παραθύρων μπορεί να αναπτύσσουν θερμοκρασία έως 66C το καλοκαίρι και η θερμοκρασία να πέφτει το χειμώνα έως -30C, επομένως είναι σημαντικό η κόλλα να πληροί συγκεκριμένες προδιαγραφές.

Παρόλο που με μια ακατάλληλη εφαρμογή του εύρους του αισθητήρα στο παράθυρο, μπορεί ο αισθητήρας να είναι εκτός της αποτελεσματικής περιοχής ανίχνευσης, η πιο χαρακτηριστική ανεπάρκεια εμφανίζεται όταν τα ακουστικά χαρακτηριστικά του δωματίου έρχονται σε σύγκρουση με τις προδιαγραφές απόδοσης του αισθητήρα. Έτσι, μπορούμε να έχουμε ακουστικά «εξασθενημένα» δωμάτια (π.χ. καλυμμένα με ταπετσαρία και κουρτίνες στα παράθυρα) που

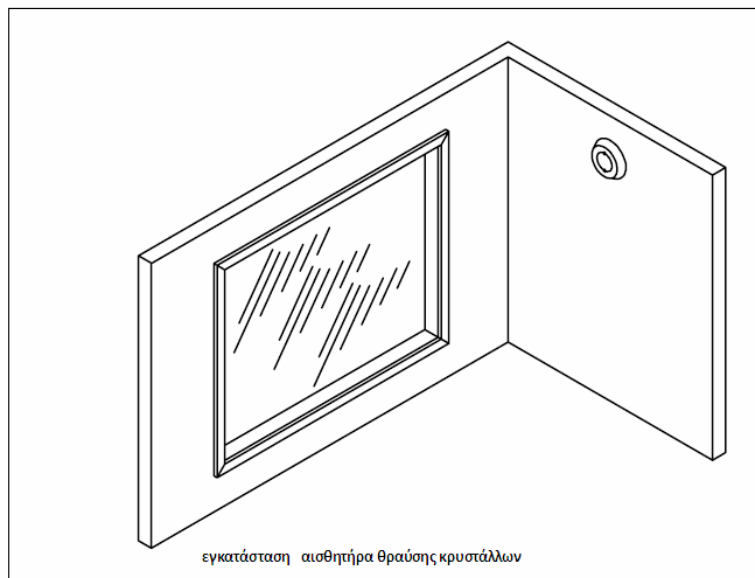
απορροφούν τις δονήσεις, ή που αλλάζουν τα ακουστικά χαρακτηριστικά ενός δωματίου σε «ενισχυμένα»(π.χ. προσθέτοντας παραθυρόφυλλα , στόρια, κουρτίνες χαλιά) όταν αυτά τα αντικείμενα προστίθενται αφού έχει ρυθμιστεί ο αισθητήρας , μπορούν να προκαλέσουν ανεπάρκεια ανίχνευσης στον αισθητήρα.

Οπότε, για προληπτικούς λόγους όλα τα παράθυρα θα πρέπει να ελεγχθούν για ρωγμές πριν την εγκατάσταση του αισθητήρα για να βεβαιωθεί ότι ένα σήμα κατάλληλης συχνότητας θα δημιουργηθεί αν σπάσει το παράθυρο.

Αιτίες ψευδών συναγερμών θα μπορούσαν να είναι η ακατάλληλη ρύθμιση ή εγκατάσταση ενός ακουστικού ανιχνευτή θραύσης κρυστάλλων που μπορεί να προκαλέσει ψευδείς συναγερμούς. Επιπλέον παρεμβολές και η επίδραση έντονων θορύβων μπορούν να προκαλέσουν εσφαλμένους συναγερμούς. Επίσης, ακατάλληλη τοποθέτηση του αισθητήρα ή ήχοι που προέρχονται από το πίσω μέρος του αισθητήρα (σε σχέση με τον προσανατολισμό του) όπως από μηχανήματα γραφείου, καθαριότητας ή βιομηχανικά μηχανήματα , μπορούν να δημιουργήσουν ήχους στην συχνότητα ανίχνευσης του αισθητήρα.

Κάποια από τα συνήθη μέτρα ματαίωσης είναι η κοπή και δημιουργία ενός ανοίγματος στο παράθυρο ή αφαίρεση ενός παραθύρου (με ή χωρίς ένα αισθητήρα τοποθετημένο πάνω σε αυτό), που μπορούν να παρακάμψουν τον αισθητήρα. Η συχνότητα επίσης που προκαλείται από το σπάσιμο μπορεί να αλλοιωθεί με τον περιορισμό του ήχου καθώς σπάει το τζάμι, κάτι που μειώνει την πιθανότητα σωστής λήψης συχνότητας από τον αισθητήρα.

αισθητήρας θραύσης κρυστάλλων



## 2.4 ΦΩΤΗΛΕΚΤΡΙΚΗ ΔΕΣΜΗ



Οι αισθητήρες φωτοηλεκτρικής δέσμης εκπέμπουν μια δέσμη υπέρυθρου φωτός σε ένα απομακρυσμένο δέκτη, δημιουργώντας έναν –θα μπορούσαμε να πούμε, «ηλεκτρονικό φράκτη». Αυτοί οι αισθητήρες συχνά χρησιμοποιούνται για την κάλυψη πορτών , διαδρόμων, εξωτερικών χώρων ενεργώντας ουσιαστικά σαν ένα πλέγμα προστασίας. Τη στιγμή που η δέσμη διακοπεί, δημιουργείται ένα σήμα συναγερμού.

Οι αισθητήρες φωτοηλεκτρικής δέσμης αποτελούνται από δύο μέρη : το πομπό και το δέκτη. Ο πομπός χρησιμοποιεί μια δίοδο LED σαν πηγή φωτός και εκπέμπει μια υπέρυθη δέσμη φωτός στο δέκτη. Ο δέκτης, αποτελείται από ένα φωτοηλεκτρικό στοιχείο το οποίο ανιχνεύει την παρουσία της δέσμης. Όταν η φωτοηλεκτρική δέσμη δεν λαμβάνεται το λιγότερο κατά 90% από την στάθμη σήματος που εκπέμπεται και διακόπτεται για ένα σύντομο χρονικό διάστημα των 75ms, (χρόνος που ένας εισβολέας διασχίζει την δέσμη), τότε δημιουργείται ένα σήμα συναγερμού.

Η δέσμη διαμορφώνεται σε μια πολύ υψηλή συχνότητα η οποία εναλλάσσεται πάνω από 1000 φορές το δευτερόλεπτο και με τέτοιο τρόπο, έτσι ώστε ο δέκτης να προστατευτεί ενάντια σε προσπάθειες παράκαμψης χρησιμοποιώντας μια υποκατάστατη πηγή φωτός.

Για να παρακαμφθεί ο αισθητήρας, η γωνία της δέσμης και η συχνότητα διαμόρφωσης θα πρέπει να ταιριάζουν τέλεια. Επίσης, οι εκδόσεις των συσκευών των δεσμών μπορεί να είναι με διπλή ,τριπλή τετραπλή δέσμη κτλ μεταξύ πομπού και δέκτη έτσι ώστε να μειώνεται η πιθανότητα εσφαλμένων συναγερμών, καθώς και να αυξάνεται η αξιοπιστία του συστήματος .

### **Εφαρμογές και παράμετροι ορθής λειτουργίας:**

Ο αισθητήρας αυτός τοποθετείται για να προστατέψει εισόδους, διαδρόμους, μεγάλες επιφάνειες τοίχων, εξωτερικούς χώρους κτλ. Ο πομπός και ο δέκτης μπορούν να έχουν απόσταση έως και πάνω από 1000 πόδια και να εξακολουθούν να παρέχουν επαρκή κάλυψη. Μία φωτοηλεκτρική δέσμη φωτός δεν επηρεάζεται από αλλαγές στην θερμική ακτινοβολία από λάμπες φθορισμού ή από ραδιοσυχνότητες. Ο φωτοηλεκτρικός αισθητήρας παρουσιάζει υψηλή πιθανότητα ανίχνευσης και χαμηλό ρυθμό εμφάνισης ψευδών συναγερμών. Η διαδρομή της δέσμης μπορεί να μεταβληθεί χρησιμοποιώντας κάτοπτρα, για να δημιουργηθεί ένα λιγότερο προβλέσιμο φράγμα ανίχνευσης, ωστόσο η χρήση των κατόπτρων μειώνει την ισχύ του σήματος της δέσμης και περιορίζει το αποτελεσματικό μήκος της δέσμης. Ένα κοινό πρόβλημα με τα κάτοπτρα είναι ότι συχνά χτυπιούνται κατά τύχη και βγαίνουν εκτός ευθυγράμμισης, δημιουργώντας την ανάγκη να ρυθμίζονται και να βαθμονομούνται ανά τακτά χρονικά διαστήματα.

Οτιδήποτε μπορεί να διαταράσσει την εκπομπή του φωτός, μπορεί να επηρεάσει την αξιοπιστία ανίχνευσης του αισθητήρα. Παράγοντες όπως η ομίχλη, ο καπνός, η υδρονέφωση, η σκόνη και σωματίδια που ανακλούν το φως, έχουν ως αποτέλεσμα τα σωματίδια φωτός να διαθλώνται ή να διασπείρονται. Αν από τους παραπάνω λόγους μπορεί να δημιουργείται μια μείωση 10% ή παραπάνω, τότε έχουμε την δημιουργία σήματος συναγερμού. Ακραίες διακυμάνσεις στο φωτισμό που μπορεί να προέρχονται από το φόντο του αισθητήρα σε σχέση με την θέση του αισθητήρα ή ακόμα και διακυμάνσεις του ηλιακού φωτός, μπορούν επίσης να μειώσουν την ευαισθησία.

Οποιοδήποτε αντικείμενο οποίο μπορεί να διακόψει την δέσμη, όπως πουλιά, ζώα φύλλα ή χαρτιά που τα μετακινεί ο αέρας θα διακόψουν την εκπεμπόμενη δέσμη και θα έχουν ως αποτέλεσμα τη δημιουργία κατάστασης συναγερμού. Επιπροσθέτως, η κακή ευθυγράμμιση του πομπού και του δέκτη ή των κατόπτρων μπορεί να προκαλέσουν κάποιο εσφαλμένο συναγερμό. Οι καθρέπτες επίσης μπορούν να συλλέγουν σκόνη, και αυτό να έχει αποτέλεσμα τη διάθλαση της ανακλώμενης δέσμης.



Πατώντας πάνω από τον αισθητήρα ή περνώντας κάτω από την διαδρομή της ακτίνας, μπορεί να προσπελαστεί ο αισθητήρας. Ωστόσο, τα κάτοπτρα μπορούν να χρησιμοποιηθούν στην αντιμετώπιση αυτής της ευπάθειας δημιουργώντας ένα φράγμα δέσμης με σχήμα πολλαπλών διαρομών ζικ-ζακ.

## 2.5 ΑΙΣΘΗΤΗΡΑΣ ΗΧΟΥ



Οι αισθητήρες ήχου ανιχνεύουν ήχους που δημιουργούνται από την είσοδο ενός εισβολέα εντός του προστατευμένου χώρου και συνήθως χρησιμοποιούνται σε εσωτερικές εφαρμογές, αλλά σε μερικές περιπτώσεις και σε εξωτερικούς χώρους.

Ο αισθητήρας ήχου αποτελείται από δύο αισθητήρες λήψης που τοποθετούνται στους τοίχους ή στις οροφές των παρακολουθούμενων περιοχών, και 1 μονάδα ενίσχυσης η οποία περιλαμβάνεται στο κύκλωμα επεξεργασίας σήματος του αισθητήρα. Οι μονάδες λήψης ήχου είναι μικρόφωνα που αφουγκράζονται θορύβους. Αυτά τα μικρόφωνα συλλέγουν ήχους για ανάλυση από την μονάδα επεξεργασίας, η οποία μπορεί να εξακριβώσει αν ένα όριο θορύβου, είναι χαρακτηριστικό μιας ενέργειας εισβολής. Αν μια συγκεκριμένη τιμή θορύβου ανιχνευθεί από μια ελεγχόμενη περιοχή εντός μιας συγκεκριμένης χρονικής περιόδου, τότε δημιουργείται ένα σήμα συναγερμού.

### **Εφαρμογές και παράμετροι ορθής λειτουργίας:**

Οι αισθητήρες ήχου θα πρέπει να τοποθετούνται σε περιοχές όπου ο θόρυβος κατά την διάρκεια μιας εισβολής αναμένεται να είναι μεγαλύτερος από τον θόρυβο υπό κανονικές συνθήκες. Αν υπάρχει παρασιτικός θόρυβος και αν η ρύθμιση του αισθητήρα δεν είναι σε θέση να τον αντισταθμίσει, Σάλλας Παντελής – Λαμπρινός Κωνσταντίνος

τότε το μικρόφωνο μπορεί να μην είναι σε θέση να ανιχνεύσει τον θόρυβο μιας εισβολής. Στην παρουσία υπερβολικού θορύβου ο αισθητήρας ήχου δεν θα πρέπει να συμπεριλαμβάνεται στην προστασία του χώρου.

Κανονικά οι αισθητήρες ήχου χρησιμοποιούνται σε συνδυασμό με άλλους αισθητήρες (όπως αισθητήρες υπέρυθρων , υπέρηχων, μικροκυμάτων κτλ) για να παρέχουν καλύτερη πιθανότητα ανίχνευσης.

Δεδομένου ότι ένας ακουστικός αισθητήρας είναι απρόσβλητος από τις αλλαγές στο θερμικό περιβάλλον και τα φώτα φθορισμού δεν έχουν καμία επίδραση στα χαρακτηριστικά ανίχνευσης του αισθητήρα, η χρήση του αισθητήρα με ένα αισθητήρα ανίχνευσης κίνησης θερμικής λήψης μπορεί να παρέχει μαζί ηχητική και θερμική κάλυψη κατά των εισβολέων στον προστατευόμενο χώρο.

Συνθήκες για μη αξιόπιστη ανίχνευση θα μπορούσαν να είναι οι μη αποτελεσματικές ρυθμίσεις ευαισθησίας του αισθητήρα αλλά και ο θόρυβος από εξωγενείς πηγές θορύβου, όπως εξοπλισμός γραφείου, λέβητες και κλιματιστικά που βρίσκονται μόνιμα στον χώρο όλο το 24ωρο.

Αίτια ψευδών συναγερμών μπορεί να είναι ο υπερβολικός παρασιτικός θόρυβος, όπως ο θόρυβος από αεροπλάνα, τραίνα, ή καιρικά φαινόμενα μπορεί να προκαλέσουν σημαντικό θόρυβο έτσι ώστε να δημιουργηθεί συναγερμός. Με την παρουσία αυτών των παραγόντων, ιδιαίτερη προσοχή θα πρέπει να δοθεί για να διαπιστωθεί κατά πόσο ο αισθητήρας είναι κατάλληλος για τη συγκεκριμένη εφαρμογή.

Ένας εισβολέας που κάνει μια ήσυχη και αργή είσοδο σκόπιμα, και λαμβάνει μέτρα για να καλύπτει τους θορύβους κατά την κίνηση του και επιπρόσθετα αφήνει ένα διάστημα καθυστέρησης χρόνου ανάμεσα στις ενέργειες του που δημιουργούν θόρυβο, μπορούν να τον βοηθήσουν να αποφύγει την ανίχνευση από ακουστικό αισθητήρα.

## 2.6 ΥΠΕΡΥΘΡΟΣ ΠΑΘΗΤΙΚΟΣ ΑΙΣΘΗΤΗΡΑΣ (PIR)



Όπως δηλώνει και η ονομασία τους, οι παθητικοί ανιχνευτές υπέρυθρων ονομάζονται παθητικοί γιατί δεν εκπέμπουν κάποιο σήμα, αλλά απλά λαμβάνουν την υπέρυθρη ακτινοβολία του χώρου. Οι παθητικοί υπέρυθροι αισθητήρες (μερικές φορές λέγονται και πυροηλεκτρικοί αισθητήρες) έχουν την ιδιότητα το αισθητήριο στοιχείο τους συνήθως να διαιρείτε σε πολλούς τομείς. Η ανίχνευση κίνησης προκαλείται όταν μια πηγή θερμότητας (π.χ. η θερμότητα του ανθρώπινου σώματος) διασχίσει δύο γειτονικά όρια του τομέα ή διασχίσει το ίδιο όριο δύο φορές σε ένα συγκεκριμένο χρονικό διάστημα.

Οι παθητικοί υπέρυθροι αισθητήρες ανιχνεύουν την ηλεκτρομαγνητική ακτινοβολία που παράγεται από τη θερμοκρασία των σωμάτων, που ονομάζεται και υπέρυθρη ακτινοβολία και βρίσκεται χαμηλότερα του οπτικού φάσματος. Οι παθητικοί υπέρυθροι αισθητήρες δεν μετρούν το ποσό της λαμβανόμενης υπέρυθρης ενέργειας ανά δευτερόλεπτο, αλλά τις αλλαγές της θερμικής ακτινοβολίας. Οι αισθητήρες αυτοί ανιχνεύουν τις «θερμές» απεικονίσεις αντιλαμβανόμενοι τη διαφορά που υπάρχει στην «θερμή» λαμβανόμενη εικόνα και στο «ψυχρό» φόντο κάτι που απορρέει από τη λαμβανόμενη υπέρυθρη ακτινοβολία του χώρου.

Το μήκος κύματος της υπέρυθρης ακτινοβολίας μετράται σε μικρόμετρα, με την παραγόμενη υπέρυθρη ακτινοβολία να βρίσκεται στην περιοχή μεταξύ 7 έως 14 μικρομέτρων. Οι περισσότεροι υπέρυθροι παθητικοί αισθητήρες επικεντρώνονται σε αυτό το ιδιαίτερα στενό εύρος ζώνης. Προκειμένου να αποφύγουμε τις θερμικές αποκλίσεις λόγω των περιβαλλοντικών αλλαγών, χρησιμοποιείται ένα κύκλωμα μέτρησης του ρυθμού αλλαγής της υπέρυθρης ακτινοβολίας που εκπέμπεται από την

θερμότητα των σωμάτων ή ένα κύκλωμα σύγκρισης της διαφοροποίησης της ακτινοβολίας των επιμέρους ζωνών των τομέων του αισθητήρα .

Το κύκλωμα επεξεργασίας αξιολογεί το μετρούμενο ρυθμό αλλαγής της υπέρυθρης ακτινοβολίας στο οπτικό πεδίο του αισθητήρα και το συγκρίνει με το αρχικό μοτίβο της θερμοκρασίας στον χώρο. Η κίνηση ενός εισβολέα προκαλεί ένα πολύ γρήγορο ρυθμό αλλαγής ενώ η σταδιακή διακύμανση της θερμοκρασίας από την άλλη, προκαλεί ένα πολύ αργό ρυθμό αλλαγής της λαμβανόμενης υπέρυθρης ακτινοβολίας. Στην τεχνική της μέτρησης διαφοροποίησης ελέγχεται στην πραγματικότητα αν ένας εισβολέας ή ένα αντικείμενο είναι στην πραγματικότητα εκεί ή δεν είναι. Διαφορετικοί τομείς του αισθητήρα δημιουργούν επιμέρους ζώνες ανίχνευσης και ανάλογα από τον χώρο τα λαμβανόμενα σήματα παρουσιάζουν διαφοροποιήσεις μεταξύ τους σε σχέση με τις ζώνες από τις οποίες εκπέμπονται. Αυτές τις ζώνες χρησιμοποιεί ο αισθητήρας συγκρίνοντας τις για την ανίχνευση κίνησης. Ένας ακάλυπτος άνθρωπος που εισέρχεται στο οπτικό πεδίο του αισθητήρα με μια συνήθη ταχύτητα (με μια ταχύτητα βαδίσματος ή και παραπάνω) υπό κανονικές συνθήκες, θα εκπέμψει κάποια σήματα τα οποία ανιχνεύονται.

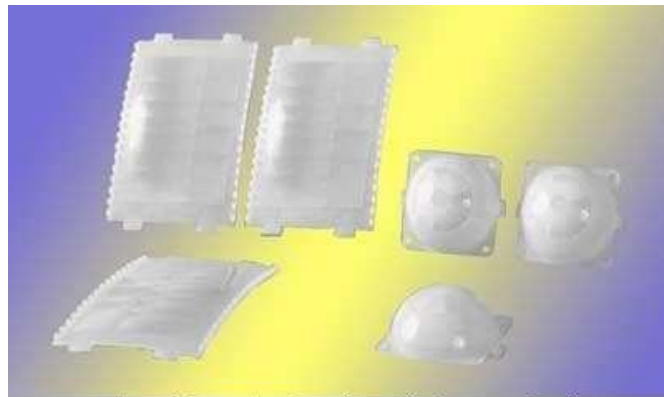
Έτσι, η αλλαγή της ακτινοβολίας λαμβάνεται από τους φακούς του αισθητήρα και αν η ποσότητα της λαμβανόμενης ακτινοβολίας ξεπερνά μια προκαθορισμένη τιμή, ο θερμικός αισθητήρας παράγει ένα ηλεκτρικό σήμα το οποίο στέλνεται στο κύκλωμα επεξεργασίας για αξιολόγηση και την πιθανή δημιουργία συναγερμού.

Το μήκος κύματος υποδιαιρείται σε 2 κύριες κατηγορίες: η 1 καλύπτει την περιοχή ακτινοβολίας που εκπέμπεται από τηλεχειριστήρια τηλεοράσεων και άλλων συσκευών και η άλλη την περιοχή της υπέρυθρης ακτινοβολίας που εκπέμπεται από την θερμοκρασία του ανθρώπινου σώματος που σε αυτήν την περιοχή λειτουργούν εφαρμογές των συστημάτων ασφαλείας.

Οι αρχές της οπτικής και των ανακλάσεων παίζουν ένα πολύ σημαντικό ρόλο στον σχεδιασμό και τη λειτουργία των παθητικών υπέρυθρων αισθητήρων. Λόγω της ανάγκης της ακριβούς εστίασης της θερμικής ακτινοβολίας, η αντανάκλαση/εστίαση της θερμικής ακτινοβολίας γίνεται με δύο τρόπους : την εστίαση μέσω αντανάκλασης και την μέθοδο φακών Fresnel.

Στην εστίαση μέσω ανάκλασης τα κύματα θερμικής ακτινοβολίας ανακλώνται μέσω ενός κούλου κατόπτρου και κατευθύνονται στο στοιχείο ανίχνευσης. Από την άλλη, με

του φακούς Fresnel η θερμική ακτινοβολία κατευθύνονται απευθείας στον αισθητήρα. Όμως, και στις 2 περιπτώσεις υπάρχει μια μικρή απώλεια ενέργειας αλλά και οι δύο αισθητήρες λειτουργούν ικανοποιητικά.



Φακοί Fresnel παθητικών υπέρυθρων αισθητήρων



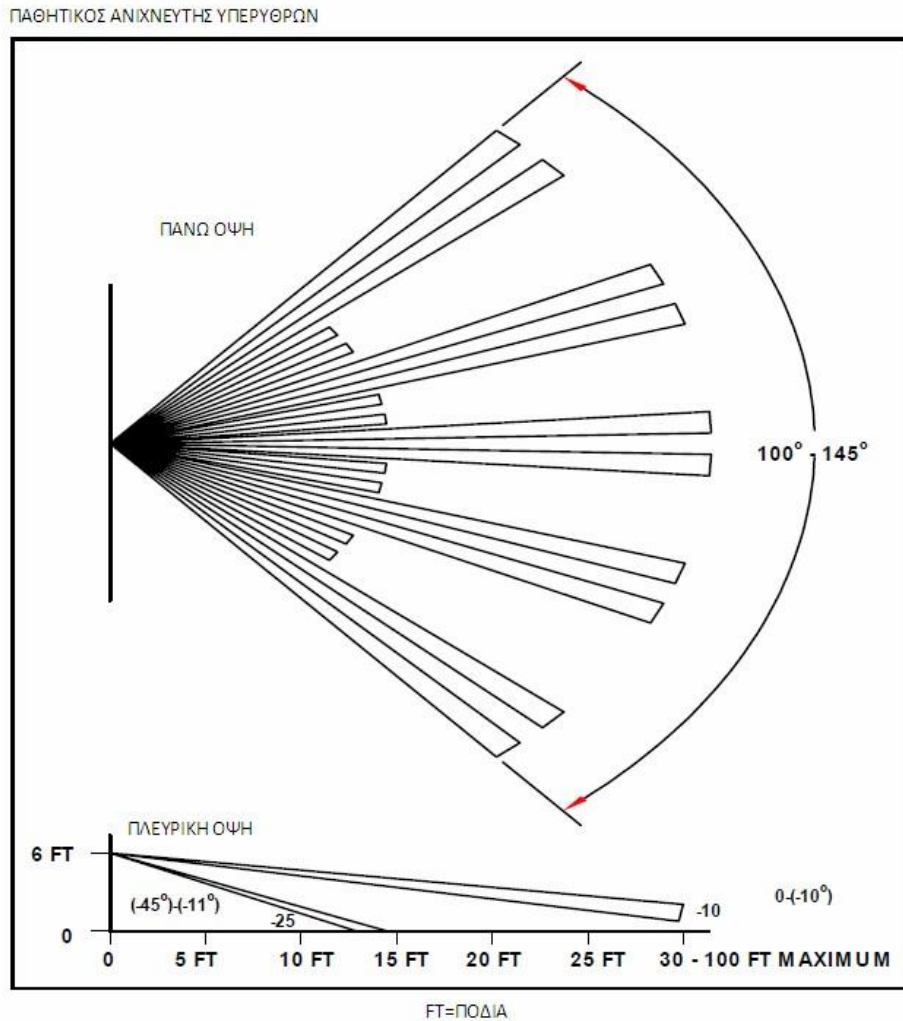
Κοιλότητα ανάκλασης σε παθητικό υπέρυθρο αισθητήρα

### Εφαρμογές και παράμετροι ορθής λειτουργίας:

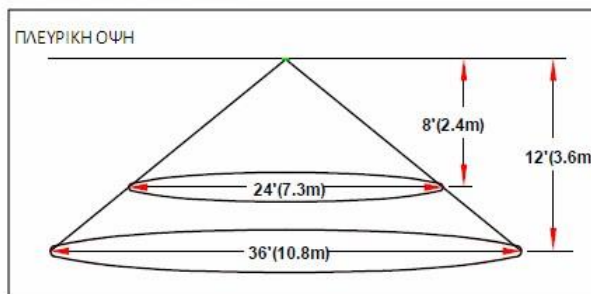
Οι παθητικοί αισθητήρες υπέρυθρων εγκαθίστανται σε τοίχους ή οροφές και η περιοχή ανίχνευση τους θα πρέπει να καλύπτει τις ζώνες εκείνες που είναι πιθανό να υπάρξει εισβολή.

Η ζώνη ανίχνευσης μπορεί να έχει το σχήμα του παρακάτω διαγράμματος και να διαφέρει ανάλογα τον αισθητήρα. Όπως βλέπουμε η περιοχή κάλυψης που ξεκινά από τον αισθητήρα είναι αρχικά στενή και ανοίγει όσο απομακρύνεται από τον Σάλλας Παντελής – Λαμπρινός Κωνσταντίνος

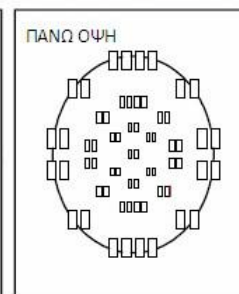
αισθητήρα. Επίσης, στο διάγραμμα μπορούμε και παρατηρούμε ότι υπάρχουν σημεία που φωτίζονται αλλά και νεκρά σημεία σκίασης.



ΣΥΝΗΘΗ ΚΑΛΥΨΗ ΠΑΘΗΤΙΚΟΥ ΑΙΣΘΗΤΗΡΑ ΥΠΕΡΥΘΡΩΝ ΤΟΠΟΘΕΤΗΜΕΝΟΥ ΣΕ ΟΡΟΦΗ



ΔΙΑΓΡΑΜΜΑ ΚΑΛΥΨΗΣ ΠΑΤΩΜΑΤΟΣ ΜΕ ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΔΕΣΜΩΝ ΤΩΝ ΕΠΙΜΕΡΟΥΣ ΤΟΜΕΩΝ ΤΟΥ ΑΙΣΘΗΤΗΡΑ



Έτσι, λαμβάνοντας υπόψη την περιοχή κάλυψης του αισθητήρα που αναγράφονται στα χαρακτηριστικά του, κατά την εγκατάσταση πρέπει να εστιάζεται ο αισθητήρας σε περιοχές που χρειάζονται προστασία και να αγνοηθούν οι άλλες περιοχές για την αποφυγή ψευδών συναγερωμών. Οι παθητικοί αισθητήρες υπερύθρων θεωρητικά έχουν διάγραμμα κάλυψης 360 μοιρών.

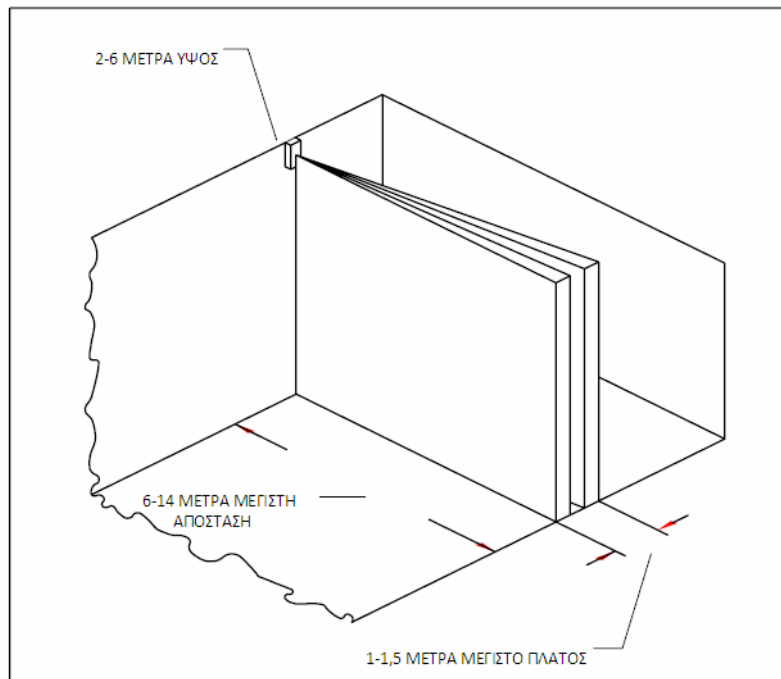
Η εναλλαγή διαφορετικών φακών και ανακλαστήρων/κατόπτρων επιτρέπει την αλλαγή του οπτικού πεδίου των αισθητήρων και κατ' επέκταση το διάγραμμα κάλυψης των επιτηρούμενων περιοχών. Ο σχεδιασμός των παθητικών ανιχνευτών υπερύθρων περιλαμβάνει και φακούς «κουρτίνας» με χαρακτηριστικά που παρέχουν ένα πλήρη φράκτη κάλυψης των επιτηρούμενων ζωνών εξαλείφοντας όσο το δυνατόν περισσότερο τα νεκρά σημεία.

Αισθητήρες με αυτά τα χαρακτηριστικά είναι ιδανικοί για την προστασία σημείων εισόδου ή διαδρόμων.

Επειδή οι παθητικοί αισθητήρες ανίχνευσης υπερύθρων ψάχνουν για την ύπαρξη θερμικής ακτινοβολίας έναντι ενός ψυχρότερου φόντου, η ανίχνευση βασίζεται στην ύπαρξη θερμοκρασίας. Όσο η θερμοκρασία του περιβάλλοντος χώρου προσεγγίζει την θερμοκρασία του σώματος του εισβολέα, η ευαισθησία του αισθητήρα μειώνεται. Αυτό ισχύει ιδιαίτερα για περιβάλλοντα που η θερμοκρασία κυμαίνεται μεταξύ 27 και 38 βαθμών Κελσίου. Θεωρητικά, αν ένα άτομο έχει την ίδια θερμοκρασία με τον περιβάλλοντα χώρο δεν θα τον αντιληφθεί ο αισθητήρας. Για τον λόγο αυτό ένας αισθητήρας διαφορετικής τεχνολογίας θα πρέπει να χρησιμοποιείται σε συνδυασμό με των παθητικό αισθητήρα ανίχνευσης υπερύθρων.

Αιτία ψευδών συναγερμών θα μπορούσε να είναι η εκπεμπόμενη θερμότητα από μικρά ζώα ή τρωκτικά που μπορεί να προκαλέσει ψευδείς συναγερμούς. Κατά τον χρόνο θέρμανσης καλοριφέρ, φούρνων ή βραστήρων μπορεί επίσης να προκληθούν ψευδείς συναγερμοί αν οι παραπάνω συσκευές βρίσκονται εντός της περιοχής κάλυψης του αισθητήρα. Επιπλέον, αν οι αισθητήρες δεν έχουν σχεδιαστεί με την ικανότητα να φιλτράρουν το ορατό φως, μπορεί να έχουμε ψευδείς συναγερμούς από φώτα, προβολείς ή άλλες πηγές φωτός. Αν και η υπέρυθη ακτινοβολία φιλτράρεται από το κοινό γυαλί των παραθύρων, αντικείμενα εντός του χώρου που βρίσκεται ο αισθητήρας μπορεί σταδιακά να θερμανθούν και να εκπέμπουν υπέρυθη ακτινοβολία ή αυτή η ακτινοβολία να ανακλάτε από άλλα αντικείμενα προς την κατεύθυνση του αισθητήρα. Αν αυτή ακτινοβολία διαταραχθεί π.χ. λόγω της κίνησης νεφών που θα διακόψουν την ηλιακή ακτινοβολία, μπορεί να δημιουργήσουν μια τυχαία κατάσταση διαφοροποίησης των λαμβανόμενων σημάτων και να έχουμε την δημιουργία ψευδών συναγερμών.

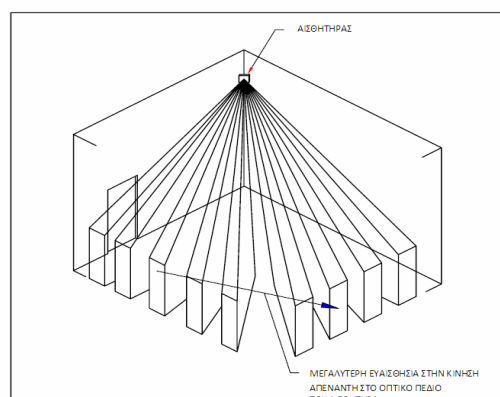
ΣΗΜΗΘΕΣ ΔΙΑΓΡΑΜΜΑ ΚΑΛΥΨΗΣ ΠΛΑΗΤΙΚΟΥ ΑΝΙΧΝΕΥΤΗ ΥΠΕΡΥΘΡΩΝ «ΚΟΥΡΤΙΝΑΣ»



Σκίαση , απόκρυψη ή συγκάλυψη της πηγής θερμότητας ενός εισβολέα εντός της περιοχής κάλυψης του αισθητήρα μειώνει την πιθανότητα ανίχνευσης λόγω της μειωμένης ακτινοβολήσης θερμότητας. Αυτό έχει ως αποτέλεσμα ο αισθητήρας να μην λαμβάνει επαρκές βαθμό σήματος υπέρυθρης ακτινοβολίας και έτσι να μη δίνει συναγερμό.

Επιπροσθέτως, γνωρίζοντας τα νεκρά σημεία του διαγράμματος κάλυψης, κάποιος μπορεί να παρακάμψει την κάλυψη του αισθητήρα στην ενεργή περιοχή. Περπατώντας προς τον αισθητήρα απέναντι από το οπτικό πεδίο του, μπορεί επίσης να μειώσει την ικανότητα ανίχνευσης του αισθητήρα μη επιτρέποντας να σπάσουν τα όρια ανίχνευσης των δεσμών (νοητές δέσμες που αντιπροσωπεύουν ζώνες ανίχνευσης των επιμέρους τομέων του αισθητήρα).

ΠΛΑΗΤΙΚΟΣ ΑΙΣΘΗΤΗΡΑΣ ΥΠΕΡΥΘΡΩΝ





## 2.7 ΑΝΙΧΝΕΥΤΗΣ ΚΑΠΝΟΥ(ΠΥΡΑΝΙΧΝΕΥΤΗΣ)



Ο ανιχνευτής καπνού είναι ικανός να αντιλαμβάνεται την ύπαρξη σωματιδίων καπνού στον αέρα προκειμένου να ενεργοποιηθεί συναγερμός για την ύπαρξη φωτιάς.

Υπάρχουν δύο βασικοί τύποι ανιχνευτών καπνού : οι ανιχνευτές ιονισμού και οι φωτοηλεκτρικοί ανιχνευτές. Ένα σύστημα συναγερμού ή πυροπροστασίας μπορεί να χρησιμοποιεί τον ένα ή και τους δύο τύπους ανιχνευτών και μερικές φορές επιπλέον ανιχνευτές θερμοκρασίας για την προειδοποίηση ύπαρξης πυρκαγιάς.

Οι ανιχνευτές ιονισμού έχουν ένα θάλαμο ιονισμού και μια πηγή ιονίζουσας ακτινοβολίας από μια μικρή ποσότητα αμερίκιου-241 το οποίο είναι μια πηγή σωματιδίων άλφα (πυρήνες ηλίου). Ο θάλαμος ιονισμού αποτελείται από δύο πλάκες που απέχουν περίπου κατά ένα εκατοστό. Τάση εφαρμόζεται στις πλάκες , φορτίζοντας την μια πλάκα θετικά και την άλλη αρνητικά. Τα σωματίδια άλφα συνεχώς απελευθερώνονται από το αμερίκιο χτυπώντας ηλεκτρόνια από τα άτομα του αέρα ιονίζοντας τα άτομα του οξυγόνου και του αζώτου στο θάλαμο. Τα θετικά φορτισμένα άτομα οξυγόνου και αζώτου έλκονται από την αρνητική πλάκα και τα ηλεκτρόνια έλκονται από την θετική πλάκα, δημιουργώντας ένα μικρό συνεχές ηλεκτρικό ρεύμα. Όταν καπνός εισέρχεται στον θάλαμο ιονισμού, τα σωματίδια καπνού συνδέονται με τα ιόντα και τα εξουδετερώνουν και έτσι δεν φτάνουν στην πλάκα. Η πτώση στο ρεύμα ανάμεσα στις πλάκες ενεργοποιεί το συναγερμό.

Στον τύπο φωτοηλεκτρικού συστήματος , ο καπνός μπορεί να εμποδίσει μια ακτίνα φωτός. Σε αυτήν την περίπτωση , η μείωση του φωτός που φτάνει στο φωτοκύτταρο διεγείρει τον συναγερμό. Στον πιο κοινό τύπο φωτοηλεκτρικής μονάδας, το φως διασκορπίζεται από τα σωματίδια καπνού και όταν το φωτοκύτταρο λαμβάνει φως, διεγείρει τον συναγερμό. Σε αυτόν τον τύπο ανιχνευτή ο θάλαμος έχει σχήμα T με μια δίοδο LED η οποία εκπέμπει μια ακτίνα φωτός κατά μήκος της οριζόντιας γραμμής του T. Ένα φωτοκύτταρο βρίσκεται στο κάτω μέρος της κάθετης βάσης του T που δημιουργεί ένα ρεύμα όταν εκτίθεται στο φως. Σε συνθήκες μη ύπαρξης καπνού η ακτίνα φωτός διασχίζει την οριζόντια γραμμή του T σε μια μη διακοπτόμενη ευθεία χωρίς να προσπέσει στο φωτοκύτταρο που είναι τοποθετημένο σε ορθή γωνία κάτω από την δέσμη. Όταν έχουμε την ύπαρξη καπνού , το φως διασκορπίζεται από τα σωματίδια καπνού και μέρος του φωτός κατευθύνεται προς τα κάτω στο κάθετο μέρος του T και προσπίπτει στο φωτοκύτταρο. Όταν επαρκής φως χτυπά το κύτταρο έχουμε ενεργοποίηση του συναγερμού.

### **Εφαρμογές και παράμετροι ορθής λειτουργίας:**

Οι ανιχνευτές καπνού τοποθετούνται σε εσωτερικούς χώρους σε οικίες , επαγγελματικούς χώρους, βιομηχανίες, γραφεία κ.ά. Οι ανιχνευτές καπνού πρέπει να τοποθετούνται στην οροφή του χώρου που επιτηρούν και η μεταξύ τους απόσταση ανάμεσα σε πυρανιχνευτές θα πρέπει να είναι κατάλληλη ανάλογα την ακτίνα κάλυψης που δίνει ο κατασκευαστής του κάθε πυρανιχνευτή. Σε οροφές που δεν είναι επίπεδες οι ανιχνευτές καπνού θα πρέπει να τοποθετούνται κατάλληλα στο ψηλότερο σημείο της οροφής λόγο του ότι τα σωματίδια καπνού ανεβαίνουν προς τα επάνω και συγκεντρώνονται στην οροφή του κτιρίου. Όταν ο καπνός που συγκεντρώνεται στην οροφή έχει κατάλληλη πυκνότητα, ανιχνεύεται από τον αισθητήρα καπνού. Με αυτόν τον τρόπο, έχουμε την ταχύτερη ανίχνευση καπνού όταν οι ανιχνευτές καπνού είναι τοποθετημένοι στην οροφή. Επίσης, αν η οροφή χωρίζεται σε πλαίσια π.χ. λόγο της ύπαρξης δοκαριών θα πρέπει σε κάθε πλαίσιο να είναι τοποθετημένοι ξεχωριστά ανιχνευτές καπνού ώστε να υπάρχει προστασία στο συγκεκριμένο σημείο, λόγω του ότι ο καπνός δεν θα περάσει από πλαίσιο σε πλαίσιο λόγω της ύπαρξης των δοκαριών. Οι ανιχνευτές ιονισμού και οι φωτοηλεκτρικοί ανιχνευτές είναι αποτελεσματικοί ανιχνευτές καπνού. Οι ανιχνευτές ιονισμού ανταποκρίνονται γρηγορότερα σε πυρκαγιές με φλόγες με μικρότερα σωματίδια

Σάλλας Παντελής – Λαμπρινός Κωνσταντίνος

καύσης, οι φωτοηλεκτρικές ανιχνευτές ανταποκρίνονται γρηγορότερα σε φωτιές που σιγοκαίνε και δημιουργούν πυκνό καπνό με μεγαλύτερα σωματίδια.

Συνθήκες μη αξιόπιστης ανίχνευσης μπορούν να αποφευχθούν όταν οι ανιχνευτές καπνού είναι τοποθετημένοι κατάλληλα σύμφωνα με τα κριτήρια που δίνουν οι κατασκευαστές, σε σημεία και σε αποστάσεις μεταξύ τους έτσι ώστε να μην δημιουργούνται νεκρές ζώνες λόγω του ότι αν υπάρξει φωτιά σε αυτά τα σημεία, δεν θα ανιχνευθεί μέχρι αν περάσει σε άλλα ορατά σημεία. Επίσης, θα πρέπει να συντηρούνται κατάλληλα και να γίνεται έλεγχος σε τακτικά χρονικά διαστήματα.

Αίτια ψευδών συναγερμών μπορούν να υπάρξουν σε κάθε είδος ανιχνευτή, που ο ατμός ή η υψηλή υγρασία μπορεί να οδηγήσει στην συμπύκνωση υδρατμών εντός του ανιχνευτή, προκαλώντας την δημιουργία συναγερμού. Για το λόγο αυτό, θα πρέπει να αποφεύγεται η τοποθέτηση τους πάνω από μαγειρικές εστίες και φούρνους και γενικά κοντά σε πηγές ατμού για την αποφυγή ψευδών συναγερμών. Τέλος, κάτι εξαιρετικά δύσκολο αλλά όχι απίθανο, η ύπαρξη σκόνης στον αέρα μπορεί να προκαλέσει ψευδείς συναγερμούς.

## ΚΕΦΑΛΑΙΟ 3

### ARDUINO

Το Arduino είναι ένας single-board μικροελεγκτής. Πρόκειται δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και 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++).

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 ενσωμάτωση 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 εργοστασιακής κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Σε εννοιολογικό επίπεδο, στην χρήση του Arduino software stack, όλα τα boards προγραμματίζονται με μία RS-232 σειριακή σύνδεση, αλλά ο τρόπος που επιτυγχάνεται αυτό διαφέρει σε κάθε hardware εκδοχή. Οι σειριακές πλάκες Arduino περιέχουν ένα απλό level shifter κύκλωμα για να μετατρέπει μεταξύ σήματος επιπέδου RS-232 και TTL. Τα τωρινά Arduino προγραμματίζονται μέσω USB, αυτό καθίσταται δυνατό μέσω της εφαρμογής προσαρμοστικών chip USB-to-Serial όπως το FTDI FT232. Κάποιες παραλλαγές, όπως το Arduino mini και το ανεπίσημο Boarduino, χρησιμοποιούν ένα αφαιρούμενο USB-to-Serial καλώδιο ή board, Bluetooth ή άλλες μεθόδους.

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.

Λόγω της φύσης του ανοικτού κώδικα, το arduino διαθέτει μια μεγάλη γκάμα από αισθητήρες οι οποίοι ανάλογα με το είδος τους, μεταδίδουν δεδομένα στο arduino και ανάλογα πως θέλουμε να τα εκμεταλλευτούμε, έχουμε την δυνατότητα αυτοματοποιημένων ενεργειών σε πολύ χαμηλό κόστος και αρκετά μεγάλη ακρίβεια.

## 3.1 Μοντέλα Arduino

Τα μοντέλα Arduino που κυκλοφορούν στην αγορά, παρουσιάζονται στην παρακάτω εικόνα . Στη συγκεκριμένη εργασία, επιλέχθηκε το μοντέλο Arduino mega μιας και καλύπτει όλες τις ανάγκες της εργασίας, καθώς τα pins που διαθέτει επαρκούν για όλες τις λειτουργίες, όπως και η μνήμη του μικροελεγκτή είναι αρκετή για να αποθηκεύσει τον κώδικα που κατασκευάστηκε.



Arduino Diecimila



Arduino Duemilanove (rev 2009b)



Arduino UNO



Arduino Leonardo



Arduino Mega



Arduino Nano



Arduino Mini





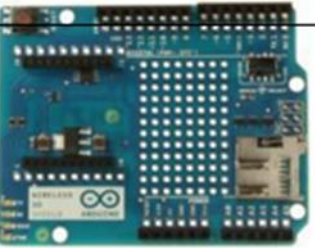
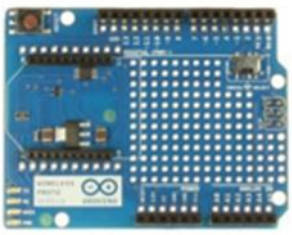
LilyPad Arduino

### Εκδόσεις:

- Τον Σεπτέμβριο του 2006 ανακοινώθηκε το Arduino Mini
- Τον Οκτώβρη του 2008 ανακοινώθηκε το Arduino Duemilanove. Αρχικά βασίστηκε στο Atmel Atmega168, αλλά μετά στάλθηκε με το ATmega328
- Τον Μάρτιο του 2009 ανακοινώθηκε το Arduino Mega. Είναι βασισμένο στο Atmel
  - ATmega1280
- Από τον Μάιο του 2011 πάνω από 300,000 Arduino ήταν σε χρήση σε όλο τον κόσμο
- Τον Ιούλιο του 2012 ανακοινώθηκε το Arduino Leonardo. Είναι βασισμένο στο
  - Atmel ATmega32u4
- Τον Οκτώβριο του 2012 ανακοινώθηκε το Arduino Due. Είναι βασισμένο στο Atmel SAM3X8E, που είχε πυρήνα ARM Cortex-M3
- Τον Νοέμβριο του 2012 ανακοινώθηκε το Arduino Micro. Είναι βασισμένο στο
  - Atmel ATmega32u4
- Τον Μάιο του 2013 ανακοινώθηκε το Arduino Robot. Είναι βασισμένο στο Atmel ATmega32u4 και ήταν το πρώτο επίσημο Arduino με ρόδες
- Μάιο του 2013 ανακοινώθηκε το Arduino Yun. Είναι βασισμένο στο ATmega32u4 και στο Atheros AR9331 και ήταν το πρώτο προϊόν wifi που συνδύαζε το Arduino.

### 3.1.2 Arduino Shields

Τα Shields, είναι πλακέτες που μπορούν να συνδεθούν στην κορυφή ενός μικροελεγκτή Arduino, επεκτείνοντας τις δυνατότητές του. Τα διαφορετικά shields, ακολουθούν την ίδια φιλοσοφία, δηλαδή την ευκολία στην τοποθέτηση και το μικρό κόστος στην παραγωγή τους. Μερικά από αυτά φαίνονται στον ακόλουθο πίνακα.

<p>Arduino WiFi Shield</p>		<p>Το Arduino WiFi Shield συνδέει το Arduino στο διαδίκτυο αυτόματα.</p>
<p>Arduino Ethernet Shield</p>		<p>Το Arduino Ethernet Shield συνδέει το Arduino στο διαδίκτυο με ένα RJ45 καλώδιο.</p>
<p>Wireless SD Shield</p>		<p>Το Wireless SD Shield επιτρέπει σε 1 πλακέτα Arduino να επικοινωνεί ασύρματα με 1 ασύρματη μονάδα. Η μονάδα μπορεί αν επικοινωνήσει ως και 100 πόδια σε εσωτερικούς χώρους ή 300 πόδια σε εξωτερικούς χώρους. Η μονάδα περιέχει 1 θύρα υποδοχής SD.</p>
<p>Wireless Proto Shield</p>		<p>Το Wireless Proto shield επιτρέπει στο Arduino να επικοινωνεί ασύρματα με μία ασύρματη μονάδα. Η μονάδα μπορεί αν επικοινωνήσει ως και 100 πόδια σε εσωτερικούς χώρους ή 300 πόδια σε εξωτερικούς χώρους. Η μονάδα δεν περιέχει θύρα υποδοχής SD.</p>

<p>Arduino Motor Shield</p>		<p>Το Arduino Motor Shield επιτρέπει την οδήγηση δύο DC κινητήρων από την ίδια συσκευή, ελέγχοντας την ταχύτητα και την κατεύθυνση του καθενός ξεχωριστά.</p>
---------------------------------	---	---

### 3.1.3 Ακροδέκτες

**VIN** : Η τάση εισόδου που τροφοδοτείται ο Arduino, όταν χρησιμοποιούμε μια εξωτερική πηγή ισχύος (και όχι τα 5V από τη σύνδεση μέσω USB ή άλλων ρυθμιζόμενων πηγών ενέργειας).

**V** : Η ρυθμιζόμενη παροχή ηλεκτρικού ρεύματος που χρησιμοποιείται για την τροφοδοσία του μικροελεγκτή και άλλα στοιχεία του Arduino. Αυτή μπορεί να προέλθει είτε από VIN μέσω του ρυθμιστή τάσης πάνω στον Arduino, είτε μέσω USB.

**3V3** : Η 3.3 προμήθεια Volts που παράγεται από το ενσωματωμένο ρυθμιστή. Μέγιστο ρεύμα είναι 50 mA.

**GND** : Οι Ακίδες γείωσης. (Ground)

**IOREF**: Αυτή η καρφίτσα στη πλακέτα Arduino που παρέχει την τάση αναφοράς με το οποίο λειτουργεί το μικροελεγκτή. Μια σωστά ρυθμισμένη ασπίδα μπορεί να διαβάσει την τάση pin IOREF και επιλέγοντας την κατάλληλη πηγή ή να επιτρέπει στους μεταφραστές τάσης στις εξόδους για την εργασία με την 5V ή 3.3V.



### 3.1.3 Μνήμη Arduino

Ο ATmega2560 έχει 256 KB μνήμη flash για την αποθήκευση κώδικα (εκ των οποίων 8 KB χρησιμοποιείται για τον φορτωτή εκκίνησης "Boot Loader"), 8 KB SRAM και 4 KB EEPROM (πού μπορεί να διαβάσει και να γράψει με την βιβλιοθήκη EEPROM).

### 3.1.4 Είσοδοι και Έξοδοι

Το Mega2560 έχει 16 αναλογικές εισόδους, καθεμία από τις οποίες παρέχουν 10 bits (δηλαδή 1.024 διαφορετικές τιμές). Από προεπιλογή μετρούν από το 0 έως 5 volts, ωστόσο, είναι δυνατόν να αλλάξει το εύρος τους, χρησιμοποιώντας την συνάρτηση AREF και analogReference ().

Κάθε μια από της 54 ψηφιακές ακίδες του Arduino mega 2560 μπορεί να χρησιμοποιηθούν ως είσοδοι ή έξοδοι, χρησιμοποιώντας τις pinMode() , digitalWrite() , και digitalRead() λειτουργίες. Λειτουργούν σε 5 volt. Κάθε ακίδα μπορεί να προσφέρει ή να λάβει μέγιστο 40mA και έχει εσωτερική αντίσταση 20-50 kOhms.

Επιπλέον, μερικά Pins έχουν εξειδικευμένες λειτουργίες:

0 (RX) και 1(TX) ως serial 1, 19(RX) και 18(TX) ως serial 2, 17(RX) και 16(TX) ως serial 3, ως 15(RX) και 14(TX) Serial 4. Επίσης, μπορούν να χρησιμοποιηθούν για τη λήψη (RX) και μετάδοση (TX) TTL (transistor – transistor logic) σειράς δεδομένων. Οι Καρφίτσες 0 και 1 συνδέονται επίσης και με τις αντίστοιχους ακροδέκτες του ATmega8U2 USB-TTL Serial chip.

**PWM:** από 0 έως 13, για παροχή 8 -bit εξόδου PWM (Pulse -width modulation) με την συνάρτηση analogWrite().

**SPI:** Serial Peripheral Interface 50(MISO) Master In Slave Out, 51(MOSI) Master Out Slave In, 52(SCK) Serial Clock, 53(SS) Slave Select. Αυτές είναι οι ακίδες SPI και υποστηρίζονται μέσω των βιβλιοθηκών SPI .

**LED:** Υπάρχει ενσωματωμένο LED που συνδέετε ψηφιακά με τον ακροδέκτη 13. Όταν η τιμή της εξόδου είναι υψηλής αξίας, το LED είναι αναμμένο, και όταν η τιμή της εξόδου του είναι χαμηλής αξίας το LED είναι σβηστό.

**I<sup>2</sup>C:** 20(SDA) και 21(SCL). Η επικοινωνία τον I<sup>2</sup>C(TWI) γίνεται με την υποστήριξη των βιβλιοθηκών Wire.

**AREF:** Τάσης αναφοράς για τις αναλογικές εισόδους. Χρησιμοποιείται με την συνάρτηση analogReference().

**RESET:** Το κουμπί reset χρησιμοποιείτε για να επαναφέρουμε τον μικροελεγκτή στην αρχική κατάσταση.

Ο Arduino Mega 2560 διαθέτει 16 αναλογικές εισόδους, κάθε μια από της οποίες παρέχουν 10bits (δηλαδή 1024 διαφορετικές τιμές).

### 3.1.5 Τρόποι επικοινωνίας

Ο Arduino Mega2560 έχει μια σειρά από εγκαταστάσεις για την επικοινωνία με ένα υπολογιστή, ένα άλλο Arduino ή άλλους μικροελεγκτές.

Ο ATmega2560 παρέχει τέσσερα UART (Universal Asynchronous Receiver/Transmitter) υλικού για TTL (5V) μέσω σειριακής επικοινωνίας. Μέσω του ATmega8U2 πάνω στην πλακέτα τον Arduino γίνεται επικοινωνία με σύνδεση USB και παρέχει μια εικονική θύρα COM για το λογισμικό στον υπολογιστή. Το λογισμικό Arduino περιλαμβάνει μια σειριακή οθόνη που επιτρέπει την απλή μορφή κειμένου δεδομένων που αποστέλλονται από και προς τον Arduino.

Οι RX και TX λυχνίες LED πάνω στην πλακέτα αναβοσβήνουν όταν τα δεδομένα μεταδίδονται μέσω τον ATmega8U2 και από την USB σύνδεση, Με την SoftwareSerial βιβλιοθήκη που επιτρέπει την σειριακή επικοινωνία σε κάποιους από τους ψηφιακούς ακροδέκτες όπως προαναφέραμε του Arduino Mega2560.

Ο ATmega2560 υποστηρίζει επίσης I2C(TWI) και SPI επικοινωνία. Το λογισμικό περιλαμβάνει μια βιβλιοθήκη Wire για να απλοποιήσει τη χρήση των I2C. Για επικοινωνία με το SPI, χρησιμοποιείτε η βιβλιοθήκη SPI.

### 3.1.6 Προγραμματισμός Arduino

Ο Arduino mega 2560 μπορεί να προγραμματιστεί με το λογισμικό Arduino IDE που είναι διαθέσιμο για download στην επίσημη ιστοσελίδα του.

Ο ATmega2560 Arduino έρχεται preburned με ένα bootloader που μας επιτρέπει να φορτώσουμε το νέο κώδικα χωρίς τη χρήση εξωτερικού προγραμματιστή υλικού. Επικοινωνεί χρησιμοποιώντας το αρχικό πρωτόκολλο STK500.

Μπορούμε επίσης να παρακάμψουμε το bootloader και το πρόγραμμα του μικροελεγκτή μέσω του ICSP (In Circuit Serial Programming) Pin.

Πηγαίος κώδικας του ATmega8U2 είναι διαθέσιμος στην επίσημη ιστοσελίδα του Arduino. Στον ATmega8U2 είναι φορτωμένο έναν bootloader DFU (Device Firmware Upgrade), το οποίο μπορεί να ενεργοποιηθεί από τη σύνδεση του βραχυκυκλωτήρα με κολλήσεις στο πίσω μέρος της πλακέτας και στη συνέχεια επαναφορά του 8U2. Μπορεί έπειτα να χρησιμοποιηθεί το λογισμικό FLIP της Atmel για Windows ή ο προγραμματιστής DFU για Mac OS X και Linux για να φορτωθεί ένα νέο firmware. Ακόμα, μπορεί να χρησιμοποιηθεί το ISP με εξωτερικό προγραμματιστή αντικαθιστώντας τον bootloader DFU.

## ΚΕΦΑΛΑΙΟ 4

### ΤΑ ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ ΓΙΑ ΤΗΝ ΕΡΓΑΣΙΑ

Τα προγράμματα που χρησιμοποιήθηκαν για τους σκοπούς της εργασίας μας είναι ανοιχτού ή ελεύθερου λογισμικού. Τα κύρια εργαλεία που χρησιμοποιήθηκαν κατά τη δημιουργία της εργασίας είναι τα παρακάτω:

#### 4.1 Xampp

Το XAMPP είναι πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξάρτητου πλατφόρμας το οποίο περιέχει τον εξυπηρετητή ιστοσελίδων http "Apache", τη βάση δεδομένων "MySQL" και ένα διερμηνέα για σενάρια, γραμμένα σε γλώσσες προγραμματισμού PHP και Perl.

Το XAMPP 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, που μπορεί να εξυπηρετεί και δυναμικές ιστοσελίδες τεχνολογίας PHP/MySQL. Είναι ανεξάρτητο από την πλατφόρμα 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.



## 4.2 Apache Web Server

Ο Apache HTTP γνωστός σαν Apache, είναι ένας εξυπηρετητής του Παγκόσμιου Ιστού (Web Server). Όποτε ένας χρήστης επισκέπτεται ένα ιστότοπο, το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης.

Ο Apache αποτελεί έναν ανοικτού κώδικα λογισμικό, το οποίο υποστηρίζει ένα σύνολο από διάφορα λειτουργικά συστήματα, όπως Linux, Unix, Windows κ.ά. Είναι γραμμένος σε 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 στατικές και δυναμικές ιστοσελίδες. Ο APACHE, χρησιμοποιείται πολλές φορές σε περιπτώσεις που είναι αναγκαίο διάφορες πληροφορίες να γίνουν προσβάσιμες με ένα ασφαλή και αξιόπιστο τρόπο.

Κυκλοφόρησε υπό την άδεια λογισμικού Apache και συντηρείται από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation). Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και την πλατφόρμα .NET. Τον Οκτώβριο του 2007 το μερίδιό του ήταν 47.73% από όλους τους ιστοτόπους. 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 εργασία. Ένας ακόμη λόγος,

είναι γιατί υποστηρίζει τη γλώσσα προγραμματισμού PHP, η οποία χρησιμοποιήθηκε για την υλοποίηση των λειτουργιών της ιστοσελίδας.

### 4.3 Arduino IDE

Το Arduino είναι ένας single-board μικροελεκτήρας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και 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 προσυναρμολογημένες· 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 2005, ένα σχέδιο κίνησε προκειμένου να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzì και David Cueartielles ονόμασαν το σχέδιο από τον Arduino της Invea και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας- την ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti.

Το σχέδιο Arduino είναι μία διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring, παρόμοια με την C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE).

Για τη διαχείριση του Arduino από τον υπολογιστή χρησιμοποιείται το Arduino IDE.

Το Arduino IDE είναι βασισμένο σε Java και παρέχει:

- Γραφικό περιβάλλον για τη συγγραφή των προγραμμάτων (τα οποία ονομάζονται sketch στην ορολογία του Arduino) με συντακτική χρωματική σήμανση.
- Κάποιες έτοιμες βιβλιοθήκες για προέκταση της.
- Compiler για τη μεταγλώττιση των sketch.

- Serial monitor που παρακολουθεί την επικοινωνία της σειριακής θύρας (USB), αναλαμβάνει να στείλει αλφαριθμητικά στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για το debugging των sketches και την επιλογή για ανέβασμα των μεταγλωττισμένων sketches στο Arduino.

#### 4.4 VBScript

Η VBScript (Visual Basic Scripting Edition) είναι μια γλώσσα που αναπτύχθηκε από τη Microsoft, που εμπνευσμένη από τη Visual Basic. Είναι σχεδιασμένη ως μια «ελαφριά» γλώσσα με ένα γρήγορο διερμηνέα για χρήση σε μια ευρεία ποικιλία του Microsoft περιβάλλοντος. Η VBScript χρησιμοποιεί τη αντικειμενοστραφή μέθοδο για να αποκτήσει πρόσβαση σε στοιχεία του περιβάλλοντος εντός του οποίου λειτουργεί. Για παράδειγμα, η FileSystemObject (FSO) χρησιμοποιείται για να δημιουργήσει, να διαβάσει, να ενημερώσει και να διαγράψει αρχεία. Η VBScript είναι εγκατεστημένη από προεπιλογή σε κάθε έκδοση επιφάνειας εργασίας των Microsoft Windows από τα Windows 98 και προαιρετικά με Windows CE (ανάλογα με τη συσκευή που είναι εγκατεστημένη).

Ένα script της VBScript πρέπει να εκτελεστεί μέσα σε ένα περιβάλλον υποδοχής, με κάποια να παρέχονται με τα Microsoft Windows, συμπεριλαμβανομένων των: Windows Script Host (WSH), ο Internet Explorer , και υπηρεσίες Internet Information Services (IIS). Επιπλέον, η VBScript μπορεί να ενσωματωθεί σε άλλα προγράμματα, μέσω τεχνολογιών όπως η Microsoft Script (msscript.ocx).

Την VB μπορούμε να την εκμεταλλευτούμε με το να διαβάζουμε τα δεδομένα από την σειριακή θύρα USB που είναι συνδεδεμένος ο Arduino σε πραγματικό χρόνο. Έτσι δημιουργείτε ένα μικρό script το οποίο τρέχει στον Host υπολογιστή και διαβάζει όλα τα εξερχόμενα δεδομένα από τον μικροελεγκτή και τα αποθηκεύει σε ένα αρχείο txt που το έχουμε σαν data logger.

## ΚΕΦΑΛΑΙΟ 5

### ΤΟ ΛΟΓΙΣΜΙΚΟ ΜΕΡΟΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Το λογισμικό μέρος του συστήματος χωρίζεται σε δυο βασικά τμήματα. Το πρώτο τμήμα το οποίο ονομάζεται ως «διαδικτυακό τμήμα» δίνει τη δυνατότητα παρακολούθησης 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 ο προγραμματισμός του μικροελεγκτή πραγματοποιούνται σε αυτό το τμήμα.

#### 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 3 μέρη: δομή (structure), τιμές (values) και συναρτήσεις (functions).

#### 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 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, στη γλώσσα του Arduino, μπορούν να χρησιμοποιηθούν ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπους 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.



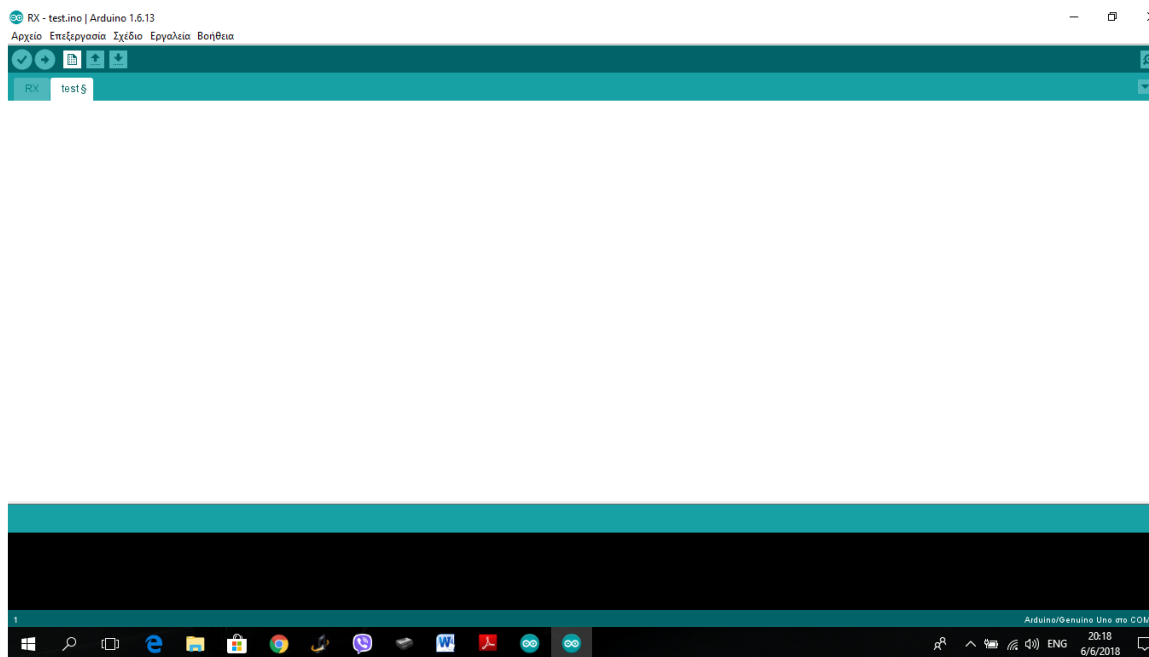
Οι πιο σημαντικές εντολές επεξηγούνται στον πίνακα που ακολουθεί:

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

<, >, <=, >=	volatile, const	interrupts(), noInterrupts()
<b>Λογικοί τελεστές</b>	<b>Εργαλεία</b>	<b>Επικοινωνία</b>
&&,   , !	sizeof()	Serial

## 5.1 Το περιβάλλον ανάπτυξης

Το περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία πολυπλατφορμική εφαρμογή γραμμένη σε Java και βασίζεται στο περιβάλλον της γλώσσας προγραμματισμού Processing.



Εικόνα 4.9: Περιβάλλον ArduinoIDE

### 5.1.1 Ρυθμίσεις του περιβάλλοντος ανάπτυξης

Οι βασικές ρυθμίσεις που πρέπει να κάνουμε από την στιγμή που συνδέσουμε το Arduino στο σύστημα μας είναι:

1. Επιλογή πλακέτας. Από το μενού Tools -> Board επιλέγουμε την πλακέτα που έχουμε.
2. Επιλογή σειριακής θύρας. Από το μενού Tools -> Serial Port επιλέγουμε την σειριακή θύρα ή θύρα USB που έχουμε συνδεδεμένο το Arduino.

Ρυθμίσεις που αφορούν το μέγεθος του κειμένου, τον φάκελο αποθήκευσης, χρήση εξωτερικού κειμενογράφου βρίσκονται στη καρτέλα Preferences ( File -> Preferences ). Για περισσότερες ρυθμίσεις μπορούμε να κάνουμε αλλαγές το αρχείο preferences.txt (βρίσκεται στον φάκελο του χρήστη - /.arduino/preferences.txt στο ΛΣ Linux).

### 5.1.2 Δομή προγράμματος

Ένα πρόγραμμα του Arduino θα μπορούσε να έχει την εξής δομή:

```
// δηλώσεις μεταβλητών
void setup() {
  αρχικοποιήσεις }
void loop() {
// ...
}
```

Υπάρχουν δυο βασικές συναρτήσεις σε ένα τυπικό πρόγραμμα. Η συνάρτηση setup() εκτελείται στην αρχή του προγράμματος και για μία μόνο φορά. Χρησιμοποιείται για τις αρχικοποιήσεις των μεταβλητών, τις δηλώσεις των pin (αν θα είναι είσοδος ή έξοδος) και τις αρχικοποιήσεις των βιβλιοθηκών.

Η συνάρτηση loop() κάνει αυτό που λέει και το όνομά της, Ο κώδικας που γράφεται μέσα στη συνάρτηση αυτή επαναλαμβάνεται συνεχώς δίνοντας την δυνατότητα στο πρόγραμμα μας να αλλάζει τιμές και το Arduino να ανταποκρίνεται ανάλογα.

Σε αυτό το σημείο θα παρουσιάσουμε αναλυτικά τον κώδικα που γράψαμε στο Arduino IDE για να υλοποιεί τις προγραμματιστικές ανάγκες που θέσαμε στο σύστημα συναγερμού δίνοντας εντολές μέσω συναρτήσεων που σχεδιάστηκαν για κάθε φυσικό εξάρτημα ξεχωριστά που είναι συνδεδεμένο στη μητρική πλακέτα και το σύνολο αυτών απαρτίζουν το ολοκληρωμένο σύστημα συναγερμού που κατασκευάσαμε.

## ΚΕΦΑΛΑΙΟ 6

### ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΕΞΑΡΤΗΜΑΤΑ ΕΡΓΑΣΙΑΣ

Στα ακόλουθα υποκεφάλαια παρουσιάζονται τα εξαρτήματα που δομούν το πρακτικό μέρος της παρούσας εργασίας.

#### 6.1 Arduino Mega



Εικόνα 6.1: Η πλακέτα Arduino Uno

Πρόκειται για την υπολογιστική πλατφόρμα στην οποία αναφερθήκαμε στο κεφάλαιο 3, η οποία αποτελείται από τον μικροελεγκτή ATmega328.

## 6.2 Μαγνητικοί αισθητήρες

Πρόκειται για τους αισθητήρες που αναφέραμε στο κεφάλαιο 2.



## 6.3 Αισθητήρας υπερήχων

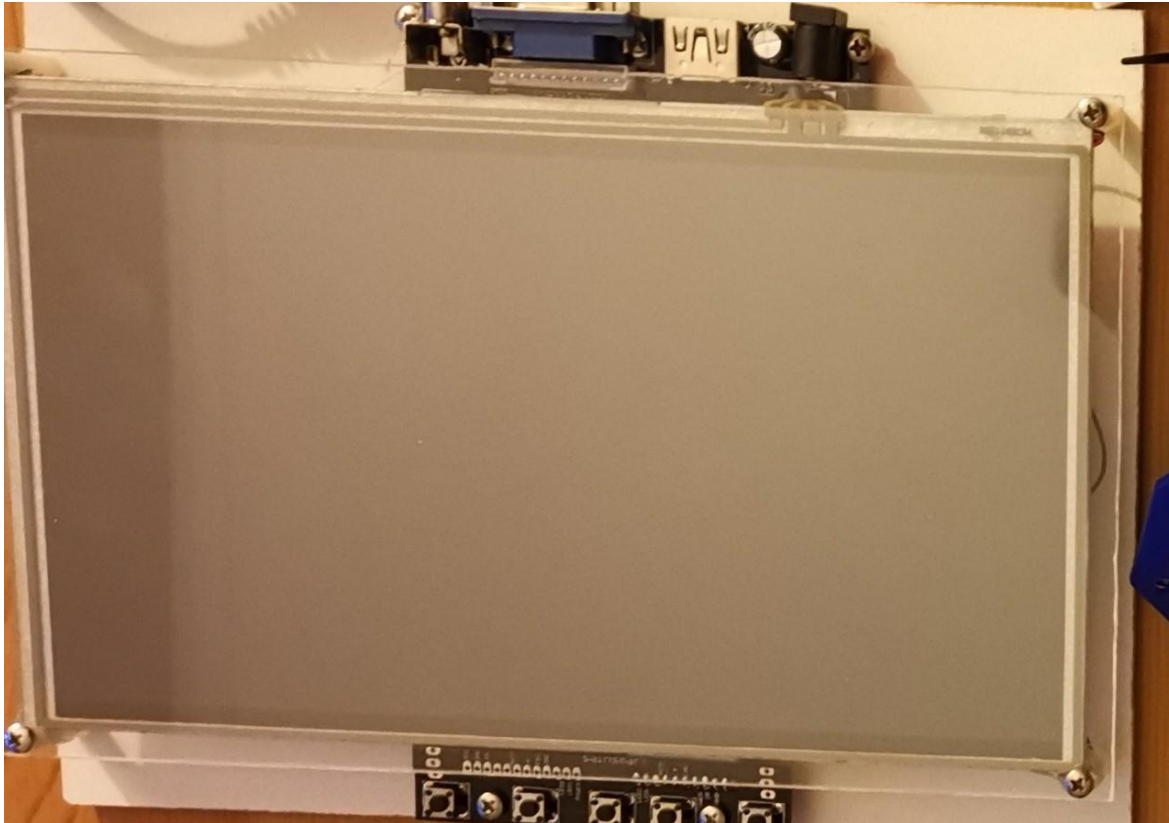
Πρόκειται για τους αισθητήρες που αναφέραμε επίσης στο κεφάλαιο 2.



Εικόνα 6.3: Ο αισθητήρας υπερήχων

#### 6.4 Οθόνη 9”

Η οθόνη που χρησιμοποιήθηκε για την προβολή των ανακοινώσεων:



Εικόνα 6.4: Η οθόνη 9 ιντσών

#### 6.4 Πλαστικό Κουτί με Πληκτρολόγιο 4x4



Εικόνα 6.4: Πλαστικό Κουτί με Πληκτρολόγιο 4x4



## 6.5 Ιστοσελίδα διαχείρισης ανακοινώσεων

**Απομακρυσμένος πίνακας ανακοινώσεων**

**+ Προσθήκη ανακοίνωσης**   **✕ Διαγραφή ανακοίνωσης**

~ Πίνακας Ανακοινώσεων ~

Θέση/Σελίδα	Ανακοίνωση
-------------	------------

**+ Ρύθμιση e-mail συστήματος συναγεμμού**

sallas\_p@hotmail.com

Μέρος πτυχιακής εργασίας των Σάλλα Παντελή και Λαμπρινού Κων/νου  
Τμήμα Μηχανικών Η/Υ Συστημάτων - Α.Ε.Ι. Πειραιά Τ.Τ.

## ΚΕΦΑΛΑΙΟ 7

### ΠΑΡΑΡΤΗΜΑ Α' – ΚΩΔΙΚΑΣ ΣΥΣΤΗΜΑΤΟΣ ΑΣΦΑΛΕΙΑΣ

```
//Libraries
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <UIPEthernet.h>
#include <Ultrasonic.h>

/*-----KEYPAD-----*/
const byte numRows= 4; //number of rows on the keypad
const byte numCols= 4; //number of columns on the keypad
char keypressed;

//keymap defines the key pressed according to the row and columns just as
appears on the keypad
char keymap[numRows][numCols]=
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};

//Code that shows the the keypad connections to the arduino terminals
byte rowPins[numRows] = {37,36,35,34}; //Rows 0 to 3
byte colPins[numCols] = {33,32,31,30}; //Columns 0 to 3

//initializes an instance of the Keypad class
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins,
numRows, numCols);
```

```
/*-----Ethernet and XML process-----*/  
  
// Max string length may have to be adjusted depending on data to be extracted  
#define MAX_STRING_LEN 100  
int index=0; //for notes reading  
  
// Setup vars  
char tagStr[MAX_STRING_LEN] = "";  
char dataStr[MAX_STRING_LEN] = "";  
char tmpStr[MAX_STRING_LEN] = "";  
char endTag[3] = {'<', '/', '\0'};  
int len;  
  
// Flags to differentiate XML tags from document elements (ie. data)  
boolean tagFlag = false;  
boolean dataFlag = false;  
  
// Ethernet vars  
byte mac[] = { 0x00, 0x0B, 0xAC, 0x1D, 0xFD, 0x67 };  
byte ip[] = { 192, 168, 1, 192 };  
byte DNS[] = {192, 168, 1, 1};  
byte gateway[] = {192, 168, 1, 1};  
byte subnet[] = {255, 255, 255, 0};  
//byte serverip[] = {31,170,164,65 }; // hol.es server <-----  
byte serverip[]={192,168,1,200};  
EthernetServer server(80); //server port default 80  
EthernetClient client;  
  
/*-----CONSTANTS-----*/  
const int buzzer = 38; //Buzzer/small speaker  
const int doorMagSen = 40; //Door magnetic sensor  
const int windowMagSen1 = 41; //Window magnetic sensor 1  
const int windowMagSen2 = 42; //Window magnetic sensor 2
```

```
Ultrasonic ultrasonic(A0,A1); //Ultrasonic sensor (trig,echo)
LiquidCrystal lcd(6, 12, 10, 9, 8, 7); //lcd ((RS, E, D4, D5, D6, D7)

/*-----VARIABLES-----*/

String password="2580"; //Variable to store the current password
String tempPassword=""; //Variable to store the input password
int doublecheck;

boolean armed = false; //Variable for system state (armed:true /
unarmed:false)

boolean input_pass; //Variable for input password (correct:true / wrong:false)

boolean storedPassword = true;

boolean changedPassword = false;

boolean checkPassword = false;

boolean readXML = false;

int distance;

int i = 1; //variable to index an array

String email;

String id[4];

String text[4];

String readString;

String connectionState="Not connected";

/*****/

void setup() {
  Serial.begin(9600);          //Start serial communication
  Ethernet.begin(mac, ip, DNS, gateway, subnet); //Start ethernet
  server.begin();             //Start the server and wait for connections
  delay(500);                 //Just a small delay
  lcd.begin(16, 2);           //Setup the LCD's number of columns and rows
  pinMode(doorMagSen,INPUT_PULLUP); //Set all magnetic sensors as
```

```
input withn internal pullup resistor
pinMode(windowMagSen1,INPUT_PULLUP);
pinMode(windowMagSen2,INPUT_PULLUP);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Please wait");
lcd.setCursor(0,1);
lcd.print("Connecting...");
while(i<5){           //Try 5 times to connect with internet/local server
if (client.connect(serverip, 80)) {
    connectionState="Connected";
    client.println("GET /noteTable.xml HTTP/1.0");
    client.println();
    break;
}
    client.stop();
    i++;
}
i=1; //set i as default 1
}

void loop() { //Main loop
    if (armed){
        systemIsArmed(); //Run function to activate the system
    }
    else if (!armed){
        systemIsUnarmed(); //Run fuction to de activate the system
    }
}
}
```

```
/******FUNCTIONS******/

//While system is unarmed
void systemIsUnarmed(){
  int screenMsg=0;

  lcd.clear();          //Clear lcd

  unsigned long previousMillis = 0;    //To make a delay by using millis()
function

  const long interval = 5000;          //delay will be 5 sec.

          //every "page"-msg of lcd will change every 5 sec
  while(!armed){          //While system is unarmed do...

    httpR();          //Call httpR() to handle the incoming data from ethernet

    unsigned long currentMillis = millis(); //Store the current run-time of the
system (millis function)

    if (currentMillis - previousMillis >= interval) {
      previousMillis = currentMillis;
      if(screenMsg==0){          //First page-message of lcd
        lcd.setCursor(0,0);
        lcd.print("SYSTEM ALARM OFF");
        lcd.setCursor(0,1);
        lcd.print("-----");
        screenMsg=1;
      }
      else if (screenMsg==1){          //Second page-message of lcd
        lcd.setCursor(0,0);
        lcd.print("A to arm ");
        lcd.setCursor(0,1);
        lcd.print("B to change pass");
        screenMsg=2;
      }
    }
  }
}
```

```
    }  
    else{          //Third page-message of lcd  
        lcd.setCursor(0,0);  
        lcd.print("C Is connected? ");  
        lcd.setCursor(0,1);  
        lcd.print("D e-mail address");  
        screenMsg=0;  
    }  
}  
  
readkey();          //Read if button is pressed by calling the readkey  
function  
  
if (keypressed =='A'){          //If A is pressed, activate the system  
    tone(buzzer,500,200);  
    systemIsArmed();          //by calling the systemIsArmed function  
}  
  
else if (keypressed =='B'){//If B is pressed, change current password  
    doublecheck=0;  
    tone(buzzer,500,200);  
    storedPassword=false;  
    if(!changedPassword){          //By calling the changePassword function  
        changePassword();  
    }  
}  
  
else if (keypressed =='C'){  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("Connection State");  
    lcd.setCursor(0,1);  
    lcd.print(connectionState);  
    delay(5000); //Show message for 5 sec
```

```
    }
    else if (keypressed == 'D'){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("e-mail address:");
        lcd.setCursor(0,1);
        lcd.print(email);
        delay(5000); //Show message for 5 sec
    }
}
}

//While system is armed
void systemIsArmed(){
    lcd.clear();
    int count=10;          //Count 10sec before activate the system
    unsigned long previousMillis = 0;
    const long interval = 1000;
    while(!armed){
        distance = ultrasonic.Ranging(CM); //Store distance from sensor only for first
time
        //While system is unarmed - for 10sec do...
        lcd.setCursor(0,0);
        lcd.print(" SYSTEM WILL BE "); //Print message to lcd with 10 sec timer
        lcd.setCursor(0,1);
        lcd.print("  ARMED IN ");
        unsigned long currentMillis = millis();
        if (currentMillis - previousMillis >= interval) {
            previousMillis = currentMillis;
            //Screen counter 10sec
        }
    }
}
```



```
    if (count>1){
        count--;      //Countdown timer
    }
    else{
        armed=true;   //Activate the system!
        break;
    }
}
lcd.setCursor(12,1);
lcd.print(count);   //show the timer at lcd second line 13 position
}
while (armed){      //While system is armed do...
    httpR();
    lcd.setCursor(0,0);
    lcd.print("SYSTEM IS ARMED!");
    lcd.setCursor(0,1);
    lcd.print("-----");
    int door = digitalRead(doorMagSen); //Read magnetic sensors and
ultrasonic sensor

    int window1 = digitalRead(windowMagSen1);
    int window2 = digitalRead(windowMagSen2);
    int curr_distanse = ultrasonic.Ranging(CM);

    //Elegxos gia paraviasi
    if ((window1==HIGH) || (window2==HIGH)){
        alarmFunction(); //Sinagermos
    }
    if (door==HIGH){
        //An anixei i porta tote perimenei na mpei kodikos
        unlockPassword();
    }
}
```

```
    }  
    //Ultrasonic sensor code  
    if (curr_distanse < (distance -5)){ //Check distanse (minus 5 for safety) with  
current distanse  
        alarmFunction();  
    }  
}  
}  
//Door is opend, unlcok the system!  
void unlockPassword() {  
    int count=21;        //20 sec for alarm!  
    retry:               //label for goto, retry in case of wrong password  
        tempPassword="";    //reset temp password (typing...)  
    lcd.clear();        //clear lcd  
    i=6;                //variable to put * while typing pass  
    unsigned long previousMillis = 0;  
    const long interval = 1000;  
    boolean buzzerState = false;    //variable to help us make a beep tone  
    while(!checkPassword){        //While waiting for correct password do...  
        unsigned long currentMillis = millis();  
        if (currentMillis - previousMillis >= interval) {  
            previousMillis = currentMillis; //play beep tone every 1 sec  
            if (!buzzerState){  
                tone(buzzer, 700);  
                buzzerState=true;  
            }  
            else{  
                noTone(buzzer);  
                buzzerState=false;  
            }  
        }  
    }  
}
```

```
    if (count>0){          //Screen counter 20sec
        count--;
    }
    else{
        alarmFunction();   //Times is up, ALARM!
        break;
    }
}
readkey();                //Read for pressed button
lcd.setCursor(0,0);
lcd.print("ALARM IN: ");
//For screen counter - 20sec
if (count>=10){
    lcd.setCursor(14,0);
    lcd.print(count);     //print countdown timer at lcd
}
else{                    //catch '0'bellow 10 (eg 09)
    lcd.setCursor(14,0);
    lcd.print(" ");
    lcd.print(count);
}
lcd.setCursor(0,1);
lcd.print("PASS>");
if (keypressed != NO_KEY){ //Accept only numbers and * from keypad
    if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed
    == '3' ||
        keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed ==
    '7' ||
        keypressed == '8' || keypressed == '9' ){
        tempPassword += keypressed;
    }
}
```

```
    lcd.setCursor(i,1);
    lcd.print("*");    //Put * on lcd
    i++;
    tone(buzzer,500,200); //Button tone
}
else if (keypressed == '*'){ //Check for password
    if (password==tempPassword){//If it's correct unarmed the system
        armed=false;
        tone(buzzer,700,500);
        break;
    }
    else{    //if it's false, retry
        tempPassword="";
        tone(buzzer,200,200);
        delay(300);
        tone(buzzer,200,200);
        goto retry;
    }
}
}
}

//Alarm
void alarmFunction(){
    alarmMail();    //Send notification email by calling the alarmMail()
    function
    retry: //label for goto
    tempPassword="";
```

```
lcd.clear();
i=6;
unsigned long previousMillis = 0;
const long interval = 500;
boolean buzzerState = false;
while(!checkPassword){ //Waiting for password to deactivate the alarm...
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis; //Play a beep tone every 0.5 second
    if (!buzzerState){
      tone(buzzer, 700);
      buzzerState=true;
    }
    else{
      noTone(buzzer);
      buzzerState=false;
    }
  }
  readkey(); //Read for keypressed
  lcd.setCursor(0,0);
  lcd.print(" !!! ALARM !!! ");
  lcd.setCursor(0,1);
  lcd.print("PASS>");
  if (keypressed != NO_KEY){ //Accept only numbers and *
    if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed
    == '3' ||
    keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed ==
    '7' ||
    keypressed == '8' || keypressed == '9' ){
      tempPassword += keypressed;
    }
  }
}
```

```
    lcd.setCursor(i,1);
    lcd.print("*");
    i++;
}
else if (keypressed == '*'){
    Serial.print(tempPassword);
    Serial.print("\t");
    Serial.println(password);
    if (password==tempPassword){
        armed=false;
        tone(buzzer,700,500);
        break;
    }
    else{
        tempPassword="";
        tone(buzzer,200,200);
        delay(300);
        tone(buzzer,200,200);
        goto retry;
    }
}
}
}
}
}
//Change current password
void changePassword(){
    retry: //label for goto
    tempPassword="";
    lcd.clear();
    i=1;
```

```

while(!changedPassword){           //Waiting for current password
  readkey();
  lcd.setCursor(0,0);
  lcd.print("CURRENT PASSWORD");
  lcd.setCursor(0,1);
  lcd.print(">");
  if (keypressed != NO_KEY){
    if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed
    == '3' ||
    keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed ==
    '7' ||
    keypressed == '8' || keypressed == '9' ){
      tempPassword += keypressed;
      lcd.setCursor(i,1);
      lcd.print("*");
      i++;
      tone(buzzer,500,200);
    }
    else if (keypressed == '*'){
      i=1;
      if (password==tempPassword){
        storedPassword=false;
        tone(buzzer,500,200);
        newPassword();           //Password is corrent, so call the newPassword
function
        break;
      }
      else{           //Try again
        tempPassword="";
        tone(buzzer,200,200);
        delay(300);

```

```
        tone(buzzer,200,200);
        goto retry;
    }
}
}
}
}
String firstpass;
//Setup new password
void newPassword(){
    tempPassword="";
    changedPassword=false;
    lcd.clear();
    i=1;
    while(!storedPassword){
        readkey();
        if (doublecheck==0){
            lcd.setCursor(0,0);
            lcd.print("SET NEW PASSWORD");
            lcd.setCursor(0,1);
            lcd.print(">");
        }
        else{
            lcd.setCursor(0,0);
            lcd.print("One more time...");
            lcd.setCursor(0,1);
            lcd.print(">");
        }
        if (keypressed != NO_KEY){
            if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed
```



```
== '3' ||
  keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed ==
'7' ||
  keypressed == '8' || keypressed == '9' ){
  tempPassword += keypressed;
  lcd.setCursor(i,1);
  lcd.print("*");
  i++;
  tone(buzzer,500,200);
}
else if (keypressed == '*'){
  if (doublecheck == 0){
    firstpass=tempPassword;
    doublecheck=1;
    newPassword();
  }
  if (doublecheck==1){
    doublecheck=0;
    if (firstpass==tempPassword){
      i=1;
      firstpass="";
      password = tempPassword; // New password saved
      tempPassword="";//erase temp password
      lcd.setCursor(0,0);
      lcd.print("PASSWORD CHANGED");
      lcd.setCursor(0,1);
      lcd.print("-----");
      storedPassword=true;
      tone(buzzer,500,400);
      delay(1000);
```



```

        readString += c;    //store characters to string
    }
    //if HTTP request has ended
    if (c == '\n') {
        //now output HTML data header
        if(readString.indexOf('?') >=0) { //don't send new page
            client.println("HTTP/1.1 204 Zoomkat");
            client.println();
            client.println();
        }
        client.stop();
        if(readString.indexOf("update") >0){ // if get has /?update
            Serial.println("updating xml...");
            index=0;
            readXML=true;
            readXMLfile();    //Read XML file
        }
        if(readString.indexOf("mail") >0){ // if get has /?update
            Serial.println("updating email...");
            index=0;
            readXML=true;
            readXMLmail();    //Read XML file
        }
        readString="";    //Reset variable
    }
}
}
}

/*****PRINT NOTES EVERY 10sec*****/
unsigned long curMillis = millis();

```

```
if (curMillis - prevMillis >= 5000) {
    // save the last time you blinked the LED
    prevMillis = curMillis;
    index++;
    flagNote=0;
    if (index==5){
        index=1;
    }
}
if (index==1){
    if (flagNote==0){
        if (text[index-1].length()>1 && text[index-1].length()<71){
            delay(100);
            Serial.print('<');
            delay(2);
            Serial.print(text[index-1].substring(0,10)); //Cut string to pieces of 10
chars
            delay(5);
            Serial.print(text[index-1].substring(10,20));
            delay(5);
            Serial.print(text[index-1].substring(20,30));
            delay(5);
            Serial.print(text[index-1].substring(30,40));
            delay(5);
            Serial.print(text[index-1].substring(40,50));
            delay(5);
            Serial.print(text[index-1].substring(50,60));
            delay(5);
            Serial.print(text[index-1].substring(60,70));
            delay(5);
```

```
        Serial.println('>');
    delay(5);
        Serial.println("null");// gia na xekolaei
    delay(10);
    }
    flagNote=1;
}
}
else if (index==2){
    if (flagNote==0){
        if (text[index-1].length()>1 && text[index-1].length()<71){
            delay(100);
            Serial.print('<');
            delay(2);
            Serial.print(text[index-1].substring(0,10)); //Cut string to pieces of 10
chars
            delay(5);
            Serial.print(text[index-1].substring(10,20));
            delay(5);
            Serial.print(text[index-1].substring(20,30));
            delay(5);
            Serial.print(text[index-1].substring(30,40));
            delay(5);
            Serial.print(text[index-1].substring(40,50));
            delay(5);
            Serial.print(text[index-1].substring(50,60));
            delay(5);
            Serial.print(text[index-1].substring(60,70));
            delay(5);
            Serial.println('>');
```

```
    delay(5);
    Serial.println("null");// gia na xekolaei
    delay(10);
  }
  flagNote=1;
}
}
else if (index==3){
  if (flagNote==0){
    if (text[index-1].length()>1 && text[index-1].length()<71){
      delay(100);
      Serial.print('<');
      delay(2);
      Serial.print(text[index-1].substring(0,10)); //Cut string to pieces of 10
chars
      delay(5);
      Serial.print(text[index-1].substring(10,20));
      delay(5);
      Serial.print(text[index-1].substring(20,30));
      delay(5);
      Serial.print(text[index-1].substring(30,40));
      delay(5);
      Serial.print(text[index-1].substring(40,50));
      delay(5);
      Serial.print(text[index-1].substring(50,60));
      delay(5);
      Serial.print(text[index-1].substring(60,70));
      delay(5);
      Serial.println('>');
      delay(5);
```

```
        Serial.println("null");// gia na xekolaei
    delay(10);
}
    flagNote=1;
}
}
else if (index==4){
    if (flagNote==0){
        if (text[index-1].length()>1 && text[index-1].length()<71){
            delay(100);
            Serial.print('<');
            delay(2);
            Serial.print(text[index-1].substring(0,10)); //Cut string to pieces of 10
chars
            delay(5);
            Serial.print(text[index-1].substring(10,20));
            delay(5);
            Serial.print(text[index-1].substring(20,30));
            delay(5);
            Serial.print(text[index-1].substring(30,40));
            delay(5);
            Serial.print(text[index-1].substring(40,50));
            delay(5);
            Serial.print(text[index-1].substring(50,60));
            delay(5);
            Serial.print(text[index-1].substring(60,70));
            delay(5);
            Serial.println('>');
            delay(5);
            Serial.println("null");// gia na xekolaei
```

```
        delay(10);
    }
    flagNote=1;
}
}
if (text[0]==" " && text[1]==" " && text[2]==" " && text[3]==" " && index==1 &&
flagNote==1 ){
    flagNote=2;
    flagNoNote=0;
}
if (flagNoNote==0){
    delay(100);
    Serial.print('<');
    Serial.print("No notes... Thats good!");
    Serial.println('>');
    delay(10);
    flagNoNote=1;
}
}
//Send notification email
void alarmMail(){
    if (client.connect(serverip, 80)) {
        client.println("GET http://notetable.hol.es/alarm.php HTTP/1.0");
        client.println();
    }
    else{
        client.stop();
    }
}
```



```
////////////////////////////////PROCESS XML FILE FUNCTIONS////////////////////////////////
```

```
void readXMLfile(){
  while(readXML){
    // Read serial data in from web:
    while (client.available()) {
      serialEvent();
      if (index==4){
        index=0;
        readXML=false;
        break;
      }
    }
    if (!client.connected()) {
      client.stop();
      delay(2000);
      if (client.connect(serverip, 80)) {
        Serial.println("read noteTable.xml file");
        client.println("GET /noteTable.xml HTTP/1.0");
        client.println();
        delay(2000);
      }
      else {
        Serial.println("Reconnect failed");
      }
    }
  }
}

void readXMLmail(){
  while(readXML){
    // Read serial data in from web:
```

```
while (client.available()) {
    serialEvent();
    if (index==4){
        index=0;
        readXML=false;
        break;
    }
}

if (!client.connected()) {
    client.stop();
    delay(2000);
    if (client.connect(serverip, 80)) {
        client.println("GET /config.xml HTTP/1.0");
        client.println();
        delay(2000);
    }
    else {
        Serial.println("Reconnect failed");
    }
}
}
```

// Process each char from web

```
void serialEvent() {
    // Read a char
    char inChar = client.read();
    ///Serial.print(".");
    ///Serial.print(inChar);
    if (inChar == '<') {
```

```
    addChar(inChar, tmpStr);
    tagFlag = true;
    dataFlag = false;
}
else if (inChar == '>') {
    addChar(inChar, tmpStr);
    if (tagFlag) {
        strncpy(tagStr, tmpStr, strlen(tmpStr)+1);
    }
// Clear tmp
clearStr(tmpStr);
tagFlag = false;
dataFlag = true;
}
else if (inChar != 10) {
    if (tagFlag) {
        // Add tag char to string
        addChar(inChar, tmpStr);
        // Check for </XML> end tag, ignore it
        if ( tagFlag && strcmp(tmpStr, endTag) == 0 ) {
            clearStr(tmpStr);
            tagFlag = false;
            dataFlag = false;
        }
    }
    if (dataFlag) {
        // Add data char to string
        addChar(inChar, dataStr);
    }
}
```

```
// If a LF, process the line
if (inChar == 10 ) {
    // Find specific tags and print data
    if (matchTag("<email>")){
        email=dataStr;
        index=4;
    }
    if (matchTag("<id>")) {
        id[index]=dataStr;
    }
    if (matchTag("<text>")) {
        text[index]=dataStr;
        index++;
    }
    // Clear all strings
    clearStr(tmpStr);
    clearStr(tagStr);
    clearStr(dataStr);
    // Clear Flags
    tagFlag = false;
    dataFlag = false;
}
}
```

```
// Function to clear a string
void clearStr (char* str) {
    int len = strlen(str);
    for (int c = 0; c < len; c++) {
        str[c] = 0;
    }
}
```

```
}  
  
//Function to add a char to a string and check its length  
void addChar (char ch, char* str) {  
    char *tagMsg = "<TRUNCATED_TAG>";  
    char *dataMsg = "-TRUNCATED_DATA-";  
  
    // Check the max size of the string to make sure it doesn't grow too  
    // big. If string is beyond MAX_STRING_LEN assume it is unimportant  
    // and replace it with a warning message.  
    if (strlen(str) > MAX_STRING_LEN - 2) {  
        if (tagFlag) {  
            clearStr(tagStr);  
            strcpy(tagStr,tagMsg);  
        }  
        if (dataFlag) {  
            clearStr(dataStr);  
            strcpy(dataStr,dataMsg);  
        }  
  
        // Clear the temp buffer and flags to stop current processing  
        clearStr(tmpStr);  
        tagFlag = false;  
        dataFlag = false;  
    } else {  
        // Add char to string  
        str[strlen(str)] = ch;  
    }  
}  
  
// Function to check the current tag for a specific string  
boolean matchTag (char* searchTag) {  
    if ( strcmp(tagStr, searchTag) == 0 ) {  
        return true;  
    }
```

```
}  
else {  
    return false;  
}  
}
```

## ΚΕΦΑΛΑΙΟ 8

### ΠΑΡΑΡΤΗΜΑ Β' – ΠΙΝΑΚΑΣ ΑΝΑΚΟΙΝΩΣΕΩΝ

```
#include <TVout.h>
#include <pollserial.h>
#include <fontALL.h>

TVout TV;    //Init TVout variable
pollserial pserial; //Init pollserial variable

//Variables
bool started= false;//True: Message is strated
bool ended = false;//True: Message is finished
char incomingByte ; //Variable to store the incoming byte
char msg[80]; //Message - array from 0 to 79 (80 chars max)
byte index;    //Index of array

void setup() {
  TV.begin(_NTSC,120,96); //Start TV library as NTSC type resolution 120x96
  TV.select_font(font6x8); //Set font 6x8
  TV.print(" ~ Note Table ~ "); //Print a header message
  TV.print("-----");
  TV.set_hbi_hook(pserial.begin(9600)); //Start pollserial at 9600
}

void loop() {
  // While pollserial is available...
```

```
while (pserial.available(>0)){
  //Read the incoming byte
  incomingByte = pserial.read();
  //Start the message when the '<' symbol is received
  if(incomingByte == '<'){
    started = true;
    index = 0;
    msg[index] = '\0'; // Throw away any incomplete packet
  }
  //End the message when the '>' symbol is received
  else if(incomingByte == '>'){
    ended = true;
    break; // Done reading - exit from while loop!
  }
  //Read the message!
  else{
    if(index < 81){ // Make sure there is room
      msg[index] = incomingByte; // Add char to array
      index++;
      msg[index] = '\0'; // Add NULL to end
    }
  }
}
//Print msg to screen
if(started && ended){
  TV.clear_screen(); //Clear previous screen
  TV.print(" ~ Note Table ~ ");
  TV.print("-----");
  TV.print(" "); //Blank line
  TV.print(msg); //Print message
```



```
index = 0;          //Reset to start reading again
msg[index] = '\0';
started = false;
ended = false;
}
}
```

## ΚΕΦΑΛΑΙΟ 9

### ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ

Ολοκληρώνοντας την παρούσα πτυχιακή εργασία, έγιναν κατανοητές οι δυσκολίες της μελέτης και της υλοποίησης μιας κατασκευής. Ακόμη, έγιναν κατανοητά και τα πλεονεκτήματά της. Τα υλικά και τα εξαρτήματα που χρησιμοποιήθηκαν για την υλοποίηση της είναι χαμηλού κόστους, γεγονός που είναι δελεαστικό για να πραγματοποιηθεί και κάτι παραπάνω κάποια άλλη στιγμή. Επίσης στο διαδίκτυο βρίσκονται αμέτρητες πληροφορίες περί κατασκευών και προγραμματισμού Arduino το οποίο σημαίνει πως η εξέλιξη της εν λόγω εργασίας είναι μεγάλη. Βέβαια αρκετές φορές είναι λίγο δύσκολη η υλοποίηση των εκάστοτε project.

Μία λογική συνέχεια της εργασίας – κατασκευής που παρουσιάστηκε θα ήταν η εγκατάσταση περαιτέρω αισθητήρων για καλύτερη παρακολούθηση και ασφάλεια του χώρου. Θα μπορούσαμε επίσης, να τοποθετήσουμε μια κανονική σειρά για την άμεση αποτροπή οποιασδήποτε κακόβουλης ενέργειας, καθώς και καλύτερη θέση τοποθέτησης του «κουτιού» εισαγωγής κωδικών, έπειτα από προσεκτική μελέτη του χώρου.



## ΚΕΦΑΛΑΙΟ 10

### ΒΙΒΛΙΟΓΡΑΦΙΑ

1. <https://barrgroup.com/Embedded-Systems/Books/Programming-Embedded-Systems>
2. <https://www.apachefriends.org/index.html>
3. <https://www.arduino.cc/en/Guide/Introduction>
4. <https://el.wikipedia.org/wiki/Arduino>
5. <https://playground.arduino.cc/Main/TVout/>
6. <http://playground.arduino.cc/Code/Keypad>
7. <http://playground.arduino.cc/Main/LiquidCrystal>
8. <https://github.com/JRodrigoTech/Ultrasonic-HC-SR04>
9. [https://github.com/ntruchsess/arduino\\_uip](https://github.com/ntruchsess/arduino_uip)
10. <https://www.arduino.cc/en/Main/Software>
11. <http://www.php.net>
12. Γαρύφαλλος Α. Γιώργος, Ηλεκτρονικά συστήματα ασφάλειας, Εκδόσεις ΙΩΝ, 2009
13. Noble, J., Programming interactivity: A Designer's Guide to Processing, Arduino and openFramework, O' Reilly Media, 2009
14. Lenz JE. *A review of magnetic sensors*, IEEE, 1990
15. Barr, M., *Programming Embedded Systems in C and C ++*, 1999
16. Evans, B., *Arduino Programming Notebook*, 2007.

