



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

ΤΕΙ ΗΠΕΙΡΟΥ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

**ΜΕΛΕΤΗ ΚΑΙ ΚΑΤΑΣΚΕΥΗ ΣΥΣΤΗΜΑΤΟΣ ΑΝΙΧΝΕΥΣΗΣ
ΑΝΘΡΩΠΙΝΗΣ ΠΑΡΟΥΣΙΑΣ ΣΕ ΑΠΟΜΑΚΡΥΣΜΕΝΕΣ ΘΕΣΕΙΣ
ΜΕ ΑΣΥΡΜΑΤΗ ΜΕΤΑΔΟΣΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΚΕΝΤΡΙΚΟ
ΠΙΝΑΚΑ ΕΛΕΓΧΟΥ**



ΣΑΒΒΑΡΗΣ ΜΙΧΑΛΗΣ

10ο εξάμηνο αμ:14005, e-mail: mixalis.teihp@yahoo.gr

ΥΠΟΓΡΑΦΗ

X

Επιβλέπων Καθηγητής: Ράπτης Βασίλειος



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

ΤΕΙ ΗΠΕΙΡΟΥ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

**STUDY AND CONSTRUCTION OF A HUMAN SENSING SYSTEM INTENDED
FOR DISTANT WIRELESS DATA CONNECTION TO A CENTRAL CONTROL
UNIT**

ΠΕΡΙΕΧΟΜΕΝΑ ΚΕΦΑΛΑΙΩΝ

1) Εισαγωγικές Έννοιες.....	8
1.1) Τα ενσωματωμένα συστήματα ως μέρος της καθημερινότητας.....	8
1.2) Ανίχνευση κίνησης.....	8
1.3) Μετάδοση με ραδιοκύματα.....	8
2) Μικροελεγκτές.....	10
2.1) Τι είναι ο μικροελεγκτής.....	10
2.2) Διαφορές μικροελεγκτή με μικροεπεξεργαστή.....	10
2.3) Υποσυστήματα.....	10
2.4) Πρόσθετες λειτουργίες.....	11
2.5) Κατηγορίες μικροελεγκτών.....	11
2.6) Γλώσσες προγραμματισμού και εργαλεία ανάπτυξης.....	12
2.7) Κατασκευαστές.....	12
2.8) Ο υπολογιστής Raspberry Pi.....	13
2.9) Τι είναι το ZigBee.....	14
3) Η ηλεκτρονική πλατφόρμα Arduino.....	15
3.1) Ιστορική αναδρομή.....	15
3.2) Τι είναι το Arduino.....	15
3.3) Το hardware.....	16
3.4) Μικροελεγκτής- Η καρδιά του Arduino.....	16
3.5) Χαρακτηριστικά του Arduino.....	17
3.5.1) Είσοδοι – Εξοδοι.....	17
3.5.2) Τροφοδοσία.....	18
3.5.3) Ενσωματωμένα κουμπιά και Leds.....	19
3.6) Το software.....	19
3.7) Shields.....	21
4) Υλοποίηση.....	22
4.1) Δυνατότητες και επεξήγηση των λειτουργιών του συστήματος.....	22
4.2) Υλοποίηση συστήματος.....	23
4.3) Υπηρεσία αποστολής email.....	28
4.4) Web εφαρμογή για τον απομακρυσμένο έλεγχο του συστήματος.....	29
4.5) Υλοποίηση ενεργειακής αυτονομίας.....	31

5) Αναλυτική περιγραφή εξαρτημάτων.....	35
5.1) Παθητικός αισθητήρας υπέρυθρης ακτινοβολίας HC-SR501.....	35
5.2) RF 433 MHz Πομπός /Δέκτης.....	38
5.3) Ethernet shield W5100.....	39
5.4) Arduino MEGA2560 REV3.....	40
5.5) Arduino Nano ATmega328.....	41
5.6) LCD display 1602.....	42
5.7) Active Buzzer.....	42
5.8) Solar panel.....	43
5.9) MT3608 Boost converter Step up DC-DC.....	43
5.10) Battery charger TP4056.....	44
5.11) RTC DS3231 AT2432 IIC Clock Module.....	44
5.12) Άλλα εξαρτήματα.....	45
6) Προγραμματισμός.....	49
7) Συμπεράσματα – Μελλοντικές επεκτάσεις.....	68
7.1) Συμπεράσματα.....	68
7.2) Μελλοντικές επεκτάσεις.....	68
8) Παραρτήματα.....	69
8.1) Παράρτημα κοστολόγησης εξαρτημάτων.....	69
8.2) Παράρτημα φωτογραφιών.....	70
9) Βιβλιογραφία.....	75

ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ

Εικόνα 1 Κάτοψη του υπολογιστή Raspberry Pi.....	13
Εικόνα 2 Τα πλεονεκτήματα του ZigBee σε σχέση με άλλα πρότυπα.....	14
Εικόνα 3 Κάτοψη της ηλεκτρονικής πλατφόρμας Arduino UNO.....	15
Εικόνα 4 Τα χαρακτηριστικά της ηλεκτρονικής πλατφόρμας.....	16
Εικόνα 5 Ψηφιακές υποδοχές του Arduino UNO.....	18
Εικόνα 6 Αναλογικές υποδοχές του Arduino UNO.....	18
Εικόνα 7 Εξωτερική τροφοδοσία της πλακέτας Arduino (Μπαταρία/ Μετασχηματιστής).....	19
Εικόνα 8 Arduino Shields ενσωματωμένες σε πλακέτες Arduino.....	21
Εικόνα 30 Συνδεσμολογία κυκλώματος του κεντρικού πίνακα ελέγχου.....	24
Εικόνα 31 Συνδεσμολογία κυκλώματος 1 ^{ου} υποσυστήματος.....	26
Εικόνα 32 Συνδεσμολογία κυκλώματος 2 ^{ου} υποσυστήματος.....	27
Εικόνα 33 Υπηρεσία SMTP2GO.....	28
Εικόνα 34 Γραφικό περιβάλλον της υπηρεσίας SMTP2GO.....	29
Εικόνα 35 Γραφικό περιβάλλον της Web Εφαρμογής.....	30

Εικόνα 36 Ένδειξη παραβίασης πόρτας 1 και πόρτας 2 στη Web Εφαρμογή	31
Εικόνα 37 Συνδεσμολογία του κυκλώματος ενεργειακής αυτονομίας (Το κόκκινο καλώδιο προσδιορίζει τη τάση ενώ το μαύρο τη γείωση)	32
Εικόνα 9. Παθητικός αισθητήρας υπέρυθρης ακτινοβολίας	35
Εικόνα 10 Ρυθμίσεις του Παθητικού αισθητήρα υπέρυθρης ακτινοβολίας	35
Εικόνα 11 Ρύθμιση της χρονικής καθυστέρησης μέσω του ποτενσιόμετρου	36
Εικόνα 12 Ρύθμιση της ευαισθησίας μέσω του ποτενσιόμετρου	36
Εικόνα 13 Ρύθμιση του βραχυκυκλωτήρα	37
Εικόνα 14 RF μονάδες 443 MHZ transmitter / receiver	38
Εικόνα 15 Κάτοψη πλακέτας Ethernet Shield W5100	39
Εικόνα 16 Κάτοψη πλακέτας Arduino Mega	40
Εικόνα 17 Κάτοψη πλακέτας Arduino Nano	41
Εικόνα 18 Οθόνη LCD 1602	42
Εικόνα 19 Active Buzzer	42
Εικόνα 20 Solar Panel	43
Εικόνα 21 Boost Converter Step up DC - DC	43
Εικόνα 22 Battery charger TP4056	44
Εικόνα 23 Πλακέτα εξωτερικού ρολογιού RTC	45
Εικόνα 24 Σχεδιάγραμμα αγωγιμότητας ενός Breadboard	45
Εικόνα 26 Καλώδια συνδεσμολογίας	46
Εικόνα 25 Αντιστάσεις	46
Εικόνα 27 Καλώδια Ethernet, USB τύπου A – B	47
Εικόνα 28 Δίοδοι led	47
Εικόνα 29 Διακόπτης με 4 ακροδέκτες	48
Εικόνα 38 Στιγμιότυπο αποσφαλμάτωσης προγράμματος	67

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

Πίνακας 1 Συναρτήσεις, εντολές και σταθερές για τον προγραμματισμό της πλακέτας Arduino. 20	
Πίνακας 2 Πίνακας τεχνικών προδιαγραφών Receiver / Transmitter	38
Πίνακας 3 Βασικά Χαρακτηριστικά Arduino Mega 2560	41
Πίνακας 4 Βασικά Χαρακτηριστικά Arduino Nano	41

ΛΙΣΤΑ ΠΑΡΑΡΤΗΜΑΤΩΝ

Παράρτημα Α Συνολικό κόστος εξαρτημάτων όλου του συστήματος	69
Παράρτημα Β Κάτοψη συστήματος κεντρικού πίνακα ελέγχου	70
Παράρτημα C Κάτοψη 1 ^{ου} υποσυστήματος	71
Παράρτημα D Κάτοψη 2 ^{ου} υποσυστήματος	71
Παράρτημα Ε Κάτοψη συστήματος κεντρικού πίνακα ελέγχου (τελικό αποτέλεσμα)	72
Παράρτημα F Κάτοψη 1 ^{ου} υποσυστήματος (τελικό αποτέλεσμα)	72
Παράρτημα G Κάτοψη 2 ^{ου} υποσυστήματος (τελικό αποτέλεσμα)	73

ΔΗΛΩΣΗ ΠΝΕΥΜΑΤΙΚΗΣ ΙΔΙΟΚΤΗΣΙΑΣ

Η παρούσα εργασία αποτελεί προϊόν αποκλειστικά δικής μου προσπάθειας. Όλες οι πηγές που χρησιμοποιήθηκαν περιλαμβάνονται στη βιβλιογραφία και γίνεται ρητή αναφορά σε αυτές μέσα στο κείμενο όπου έχουν χρησιμοποιηθεί.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέπων καθηγητή μου κ. Ράπτη Βασίλειο, για την βοήθεια, τη καθοδήγηση και την πολύ καλή συνεργασία που είχαμε καθ' όλο το χρονικό διάστημα εκπόνησης της διπλωματικής αυτής εργασίας. Οι συμβουλές του, οι οδηγίες του αλλά και οι κατευθύνσεις του συνέβαλαν στο μέγιστο για την επιτυχία κατά την άποψη μου ολοκλήρωση της εργασίας αυτής. Θα ήθελα επίσης να ευχαριστήσω όλους τους καθηγητές του τμήματος, για την πολύτιμη γνώση που μου μεταλαμπάδευσαν και τη βοήθεια που μου πρόσφεραν κάθε φορά που ζητήθηκε.

ΠΕΡΙΛΗΨΗ

Σε αυτήν την εργασία θα παρουσιαστεί η μελέτη, ο σχεδιασμός και η κατασκευή των υποσυστημάτων ανίχνευσης κίνησης και την ασύρματη επικοινωνία με ένα κεντρικό πίνακα ελέγχου με χρήση ηλεκτρονικών συστημάτων Arduino. Θα γίνει λεπτομερής αναφορά στους αισθητήρες και σε όλα τα εξαρτήματα που χρησιμοποιήθηκαν για την λειτουργία των συστημάτων. Θα αναφερθούμε στα ενσωματωμένα συστήματα, στην ασύρματη μετάδοση δεδομένων με ραδιοσυχνότητες, στα πλεονεκτήματα των μικροελεγκτών και πως αυτοί συμβάλλουν και επιδρούν στη καθημερινότητα των πολιτών. Επιπλέον θα αναφερθούμε στη μελέτη που έγινε για την ενεργειακή αυτονομία, τη κατανάλωση και τη διάρκεια ζωής των υποσυστημάτων με τη χρήση ηλεκτρικών συσσωρευτών. Τέλος θα γίνει εκτενής αναφορά στο προγραμματισμό του project και ανάλυση του κώδικα.

Λέξεις κλειδιά: Ενσωματωμένα συστήματα, μικροελεγκτές, Arduino, γλώσσα wiring

Η εργασία χωρίζεται σε επτά κεφάλαια. Συγκεκριμένα στο επόμενο κεφάλαιο γίνεται μια εισαγωγή στα ενσωματωμένα συστήματα, στον τρόπο που ανιχνεύεται η ανθρώπινη κίνηση και στα ραδιοκύματα.

Το δεύτερο κεφάλαιο αναφέρεται στη δομή, τα πλεονεκτήματα και τα υποσυστήματα που έχει ένας μικροελεγκτής. Επίσης γίνεται μια μικρή εισαγωγή στον υπολογιστή Raspberry Pi και στο πρότυπο ZigBee.

Στο τρίτο κεφάλαιο παρουσιάζεται η ηλεκτρονική πλατφόρμα Arduino, τα χαρακτηριστικά, η δομή και ο τρόπος λειτουργίας της πλακέτας. Γίνεται αναφορά στο Software, τη γλώσσα προγραμματισμού, τις κυριότερες εντολές και τα Shields που χρησιμοποιεί η πλακέτα Arduino.

Στο τέταρτο κεφάλαιο γίνεται αναλυτική περιγραφή για τον τρόπο λειτουργίας της κατασκευής που υλοποιήσαμε. Επίσης γίνεται εκτενής αναφορά στην υπηρεσία SMTP2GO και στην web εφαρμογή. Τέλος εμφανίζεται η μελέτη για την ενεργειακή αυτονομία των υποσυστημάτων.

Στο πέμπτο κεφάλαιο παρουσιάζονται τα εξαρτήματα που χρησιμοποιήθηκαν για τη κατασκευή του συστήματος, τα τεχνικά χαρακτηριστικά και διάφορες ρυθμίσεις που συντέλεσαν στη καλύτερη λειτουργία του συστήματος.

Στο έκτο κεφάλαιο γίνεται λεπτομερής αναφορά στον προγραμματισμό του συστήματος καθώς και μια πλήρης επεξήγηση του κώδικα.

Στο έβδομο κεφάλαιο επισημαίνονται κάποια συμπεράσματα αλλά και κάποιες μελλοντικές επεκτάσεις που μπορούν να υλοποιηθούν ώστε να αναβαθμιστεί το σύστημα.

Και στο τελευταίο κεφάλαιο παρουσιάζονται τα παραρτήματα του κώδικα όλου του συστήματος, η κοστολόγηση και φωτογραφίες της κατασκευής.

1. ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ

1.1. Τα ενσωματωμένα συστήματα ως μέρος της καθημερινότητας

Τη τελευταία δεκαετία τα ενσωματωμένα συστήματα παρόλο που είχαν καθιερωθεί σε προϊόντα χαμηλής λειτουργικότητας (π.χ. ρολόγια, ηλεκτρονικοί υπολογιστές τσέπης), σήμερα χρησιμοποιούνται σε πολλές και πολύπλοκες συσκευές (π.χ. φορητά τερματικά πολυμέσων). Μάλιστα ερευνητές έχουν χαρακτηρίσει τα ενσωματωμένα συστήματα ως την επανάσταση των ηλεκτρονικών συσκευών και αυτό διότι κατασκευάζονται δισεκατομμύρια ενσωματωμένα συστήματα καθημερινά για διάφορες λειτουργίες. Συνήθως χρησιμοποιούνται για εξειδικευμένες εφαρμογές και έχουν περιορισμένη χρήση σε αντίθεση με τους υπολογιστές γενικού σκοπού. Ένας ορισμός που θα μπορούσε να δοθεί είναι ότι ενσωματωμένο σύστημα είναι κάθε ψηφιακό σύστημα που εκτελεί κάποια λειτουργία και δεν είναι φορητός ή προσωπικός υπολογιστής, ή κεντρικός διακομιστής (mainframe). Μερικά παραδείγματα ενσωματωμένων συστημάτων είναι:

- Τηλεοράσεις και βίντεο: Οι τηλεοράσεις περιέχουν ενσωματωμένους επεξεργαστές για τη ρύθμιση καναλιών, τον έλεγχο της εικόνας, την εκτύπωση μηνυμάτων στην οθόνη κ.α.
- Συσκευές τηλε-ελέγχου: Οι συσκευές τηλε-ελέγχου (telecontrol) χρησιμοποιούν ενσωματωμένους επεξεργαστές για να μετατρέπουν τις εντολές του χρήστη σε σήματα υπερύθρων.
- Κινητά / Ασύρματα τηλέφωνα: Τα κινητά τηλέφωνα περιέχουν ενσωματωμένους επεξεργαστές για την αποκωδικοποίηση / κωδικοποίηση της φωνής και την εφαρμογή των πρωτοκόλλων επικοινωνίας τους.
- Οχήματα: Επιπλέον τα σύγχρονα οχήματα λειτουργούν με ενσωματωμένους επεξεργαστές για τη ασφάλεια και διευκόλυνση της οδήγησης των επιβατών.
- Εκτυπωτές / Φαξ: Ενσωματωμένοι επεξεργαστές χρησιμοποιούνται και σε αυτές τις συσκευές για τη διαδικασία και τον έλεγχο της εκτύπωσης. [1]

1.2. Ανίχνευση κίνησης

Σαν ορισμό θα αναφέραμε, ότι ανίχνευση κίνησης είναι η διαδικασία που επιβεβαιώνει την αλλαγή θέσης ενός αντικείμενου σε σχέση με το περιβάλλον του ή την αλλαγή στο περιβάλλοντα χώρο σε σχέση με ένα αντικείμενο. Υπάρχουν δύο μέθοδοι ανίχνευσης, οι μηχανικές (πλήκτρα σε πληκτρολόγιο) και οι ηλεκτρονικές (π.χ. κάμερα, μικρόφωνο, υπέρυθρες, laser). Επιπλέον μπορεί να πάρει διακριτές τιμές, δηλαδή αν υπάρχει κίνηση τον αριθμό 1 ή αν δεν υπάρχει τον αριθμό 0 είτε μπορεί να αποτελείται από ανίχνευση μεγέθους που μπορεί να μετρήσει και να ποσοτικοποιήσει τη δύναμη, τη ταχύτητα της κίνησης. Επίσης η κίνηση μπορεί να ανιχνευθεί από αισθητήρες ήχου, υπέρυθρους, οπτικούς, μαγνητικούς, αισθητήρες υπερήχων και αισθητήρες ραντάρ μικροκυμάτων ή από ανιχνευτές ηλεκτρομαγνητικής επαγωγής. [2]

1.3. Μετάδοση με ραδιοκύματα

Τα ραδιοκύματα χρησιμοποιούνται ευρύτατα σε επικοινωνίες επειδή παράγονται εύκολα, μπορούν να διανύσουν μεγάλες αποστάσεις και λειτουργούν σωστά τόσο σε εσωτερικούς όσο και σε εξωτερικούς χώρους. Είναι μη κατευθυντικά, δηλαδή από τη πηγή ταξιδεύουν προς όλες τις κατευθύνσεις με αποτέλεσμα να μην χρειάζονται ευθυγράμμιση οι πομποδέκτες. Παρ όλα αυτά υπάρχει και ένα ελάττωμα στη μη κατευθυντικότητα, κάτι ανάλογο βίωσε η General Motors τη δεκαετία του 1970 όταν προσπάθησε να εφαρμόσει στις νέες κάντιλακ φρένα που θα ελέγχονταν από υπολογιστή και θα είναι κατά του μπλοκαρίσματος. Δηλαδή όταν ο οδηγός πατούσε το

πεντάλ του φρένου ενεργοποιούσε και απενεργοποιούσε ο υπολογιστής τα φρένα χωρίς να τα μπλοκάρει. Η λειτουργία αυτή είχε ως αποτέλεσμα, στον αυτοκινητόδρομο του Οχάιο να παρεμβάλλεται το σύστημα του αυτοκινήτου με τους ασύρματους της τροχαίας με τρόπο που έκανε το όχημα να συμπεριφέρεται παράξενα. Μετά από πολλή αναζήτηση η εταιρεία ανακάλυψε ότι η καλωδίωση της κάντιλακ σχημάτιζε μια άριστη κεραία για τη συχνότητα που χρησιμοποιούσε το νέο σύστημα ασυρμάτων της τροχαίας στους αυτοκινητόδρομους του Οχάιο. Οι ιδιότητες των ραδιοκυμάτων εξαρτώνται από τη συχνότητα. Στις χαμηλές συχνότητες τα ραδιοκύματα διαπερνούν τα εμπόδια όμως μειώνεται η ισχύς ανάλογα με την απόσταση προέλευσης, αυτή η εξασθένιση ονομάζεται απώλεια διαδρομής. Ενώ στις υψηλές συχνότητες τα ραδιοκύματα μεταφέρονται σε ευθείες γραμμές χωρίς εμπόδια, ανακλώνται σε αυτά, έχουν απώλειες ισχύος λόγω της απώλειας διαδρομής και απορροφούνται από τη βροχή σε μεγαλύτερο βαθμό από τις χαμηλότερες συχνότητες. Επίσης ενδιαφέρουσα είναι η σύγκριση της εξασθένισης των ραδιοκυμάτων με τα σήματα των κατευθυνόμενων μέσων. Στις οπτικές ίνες, στα ομοαξονικά καλώδια και στα καλώδια σύστροφου ζεύγους παρατηρείται ότι το σήμα μειώνεται κατά το ίδιο κλάσμα ανά μονάδα απόστασης, για παράδειγμα 20 dB ανά 100 μέτρα στα καλώδια συνεστραμμένων ζευγών. Ενώ στα ραδιοκύματα το σήμα ελαττώνεται κατά το ίδιο κλάσμα καθώς διπλασιάζεται η απόσταση, για παράδειγμα 6 dB ανά διπλασιασμό ελεύθερου χώρου. Αυτό σημαίνει ότι τα ραδιοκύματα που ταξιδεύουν σε μεγάλες αποστάσεις παρεμβάλλονται συχνότερα από άλλες πηγές ή παράσιτα. Γι αυτό το λόγο οι κυβερνήσεις ελέγχουν αυστηρά τη χρήση πομπών ραδιοκυμάτων. Τέλος στις ζώνες VLF, LF, MF τα ραδιοκύματα ακολουθούν την επιφάνεια της γης και σε χαμηλές συχνότητες μπορούν να ανιχνευθούν σε απόσταση μεγαλύτερες των 1000 km ενώ στις υψηλές συχνότητες σε μικρότερη απόσταση. Αντίθετα στις ζώνες HF, VHF κάποια σήματα απορροφούνται στο έδαφος και τα υπόλοιπα ταξιδεύουν στην ιονόσφαιρα, διαθλώνται και επιστρέφουν στην γη. [3]

2. ΜΙΚΡΟΕΛΕΓΚΤΕΣ

2.1. Τι είναι ο μικροελεγκτής

Μικροελεγκτής είναι ένας τύπος επεξεργαστή που μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα επειδή περιέχει ενσωματωμένα πολλά υποσυστήματα. Χρησιμοποιείται ευρέως σε όλα τα ενσωματωμένα συστήματα αυτοματισμών, ηλεκτρονικών και ηλεκτρικών συσκευών είτε χαμηλού είτε υψηλού κόστους. [4]

2.2. Διαφορές μικροελεγκτή με μικροεπεξεργαστή

Οι μικροεπεξεργαστές για μη ενσωματωμένα συστήματα (πχ. των προσωπικών υπολογιστών) εκτελούν πολλές διαφορετικές εφαρμογές αφού η λειτουργικότητα του τελικού συστήματος καθορίζεται από τα εξωτερικά περιφερειακά τα οποία συνδέονται στη κεντρική μονάδα (μικροεπεξεργαστής) γι αυτό και δίνεται μεγάλο βάρος στην υπολογιστική ισχύ. Αντίθετα οι μικροελεγκτές στα ενσωματωμένα συστήματα χρειάζονται χαμηλή υπολογιστική ισχύ επειδή δεν συνεργάζονται με εξωτερικά περιφερειακά και οι εφαρμογές που εκτελούν είναι λιγιστές. Κυρίως δίνουν έμφαση στα ολοκληρωμένα κυκλώματα που απαιτούνται για τη λειτουργία μιας συσκευής, το χαμηλό κόστος και την εξειδίκευση.

Τα πλεονεκτήματα ενός μικροελεγκτή είναι τα εξής:

- Αυτονομία. Πολλοί μικροελεγκτές επειδή ενσωματωμένα περιέχουν μνήμες και θύρες επικοινωνίας δεν χρειάζονται άλλα ολοκληρωμένα κυκλώματα για να λειτουργήσουν.
- Η ενσωμάτωση περιφερειακών σημαίνει ευκολότερη εκτέλεση εφαρμογών, χαμηλή κατανάλωση ισχύος, μεγιστοποίηση της φορητότητας.
- Χαμηλό κόστος.
- Μεγάλη αξιοπιστία λόγω λιγότερων διασυνδέσεων.
- Λόγω των χαμηλών ταχυτήτων λειτουργίας και του μικρού μήκους των ενσωματωμένων εξωτερικών διασυνδέσεων ο μικροελεγκτής έχει μειωμένη ευαισθησία και μειωμένες εκπομπές παρεμβολών από άλλες ηλεκτρονικές συσκευές.
- Διαθέτει περισσότερους ακροδέκτες αφού δεν δεσμεύονται για εξωτερικά περιφερειακά.
- Μικρό μέγεθος συνολικού υπολογιστικού συστήματος.

Η αρχιτεκτονική των μικροελεγκτών είναι σχετικά κοινή με των μικροεπεξεργαστών αν και στη πρώτη επικρατεί η αρχιτεκτονική μνήμης τύπου Harvard η οποία χρησιμοποιεί διαφορετικές αρτηρίες σύνδεσης της μνήμης προγράμματος και της μνήμης δεδομένων (πχ. AVR από την ATMEL και PIC από την Microchip) ενώ στους μικροεπεξεργαστές επικρατεί η αρχιτεκτονική τύπου Φον Νοϊμαν. [4]

2.3. Υποσυστήματα

Το ολοκληρωμένο κύκλωμα του μικροεπεξεργαστή αποτελείται από τη λογική και αριθμητική μονάδα (ALU), από καταχωρητές (registers), από την υψηλής ταχύτητας προσωρινή μνήμη (cache memory) και τον ελεγκτή μνήμης (memory controller). Αντίθετα για ένα πλήρες ενσωματωμένο υπολογιστικό σύστημα απαιτούνται πολλά εξωτερικά περιφερειακά υποσυστήματα όπως:

- Μνήμη προγράμματος (ROM, FLASH, EPROM κλπ) η οποία περιέχει το λογισμικό του συστήματος.
- Μεγάλο μέγεθος RAM.
- Μόνιμη μνήμη αποθήκευσης λειτουργίας (τύπου EEPROM ή NVRAM). Πλεονέκτημα της μνήμης αυτής είναι η διαγραφή και εγγραφή μεμονωμένων bytes.

- Κύκλωμα αρχικοποίησης (reset).
- Διαχειριστή αιτήσεων διακοπής (interrupt request controller) από τα περιφερειακά.
- Κύκλωμα επιτήρησης τροφοδοσίας (brown-out detection) το οποίο παρακολουθεί την τροφοδοσία και αρχικοποιεί ολόκληρο το σύστημα αν αυτή πέσει κάτω από τα ανεκτά όρια, προλαμβάνοντας έτσι την αλλοίωση των δεδομένων.
- Τοπικό ταλαντωτή για τη παροχή παλμών χρονισμού (clock).
- Ρολόι πραγματικού χρόνου (Real Time Clock, RTC) το οποίο έχει πολύ χαμηλή κατανάλωση ρεύματος και τροφοδοτείται από ανεξάρτητη μπαταρία.
- Σειρά ανεξάρτητων ψηφιακών εισόδων και εξόδων (Parallel Input-Output, PIO).

Τα περισσότερα υποσυστήματα που προαναφέρθηκαν χρησιμοποιούνται σε όλες τις οικογένειες μικροελεγκτών με κάποιες διαφοροποιήσεις, κυρίως στη στην ύπαρξη εσωτερικής ή μη μνήμης προγράμματος. Έτσι υπάρχουν :

- Οι μικροελεγκτές ROM-less οι οποίοι είναι χωρίς μνήμη προγράμματος και συνδέονται με εξωτερικές μνήμες και RAM. Τέτοιοι μικροελεγκτές χρησιμοποιούνται περισσότερο σε μεγάλα υπολογιστικά συστήματα με μεγάλες απαιτήσεις μνήμης.
- Οι μικροελεγκτές με μνήμη ROM. Κατασκευάζονται με το λογισμικό της (Mask ROM) και είναι πολύ φτηνοί όταν αγοράζονται σε μεγάλες ποσότητες.
- Οι μικροελεγκτές με μνήμη FLASH. Είναι η πιο διαδεδομένη κατηγορία και μπορούν να προγραμματιστούν πολλές φορές πάνω στο κύκλωμα της ίδιας ενσωματωμένης εφαρμογής (δυνατότητα In Circuit Programming, ISP). Αυτοί οι μικροελεγκτές έχουν αντικαταστήσει τους παλαιότερους τύπους EPROM που έσβηναν με υπεριώδη ακτινοβολία (από το ειδικό τζαμάκι). [4]

2.4. Πρόσθετες λειτουργίες

Για τις πρόσθετες εφαρμογές ένας μικροελεγκτής μπορεί να περιέχει :

- Ασύγχρονες (UART) και σύγχρονες (I^2C , SPI, Ethernet) σειριακές θύρες επικοινωνίας.
- Εισόδους για μετατροπή από αναλογικό σε ψηφιακό σήμα (analog to digital converter).
- Μετατροπέα ψηφιακού σε αναλογικό σήμα (digital to analog converter)
- Ελεγκτή οθόνης (πχ. LCD).
- Υποσυστήματα για άμεση υποστήριξη από υλικολογισμικό (firmware) για πιο σύνθετα πρωτόκολλα επικοινωνίας (πχ. CAN, ADSL, ISDN).
- Μονάδα άμεσης εκτέλεσης πράξεων κινητής υποδιαστολής (FPU) καθώς και δυνατότητες ψηφιακής επεξεργασίας σήματος (Digital Signal Processing, DSP).
- Περιέχει υποσύστημα προγραμματισμού πάνω στο κύκλωμα (τύπου ISP) αυτό σημαίνει ότι μπορεί να συνδεθεί μία εξωτερική συσκευή πάνω στη συσκευή μας και να επαναπρογραμματιστεί. Συνήθως γίνεται με θύρα (UART RS-232) ή από το διαδίκτυο.
- Περιέχει υποσύστημα προγραμματισμού (ISP) και διάγνωσης με σκοπό τον προγραμματισμό της μνήμης προγράμματος. [4]

2.5. Κατηγορίες μικροελεγκτών

Λόγω της ενσωμάτωσης μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή αλλά και του ισχυρού ανταγωνισμού που δημιουργήθηκε, η βιομηχανία μικροελεγκτών εστίασε στη μαζική παραγωγή μοντέλων και μικροελεγκτών για πιο εξειδικευμένες εφαρμογές. Έτσι διακρίνονται σε:

- Μικροελεγκτές 4bit ή 8bit με χαμηλό κόστος και κατανάλωση ισχύος με μικρό αριθμό ακροδεκτών ώστε να χρειάζονται λίγα έως και καθόλου επιπλέον εξαρτήματα. Επιπλέον δεν επιτρέπεται η επέκταση μνήμης. Τέτοιοι μικροελεγκτές είναι των σειρών PIC (Microchip), AVR (Atmel), 8051 (Intel, Atmel, Dallas).
- Μικροελεγκτές 8bit ή 16bit ή 32bit χαμηλού κόστους με περισσότερους ακροδέκτες και διαθέτουν κοινά περιφερειακά όπως θύρες UART, I²C, SPI, CAN, μετατροπείς αναλογικού σε ψηφιακό σήμα και το αντίστροφο, κάποιιοι έχουν τη δυνατότητα επέκτασης μνήμης ενώ οι κατασκευαστές της άπω ανατολής εγκαθιστούν και ελεγκτές οθόνης υγρών κρυστάλλων και πληκτρολογίου.
- Μικροελεγκτές 32 bit που είναι πιο ακριβοί από τους προηγούμενους και διαθέτουν πολλούς ακροδέκτες, υποστηρίζουν αρκετά περιφερειακά, έχουν δυνατότητες εσωτερικής ή εξωτερικής μνήμης προγράμματος (FLASH) και RAM και εστιάζουν στην ταχύτητα εκτέλεσης εντολών.
- Μικροελεγκτές εξειδικευμένων εφαρμογών, λειτουργούν με εξειδικευμένα πρωτόκολλα επικοινωνίας και χρησιμοποιούνται σε τηλεπικοινωνιακές συσκευές (πχ. Μόντεμ).

Τέλος οι περισσότερες πωλήσεις γίνονται σε 8bit μικροελεγκτές λόγω του χαμηλού κόστους και του μικρού μεγέθους. [4]

2.6. Γλώσσες προγραμματισμού και εργαλεία ανάπτυξης

Η επιτυχία ενός μικροελεγκτή αποτελείται σε μεγάλο βαθμό από τον προγραμματισμό και τα εργαλεία ανάπτυξης. Δηλαδή πρέπει να υπάρχουν μεταφραστές που να μετατρέπουν τις γλώσσες υψηλού επιπέδου σε γλώσσα κατανοητή από ένα μικροελεγκτή (Assembly), πρέπει να υπάρχουν εργαλεία εκσφαλμάτωσης (debuggers) και προγραμματιστές εσωτερικής μνήμης. Τα εργαλεία αυτά δεν παρέχονται μέσα στο λογισμικό του μικροελεγκτή αφού δεν υπάρχει τυποποιημένος τρόπος επικοινωνίας, γι αυτό χρησιμοποιούν προγραμματιστές εσωτερικής μνήμης μέσω θύρας JTAG ή USB και έτοιμες πλακέτες με ψηφιακές ή και αναλογικές εξόδους. Επιπλέον περιέχουν λογισμικό ανάπτυξης εφαρμογών και έτοιμα παραδείγματα για την ευκολότερη χρήση του χρήστη. Η πιο διαδεδομένη γλώσσα προγραμματισμού μικροελεγκτών είναι η C και C++ παρ όλα αυτά σε εφαρμογές που απαιτείται ταχύτητα και μικρή δέσμευση μνήμης χρησιμοποιείται η Assembly. Τέλος στο τομέα εργαλείων ανάπτυξης δραστηριοποιούνται εξειδικευμένες εταιρείες πέρα από τους κατασκευαστές μικροελεγκτών. [4]

2.7. Κατασκευαστές

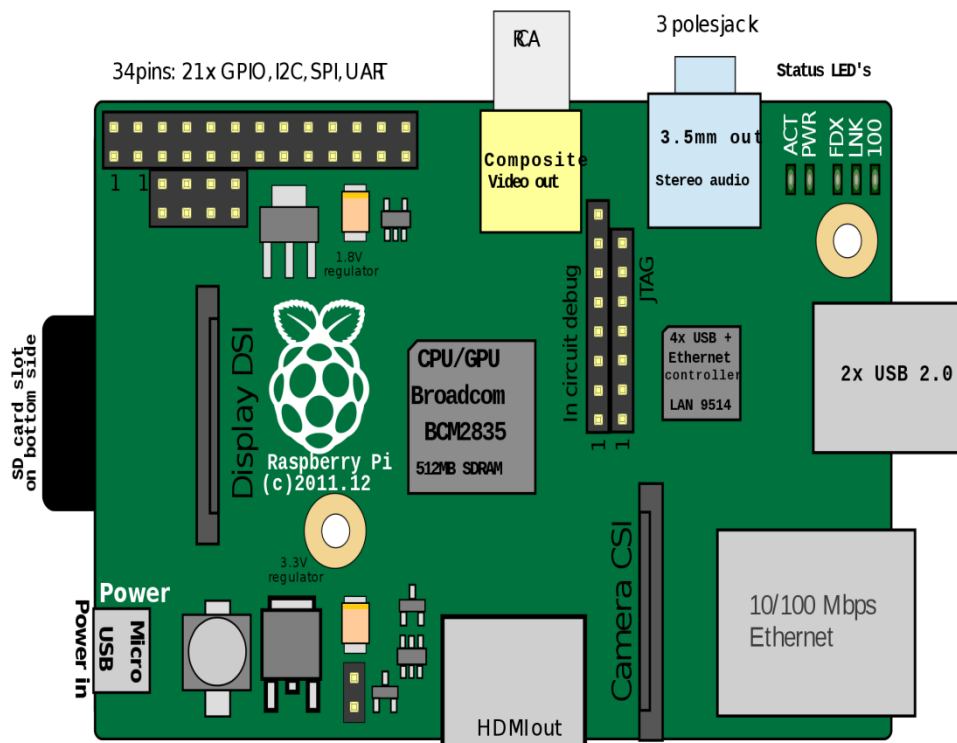
Μερικές γνωστές εταιρείες που κατασκευάζουν μικροελεγκτές είναι :

- ARM (δεν κατασκευάζει αλλά παραχωρεί δικαιώματα χρήσης του πυρήνα)
- Atmel
- Freescale Semiconductor (πρώην Motorola)
- Microchip
- Toshiba
- Texas Instruments
- NEC
- HITACHI
- EPSON
- MAXIM [4]

2.8. Ο υπολογιστής Raspberry Pi

Το Raspberry Pi δημιουργήθηκε και αναπτύχθηκε στο Ηνωμένο Βασίλειο από τον μη κερδοσκοπικό οργανισμό Raspberry Pi Foundation με σκοπό την εκπαίδευση στην επιστήμη των υπολογιστών. Κάποια από τα πλεονεκτήματα του είναι το χαμηλό κόστος, το πολύ μικρό μέγεθος (υπολογιστής σε μέγεθος πιστωτικής κάρτας), όμως η μεγαλύτερη διαφορά από έναν υπολογιστή είναι ότι διαθέτει 40 ακροδέκτες γενικού σκοπού με γενική ονομασία GPIO (General Purpose Input Output). Σε αυτούς τους ακροδέκτες μπορούν να συνδεθούν αισθητήρες και ηλεκτρονικά εξαρτήματα. Το λειτουργικό σύστημα είναι το Raspbian έκδοση του Debian Linux και μπορεί να εγκατασταθεί μέσω μιας microSD. Η προτεινόμενη γλώσσα προγραμματισμού είναι η Python (εξ ου και η ονομασία Pi) όμως μπορεί να προγραμματιστεί σε διάφορες γλώσσες και περιέχει πολλά έτοιμα παραδείγματα. Τα τεχνικά χαρακτηριστικά της πρώτης έκδοσης του είναι τα εξής:

- Τετραπύρηνο ARM επεξεργαστή στα 900 MHz.
- 1 GB μνήμη RAM.
- Υποδοχή Ethernet 100 Mbps.
- 4 θύρες USB 2.0 για σύνδεση περιφερειακών συσκευών.
- 40 pins GPIO για σύνδεση αισθητήρων και άλλων εξαρτημάτων.
- Έξοδο HDMI για σύνδεση οθόνης.
- Ειδική υποδοχή για τη σύνδεση κάμερας (επίσημη κάμερα του Raspberry Pi).
- Υποδοχή microSD.
- Υποδοχή microUSB για τη τροφοδοσία με ρεύμα. [6]



Εικόνα 1 Κάτοψη του υπολογιστή Raspberry Pi

2.9. Το πρότυπο ZigBee

Το Zigbee είναι ένα πρότυπο που αναπτύχθηκε για να καλύψει ανάγκες χαμηλού κόστους, χαμηλής ισχύος σε ασύρματα δίκτυα αισθητήρων. Προήλθε από την εταιρεία Zigbee Alliance σε συνεργασία με την επιτροπή IEEE 802.15.4. Συγκεκριμένα, το Zigbee είναι το όνομα μιας προδιαγραφής για μια ακολουθία υψηλού πρωτοκόλλου επικοινωνίας που χρησιμοποιούν οι ψηφιακοί δέκτες χαμηλής ισχύος βασισμένοι στο πρότυπο 802.15.4 για τα ασύρματα τοπικά δίκτυα. Η τεχνολογία αυτή έχει ως στόχο να λειτουργεί σε εφαρμογές ραδιοσυχνότητας, που απαιτούν ένα μικρό ρυθμό μεταφοράς δεδομένων, μεγάλη ζωή μπαταριών, εξασφαλισμένη δικτύωση και χαμηλό κόστος. Επιπλέον, χρησιμοποιείται σε ζώνες που δεν χρειάζεται άδεια σε συχνότητες 2.400 -2.484 GHz, 902 - 928 MHz και 868.0 - 868.6 MHz. Τα χαρακτηριστικά του προτύπου ZigBee είναι τα εξής:

- Χαμηλός κύκλος καθηκόντων - Μεγάλη ζωή μπαταριών.
- Χαμηλή λανθάνουσα κατάσταση.
- Άμεσο απλωμένο φάσμα ακολουθίας (DSSS).
- Χρησιμοποιείται σε αρκετές τοπολογίες: Στατικό, Δυναμικό, Πλέγμα και Αστέρι.
- Μέχρι 65.000 κόμβοι σε ένα δίκτυο.
- Αποφυγή συγκρούσεων.
- Ποιοτική εμφάνιση συνδέσεων.
- Σαφής αξιολόγηση καναλιών.
- 128-bit AES encryption κρυπτογράφηση - παρέχει ασφαλείς συνδέσεις μεταξύ των συσκευών. [5]

Market Name	ZigBee™	GPRS/GSM	Wi-Fi™	Bluetooth™
Standard	802.15.4	1xRTT/CDMA	802.11b	802.15.1
Application Focus	Monitoring & Control	Wide Area Voice & Data	Web, Email, Video	Cable Replacement
System Resources	4KB - 32KB	16MB+	1MB+	250KB+
Battery Life (days)	100 - 1,000+	1-7	.5 - 5	1 - 7
Network Size	Unlimited (2 ⁶⁴)	1	32	7
Bandwidth (K _b /s)	20 - 250	64 - 128+	11,000+	720
Transmission Range (meters)	1 - 100+	1,000+	1 - 100	1 - 10+
Success Metrics	Reliability, Power, Cost	Reach, Quality	Speed, Flexibility	Cost, Convenience

Εικόνα 2 Τα πλεονεκτήματα του ZigBee σε σχέση με άλλα πρότυπα

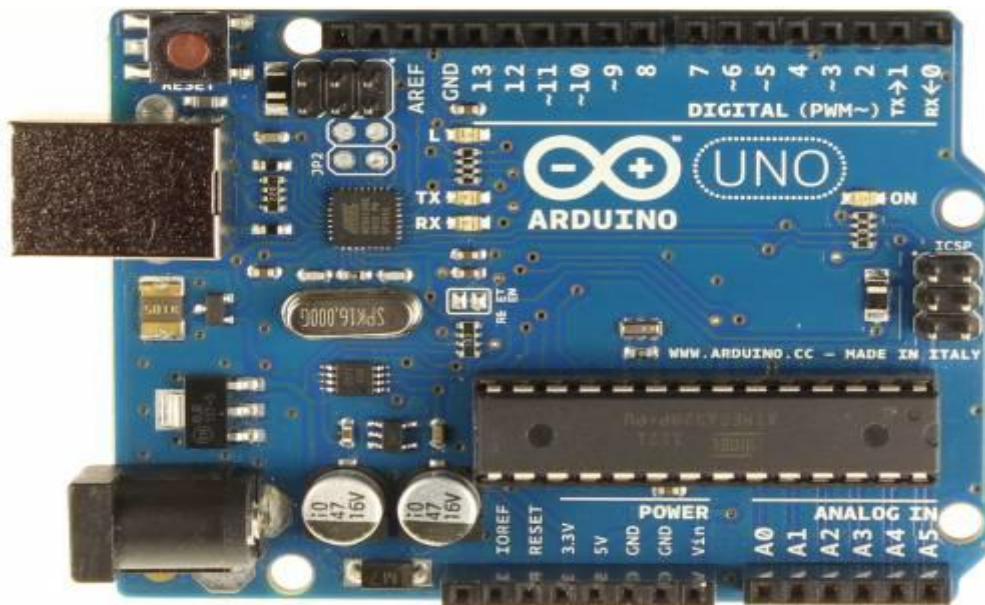
3. Η ΗΛΕΚΤΡΟΝΙΚΗ ΠΛΑΤΦΟΡΜΑ ARDUINO

3.1. Ιστορική Αναδρομή

Το Arduino πήρε το όνομα του από τον Arduin που ήταν στρατιωτικός διοικητής της Ιβρέα και αργότερα βασιλιάς της Ιταλίας. Όλα ξεκίνησαν όταν ένας φοιτητής το 2003 στη πόλη της Ιβρέα ανέπτυξε το wiring project. Το wiring project ήταν ένα απλό σύστημα που αποτελούνταν από μικροελεγκτή και λειτουργούσε για την ανάπτυξη εύκολων και γρήγορων εφαρμογών. Το Arduino ήταν μια επέκταση του wiring project που είχε ως σκοπό να συνδέσει τους ανθρώπους με τη τεχνολογία και γρήγορα χρησιμοποιήθηκε από πολύ κόσμο (φοιτητές, μηχανικούς κ.α). [7]

3.2. Τι είναι το Arduino

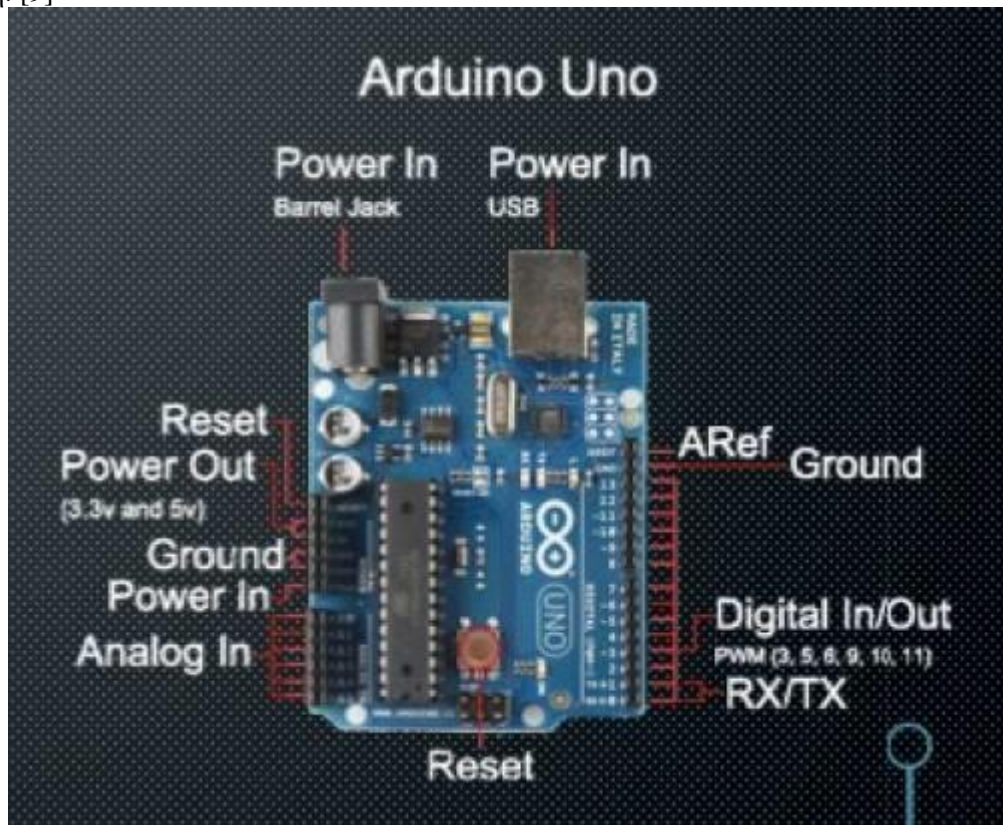
Το Arduino είναι μια ηλεκτρονική πλατφόρμα “ανοιχτού κώδικα” το οποίο μπορεί κάποιος να χρησιμοποιήσει για να κατασκευάσει εφαρμογές ρομποτικής και συστήματα αυτοματισμού. Βασικό πλεονέκτημα αυτής της πλακέτας είναι ότι δεν χρειάζεται ιδιαίτερες γνώσεις προγραμματισμού και υποστηρίζεται από πολύ μεγάλη κοινότητα. Το Arduino αποτελείται από τον μικροεπεξεργαστή Atmega της Atmel και έχει τη δυνατότητα να συνδέει μονάδες εισόδου/εξόδου οι οποίες χωρίζονται σε ψηφιακές και αναλογικές. Η πλακέτα Arduino είναι συμβατή με μεγάλη πληθώρα συσκευών. Κάποιες από αυτές είναι αισθητήρες υγρασίας, θερμοκρασίας, θερμότητας, απόστασης, δύναμης, γυροσκόπια, επιτάχυνσης, πίεσης κ.α. Επιπλέον μέσω του Arduino μπορούμε να ελέγξουμε φώτα (220V), leds, servo κινητήρες, stepper motors, DC motors, ρελέ κ.α. Ο προγραμματισμός του Arduino μπορεί να γίνει από τον υπολογιστή μέσω της σειριακής θύρας που υποστηρίζει ο μικροεπεξεργαστής Atmega από τη θύρα USB του υπολογιστή. Η σειριακή σύνδεση (Serial over USB) χρησιμοποιείται για τη μεταφορά προγραμμάτων από τον υπολογιστή προς τη πλακέτα αλλά και το αντίστροφο αφού λαμβάνει δεδομένα από τις συσκευές και τα μεταφέρει στον υπολογιστή. [8]



Εικόνα 3 Κάτοψη της ηλεκτρονικής πλατφόρμας Arduino UNO

3.3. Το Hardware

Το Arduino είναι σχεδιασμένο πάνω σε μια πλακέτα PCB (printed circuit board) στο οποίο είναι τυπωμένες όλες οι διασυνδέσεις του κυκλώματος. Οι διαστάσεις του είναι 2.7 x 2.1 ίντσες. Το μέγεθος αυτό κάνει τη πλακέτα πολύ εύχρηστη αφού μπορεί να τοποθετηθεί με ευκολία σε μικρές αυτόνομες συσκευές (πχ. Ρομπότ, τηλεκατευθυνόμενο όχημα) ή ακόμη και σε μη εμφανή σημεία. Η πλακέτα διαθέτει τον μικροεπεξεργαστή, ψηφιακές / αναλογικές υποδοχές, σταθεροποιητή τάσης, υποδοχές για τροφοδοσία, κουμπί επαναφοράς (reset) και υποδοχές γείωσης. Επιπλέον η τροφοδοσία του συστήματος μπορεί να επιτευχθεί είτε με καλώδιο USB είτε με κυλινδρικό κονέκτορα (barrel jack) και το Arduino UNO διαθέτει ένα παθητικό ηλεκτρονικό εξάρτημα γνωστό ως resettable polyfuse το οποίο προστατεύει από βραχυκυκλώματα τις θύρες USB του υπολογιστή. [9]



Εικόνα 4 Τα χαρακτηριστικά της ηλεκτρονικής πλατφόρμας

3.4. Μικροελεγκτής – Η καρδιά του Arduino

Το Arduino βασίζεται στο μικροελεγκτή ATmega328 ο οποίος είναι 8 bit RISC, χρονίζεται στα 16 MHz και διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 2 kb SRAM μνήμη η οποία χρησιμοποιείται στα προγράμματα για την αποθήκευση μεταβλητών, πινάκων κτλ. Αυτή η μνήμη λειτουργεί όπως και στον υπολογιστή δηλαδή χάνει τα δεδομένα της αν σταματήσει η παροχή ρεύματος ή γίνει reset στη πλακέτα Arduino.
- 1 kb μνήμης EEPROM χρησιμοποιείται για εγγραφή ή ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματα κατά το runtime. Αντίθετα με την SRAM δεν χάνονται τα δεδομένα αν σταματήσει η παροχή ρεύματος ή αν γίνει reset στη πλακέτα. Είναι κάτι ανάλογο του σκληρού δίσκου.

- 32 kb flash μνήμη, εκ των οποίων τα 2kb χρησιμοποιούνται για το firmware ή αλλιώς bootloader στην ορολογία του Arduino και είναι εγκατεστημένο από το κατασκευαστή του. Επίσης το firmware είναι απαραίτητο για την εγκατάσταση των προγραμμάτων μας στον μικροελεγκτή μέσω της θύρας USB χωρίς να χρειάζεται εξωτερικός hardware programmer. Ενώ τα υπόλοιπα 30 kb χρησιμοποιούνται για την αποθήκευση των προγραμμάτων αφού πρώτα μεταγλωττιστούν στον υπολογιστή. Όπως η EEPROM έτσι και η flash μνήμη δεν χάνει τα δεδομένα της αν διακοπεί η παροχή ρεύματος ή αν γίνει reset στο Arduino. [10]

3.5. Χαρακτηριστικά του Arduino

3.5.1. Είσοδοι – Έξοδοι

Στη πλακέτα Arduino υπάρχει ελεγκτής serial over USB ώστε να επιτυγχάνεται σειριακή επικοινωνία μεταξύ του μικροελεγκτή και του υπολογιστή. Η σύνδεση γίνεται μέσω USB και χρησιμοποιείται για τη μεταφορά προγραμμάτων από τον υπολογιστή στο Arduino αλλά και το αντίστροφο δημιουργώντας μια αμφίδρομη επικοινωνία μεταξύ τους μέσα από το πρόγραμμα την ώρα που εκτελείται. Επιπλέον στην πάνω πλευρά του Arduino υπάρχουν 14 ακροδέκτες (pin) που λειτουργούν με 5V τάση και μπορούν να δεχτούν το μέγιστο 40mA ρεύμα. Είναι αριθμημένα από το 0 έως το 13 και μπορούν να συμπεριφερθούν ως ψηφιακοί έξοδοι ή είσοδοι. Ως ψηφιακή έξοδος κάποιο από αυτά τα pin μπορεί να τεθεί σε κατάσταση HIGH or LOW από το πρόγραμμα μας και με αυτό το τρόπο καταλαβαίνει το Arduino αν θα διοχετεύσει ρεύμα ή όχι στο pin. Για παράδειγμα αν σε ένα συγκεκριμένο pin έχουμε συνδέσει ένα led μπορούμε να το ανάψουμε ή να το σβήσουμε. Αν πάλι θέσουμε από το πρόγραμμα μας ως ψηφιακή είσοδο κάποιο από αυτά τα pin μπορούμε με την κατάλληλη εντολή να διαβάσουμε τη κατάσταση του (HIGH or LOW). Παράλληλα κάποια από τα 14 pin έχουν κι άλλη λειτουργία:

- Αν ενεργοποιήσουμε το σειριακό interface τα pin 0 και 1 λειτουργούν ως RX & TX. Αυτό σημαίνει ότι αν στέλνουμε δεδομένα από το πρόγραμμα μας μέσω του καλωδίου USB, τα δεδομένα προωθούνται στη θύρα USB μέσω του ελεγκτή serial over USB αλλά και στο pin 0 για να τα διαβάσει μία άλλη συσκευή. Το μειονέκτημα είναι ότι με αυτό το τρόπο χάνονται 2 ψηφιακά pin.
- Τα pin 2 και 3 λειτουργούν ως εξωτερικά interrupt. Δηλαδή μπορούμε να θέσουμε στα συγκεκριμένα pin να συμπεριφέρονται μόνο ως ψηφιακοί είσοδοι στις οποίες αν συμβαίνουν συγκεκριμένες αλλαγές να σταματάει η κανονική ροή του προγράμματος για να εκτελείται μια συγκεκριμένη συνάρτηση. Σε εφαρμογές που απαιτούν μεγάλη ακρίβεια τα εξωτερικά interrupt είναι πολύ χρήσιμα.
- Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και σαν ψευδοαναλογικές έξοδοι με το σύστημα PWM (pulse width modulation). Με αυτό το τρόπο μπορούμε να συνδέσουμε ένα led σε κάποιο από τα παραπάνω pin και να ελέγξουμε τη φωτεινότητα του με ανάλυση 8bit (256 καταστάσεις, 0 κλειστό – 255 πλήρως φωτεινό) αντί να είναι μόνο αναμμένο ή μόνο σβηστό όπως με τις υπόλοιπες ψηφιακές εξόδους. Βέβαια το PWM δεν είναι πραγματικό αναλογικό σύστημα αλλά στέλνει ένα παλμό που θα εναλλάσσεσαι με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

Στην κάτω πλευρά του Arduino υπάρχουν 6 Pin αριθμημένα από 0 έως 5 τα οποία λειτουργούν ως αναλογικές είσοδοι με το σύστημα ADC (Analog to Digital Converter) που είναι ενσωματωμένο στο μικροελεγκτή. Έτσι με ένα ποτενσιόμετρο μπορούμε να ρυθμίσουμε πόση τάση θα περάσει σε κάποιο pin, αν δεν κάνουμε κάποια αλλαγή η τιμή θα είναι στα 5V. Επιπλέον μέσα από το πρόγραμμα μας μπορούμε να διαβάσουμε τη τιμή του pin ως ένα ακέραιο αριθμό από το 0 ως 1023 (10 bit). Τέλος με τη κατάλληλη εντολή μπορούν τα 6 pin να μετατραπούν σε ψηφιακά pin εισόδου / εξόδου όπως τα 14 pin που αναφέρθηκαν παραπάνω. [10]



Εικόνα 5 Ψηφιακές υποδοχές του Arduino UNO



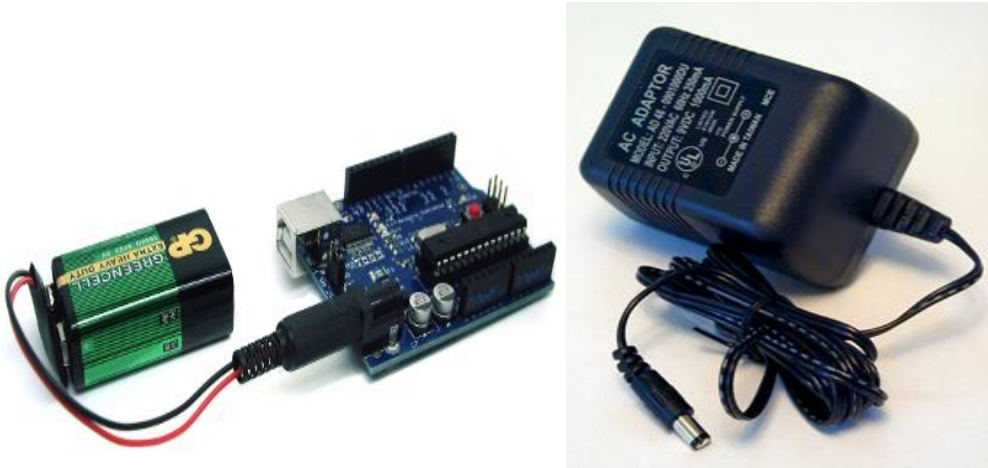
Εικόνα 6 Αναλογικές υποδοχές του Arduino UNO

3.5.2. Τροφοδοσία

Η τροφοδοσία του Arduino με ρεύμα μπορεί να επιτευχθεί είτε από εξωτερική τροφοδοσία (μπαταρία, μετασχηματιστής) είτε από τον υπολογιστή μέσω της σύνδεσης USB. Όμως για να μην υπάρξει κάποιο πρόβλημα στη πλακέτα η εξωτερική τροφοδοσία θα πρέπει να είναι από 7 ως 12V. Υπάρχουν ακόμη 6 τελευταία pin με τη σήμανση POWER και βρίσκονται δίπλα από τα αναλογικά pin. Η λειτουργία τους είναι:

- Το πρώτο που έχει την ένδειξη reset αν γειωθεί με οποιοδήποτε GND pin που βρίσκεται στο Arduino έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- Το δεύτερο με την ένδειξη 3.3V μπορεί να τροφοδοτήσει οποιοδήποτε εξάρτημα με τάση 3.3V. Η τάση αυτή προέρχεται από το ελεγκτή serial over USB και όχι από εξωτερική τροφοδοσία. Η μέγιστη ένταση που μπορεί να πάρει είναι 50 mA.
- Το τρίτο pin με την ένδειξη 5V μπορεί να τροφοδοτήσει οποιοδήποτε εξάρτημα με τάση ανάλογη των 5V. Αυτό μπορεί να επιτευχθεί είτε μέσω του USB είτε από εξωτερική τροφοδοσία αφού πρώτα περάσει από το ρυθμιστή τάσης ώστε να ρίξει τη τάση στα 5V.

- Το τέταρτο και πέμπτο pin είναι γειώσεις.
- Το έκτο και τελευταίο pin είναι το Vin και μπορεί να χρησιμοποιηθεί με 2 τρόπους. Επειδή δίπλα του υπάρχει το pin της γείωσης μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας και να αντικαταστήσει το φως των 2.1 mm. Διαφορετικά αν το Arduino τροφοδοτείται ήδη με την υποδοχή του φως μπορεί μέσω αυτού του pin να τροφοδοτήσει εξαρτήματα που χρειάζονται πλήρη τάση (7 ~ 12V) χωρίς να περάσει από τον ρυθμιστή τάσης. [10]



Εικόνα 7 Εξωτερική τροφοδοσία της πλακέτας Arduino (Μπαταρία/ Μετασχηματιστής)

3.5.3. Ενσωματωμένα κουμπιά και Leds

Τέλος στη πλακέτα Arduino υπάρχει ένας διακόπτης και 4 led επιφανειακής στήριξης. Ο διακόπτης έχει τη σήμανση reset και αν πατηθεί κάνει επανεκκίνηση το Arduino. Υπάρχει ένα led με τη σήμανση power και δηλώνει ότι η πλακέτα είναι έτοιμη για λειτουργία. Υπάρχουν ακόμη 2 led με τις σημάνσεις TX και RX τα οποία δηλώνουν ότι μεταφέρονται δεδομένα μέσω USB (προσοχή, ελέγχονται από τον ελεγκτή serial over USB και δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται μέσω των ψηφιακών Pin 0 και 1). Επιπλέον υπάρχει ένα τελευταίο led με τη σήμανση L. Οι κατασκευαστές δημιούργησαν αυτό το led και το σύνδεσαν με το ψηφιακό pin 13 ώστε να μπορεί κάποιος να υλοποιήσει κάποιο project αναβοσβήνοντας ένα led ακόμα και αν δεν έχει καμία εξωτερική συσκευή παρά μόνο τη πλακέτα Arduino. [10]

3.6. To Software

Η διαχείριση του Arduino στον υπολογιστή γίνεται με το πρόγραμμα Arduino IDE το οποίο δουλεύει και στα 3 δημοφιλέστερα λειτουργικά συστήματα. Τα πλεονεκτήματα του Arduino IDE είναι ότι έχει πρακτικό περιβάλλον με συντακτική χρωματική σήμανση για τη συγγραφή των προγραμμάτων (sketch ονομάζονται στην ορολογία του Arduino), διαθέτει παραδείγματα αλλά και πολλές έτοιμες βιβλιοθήκες για προέκταση της γλώσσας. Επίσης περιέχει τον compiler για τη μεταγλώττιση των sketch και την επιλογή να περάσει το sketch στο Arduino και ένα serial monitor για να παρακολουθεί την επικοινωνία του Arduino με τον υπολογιστή μέσω του USB καλωδίου. Βέβαια για τα δύο τελευταία θα πρέπει να έχει συνδεθεί η πλακέτα με τον υπολογιστή με ένα καλώδιο USB (τύπου A ή τύπου B όπως αυτά των εκτυπωτών) σε οποιαδήποτε θύρα και να αναγνωριστεί από το λειτουργικό σύστημα ως εικονική σειριακή θύρα. Για την αναγνώριση θα πρέπει να εγκατασταθεί ο οδηγός του FTDI chip ο οποίος υπάρχει στο φάκελο drivers του Arduino IDE. Αν η εγκατάσταση γίνει σωστά στο μενού tools του κεντρικού παραθύρου του Arduino IDE θα εμφανιστεί στην επιλογή serial port η εικονική σειριακή θύρα (COM# για Windows, /dev/ttyUSBserial## για MACOS και /dev/ttyUSB## για linux). Τέλος ο compiler που

χρησιμοποιείται είναι ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Η γλώσσα του Arduino βασίζεται στη γλώσσα wiring μια παραλλαγή της C / C++ και χρησιμοποιεί όλες τις εντολές, τελεστές, συναρτήσεις και την ίδια σύνταξη με τη C. Πέρα από αυτές όμως υπάρχουν και κάποιες ειδικές εντολές για τη διαχείριση του ειδικού hardware του Arduino, κάποιες από αυτές είναι:

Όρισμα	Είδος	Τύπος	Παράμετροι	Περιγραφή
LOW	Σταθερά	int	–	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
HIGH	Σταθερά	int	–	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
INPUT	Σταθερά	int	–	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
OUTPUT	Σταθερά	int	–	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
pinMode	Εντολή	–	(<i>pin, mode</i>)	Καθορίζει αν το συγκεκριμένο ψηφιακό <i>pin</i> θα είναι pin εισόδου ή pin εξόδου ανάλογα με την τιμή που δίνεται στην παράμετρο <i>mode</i> (INPUT ή OUTPUT αντίστοιχα).
digitalWrite	Εντολή	–	(<i>pin, pinstatus</i>)	Θέτει την κατάσταση <i>pinstatus</i> (HIGH ή LOW) στο συγκεκριμένο ψηφιακό <i>pin</i> .
digitalRead	Συνάρτηση	int	(<i>pin</i>)	Επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού <i>pin</i> (0 για LOW και 1 για HIGH) εφόσον αυτό είναι pin εισόδου.

analogRead	Συνάρτηση	int	(<i>pin</i>)	Επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο <i>pin</i> αναλογικής εισόδου στην κλίμακα 0 ως Vref.
analogWrite	Εντολή	–	(<i>pin, value</i>)	Θέτει το συγκεκριμένο ψηφιακό <i>pin</i> σε κατάσταση ψευδοαναλογικής εξόδου (PWM). Η παράμετρος <i>value</i> καθορίζει το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην κλίμακα από 0 ως 255 (π.χ. Με <i>value</i> 127, το πλάτος του παλμού είναι ίσο με μισή περίοδο).

Πίνακας 1 Συναρτήσεις, εντολές και σταθερές για τον προγραμματισμό της πλακέτας Arduino

Επιπλέον η γενική δομή του προγράμματος αποτελείται από δύο βασικές ρουτίνες:

// Ενσωματώσεις βιβλιοθηκών, δηλώσεις μεταβλητών...

void setup ()

{

// ..

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

Η ρουτίνα setup () εκτελείται μόνο μια φορά κατά την εκκίνηση του προγράμματος, ενώ η ρουτίνα loop () έχει το βασικό κορμό του προγράμματος και επαναλαμβάνεται συνεχώς. [10]

3.7. Shields

Τα shields είναι ολοκληρωμένες πλακέτες που κουμπώνουν πάνω στο Arduino προεκτείνοντας τη λειτουργία του. Με λίγα λόγια είναι σχεδιασμένα αφού κουμπώσουν να προωθούν τις υποδοχές τους ώστε να συνδεθούν από πάνω εξαρτήματα ή κι άλλα shields. Η χρησιμότητα τους είναι μεγάλη αφού συνοδεύονται πέρα από το hardware αλλά και από πολλές βιβλιοθήκες διευκολύνοντας τον προγραμματιστή. Υπάρχει όμως και ένα βασικό μειονέκτημα, χρησιμοποιεί πολλούς από τους πόρους συνδεσιμότητας του Arduino με αποτέλεσμα να μην μπορούν να χρησιμοποιηθούν απεριόριστα shields. Μερικά από τα πιο δημοφιλή shields που κυκλοφορούν είναι:

- Ethernet shield: Δίνει τη δυνατότητα στο Arduino να συνδεθεί σε κάποιο δίκτυο μέσω καλωδίου ethernet.
- Wifi shield: Όμοιο με το παραπάνω όμως με ασύρματη σύνδεση.
- Shields οθόνης: Προσθέτουν οθόνη στο Arduino.
- GPS shield: Προσθέτει GPS δυνατότητες στο Arduino. [10]



Εικόνα 8 Arduino Shields ενσωματωμένες σε πλακέτες Arduino

4. ΥΛΟΠΟΙΗΣΗ

Δημιουργήσαμε ένα σύστημα το οποίο χρησιμοποιείται για την ανίχνευση ανθρώπινης παρουσίας σε απομακρυσμένες θέσεις και μέσω ασύρματης μετάδοσης, μεταδίδεται σε ένα κεντρικό πίνακα ελέγχου αν ο χώρος έχει παραβιαστεί ή αν παραβιάζεται εκείνη την ώρα. Μπορεί να εγκατασταθεί σε διάφορους χώρους είτε αυτοί είναι οικιακά σπίτια, επιχειρήσεις, εταιρείες, αγροτικές εγκαταστάσεις και οπουδήποτε αλλού κριθεί ότι είναι απαραίτητο ένα τέτοιο σύστημα. Αρχικά τοποθετούμε το κάθε υποσύστημα σε μία πόρτα κυρίως σε υπερυψωμένο σημείο για να υπάρχει μεγαλύτερη κάλυψη από τους αισθητήρες και εγκαθιστούμε το κεντρικό σύστημα (πίνακας ελέγχου) στο χώρο που επιθυμούμε (πχ. Στο χώρο εργασίας μια μεγάλης επιχείρησης, σε ένα δωμάτιο μιας κατοικίας). Βάση της μελέτης που έγινε διαπιστώθηκε ότι η εμβέλεια για την επικοινωνία των πομπό / δεκτών όλου του συστήματος (κεντρικός πίνακας ελέγχου, υποσυστήματα) είναι δεκαπέντε μέτρα μετ' εμποδίων και γύρω στα 20 – 25 μέτρα σε ελεύθερο χώρο με τις υπάρχουσες κεραίες. Οι συνδέσεις που πραγματοποιήθηκαν και οι δυνατότητες του συστήματος αναλύονται στην συνέχεια του κεφαλαίου και τα υλικά που χρησιμοποιήθηκαν παρουσιάζονται στο επόμενο κεφάλαιο.

4.1. Δυνατότητες και επεξήγηση των λειτουργιών του συστήματος

- **Επικοινωνία 1^{ου} υποσυστήματος - κεντρικό πίνακα ελέγχου:** Ο πομπός του 1^{ου} υποσυστήματος μεταδίδει στον πίνακα ελέγχου ένα μήνυμα όταν ο αισθητήρας PIR ανιχνεύσει κίνηση και ένα μήνυμα όταν σταματήσει να υπάρχει ανθρώπινη παρουσία. Όση ώρα ανιχνεύεται κίνηση στη πόρτα ενεργοποιείται ένα φως (για τις ανάγκες της εργασίας χρησιμοποιήθηκε διόδος led) μέσω του μικροελεγκτή του υποσυστήματος. Στο πίνακα ελέγχου ο δέκτης λαμβάνει τα μηνύματα και μέσω του μικροελεγκτή αν έχει παραβιαστεί η πόρτα εμφανίζει στην LCD οθόνη «D1 OPENED h:min:sec», όπου h:min:sec είναι η ώρα που καταγράφηκε από τη μονάδα ρολογιού που βρίσκεται στο πίνακα ελέγχου όταν έλαβε το μήνυμα ο δέκτης, ενεργοποιεί ένα led το οποίο βρίσκεται ενσωματωμένο πάνω στο πίνακα ελέγχου και αντιστοιχεί για τη πόρτα1 και παραμένει σε κατάσταση HIGH μέχρι να λάβει μήνυμα ο δέκτης ότι δεν υπάρχει παραβίαση. Ταυτόχρονα ενεργοποιείται το buzzer εκπέμποντας ηχητικό σήμα «συναγερμού». Αντίστοιχα αν δεν υπάρχει ανθρώπινη παρουσία αλλάζει το μήνυμα σε «DOOR 1 CLOSED» χωρίς να υποδεικνύει τη σχετική ώρα, το led επιστρέφει σε κατάσταση LOW και το buzzer σταματάει να εκπέμπει. **Επικοινωνία πίνακα ελέγχου – 1^{ου} υποσυστήματος:** Από το πίνακα ελέγχου μπορούμε να ελέγξουμε το φως της πόρτας1 και να το ενεργοποιήσουμε ή απενεργοποιήσουμε όποια στιγμή θέλουμε μέσω ενός διακόπτη. Αυτό επιτυγχάνεται με ασύρματη μετάδοση από το πομπό του πίνακα ελέγχου στο δέκτη του υποσυστήματος.
- **Επικοινωνία 2^{ου} υποσυστήματος - κεντρικό πίνακα ελέγχου:** Ο πομπός του 2^{ου} υποσυστήματος μεταδίδει στο πίνακα ελέγχου ένα μήνυμα όταν ο αισθητήρας PIR ανιχνεύσει κίνηση και ένα μήνυμα όταν σταματήσει να υπάρχει ανθρώπινη παρουσία. Όση ώρα ανιχνεύεται κίνηση στη πόρτα ενεργοποιείται ένα φως (για τις ανάγκες της εργασίας χρησιμοποιήθηκε διόδος led) μέσω του μικροελεγκτή του υποσυστήματος. Στο πίνακα ελέγχου ο δέκτης λαμβάνει τα μηνύματα και μέσω του μικροελεγκτή αν έχει παραβιαστεί η πόρτα εμφανίζει στην LCD οθόνη «D2 OPENED h:min:sec», όπου h:min:sec είναι η ώρα που καταγράφηκε από τη μονάδα ρολογιού που βρίσκεται στο πίνακα ελέγχου όταν έλαβε το μήνυμα ο δέκτης, ενεργοποιεί ένα led το οποίο βρίσκεται ενσωματωμένο πάνω στο πίνακα ελέγχου και αντιστοιχεί για τη πόρτα2 και παραμένει σε κατάσταση HIGH μέχρι να λάβει μήνυμα ο δέκτης ότι δεν υπάρχει παραβίαση. Ταυτόχρονα ενεργοποιείται το buzzer εκπέμποντας ηχητικό σήμα «συναγερμού» (διαφορετικού τόνου από το ηχητικό σήμα της πόρτας1). Αντίστοιχα αν δεν υπάρχει ανθρώπινη παρουσία αλλάζει το μήνυμα σε «DOOR 2 CLOSED» χωρίς

να υποδεικνύει τη σχετική ώρα, το led επιστρέφει σε κατάσταση LOW και το buzzer σταματάει να εκπέμπει.

- **Επικοινωνία πίνακα ελέγχου – 2^ο υποσυστήματος:** Από το πίνακα ελέγχου μπορούμε να ελέγξουμε το φως της πόρτας2 και να το ενεργοποιήσουμε ή απενεργοποιήσουμε όποια στιγμή θέλουμε μέσω ενός διακόπτη. Αυτό επιτυγχάνεται με ασύρματη μετάδοση από το πομπό του πίνακα ελέγχου στο δέκτη του υποσυστήματος.

Ακόμα μία λειτουργία του συστήματος είναι η μετάδοση **e-mail**. Όταν ο πίνακας ελέγχου λάβει προειδοποιητικό μήνυμα από κάποιο υποσύστημα ότι ανιχνεύθηκε κίνηση, μέσω της Ethernet shield μεταδίδει e-mail στο χρήστη, για τον συγκεκριμένο χώρο που υπήρξε η παραβίαση. Δηλαδή αν παραβιαστεί η πόρτα1 φτάνει email στο παραλήπτη «DOOR 1 OPENED !!!!» αλλιώς αν παραβιαστεί η πόρτα2 το email στο παραλήπτη θα είναι «DOOR 2 OPENED !!!!».

4.2. Υλοποίηση συστήματος

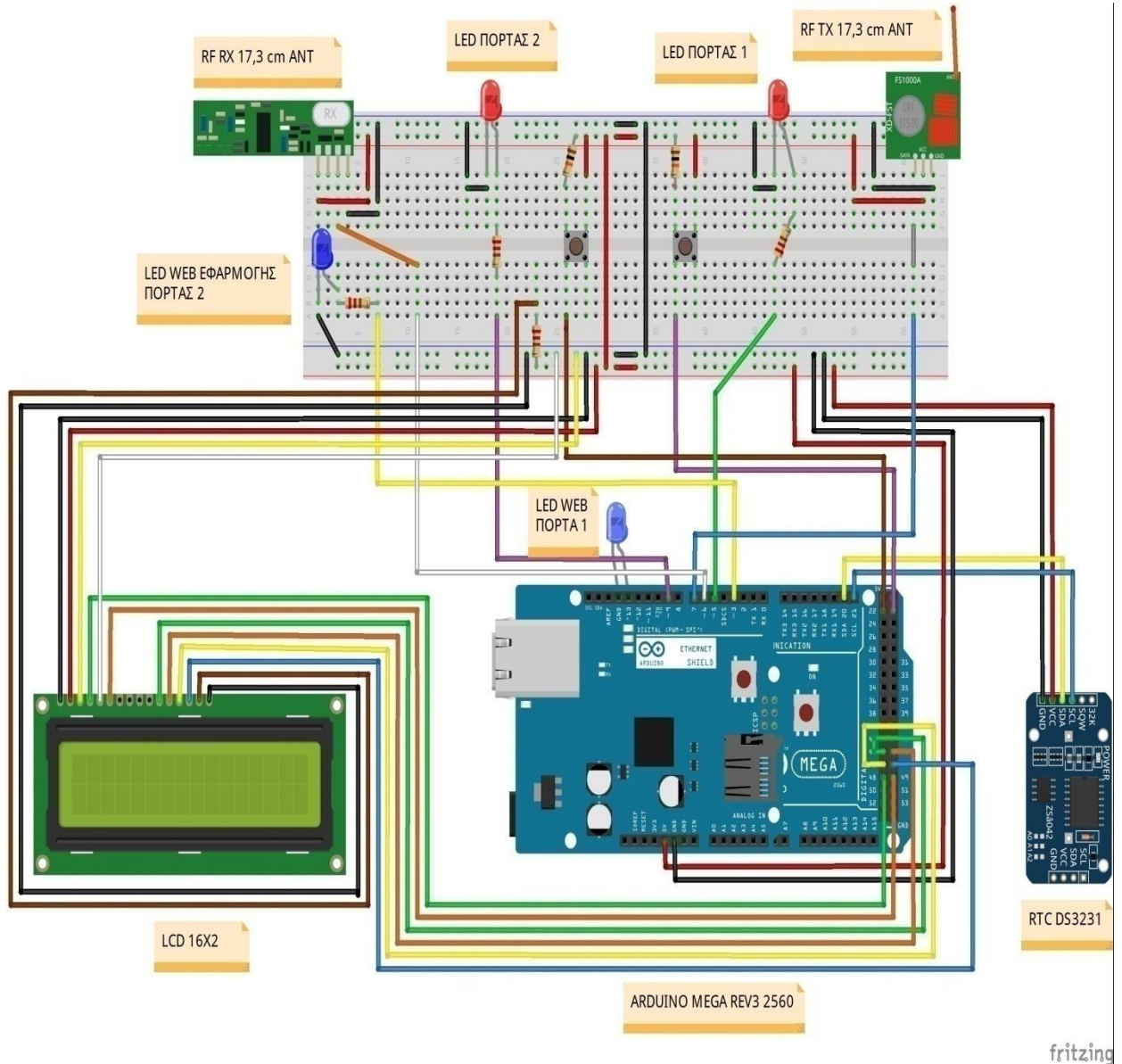
Για την ηλεκτρονική υλοποίηση του κεντρικού συστήματος χρειάστηκαν:

- Arduino MEGA2560 REV3 x1
- Ethernet shield W5100 x1
- RF 433 MHz πομπός x1
- RF 433 MHz δέκτης x1
- RTC DS3231 AT2432 IIC x1
- LCD display 16x2 x1
- Active Buzzer x1
- Buttons x2
- LEDS x4
- Resistors 220 Ohm x4
- Resistors 10 KOhm x2
- Wires
- Μπαταρία λιθίου 2025(Για το ρολόι) x1
- Breadboard x1
- Καλώδιο USB x1
- Καλώδιο Ethernet x1

Για την ηλεκτρονική υλοποίηση των απομακρυσμένων συστημάτων της εργασίας χρειάστηκαν:

- RF 433 MHz πομπός x2
- RF 433 MHz δέκτης x2
- Αισθητήρας (PIR) x2
- Arduino Nano ATmega328 x2
- Breadboard (μικρά) x3
- LEDS x2
- Wires
- Resistors 220 Ohms x2

Στο παρακάτω σχεδιάγραμμα εμφανίζεται η συνδεσμολογία του κυκλώματος που υλοποιήθηκε για το κεντρικό σύστημα. Το σχεδιάγραμμα σχεδιάστηκε με τη βοήθεια του προγράμματος Fritzing. [34]



Εικόνα 9 Συνδεσμολογία κυκλώματος του κεντρικού πίνακα ελέγχου

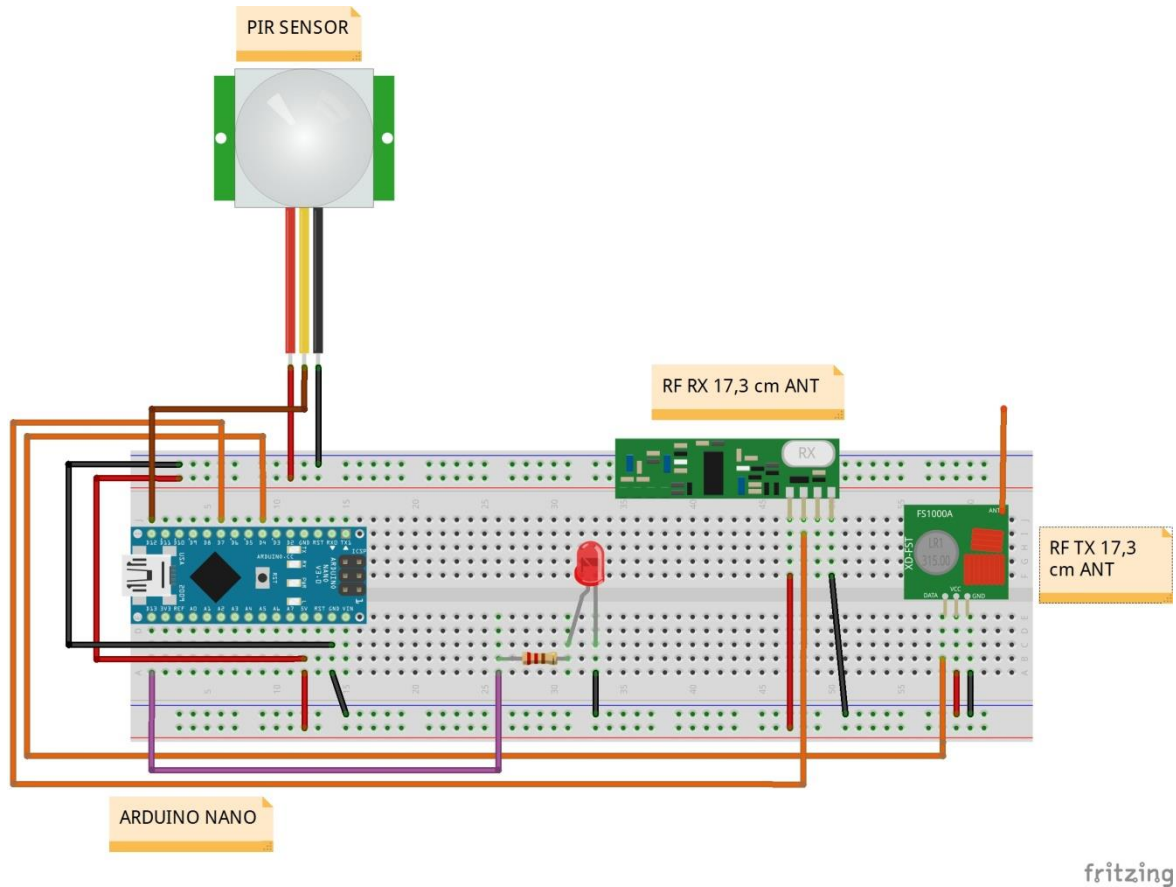
Αρχικά τοποθετήσαμε την Ethernet πλακέτα με το Arduino Mega με τέτοιο τρόπο ώστε κουμπωθούν οι ακίδες ICSP του Arduino στις υποδοχές της πλακέτας και όλοι οι ακροδέκτες της Ethernet shield στις υποδοχές του Arduino. Ύστερα συνδέθηκαν το pin τροφοδοσίας (5V) και το pin γείωσης του Arduino (καλώδια με κόκκινο χρώμα συμβολίζουν τη τάση, ενώ με μαύρο χρώμα τη γείωση) στη κάτω οριζόντια πλευρά του breadboard και με μικρά καλώδια ενώσαμε τη πάνω οριζόντια πλευρά ώστε όποιο εξάρτημα χρειαστεί τάση 5V και γείωση να συνδεθεί από το breadboard. Αυτό συμβαίνει διότι στο συγκεκριμένο κύκλωμα τα περισσότερα εξαρτήματα χρειάζονται υποχρεωτικά τροφοδοσία και γείωση.

Αφού η τροφοδοσία και η γείωση έχουν συνδεθεί για κάθε εξάρτημα, καθένα από αυτά έχει ακροδέκτες και επικοινωνία με το Arduino. Συγκεκριμένα:

- RF 433 MHz Receiver (RX): Το συγκεκριμένο εξάρτημα έχει τέσσερις ακροδέκτες εκ των οποίων χρησιμοποιούνται οι τρεις (τάση, γείωση, δεδομένα). Όπως φαίνεται και στο σχηματικό οι ακροδέκτες της τάσης και γείωσης συνδέθηκαν απευθείας στο breadboard και ο ακροδέκτης των δεδομένων (δεύτερος από αριστερά) συνδέθηκε στο

- 6^ο ψηφιακό pin του Arduino. [24]
- RF 433 MHz Transmitter (TX): Ο RF πομπός διαθέτει τρεις ακροδέκτες (τάση, γείωση, δεδομένα). Οι ακροδέκτες της τάσης και γείωσης συνδέθηκαν απευθείας στο breadboard ενώ ο ακροδέκτης των δεδομένων συνδέθηκε στο ψηφιακό pin 7 του Arduino. [24]
 - RTC DS3231: Το συγκεκριμένο εξάρτημα διαθέτει έξι ακροδέκτες εκ των οποίων για το σύστημα μας χρησιμοποιούνται οι τέσσερις. Οι ακροδέκτες με τις ενδείξεις GND & VCC συνδέθηκαν στη γείωση και τη τάση αντίστοιχα του breadboard και στη συνέχεια οι ακροδέκτες με τις ενδείξεις SDA, SCL συνδέθηκαν αντίστοιχα στις ψηφιακές υποδοχές «communication» 20 και 21 του Arduino. [26]
 - LCD 1602: Η οθόνη LCD είναι ένα εξάρτημα που διαθέτει δεκαέξι ακροδέκτες και δεσμεύει πολλές ακίδες από το Arduino. Από αυτούς στο σύστημα μας συνδέθηκαν οι έντεκα. Αρχικά οι ακροδέκτες με τη σήμανση VSS, V0, RW και K συνδέθηκαν στη γείωση και το VDD στη τάση του breadboard. Οι ακροδέκτες RS, E, D4, D5, D6 και D7 συνδέθηκαν στις ψηφιακές υποδοχές 48, 49, 44, 45, 46 και 47 αντίστοιχα του Arduino. Τέλος η ακίδα με την ένδειξη A τοποθετήθηκε με μία αντίσταση 220 Ohm στο breadboard όπως φαίνεται και στο σχηματικό. [27]
 - LEDs: Συνολικά για το κεντρικό σύστημα έχουν χρησιμοποιηθεί τέσσερις δίοδοι led. Τα θετικά άκρα των leds έχουν συνδεθεί με μία αντίσταση 220 Ohm το καθένα ενώ το άλλο άκρο των leds έχει τοποθετηθεί στη γείωση του breadboard. Τα leds με το κόκκινο φωτισμό έχουν συνδεθεί στο Arduino στις ψηφιακές υποδοχές 5 και 9. Ενώ το led με το μπλε φωτισμό συνδέθηκε στη ψηφιακή υποδοχή τρία. Παράλληλα όπως φαίνεται και στο σχεδιάγραμμα υπάρχει ακόμη ένα led με μπλε φωτισμό το οποίο τοποθετήθηκε απευθείας στην ψηφιακή υποδοχή 13 του Arduino χωρίς να χρειαστεί αντίσταση.
 - Buttons: Για τις λειτουργίες του κεντρικού συστήματος χρησιμοποιήθηκαν δύο διακόπτες. Για τη σύνδεση των buttons χρειάστηκαν οι τρεις ακροδέκτες από τους τέσσερις που διαθέτει. Οι ακροδέκτες κάτω αριστερά συνδέθηκαν για την επικοινωνία με το Arduino στις ψηφιακές υποδοχές 22 και 23. Και στα δύο buttons στον πάνω αριστερά ακροδέκτη τοποθετήθηκαν αντιστάσεις 10 KOhm ενώ ο ακροδέκτης πάνω δεξιά καταλήγει στη τάση του breadboard. Οι συνδέσεις των διακοπών φαίνονται αναλυτικά στο σχεδιάγραμμα. [7]
 - Buzzer: Το buzzer χρησιμοποιεί δύο ακροδέκτες. Ο αρνητικός ακροδέκτης τοποθετήθηκε στη γείωση της κάτω οριζόντιας πλευράς του breadboard ενώ ο θετικός ακροδέκτης συνδέθηκε στο 8^ο ψηφιακό pin του Arduino.

Παρακάτω εμφανίζεται η συνδεσμολογία των κυκλωμάτων που υλοποιήθηκαν για τα απομακρυσμένα υποσυστήματα του project. Τα σχεδιαγράμματα υλοποιήθηκαν με τη βοήθεια του προγράμματος Fritzing. [34]



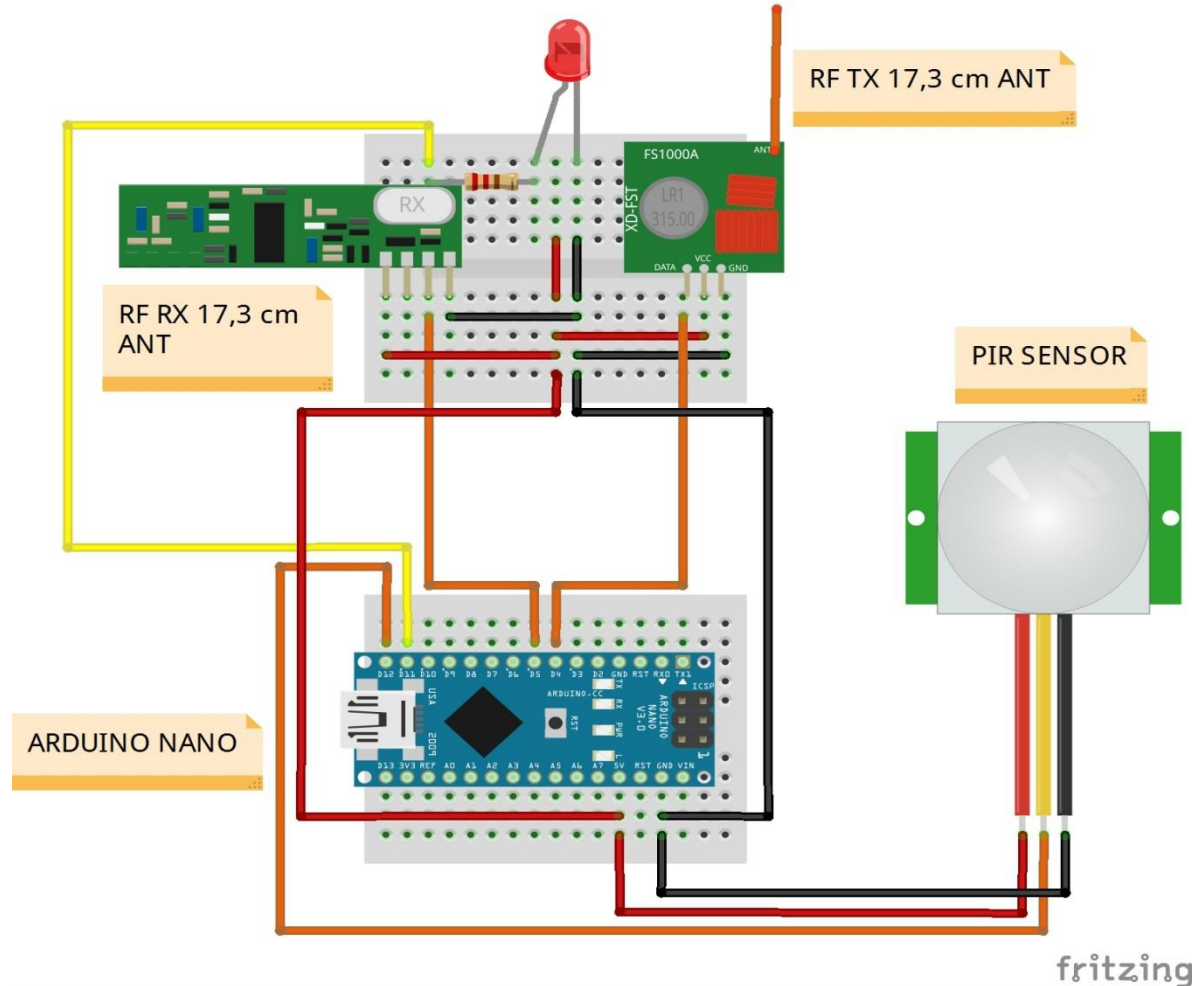
Εικόνα 10 Συνδεσμολογία κυκλώματος 1^{ου} υποσυστήματος

Όπως και στο κεντρικό σύστημα έτσι και στο πρώτο απομακρυσμένο σύστημα συνδέσαμε το pin τροφοδοσίας (5V) και το pin γείωσης του Arduino Nano (καλώδια με κόκκινο χρώμα συμβολίζουν τη τάση, ενώ με μαύρο χρώμα τη γείωση) στη πάνω οριζόντια πλευρά του breadboard και με μικρά καλώδια έχουμε ενώσει τη κάτω οριζόντια πλευρά ώστε όποιο εξάρτημα χρειαστεί τάση 5V και γείωση να συνδεθεί από το breadboard.

- Αισθητήρας PIR: Ο αισθητήρας ανίχνευσης κίνησης διαθέτει τρεις ακροδέκτες. Ένα για τη τάση (5V) και ένα για τη γείωση που συνδέθηκαν στη πάνω οριζόντια πλευρά του breadboard και ένα ακροδέκτη για τα δεδομένα που επικοινωνεί με το Arduino και συνδέθηκε στο ψηφιακό pin 12. [24]
- RF 433 MHz Receiver (RX): Το συγκεκριμένο εξάρτημα έχει τέσσερις ακροδέκτες εκ των οποίων χρησιμοποιούνται οι τρεις (τάση, γείωση, δεδομένα). Όπως φαίνεται και στο σχηματικό οι ακροδέκτες της τάσης και γείωσης συνδέθηκαν απευθείας στη κάτω οριζόντια πλευρά του breadboard και ο ακροδέκτης των δεδομένων (δευτερός από αριστερά) συνδέθηκε στο 7^ο ψηφιακό pin του Arduino. [24]
- RF 433 MHz Transmitter (TX): Ο RF πομπός διαθέτει τρεις ακροδέκτες (τάση, γείωση, δεδομένα). Το ίδιο και με τον πομπό οι ακροδέκτες της τάσης και γείωσης συνδέθηκαν απευθείας στη κάτω οριζόντια πλευρά του breadboard ενώ ο ακροδέκτης των δεδομένων συνδέθηκε στο ψηφιακό pin 4 του Arduino. [24]
- LED: Για τη λειτουργία του πρώτου υποσυστήματος χρησιμοποιείται μία διόδος led

κόκκινου χρώματος η οποία συνδέεται το αρνητικό άκρο στη γείωση του breadboard ενώ το θετικό άκρο στο ψηφιακό pin 13 μέσω μιας αντίστασης 220Ωm όπως φαίνεται στο σχεδιάγραμμα.

Τέλος στο επόμενο σχεδιάγραμμα φαίνεται η συνδεσμολογία του κυκλώματος του δεύτερου και τελευταίου υποσυστήματος που υλοποιήσαμε. Για το σχεδιάγραμμα χρησιμοποιήθηκε το πρόγραμμα Fritzing. [34]



Εικόνα 11 Συνδεσμολογία κυκλώματος 2^ο υποσυστήματος

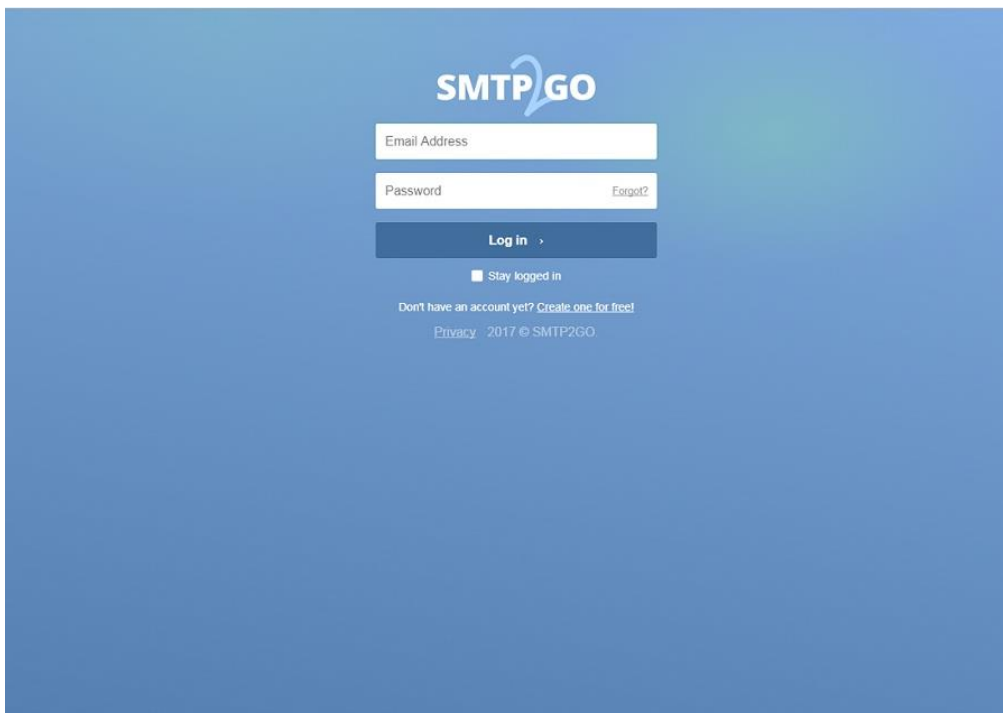
Για το δεύτερο υποσύστημα χρησιμοποιήσαμε δύο breadboards όμως η λειτουργία του είναι ίδια με το πρώτο υποσύστημα. Συνδέσαμε το pin τροφοδοσίας (5V) και το pin γείωσης του Arduino Nano (καλώδια με κόκκινο χρώμα συμβολίζουν τη τάση, ενώ με μαύρο χρώμα τη γείωση) στο δεύτερο breadboard ώστε τα εξαρτήματα που είναι τοποθετημένα εκεί να τροφοδοτούνται απευθείας από εκεί.

- Αισθητήρας PIR: Ο αισθητήρας ανίχνευσης κίνησης διαθέτει τρεις ακροδέκτες. Ένα για τη τάση (5V) και ένα για τη γείωση, που συνδέθηκαν απευθείας στη τάση (5V) και στη γείωση του Arduino και ένα ακροδέκτη για τα δεδομένα που επικοινωνεί με το Arduino και συνδέθηκε στο ψηφιακό pin 12. [24]
- RF 433 MHz Receiver (RX): Το συγκεκριμένο εξάρτημα έχει τέσσερις ακροδέκτες εκ των οποίων χρησιμοποιούνται οι τρεις (τάση, γείωση, δεδομένα). Όπως φαίνεται και στο σχηματικό οι ακροδέκτες της τάσης και γείωσης συνδέθηκαν απευθείας με μικρά καλώδια στο breadboard και ο ακροδέκτης των δεδομένων (δεύτερος από δεξιά) συνδέθηκε στο 5^ο ψηφιακό pin του Arduino. [24]

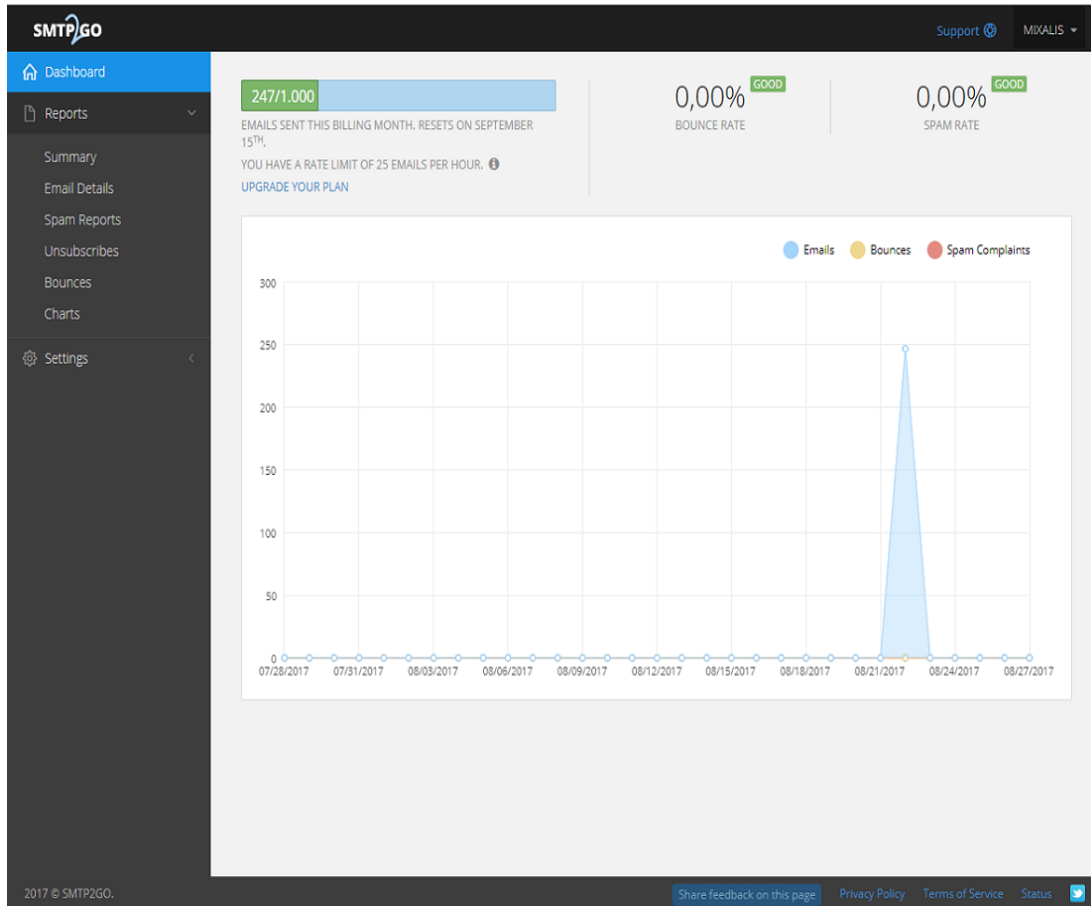
- RF 433 MHz Transmitter (TX): Ο RF πομπός διαθέτει τρεις ακροδέκτες (τάση, γείωση, δεδομένα). Το ίδιο και με τον πομπό οι ακροδέκτες της τάσης και γείωσης συνδέθηκαν απευθείας στο breadboard με μικρά καλώδια ενώ ο ακροδέκτης των δεδομένων συνδέθηκε στο ψηφιακό pin 4 του Arduino. [24]
- LED: Για τη λειτουργία του δεύτερου υποσυστήματος χρησιμοποιείται μία δίοδος led κόκκινου χρώματος η οποία συνδέεται το αρνητικό άκρο στη γείωση του breadboard ενώ το θετικό άκρο στο ψηφιακό pin 11 μέσω μιας αντίστασης 220 Ohm όπως φαίνεται στο σχεδιάγραμμα.

4.3. Υπηρεσία αποστολής email

Αυτή η λειτουργία επιτυγχάνεται μέσω της υπηρεσίας SMTP2GO. Στην δωρεάν έκδοση (αυτή χρησιμοποιήθηκε για τις ανάγκες της εργασίας) υπάρχουν γραφήματα που μπορεί να δει ο χρήστης τότε στάλθηκαν τα e-mails (την ημερομηνία και τον αριθμό των e-mails). Επιπλέον μπορούν να σταλούν έως 1000 e-mail για ένα μήνα και εν συνεχεία ανανεώνεται κάθε μήνα και υπάρχει ένας περιορισμός ότι μπορούν να σταλούν μέχρι 25 e-mail ανά ώρα. Επίσης μπορούν να συνδεθούν περισσότεροι από έναν χρήστη και να μεταδίδονται e-mails σε όλους. Για την εγγραφή στο SMTP2GO απαιτείται μόνο ένα e-mail και ένα password. [30]



Εικόνα 12 Υπηρεσία SMTP2GO



Εικόνα 13 Γραφικό περιβάλλον της υπηρεσίας SMTP2GO

Τέλος για την επιτυχή αποστολή των e-mail χρειάζεται να κωδικοποιήσουμε σε μορφή base64 με UTF-8 χαρακτήρες το email και το password με το οποίο έγινε η εγγραφή στην υπηρεσία SMTP2GO. Αυτό μπορεί να εκτελεστεί μέσω της σελίδας base64encode και θα φανεί αναλυτικά στο κεφάλαιο του προγραμματισμού (<https://www.base64encode.org/>). [31]

4.4. Web εφαρμογή για τον απομακρυσμένο έλεγχο του συστήματος

Η Ethernet shield δεν τοποθετήθηκε μόνο για τη λειτουργία του e-mail. Άλλη μία χρησιμότητα του συστήματος που προστέθηκε στη συνέχεια είναι ο απομακρυσμένος έλεγχος των φώτων και της παραβίασης του χώρου των υποσυστημάτων. Η λειτουργία αυτή μπορεί να επιτευχθεί μέσω του γραφικού περιβάλλοντος που έχουμε δημιουργήσει από ένα web browser είτε από υπολογιστή, smartphone, tablet αρκεί να γράψει την απαραίτητη IP στη μπάρα αναζήτησης του browser. Η IP διεύθυνση λειτουργεί σαν domain name της «Σελίδας» και είναι η < **192.168.1.177** >. Αν βρισκόμαστε σε ένα τοπικό δίκτυο μπορούμε να θέσουμε όποια IP διεύθυνση θέλουμε αρκεί να ελέγξουμε ποια είναι η gateway IP και ποιες IP έχουν δεσμευτεί από άλλα υπολογιστικά συστήματα. Παρακάτω εμφανίζεται το γραφικό περιβάλλον:

Security alarm system TEI of Epirus

No Movement DOOR 1!

LED On DOOR 1

LED Off DOOR 1

No Movement DOOR 2!

LED On DOOR 2

LED Off DOOR 2

Εικόνα 14 Γραφικό περιβάλλον της Web Εφαρμογής

Όπως παρατηρείται στην εικόνα ο χρήστης μπορεί να ελέγξει τα φώτα και των δύο υποσυστημάτων. Να τα ενεργοποιήσει ή να τα απενεργοποιήσει. Επιπλέον εμφανίζονται τα εξής μηνύματα σε περίπτωση που οι αισθητήρες PIR δεν ανιχνεύουν κίνηση «No Movement DOOR 1!» , «No movement DOOR 2!» ενώ αν οι αισθητήρες ανιχνεύσουν κίνηση τροποποιούνται τα μηνύματα και εμφανίζονται «Motion Detected DOOR 1!» , «Motion Detected DOOR 2!».



Εικόνα 15 Ένδειξη παραβίασης πόρτας 1 και πόρτας 2 στη Web Εφαρμογή

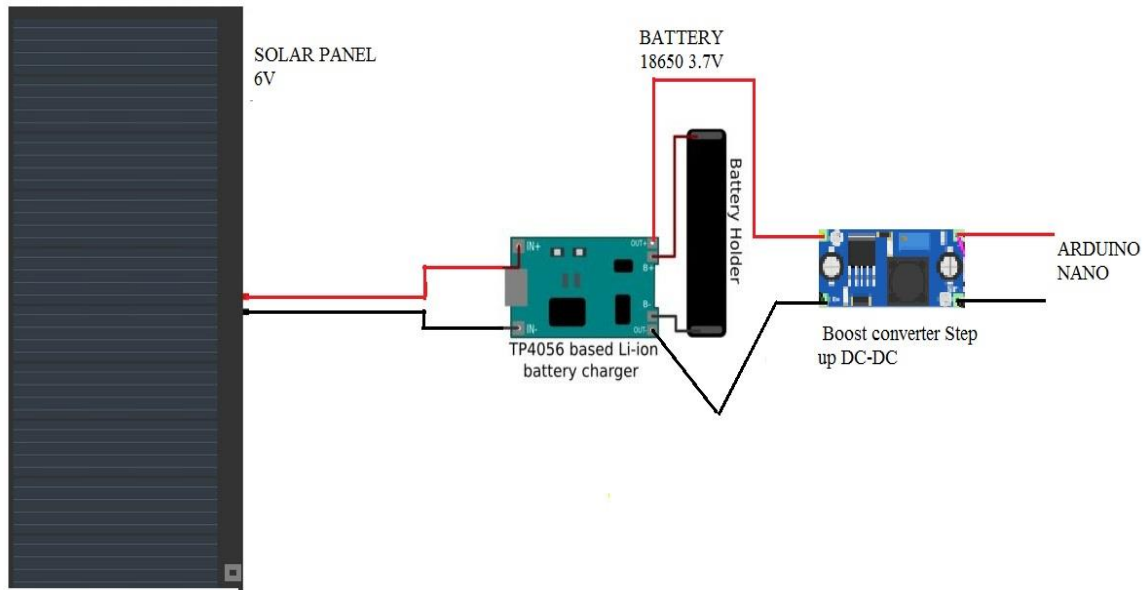
Οι λειτουργίες της web εφαρμογής είναι δύο: Αν ο κεντρικός πίνακας ελέγχου λάβει μήνυμα από τα υποσυστήματα ότι οι πόρτες παραβιάστηκαν, μέσω της Ethernet shield που επικοινωνεί με το διαδίκτυο τροποποιεί την εφαρμογή και εμφανίζει τα κατάλληλα μηνύματα όπως φαίνεται και στην εικόνα 36. Στο κεντρικό πίνακα ελέγχου ωστόσο υπάρχουν ακόμη δύο δίοδοι led χρώματος μπλε. Αν χρειαστεί να ενεργοποιήσουμε το φως της πόρτας1 ή το φως της πόρτας2 ενεργοποιούνται ταυτόχρονα τα led της αντίστοιχης πόρτας. Για παράδειγμα αν πατήσουμε το button της εφαρμογής για να ενεργοποιηθεί το φως της πόρτας1 πρώτα θα ενεργοποιηθεί το led που βρίσκεται στο πίνακα ελέγχου και αντιστοιχεί στη πόρτα1 και αφού ενεργοποιηθεί, ο πομπός RF θα μεταδώσει δεδομένα ασύρματα στο δέκτη του υποσυστήματος 1 για να ενεργοποιήσει το φως στη πόρτα1. Το ίδιο συμβαίνει για να απενεργοποιηθεί το φως και η αντίστοιχη λειτουργία συμβαίνει για τη πόρτα2. Η λογική των leds (κεντρικός πίνακας ελέγχου) χρησιμοποιήθηκε ώστε ο χρήστης να μπορεί μέσω του πίνακα ελέγχου να διαπιστώσει αν κάποιος έχει εισβάλει στο σύστημα και ελέγχει τα φώτα των υποσυστημάτων. [28]

4.5. Υλοποίηση Ενεργειακής αυτονομίας

Τα δύο υποσυστήματα υλοποιήθηκαν με χρήση ηλιακών πάνελ για να είναι ενεργειακά αυτόνομα χωρίς να χρειάζονται τη παροχή ενέργειας από ηλεκτρική πρίζα. Τα υλικά που χρειάστηκαν είναι:

- Solar Panel 6V x2
- MT3608 Boost converter Step up DC-DC x2
- Battery charger TP4056 x2
- Battery 18650 x2
- Wires
- Single battery case x2

Παρακάτω εμφανίζεται το σχεδιάγραμμα της ενεργειακής αυτονομίας για το **κάθε υποσύστημα** (να σημειωθεί ότι είναι το ίδιο και για τα 2 υποσυστήματα) και σχεδιάστηκε με το πρόγραμμα Fritzing. [34]



Εικόνα 16 Συνδεσμολογία του κυκλώματος ενεργειακής αυτονομίας (Το κόκκινο καλώδιο προσδιορίζει τη τάση ενώ το μαύρο τη γείωση)

Για τη σύνδεση των εξαρτημάτων χρησιμοποιήσαμε ηλεκτρικό κολλητήρι. Ενώσαμε το θετικό άκρο του ηλιακού πάνελ με την υποδοχή IN + και το αρνητικό άκρο του πάνελ με την υποδοχή IN- του φορτιστή μπαταρίας. Στην έξοδο του φορτιστή στις υποδοχές B+ και B- σύνδεσα τα άκρα της μπαταρίας (στο B+ το θετικό άκρο, στο B- το αρνητικό άκρο) και στις υποδοχές OUT+ και OUT- συνδέσαμε τα άκρα του Boost converter. Η λειτουργία του συστήματος έχει ως εξής: Η τάση που απορροφάται από τον ήλιο στο ηλιακό πάνελ περνάει από το φορτιστή μπαταρίας. Αν η μπαταρία είναι πλήρως φορτισμένη ο φορτιστής θα διακόψει τη τροφοδοσία ενώ αν είναι εξασθετισμένη θα επιτρέψει τη φόρτιση. Μπορούμε εύκολα να διακρίνουμε πότε είναι φορτισμένη πλήρως η μπαταρία μας αφού πάνω στη πλακέτα του φορτιστή υπάρχει δείκτης τροφοδοσίας. Επειδή όμως η μπαταρία λειτουργεί με τάση 3.7V ενώ το Arduino χρειάζεται 5V minimum παρεμβάλλεται ανάμεσα στο φορτιστή και στη μπαταρία ο ρυθμιστής τάσης. Η ιδιότητα του είναι να δέχεται τη τάση 3.7 V της μπαταρίας και μέσω ενός ποτενσιόμετρου να την αυξάνει στα 5V για να λειτουργήσει σωστά το σύστημα μας (το μέγιστο που μπορεί να φτάσει είναι 28 V). [33]

Πρέπει όμως να γνωρίζουμε αν η τροφοδοσία του ηλιακού πάνελ και η χωρητικότητα της μπαταρίας που χρησιμοποιήσαμε επαρκεί για να τροφοδοτήσει το σύστημα μας και ποια είναι η ισχύς που καταναλώνει. (Η ανάλυση που φαίνεται παρακάτω είναι ίδια και για τα δύο υποσυστήματα).

Το ρεύμα που χρειάζεται το σύστημα μας όταν βρίσκεται σε **κατάσταση λειτουργίας** είναι:

- Arduino nano ≈ 20 mA
- Πομπός (Tx) = 2 mA
- Δέκτης (Rx) ≈ 5 mA
- Led = 20 mA
- Αισθητήρας (PIR) = 65 mA

Συμπεριλαμβάνουμε όμως και το ρεύμα που δίνει το Arduino nano στις ψηφιακές υποδοχές όταν μέσα από αυτές λειτουργούν τα εξαρτήματα. Η μέγιστη τιμή ρεύματος είναι 40 mA επί 4 εξαρτήματα (led, Rx, Tx, Pir) δηλαδή ίσο με 160 mA.

$$160 + 20 + 2 + 5 + 20 + 65 = 272 \text{ mA}$$

Το ρεύμα που χρειάζεται το σύστημα μας όταν βρίσκεται σε κατάσταση ηρεμίας είναι:

Arduino nano $\approx 20 \text{ mA}$
Δέκτης (Rx) $\approx 5 \text{ mA}$
Αισθητήρας (PIR) = 0,05 mA

$$20 + 5 + 0,05 = 25,05 \approx 25 \text{ mA}$$

Αν υποθέσουμε ότι το σύστημα μας θα ενεργοποιηθεί 100 φορές κατά τη διάρκεια της ημέρας και ότι ο χρόνος λειτουργίας είναι 1 λεπτό κάθε φορά:

Συνολικά θα έχουμε $1 * 100 = 100 \text{ min} \sim 1 \text{ h} : 40 \text{ min}$
Αν το στρογγυλοποιήσουμε, συνολικά 2 hours

Για 2 ώρες συνεχής λειτουργίας χρειάζεται ρεύμα: $2 * 272 \text{ mA} = 544 \text{ mAh}$

Η ισχύς που χρειάζεται για αυτές τις ώρες συνεχούς λειτουργίας είναι:
 $5 \text{ V} * 544 \text{ mAh} = \underline{2720 \text{ mWh}}$

Αν τις υπόλοιπες ώρες ($24 \text{ h} - 2 \text{ h} = 22 \text{ hours}$) το σύστημα μας βρίσκεται σε κατάσταση ηρεμίας η ισχύς που χρειάζεται είναι:

$$5 \text{ V} * 25 \text{ mA} * 22 \text{ h} = \underline{2750 \text{ mWh}}$$

Επομένως κατά τη διάρκεια μίας ημέρας η συνολική ισχύς που καταναλώνει είναι
 $2720 + 2750 = \underline{5470 \text{ mWh}}$

Η μπαταρία που χρησιμοποιήθηκε σε κάθε υποσύστημα είναι τύπου 18650, Li-On της Samsung, με χωρητικότητα 3000 mAh.

$$\text{ισχύς μπαταρίας πλήρους εκφόρτισης} = 3.7 * 3000 \text{ mAh} = 11100 \text{ mWh}$$

Παίρνουμε το 50 % του χρόνου εκφόρτισης διότι δεν αφήνουμε τη μπαταρία να εκφορτιστεί πλήρως.

$$\text{Ισχύς που μας παρέχει η μπαταρία} = 11100 * 50\% = 5550 \text{ mWh}$$

Το ηλιακό πάνελ που χρησιμοποιήσαμε για τη παρούσα εργασία έχει τα εξής χαρακτηριστικά:

Τάση (V) = 6V

Ένταση (I) = 180 mA

Ισχύς (W) = 1100 mW

- Η ισχύς που δίνει σε μία ώρα ηλιοφάνειας είναι: $6 \text{ V} * 180 \text{ mA} = 1100 \text{ mWh}$

- Για 3 ώρες ηλιοφάνειας την ημέρα : $3h * 1100 \text{ mWh} = 3300 \text{ mWh}$
- Για 5 ώρες ηλιοφάνειας την ημέρα : $5h * 1100 \text{ mWh} = 5500 \text{ mWh}$

Επομένως συμπεραίνουμε ότι χρειάζεται 5 ώρες ηλιοφάνειας κατά τη διάρκεια της ημέρας για να φορτιστεί πλήρως η μπαταρία. Αυτό σημαίνει ότι τα υποσυστήματα μας μπορούν να είναι ενεργειακά αυτόνομα και να τροφοδοτούνται από ηλιακό πάνελ.

5. ΑΝΑΛΥΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΕΞΑΡΤΗΜΑΤΩΝ

Στα πλαίσια της υλοποίησης της εφαρμογής μας, χρησιμοποιήθηκαν τα εξαρτήματα που αναφέρονται αναλυτικά παρακάτω.

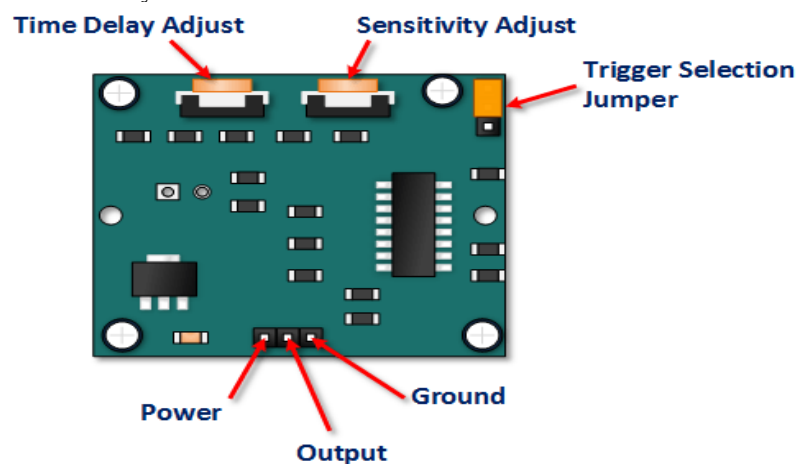
5.1. Παθητικός αισθητήρας υπέρυθρης ακτινοβολίας HC-SR501



Εικόνα 17. Παθητικός αισθητήρας υπέρυθρης ακτινοβολίας

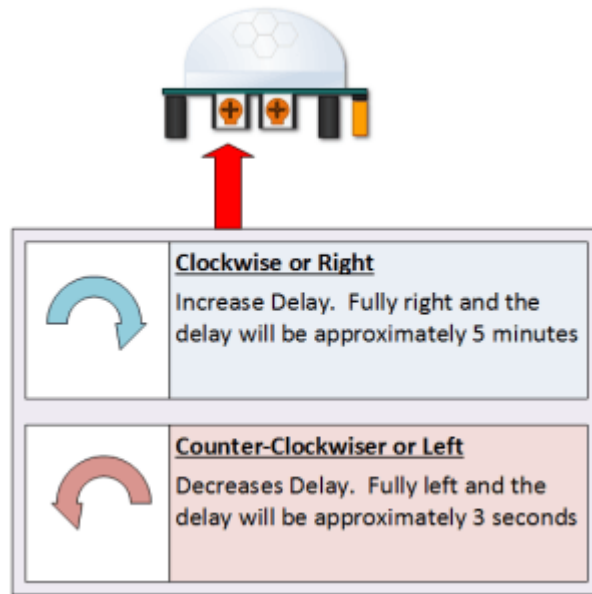
Ο αισθητήρας παθητικών υπέρυθρων (Passive Infrared Sensors, PIR) όπως υποδηλώνει και το όνομα του «παθητικό» δεν εκπέμπει κάποιο ηλεκτρομαγνητικό κύμα αλλά απαντά μόνο στην υπέρυθη ενέργεια που ακτινοβολείται από το αντικείμενο που ανιχνεύτηκε. Γι αυτό χρησιμοποιεί ένα πυροηλεκτρικό υλικό στον πυρήνα του αισθητήρα το οποίο παράγει ενέργεια όταν εκτίθεται σε υπέρυθη ακτινοβολία. Όταν διέρχεται ο άνθρωπος μπροστά από έναν αισθητήρα επειδή το ανθρώπινο σώμα λόγω θερμότητας εκπέμπει υπέρυθη ακτινοβολία, ο αισθητήρας αλλάζει με τα κατάλληλα κυκλώματα την ενέργεια σε έξοδο τάσης. Με απλά λόγια η έξοδος του αισθητήρα θα είναι HIGH όταν υπάρχει κίνηση στο οπτικό πεδίο και LOW όταν δεν υπάρχει κίνηση.

Όμως πάνω στην ηλεκτρονική πλακέτα του αισθητήρα υπάρχουν κάποιες ρυθμίσεις για τη καλύτερη λειτουργία του. Αυτές είναι:



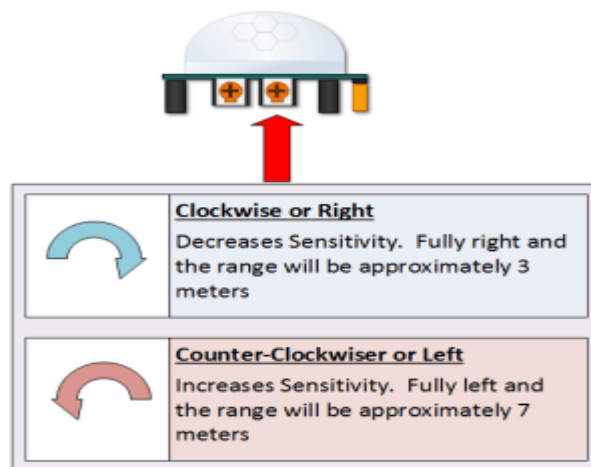
Εικόνα 18 Ρυθμίσεις του Παθητικού αισθητήρα υπέρυθρης ακτινοβολίας

- **Time delay adjust** (ρύθμιση χρονικής καθυστέρησης): Ορίζει για πόσο χρόνο θα παραμείνει ο αισθητήρας σε κατάσταση HIGH αν ανιχνεύσει κίνηση (από τρία δευτερόλεπτα έως πέντε λεπτά). Η ρύθμιση πραγματοποιείται με τη χρήση του ποτενσιόμετρου, όσο το κινούμε δεξιόστροφα αυξάνεται ο χρόνος καθυστέρησης, πλήρως δεξιά η κατάσταση HIGH θα είναι πέντε λεπτά. Αντίθετα όσο το κινούμε αριστερόστροφα μειώνεται η καθυστέρηση, πλήρως αριστερά η διάρκεια που θα παραμείνει σε κατάσταση HIGH ο αισθητήρας είναι τρία δευτερόλεπτα.



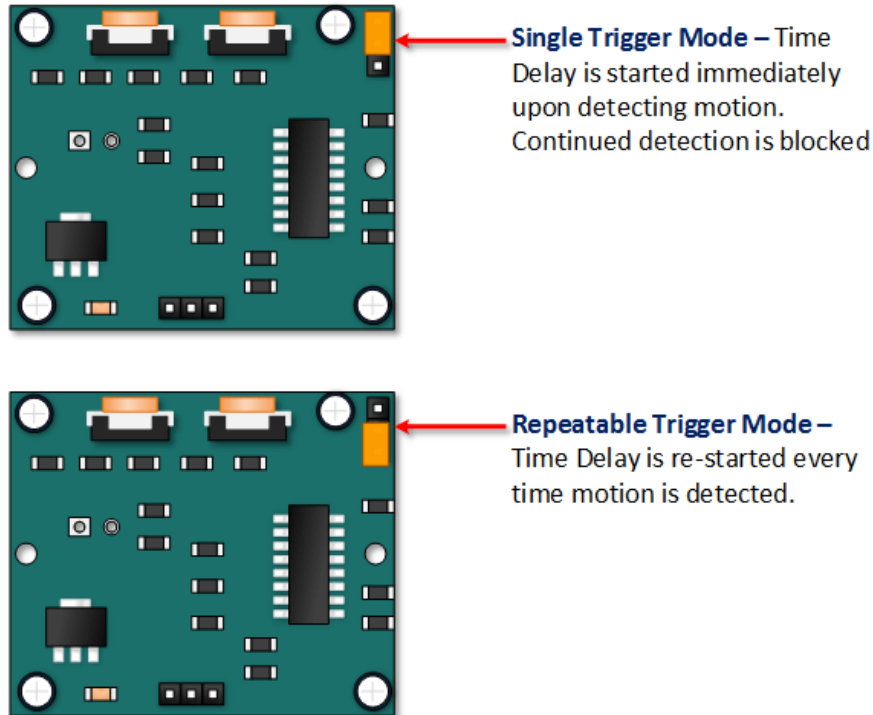
Εικόνα 19 Ρύθμιση της χρονικής καθυστέρησης μέσω του ποτενσιόμετρου

- **Sensitivity adjust** (ρύθμιση ευαισθησίας): Ορίζει την απόσταση ανίχνευσης (από τρία μέτρα έως επτά μέτρα). Η ρύθμιση πραγματοποιείται με τη χρήση του ποτενσιόμετρου, όσο το κινούμε δεξιόστροφα μειώνεται η απόσταση που θα ανιχνεύει κίνηση ο αισθητήρας (όριο τρία μέτρα). Αντίθετα όσο το κινούμε αριστερόστροφα αυξάνεται η απόσταση ανίχνευσης του αισθητήρα (όριο επτά μέτρα).



Εικόνα 20 Ρύθμιση της ευαισθησίας μέσω του ποτενσιόμετρου

- **Trigger selection jumper:** Αν ο βραχυκυκλωτήρας βρεθεί σε κατάσταση “L” (single trigger mode) και ο αισθητήρας ανιχνεύσει κίνηση θα υπολογιστεί μόνο μία φορά ο χρόνος καθυστέρησης ακόμη και αν συνεχίζει να υπάρχει κίνηση στο οπτικό πεδίο του αισθητήρα, ενώ αν βρεθεί σε κατάσταση “H” (repeatable trigger mode) θα υπολογίζεται εκ νέου ο χρόνος καθυστέρησης όσο ανιχνεύεται κίνηση.



Εικόνα 21 Ρύθμιση του βραχυκυκλωτήρα

Ένα σημαντικό μειονέκτημα του αισθητήρα είναι ότι αν από κατάσταση HIGH βρεθεί σε κατάσταση LOW, ο εντοπισμός κίνησης μπλοκάρεται για περίοδο τριών δευτερολέπτων και ύστερα είναι έτοιμο να ανιχνεύσει πάλι κίνηση. Για τις ανάγκες της εργασίας ο χρόνος που παραμένει σε κατάσταση HIGH αν ανιχνεύσει κίνηση είναι τρία δευτερόλεπτα, δηλαδή ο ελάχιστος χρόνος, η απόσταση εντοπισμού είναι ρυθμισμένη στα τρία μέτρα ενώ ο βραχυκυκλωτήρας έχει τοποθετηθεί σε Repeatable Trigger Mode για να επαναλαμβάνεται ο χρόνος καθυστέρησης όσο υπάρχει κίνηση. [11]

Τα χαρακτηριστικά του PIR αισθητήρα είναι:

- Product Type HC--SR501 Body Sensor Module
- Working voltage range: DC 4.5-20V
- Quiescent Current: 50μA
- Power Consumption: 65mA
- High output level 3.3 V / Low 0V
- Trigger L trigger can not be repeated / H repeated trigger
- Circuit board dimensions: 32 * 24 mm
- Maximum 110 ° angle sensor
- 7 m maximum sensing distance [12]

5.2. RF 433 MHz Πομπός/Δέκτης



Εικόνα 22 RF μονάδες 443 MHz transmitter / receiver

Για την παρούσα εργασία χρησιμοποιήθηκαν οι RF 433MHz (Radio Frequency) μονάδες με σκοπό την ασύρματη μετάδοση των δεδομένων του συστήματος με αμφίδρομη επικοινωνία. Λόγω της μικρής εμβέλειας τοποθετήθηκε μια κεραία σε κάθε μονάδα. Η κεραία είναι ένα μονόκλωνο καλώδιο σύρματος χαλκού με διάμετρο 1 mm και μήκος 17.3 cm. Ο τρόπος υπολογισμού του μήκους της κεραίας προκύπτει από τη διαίρεση της ταχύτητας του φωτός με τη συχνότητα για τον υπολογισμό του μήκους κύματος. Ύστερα διαιρούμε το μήκος κύματος διά τέσσερα για να βρούμε το μήκος της κεραίας.

- C: ταχύτητα του φωτός στο κενό ($3 * 10^8$ m/s)
- F: συχνότητα (433 MHz)
- λ: μήκος κύματος

$$\begin{aligned} \text{ταχύτητα του φωτός (c)} &= \text{μήκος κύματος (\lambda)} * \text{συχνότητα (f)} \\ \text{μήκος κύματος} &= (3 * 10^8) / (433 * 10^6) = 0,69284 \text{ m} \\ \text{μήκος κεραίας} &= 0,69284 / 4 = 0,1732 \text{ m} = 17,32 \text{ cm} \text{ [24]} \end{aligned}$$

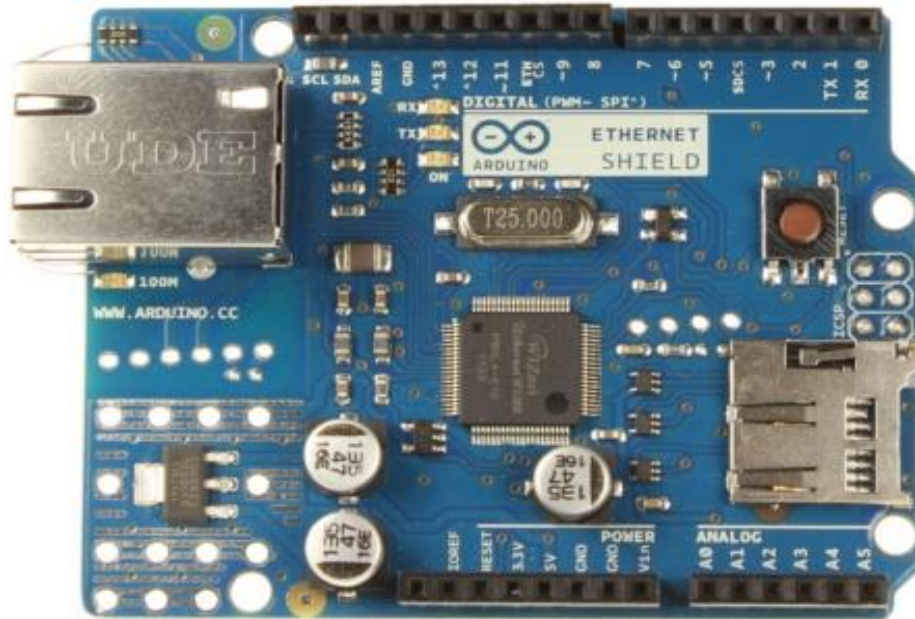
Τα χαρακτηριστικά των μονάδων είναι τα εξής:

Receiver module	Transmitter module
Product Model: XD-RF-5V	Product Model: XD-FST
voltage, different results)	Launch distance: 20-200 meters (different
Quiescent Current: 4mA	Operating voltage: 3.5-12V
Receiving frequency: 433.92MHZ	Dimensions: 19 * 19mm
Receiving frequency: 433.92MHZ	Operating mode: AM
Operating voltage: DC 5V	Transfer rate: 4KB / S
Size:30x14x7mm	Transmitting power: 10mW
	Transmitting frequency: 433M
	Pinout from left → right: (DATA; VCC; GND)

Πίνακας 2 Πίνακας τεχνικών προδιαγραφών Receiver / Transmitter

[13]

5.3. Ethernet Shield W5100



Εικόνα 23 Κάτωψη πλακέτας Ethernet Shield W5100

Για την υλοποίηση της εργασίας χρησιμοποιήθηκε η πλακέτα Ethernet για τη σύνδεση του Arduino με το δίκτυο. Η Ethernet shield βασίζεται στο τσιπ Wiznet W5100 το οποίο διαθέτει μια στοίβα δικτύου (IP) ικανή για TCP και UDP και χρησιμοποιεί τη βιβλιοθήκη Ethernet για να λειτουργήσει. Επιπλέον πάνω στη πλακέτα υπάρχει ενσωματωμένη υποδοχή για κάρτα microSD για αποθήκευση αρχείων και προβολή μέσω του δικτύου και λειτουργεί με τη βιβλιοθήκη SD. Το Arduino επικοινωνεί με το τσιπ W5100 και τη κάρτα microSD από το διάλυο SPI (μέσω του ICSP). Για το UNO δεσμεύει τις ακίδες 11, 12, 13 ενώ για το Arduino MEGA δεσμεύει τις ακίδες 50, 51, 52. Και στις δύο πλακέτες δεσμεύονται το pin 10 για το τσιπ και το pin 4 για την κάρτα SD. Όλες αυτές οι ακίδες δεν μπορούν να χρησιμοποιηθούν για γενική χρήση (είσοδοι / έξοδοι). Τέλος για την πλακέτα Arduino Mega το pin 53 είναι το SS και πρέπει να διατηρηθεί ως έξοδος για να λειτουργήσει ο διάλυος SPI. Η Ethernet Shield περιέχει μια υποδοχή RJ-45 και η επικοινωνία με το δίκτυο επιτυγχάνεται με καλώδιο κατηγορίας 5 Ethernet. Κάποια χρήσιμα leds που βρίσκονται πάνω στη πλακέτα είναι:

- PWR: Δείχνει ότι υπάρχει τροφοδοσία.
- LINK: Δείχνει ότι υπάρχει σύνδεση σε δίκτυο. Αν αναβοσβήνει το led σημαίνει ότι μεταδίδει ή λαμβάνει δεδομένα.
- FULLD: Δηλώνει ότι υπάρχει αμφίδρομη επικοινωνία.
- 100M: Σημαίνει ότι υπάρχει σύνδεση 100 Mb/s (σε αντίθεση με 10 Mb/s).
- RX & TX: Αναβοσβήνουν τα Led όταν λαμβάνουν ή εκπέμπουν δεδομένα.
- COLL: Αναβοσβήνει αν εντοπίσει σύγκρουση με το δίκτυο.

Η τάση λειτουργίας του Ethernet shield είναι 5V και η ταχύτητα σύνδεσης 10Mb / 100Mb. [14]

5.4. Arduino MEGA 2560 REV3



Εικόνα 24 Κάτοψη πλακέτας Arduino Mega

Για τις ανάγκες της εργασίας του κεντρικού συστήματος χρησιμοποιήθηκε το Arduino mega 2560 διότι το Arduino UNO δεν ήταν επαρκή σε flash memory αλλά και σε ακίδες. Το mega 2560 εκτός από διαφορετικό τύπο έχει και διαφορετική φυσική μορφή του μικροελεγκτή που χρησιμοποιεί. Επιπλέον διαθέτει 54 ψηφιακές ακίδες για είσοδο / έξοδο αντί 14 του UNO και 16 αναλογικές ακίδες αντίθετα με τις 6 που διαθέτει το UNO. Αυτό σημαίνει ότι μπορούμε να συνδέσουμε περισσότερες περιφερειακές συσκευές και να δημιουργήσουμε μεγαλύτερα projects. Τα βασικά χαρακτηριστικά του Arduino mega 2560 είναι τα εξής:

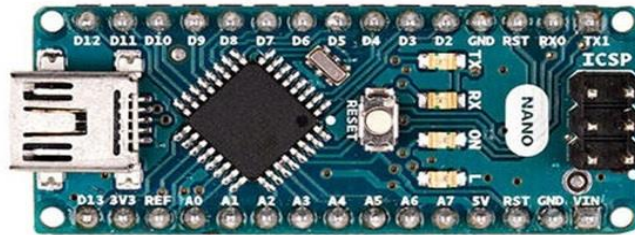
Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Πίνακας 3 Βασικά Χαρακτηριστικά Arduino Mega 2560

[36]

5.5. Arduino Nano ATmega328



Εικόνα 25 Κάτοψη πλακέτας Arduino Nano

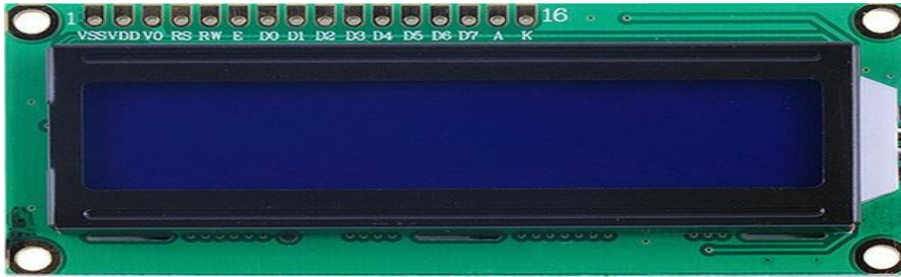
Για τις ανάγκες των υποσυστημάτων της εργασίας χρησιμοποιήθηκαν Arduino Nano. Περιέχει τον ίδιο μικροελεγκτή ATmega328 της Atmel με το Arduino UNO. Το βασικό πλεονέκτημα αυτής της πλακέτας είναι το μέγεθος της αφού είναι πολύ μικρή και μπορεί να εφαρμοστεί σε Projects με μικρό φυσικό μέγεθος. Η μόνη διαφορά είναι ότι διαθέτει 8 αναλογικές ακίδες (οι οποίες μπορούν να χρησιμοποιηθούν και σαν ψηφιακές υποδοχές) αντί για 6 και mini USB αντί του τυπικού του Arduino UNO. Τα χαρακτηριστικά του είναι τα εξής:

Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog I/O Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12 V
Digital I/O Pins	22
PWM Output	6
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7 g
Product Code	A000005

Πίνακας 4 Βασικά Χαρακτηριστικά Arduino Nano

[15]

5.6. LCD display 1602



Εικόνα 26 Οθόνη LCD 1602

Για τη παρούσα εργασία χρησιμοποιήθηκε οθόνη LCD 1602. Η Οθόνη αυτή είναι μια συσκευή εξόδου που ελέγχεται απευθείας από το Arduino. Από την προδιαγραφή του ονόματος της είναι μια οθόνη με τεχνολογία LCD 2 γραμμών και 16 χαρακτήρων η κάθε γραμμή. Διαθέτει οπίσθιο φωτισμό led και 16 ακροδέκτες για τη σύνδεση με κάποιο μικροελεγκτή. Επιπλέον οι περισσότερες οθόνες είναι συμβατές με το chipset Hitachi HD44780. [7]

5.7. Active Buzzer

Το buzzer (βομβητής) είναι μια συσκευή ήχου η οποία μπορεί να είναι μηχανική, ηλεκτρομηχανική, πιεζοηλεκτρική. Τέτοιου είδους βομβητές χρησιμοποιούνται σε συσκευές συναγερμού, χρονοδιακόπτες, χρονόμετρα κτλ. [16]



Εικόνα 27 Active Buzzer

Τα χαρακτηριστικά του βομβητή είναι τα εξής:

- Operating Voltage: 3.5V ~ 5.5 V
- Maximum Current: 30mA / 5V DC
- Resonance Frequency: 2500 Hz \pm 300 Hz [17]

5.8. Solar Panel

Για την ενεργειακή αυτονομία των υποσυστημάτων χρησιμοποιήθηκαν ηλιακά πάνελ τα εξής χαρακτηριστικά:



Εικόνα 28 Solar Panel

- Power: 1.1 W
- Voltage: 6V
- Size: 112x84 mm
- Weight: 36 g

5.9. MT3608 Boost converter Step up DC-DC

Για τις ανάγκες της εργασίας χρησιμοποιήθηκε boost converter με τα εξής χαρακτηριστικά:

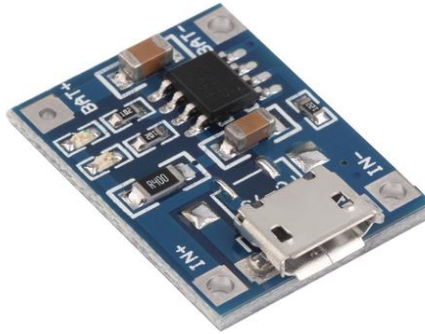


Εικόνα 29 Boost Converter Step up DC - DC

- Input Voltage: 2V ~ 24V
- Maximum output current: 2A
- output Voltage: 5V ~ 28V
- Maximum output Voltage: 28V
- Efficiency: up to 97%
- Switching frequency: 1.2 MHz [37]

5.10. Battery charger TP4056

Ένα στοιχείο ακόμη που χρησιμοποιήθηκε στα υποσυστήματα της ενεργειακής αυτονομίας είναι ένας φορτιστής μπαταρίας TP4056. Τα χαρακτηριστικά του είναι:



Εικόνα 30 Battery charger TP4056

- Input Voltage: 4V~5V
- Max input Voltage: $\leq 8V$
- Full charging Voltage: 4.2 V
- Max output current: 1000 mA
- Δείκτης τροφοδοσίας: Το κόκκινο led σημαίνει ότι η μπαταρία φορτίζει ενώ το πράσινο ότι η μπαταρία είναι πλήρως φορτισμένη.
- Διαθέτει mini USB υποδοχή. [18]

5.11. RTC DS3231 AT2432 IIC Clock Module

Για τη λειτουργία του project χρειάστηκε μονάδα ρολογιού. Το DS3231 είναι μία συσκευή χαμηλού κόστους με υψηλή ακρίβεια. Διαθέτει υποδοχή για μπαταρία και σε περίπτωση διακοπής ρεύματος αντλεί τάση από τη μπαταρία. Το RTC μετράει δευτερόλεπτα, λεπτά, ώρες και ημερομηνία καθώς και μέρα, μήνα, έτος και προσαρμόζεται αυτόματα για μήνες με λιγότερες από 31 ημέρες. Τα χαρακτηριστικά του είναι:

- Operating Voltage: 2.3 – 5V
- Clock chip: τσιπ υψηλής ακρίβειας DS 3231
- Weight: 8g
- Low - Power Consumption
- Operating Temperature Ranges
 - Commercial: 0°C to +70°C
 - Industrial: -40°C to +85°C [19]

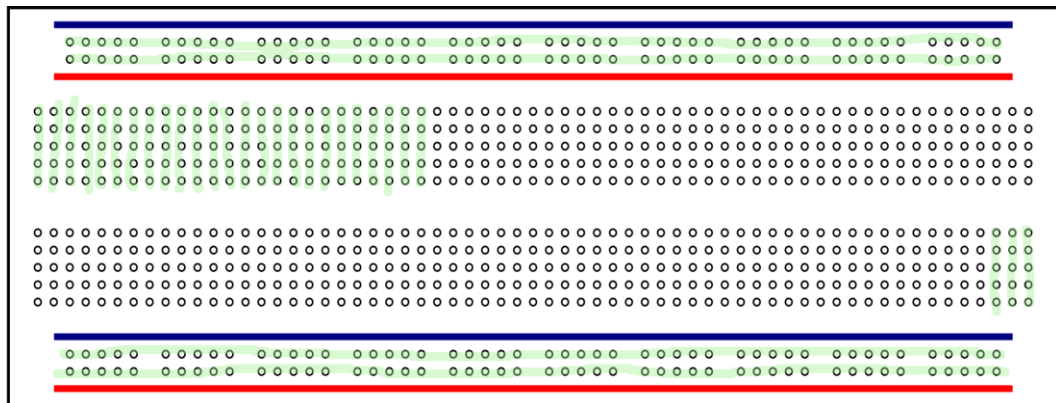


Εικόνα 31 Πλακέτα εξωτερικού ρολογιού RTC

5.12. Άλλα εξαρτήματα

Στη υλοποίηση της παρούσας εργασίας χρησιμοποιήθηκαν και τα παρακάτω εξαρτήματα:

- **Breadboard:** Το Breadboard χρησιμοποιείται για τη σύνθεση κυκλωμάτων. Η δομή του είναι η εξής: αποτελείται από κατακόρυφες πεντάδες όπου οι τρύπες είναι αγωγικές μεταξύ τους αφού στο εσωτερικό τους συνδέονται με χαλκό με αποτέλεσμα τα εξαρτήματα που συνδέονται στην ίδια πεντάδα να είναι αγωγικά μεταξύ τους. Επιπλέον έχει οριζόντιες γραμμές στη πάνω και κάτω πλευρά του breadboard με μεγάλο πλήθος από τρύπες οι οποίες έχουν “οριζόντια” αγωγιμότητα μεταξύ τους και χρησιμοποιούνται περισσότερο για την τροφοδοσία των εξαρτημάτων. [20]



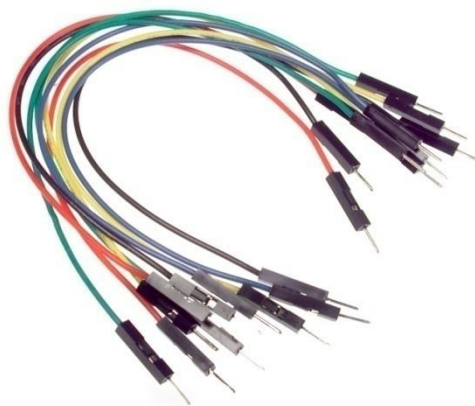
Εικόνα 32 Σχεδιάγραμμα αγωγιμότητας ενός Breadboard

- **Αντιστάσεις:** Οι αντιστάσεις στα ηλεκτρονικά κυκλώματα χρησιμοποιούνται για να περιορίσουν τη ροή ρεύματος και τη πτώση τάσης στα άκρα τους. Το μέγεθος της αντίστασης ενός υλικού εξαρτάται από τη διάμετρο, το μήκος και το στοιχείο που είναι κατασκευασμένο το υλικό. Όσο μεγαλώνει η διάμετρος τόσο μικραίνει η αντίσταση και όσο μεγαλύτερο είναι το μήκος του υλικού τόσο αυξάνεται η αντίσταση. [21]



Εικόνα 33 Αντιστάσεις

- **Wires (καλώδια συνδεσμολογίας):** Τα wires είναι απλά καλώδια με γυμνές άκρες που χρησιμοποιήθηκαν για τη σύνδεση των συστημάτων.



Εικόνα 34 Καλώδια συνδεσμολογίας

- **Καλώδια USB, Ethernet:** Χρησιμοποιήθηκε καλώδιο USB τύπου A - B για τον προγραμματισμό της πλακέτας Arduino και καλώδιο κατηγορίας E Ethernet με φισ RJ45 για τη δικτύωση του project με το δίκτυο.



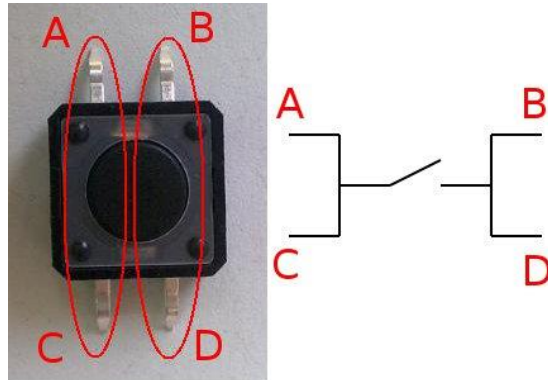
Εικόνα 35 Καλώδια Ethernet, USB τύπου A – B

- **Δίοδοι Leds:** Τα Leds εκπέμπουν φως στην ορθή πόλωση τους και κατασκευάζονται σε διάφορα χρώματα (πχ. Κόκκινο, πράσινο, κίτρινο, άσπρο). Η αρχή λειτουργίας των Leds βασίζεται στο γεγονός ότι στην ορθή πόλωση τους δημιουργούνται επανασυνδέσεις οπών και ηλεκτρονίων με αποτέλεσμα να απελευθερώνεται ενέργεια με τη μορφή ηλεκτρομαγνητικών κυμάτων. Τα πλεονεκτήματά τους είναι πολλά σε σχέση με τις ενδεικτικές λυχνίες όπως: έχουν πολύ μικρή κατανάλωση, έχουν μικρό μέγεθος και μεγάλη διάρκεια ζωής. [22]



Εικόνα 36 Δίοδοι led

- Working current: 20 mA
 - Working voltage: 1.8 – 2.2 V [23]
- **Διακόπτης (Button):** Στο Arduino συνήθως χρησιμοποιούνται διακόπτες με τέσσερις ακροδέκτες. Έναν ακροδέκτη σε κάθε γωνία. Οι ακροδέκτες είναι μεταξύ τους συνδεδεμένοι μόνο κατακόρυφα (Ο αριστερά πάνω με τον αριστερά κάτω και το ίδιο με τους δεξιούς ακροδέκτες). Έτσι το αριστερό τμήμα του διακόπτη είναι απομονωμένο από το δεξί και όταν πατηθεί το κουμπί ο διακόπτης κλείνει και αποκαθίσταται η οριζόντια σύνδεση. [7]



Εικόνα 37 Διακόπτης με 4 ακροδέκτες

6. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Για την εγγραφή του κώδικα όλου του project χρησιμοποιήθηκε η έκδοση Arduino IDE 1.8.1 και προγραμματίστηκε στη γλώσσα wiring, μια παραλλαγή της C, C++. Παρακάτω παραθέτεται ο κώδικας του project: [35]

Κώδικας του κεντρικού πίνακα ελέγχου:

- Ο κώδικας (sketch όπως ονομάζονται τα προγράμματα στο Arduino) όπως αναφέρθηκε ξεκινάει με την εισαγωγή των βιβλιοθηκών για κάθε ένα εξάρτημα ή λειτουργία που χρησιμοποιήσουμε.

//Δηλώσεις βιβλιοθηκών

```
#include <VirtualWire.h> //Δήλωση βιβλιοθήκης για την επικοινωνία των RF μονάδων.  
#include <LiquidCrystal.h> //Δήλωση βιβλιοθήκης για την LCD οθόνη.  
#include <Wire.h> //Δήλωση βιβλιοθήκης wire. Χρησιμοποιείται για επικοινωνία συσκευών I2C. Μια τέτοια συσκευή είναι το εξωτερικό ρολόι RTC στη παρούσα εργασία.  
#include "RTCLib.h" //Δήλωση βιβλιοθήκης για το εξωτερικό ρολόι RTC.  
#include <Ethernet.h> //Δήλωση βιβλιοθήκης για την Ethernet shield.  
#include <SPI.h> //Δήλωση βιβλιοθήκης για την επικοινωνία του Arduino MEGA με την Ethernet shield.
```

Με την λέξη #include εισάγουμε βιβλιοθήκη βάζοντάς την σε εισαγωγικά ή αποσιωπητικά όταν αυτή βρίσκεται σε αρχείο. Οι βιβλιοθήκες είτε παρέχονται από το λογισμικό για τις πιο γνωστές λειτουργίες είτε υπάρχουν στο διαδίκτυο για κάθε ένα από τα επιπρόσθετα εξαρτήματα που χρησιμοποιούμε.

- Στη συνέχεια δηλώνονται οι μεταβλητές που θα χρειαστούν στο πρόγραμμα:

//Δήλωση μεταβλητών

```
int pir_state=0;  
int pir_state1=0;  
char c = 0;  
char command[2] = "\0";  
int i=0;  
boolean LED_status = 0;  
int pinSpeaker = 8;  
const int led_pin = 9;  
const int ledpin = 5;  
const int receive_pin = 6;  
int transmit_pin = 7;  
const int button1 = 23;  
const int button2 = 22;
```

```
int val=0;
int val2=0;
int val3=0;
int val4=0;
```

```
RTC_DS3231 rtc;
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"}; //Δημιουργία πίνακα για το ρολόι.
```

```
LiquidCrystal lcd(48, 49, 44, 45, 46, 47); //Δημιουργία μεταβλητής τύπου
LiquidCrystal οι παράμετροι μέσα στη παρένθεση είναι οι ψηφιακές υποδοχές που
συνδέθηκαν τα καλώδια με το Arduino MEGA.
```

```
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED}; //Δήλωση της MAC address
της Ethernet Shield.
```

```
IPAddress ip(192, 168, 1, 177); //Δήλωση της διεύθυνσης IP που θα λειτουργεί η
Ethernet shield.
```

```
IPAddress gateway( 192, 168, 1, 1 ); //Δήλωση της διεύθυνσης Gateway του router του
τοπικού μου δικτύου.
```

```
IPAddress subnet( 255, 255, 255, 0 ); //Δήλωση της μάσκας υποδικτύου του τοπικού
μου δικτύου.
```

```
EthernetServer server(80); //Η θύρα 80 είναι η προεπιλεγμένη θύρα που
λειτουργεί ο HTTP server .
```

```
String HTTP_req;
```

```
char Server[] = "smtpcorp.com"; //Δημιουργία ενός πίνακα που περιέχει το όνομα
του SMTP server.
```

```
int port = 2525; //2525 είναι η θύρα που λειτουργεί η υπηρεσία
SMTP.
```

```
EthernetClient client; //Δημιουργία ενός πελάτη που θα μπορεί να
συνδεθεί με μια συγκεκριμένη IP και μια θύρα.
```

- Αφού τελείωσαν οι δηλώσεις στη συνέχεια εκτελείται η ρουτίνα setup(). Η συγκεκριμένη συνάρτηση εκτελείται μόνο μία φορά μέσα στο πρόγραμμα και χρησιμοποιείται για την αρχικοποίηση των εξαρτημάτων και τη λειτουργία των ακίδων του Arduino.

```
void setup() //Καλείται η συνάρτηση.
```

```
{
```

```
lcd.begin(16, 2); //Ορισμός διαστάσεων της οθόνης. Οι διαστάσεις είναι 2
γραμμές με 16 χαρακτήρες.
```

```
Serial.begin(9600); //Αρχικοποίηση σειριακής επικοινωνίας με το Arduino
στα 9600 bps.
```

Ethernet.begin(mac, ip, gateway, gateway, subnet); // Αρχικοποιεί την βιβλιοθήκη Ethernet και τις ρυθμίσεις δικτύου που ορίσαμε παραπάνω.

server.begin();// Ενημερώνει το διακομιστή για να αρχίσει να «ακούει» τις εισερχόμενες συνδέσεις.

Serial.println(Ethernet.localIP()); //Εκτυπώνει τη τοπική διεύθυνση IP στο serial monitor του Arduino IDE.

- Το παρακάτω κομμάτι κώδικα αναφέρεται στο RTC ρολόι. Ελέγχει αν έχει γίνει σωστή αρχικοποίηση της συνάρτησης και ορίζει το ρολόι με ημερομηνία και ώρα αλλιώς εκτυπώνει μήνυμα στο serial monitor «δεν βρέθηκε ρολόι».

```
if (! rtc.begin())
```

```
{  
  Serial.println("Couldn't find RTC");  
  while (1);  
}
```

```
if (rtc.lostPower())
```

```
{  
  Serial.println("RTC lost power, lets set the time!");  
  // following line sets the RTC to the date & time this sketch was compiled  
  rtc.adjust(DateTime(2017, 5, 13, 22, 40, 30));  
}
```

vw_set_tx_pin(transmit_pin); //Καθορίζεται η ακίδα με τη μεταβλητή transmit_pin ως ακίδα μετάδοσης δεδομένων.

vw_set_rx_pin(receive_pin); //Καθορίζεται η ακίδα με τη μεταβλητή receive_pin ως ακίδα λήψης δεδομένων.

vw_setup(400); //Αρχικοποιεί τη βιβλιοθήκη VirtualWire και δηλώνει τη ταχύτητα επικοινωνίας των δεδομένων (4000 bps). Όλες οι ακίδες πρέπει να δηλωθούν πριν από αυτή την εντολή.

vw_set_ptt_inverted(true); //Διαμορφώνει τη πολικότητα.

vw_rx_start(); //Ενεργοποιεί τη διαδικασία λήψης. Πρέπει να καλεστεί αυτή η συνάρτηση πριν οποιαδήποτε λήψη δεδομένων.

pinMode(led_pin, OUTPUT); //Καθορίζει το ψηφιακό ακροδέκτη με τη μεταβλητή led_pin ως έξοδο.

pinMode(ledpin, OUTPUT); //Καθορίζει το ψηφιακό ακροδέκτη με τη μεταβλητή ledpin ως έξοδο.

pinMode(13,OUTPUT); //Καθορίζει το ψηφιακό ακροδέκτη 13 ως έξοδο.

pinMode(3,OUTPUT); // Καθορίζει το ψηφιακό ακροδέκτη 3 ως έξοδο.

pinMode(4,OUTPUT); // Καθορίζει το ψηφιακό ακροδέκτη 4 ως έξοδο.

pinMode(pinSpeaker, OUTPUT); //Καθορίζει το ψηφιακό ακροδέκτη με τη μεταβλητή pinSpeaker ως έξοδο.

```
pinMode(button1,INPUT);           //Καθορίζει το ψηφιακό ακροδέκτη με τη
μεταβλητή button1 ως είσοδο.
pinMode(button2,INPUT);           //Καθορίζει το ψηφιακό ακροδέκτη με τη
μεταβλητή button2 ως είσοδο.
digitalWrite(4,HIGH);

}
```

- Έπειτα ξεκινάει η ρουτίνα Void loop() η οποία καταλαμβάνει το μεγαλύτερο κομμάτι κώδικα και είναι ο βασικός κορμός του προγράμματος. Η συνάρτηση αυτή επαναλαμβάνεται συνεχώς.

```
void loop()                         //Καλείται η συνάρτηση
{
  DateTime now = rtc.now();          //Καλεί τη συνάρτηση now(). Η συνάρτηση
επιστρέφει ένα αντικείμενο τύπου datetime που περιγράφει παρακάτω την ώρα, λεπτά,
δευτερόλεπτα.
  char msg[1] = {'2'};              //Δημιουργία πίνακα msg με ένα χαρακτήρα
μέσα.
  digitalWrite(button1);            //Επιστρέφει τη κατάσταση του button1.
  digitalWrite(button2);            //Επιστρέφει τη κατάσταση του button2.
```

- Δημιουργία ενός πίνακα που αντιγράφεται το μήνυμα που στέλνει ο πομπός και τύπο δεδομένων uint8_t και buflen είναι το μέγιστο μέγεθος του πακέτου. Αυτός ο πίνακας χρησιμοποιείται από την VirtualWire για τον δέκτη.

```
uint8_t buf[buflen];
uint8_t buflen = VW_MAX_MESSAGE_LEN;
```

- Το παρακάτω κομμάτι κώδικα είναι πολύ σημαντικό και αναφέρεται στα πακέτα που δέχεται ο δέκτης στο κεντρικό πίνακα ελέγχου.
- Ελέγχει το μήνυμα που έφτασε στο δέκτη. Αν το μήνυμα είναι ο χαρακτήρας «1» σημαίνει ότι παραβιάστηκε η πόρτα1 επομένως: Ενεργοποιεί το κόκκινο led που αντιστοιχεί στη πόρτα1, εμφανίζει στην lcd οθόνη το μήνυμα «D1 opened», παράλληλα εμφανίζει την συγκεκριμένη ώρα, ενεργοποιείται η συνάρτηση του ηχείου, θέτει 1 τη μεταβλητή pir_state (θα δούμε παρακάτω που είναι χρήσιμη αυτή η μεταβλητή) και ελέγχει τις συναρτήσεις sendEmail1(), sendEmail2() αν είναι σωστές. Αν είναι στέλνει το κατάλληλο email στο παραλήπτη.

```
if (vw_get_message(buf, &buflen))   //Έλεγχος του πακέτου που έλαβε ο
δέκτης.
{
  if(buf[0]=='1')                    //Αν λάβει πακέτο με το χαρακτήρα «1».
  {
    digitalWrite(ledpin,HIGH);       //Ενεργοποιεί το led.
    lcd.setCursor(0,0);              //Θέτει στο δρομέα της οθόνης να
ξεκινήσει την εγγραφή από την πρώτη στήλη και γραμμή.
    lcd.print("D1 opened");          //Εκτυπώνει η οθόνη το μήνυμα που
υπάρχει μέσα στη παρένθεση.
```



```
    lcd.print(now.hour(), DEC);           //Εκτυπώνει την συγκεκριμένη ώρα σε
    δεκαδική μορφή.
    lcd.print(':');
    lcd.print(now.minute(), DEC);       //Εκτυπώνει τα συγκεκριμένα λεπτά σε
    δεκαδική μορφή.
    lcd.print(':');
    lcd.print(now.second(), DEC);       //Εκτυπώνει τα συγκεκριμένα
    δευτερόλεπτα σε δεκαδική μορφή.
    playTone(300, 100);                 //Καλεί τη συνάρτηση του ηχείου.
    pir_state=1;                        //Θέτει τη μεταβλητή pir_state=1.
    if(sendEmail())                     //Ελέγχει τη συνάρτηση του email.
        Serial.println(F("Email sent")); //Εκτυπώνει μήνυμα στο serial monitor
    του Arduino ότι στάλθηκε το email.
    else
        Serial.println(F("Email failed")); //Εκτυπώνει μήνυμα στο serial monitor
    του Arduino ότι δεν στάλθηκε.
}
```

- Αν το μήνυμα που λάβει ο δέκτης είναι ο χαρακτήρας «0» σημαίνει ότι σταμάτησε η παραβίαση της πόρτας1 οπότε απενεργοποιεί το led που αντιστοιχεί στη πόρτα1 και αλλάζει το μήνυμα που εμφανίζει η οθόνη. Τέλος απενεργοποιεί το ηχείο.

```
if( buf[0]== '0')                     //Αν λάβει πακέτο με το χαρακτήρα «0».
{
    digitalWrite(ledpin,LOW);          //Απενεργοποιεί το led.
    lcd.setCursor(0,0);                 //Θέτει στο δρομέα της οθόνης να
    ξεκινήσει την εγγραφή από την πρώτη στήλη και γραμμή.
    lcd.print("Door 1 closed ");        //Εκτυπώνει η οθόνη το μήνυμα που
    υπάρχει μέσα στη παρένθεση.
    playTone(0, 0);                     //Καλεί τη συνάρτηση του ηχείου με
    μηδενικές παραμέτρους που σημαίνει να σταματήσει να εκπέμπει ηχητικό σήμα.
    pir_state=0;                        //Θέτει τη μεταβλητή pir_State=0.
}
```

- Αν το μήνυμα που λάβει ο δέκτης είναι ο χαρακτήρας «3» σημαίνει ότι παραβιάστηκε η πόρτα2 επομένως: Ενεργοποιεί το led που αντιστοιχεί στη πόρτα 2, εκτυπώνει στην οθόνη το κατάλληλο μήνυμα και την ώρα, ενεργοποιεί τη συνάρτηση του ηχείου και στέλνει email στο παραλήπτη.

```
if(buf[0]== '3')                       //Αν το πακέτο που λάβει ο δέκτης είναι
ο χαρακτήρας «3».
{
    digitalWrite(led_pin,HIGH);        //Ενεργοποιεί το led της πόρτας 2.
    lcd.setCursor(0,2);                 //Θέτει στο δρομέα της οθόνης να
    ξεκινήσει την εγγραφή από την πρώτη στήλη και δεύτερη γραμμή.
    lcd.print("D2 opened");             //Εκτυπώνει η οθόνη το μήνυμα που
    υπάρχει μέσα στη παρένθεση.
    lcd.print(now.hour(), DEC);         //Εκτυπώνει την συγκεκριμένη ώρα σε
    δεκαδική μορφή.
    lcd.print(':');
    lcd.print(now.minute(), DEC);       //Εκτυπώνει τα συγκεκριμένα λεπτά σε
    δεκαδική μορφή.
}
```

```
    lcd.print(':');  
    lcd.print(now.second(), DEC);           //Εκτυπώνει τα συγκεκριμένα  
    δευτερόλεπτα σε δεκαδική μορφή.  
    playTone(400, 200);                     //Καλεί τη συνάρτηση του ηχείου.  
    if(sendEmail2())                         //Ελέγχει τη συνάρτηση του email.  
        Serial.println(F("Email sent"));    //Εκτυπώνει μήνυμα στο serial monitor  
    του Arduino ότι στάλθηκε το email.  
    else  
        Serial.println(F("Email failed"));  //Εκτυπώνει μήνυμα στο serial monitor  
    του Arduino ότι δεν στάλθηκε.  
    pir_state1=1;                           //Θέτει τη μεταβλητή pir_State1=1.  
    }
```

- Αν το μήνυμα που λάβει ο δέκτης είναι ο χαρακτήρας «4» σημαίνει ότι σταμάτησε η παραβίαση της πόρτας 2 οπότε απενεργοποιεί το led που αντιστοιχεί στη πόρτα 2 και αλλάζει το μήνυμα που εμφανίζει η οθόνη. Τέλος απενεργοποιεί το ηχείο.

```
    if(buf[0]=='4')                          //Αν λάβει πακέτο με το χαρακτήρα «4».  
    {  
        digitalWrite(led_pin,LOW);          //Απενεργοποιεί το led.  
        lcd.setCursor(0,2);                 //Θέτει στο δρομέα της οθόνης να  
        ξεκινήσει την εγγραφή από την πρώτη στήλη και δεύτερη γραμμή.  
        lcd.print("Door 2 closed ");        //Εκτυπώνει η οθόνη το μήνυμα που  
        υπάρχει μέσα στη παρένθεση.  
        playTone(0, 0);                     //Καλεί τη συνάρτηση του ηχείου με  
        μηδενικές παραμέτρους που σημαίνει να σταματήσει να εκπέμπει ηχητικό σήμα.  
        pir_state1=0;                       //Θέτει τη μεταβλητή pir_State1=0.  
    }  
}
```

- Το παρακάτω τμήμα κώδικα αναφέρεται στη web εφαρμογή που έχω υλοποιήσει και περιέχει τη σύνδεση της Ethernet shield με το δίκτυο, το γραφικό περιβάλλον της εφαρμογής και τον έλεγχο των φώτων μέσω web στα 2 υποσυστήματα.

```
EthernetClient client = server.available(); //Για ένα πρόγραμμα-πελάτη  
υπάρχουν διαθέσιμα δεδομένα στο διακομιστή.  
boolean current_line_is_first = true;
```

//Αν ο πελάτης έχει συνδεθεί χωρίς σφάλματα στο web εμφανίζεται στην οθόνη του τερματικού το γραφικό περιβάλλον το οποίο αναλύεται παρακάτω:

```
if (client) {  
    boolean current_line_is_blank = true;  
    while (client.connected()) {  
        if (client.available()) {  
            char c = client.read();  
            if (c == '\n' && current_line_is_blank) {
```

```
if(HTTP_req.indexOf("b=1")>-1) {
    LED_status=1;
}
if(HTTP_req.indexOf("b=2")>-1) {
    LED_status=0;
}
```

//Δημιουργία του γραφικού περιβάλλοντος της ιστοσελίδας. Ο προγραμματισμός μέσα στις συναρτήσεις client.println έγινε με τη χρήση της γλώσσας HTML. Ο προγραμματισμός ξεκινάει όταν ανοίγει <html> και τελειώνει όταν κλείσει </html>.

```
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html"); //Δημιουργία της HTML
σελίδας.
client.println();
client.println("<META HTTP-EQUIV=REFRESH CONTENT=10 URL=>");
//Αυτή η εντολή σημαίνει ότι κάνει refresh την ιστοσελίδα ανά 10 δευτερόλεπτα.
client.println("<html>");
client.println("<head>");
client.println("<title> LED Control Security Alarm System Tei of Epirus
</title>"); //Δημιουργία τίτλου που εκτυπώνεται στο tab της ιστοσελίδας.
client.println("</head>");
client.println("<body style='background-color:lightgrey'>"); //Καλύπτει το
background της ιστοσελίδας με γκρι χρώμα.
client.println("<center><p><h1>Security alarm system TEI of Epirus
</h1></p><center><hr><br />"); //εκτυπώνει τον παραπάνω τίτλο στο κέντρο της
ιστοσελίδας.
client.println("<form method='get'>");
```

//Στο συγκεκριμένο τμήμα κώδικα εμφανίζεται μήνυμα στην ιστοσελίδα αν υπάρχει κίνηση στη πόρτα1 ή αν δε υπάρχει κίνηση. Το μήνυμα αυτό τροποποιείται λόγω των μεταβλητών pir_state=1 και pir_state=0 που θέσαμε παραπάνω.

```
if (pir_state==1)
{
    client.println("<p><h2><font color=red>Motion Detected DOOR
1!</font></h2></p>");
}
else
{
    client.println("<p><h2><font color=green>No Movement DOOR
1!</font></h2></p>");
}
```

//Στο συγκεκριμένο τμήμα κώδικα εμφανίζεται μήνυμα στην ιστοσελίδα αν υπάρχει κίνηση στη πόρτα2 ή αν δε υπάρχει κίνηση. Το μήνυμα αυτό τροποποιείται λόγω των μεταβλητών pir_state1=1 και pir_state1=0 που θέσαμε παραπάνω.

```
if (pir_state1==1)
{
    client.println("<p><h2><font color=red>Motion Detected DOOR
2!</font></h2></p>");
}
```

```
    }
    else
    {
        client.println("<p><h2><font color=green>No Movement DOOR
2!</font></h2></p>");
    }

    client.println("</form>");
    client.println("</body>");
    client.println("</html>"); //Κλείσιμο του γραφικού περιβάλλοντος της
ιστοσελίδας.
    HTTP_req = "";
    break;
}
if (c == '\n') {
    current_line_is_first = false;
    current_line_is_blank = true;
}
else if (c != '\r') {
    current_line_is_blank = false;
}
if (current_line_is_first && c == '=') {
    for (int i = 0; i < 1; i++) {
        c = client.read();
        command[i] = c;
    }
}

}

}

delay(1); //Δίνει μια μικρή καθυστέρηση στο
browser να λάβει τα δεδομένα.
client.stop(); //Γίνεται αποσύνδεση του πελάτη από το
Web.
Serial.println("client disconnected"); //Εκτυπώνει στο serial monitor την
κατάσταση της σύνδεσης. Δηλαδή ότι ο πελάτης αποσυνδέθηκε.
}

}
```

- ο Η συνάρτηση playTone είναι μια έτοιμη συνάρτηση που χρησιμοποιήθηκε για την εκπομπή ηχητικών σημάτων από το buzzer. Ρυθμίζοντας τις παραμέτρους της συνάρτησης τροποποιείται και το ηχητικό σήμα.

```
void playTone(long duration, int freq)
{
    duration *= 1000;
    int period = (1.0 / freq) * 100000;
    long elapsed_time = 0;
    while (elapsed_time < duration)
```

```
{
  digitalWrite(pinSpeaker,HIGH);
  delayMicroseconds(period / 2);
  digitalWrite(pinSpeaker, LOW);
  delayMicroseconds(period / 2);
  elapsed_time += (period);
}
}
```

- ο Το τελευταίο τμήμα του κώδικα του κεντρικού πίνακα ελέγχου είναι οι συναρτήσεις `sendEmail1()`, `sendEmail2()` όπου χρησιμοποιούνται για την αποστολή e-mail αν κάποιο από τα υποσυστήματα δηλώσει παραβίαση πόρτας. Αυτό το κομμάτι είναι τυποποιημένο από την επίσημη ιστοσελίδα του Arduino.cc οπότε θα αναφερθούμε μόνο σε συγκεκριμένα μέρη που χρειάζεται επεξήγηση.

// Συνάρτηση για την αποστολή email αν παραβιαστεί η πόρτα1.

```
byte sendEmail1()
```

```
{
  byte thisByte = 0;
  byte respCode;

  if(client.connect(Server,port) == 1) {
    Serial.println(F("connected"));
  } else {
    Serial.println(F("connection failed"));
    return 0;
  }
}
```

```
if(!eRcv()) return 0;
```

```
Serial.println(F("Sending hello"));
```

```
// replace 1.2.3.4 with your Arduino's ip
```

```
client.println("EHLO 192.168.1.177");
```

//Θέτουμε την διεύθυνση IP που λειτουργεί η Ethernet Shield.

```
if(!eRcv()) return 0;
```

```
Serial.println(F("Sending auth login"));
```

```
client.println("auth login");
```

```
if(!eRcv()) return 0;
```

```
Serial.println(F("Sending User"));
```

```
// Change to your base64 encoded user
```

```
client.println("bWl4b3Nwd25AZ21haWwuY29t"); //Γράφουμε το e-mail με το
```

οποίο κάναμε εγγραφή στην υπηρεσία SMTP2GO σε κωδικοποιημένη μορφή base64 (όπως αναφέρθηκε σε προηγούμενο κεφάλαιο).

```
if(!eRcv()) return 0;
```

```
Serial.println(F("Sending Password"));
// change to your base64 encoded password
client.println("V0ρpnNzZSWWIFVjNy"); //Γράφουμε το password με το οποίο
κάναμε εγγραφή στην υπηρεσία SMTP2GO σε κωδικοποιημένη μορφή base64 (όπως
αναφέρθηκε σε προηγούμενο κεφάλαιο).

if(!eRcv()) return 0;

// change to your email address (sender)
Serial.println(F("Sending From"));
client.println("MAIL From: <mixospwn@gmail.com>"); //Γράφουμε το e-mail
του αποστολέα.
if(!eRcv()) return 0;

// change to recipient address
Serial.println(F("Sending To"));
client.println("RCPT To: < mixospwn@gmail.com >"); //Γράφουμε το e-mail
του παραλήπτη.
if(!eRcv()) return 0;

Serial.println(F("Sending DATA"));
client.println("DATA");
if(!eRcv()) return 0;

Serial.println(F("Sending email"));

// change to recipient address
client.println("To: < mixospwn@gmail.com >"); //Γράφουμε το e-mail του
παραλήπτη.

// change to your address
client.println("From: <mixospwn@gmail.com>"); //Γράφουμε το e-mail του
αποστολέα.

client.println("Subject: Arduino Alarm System\r\n"); //Γράφουμε το θέμα του e-mail
που θέλουμε να στείλει ο μικροελεγκτής μέσω της Ethernet shield.

client.println("DOOR 1 OPENED !!!"); //Γράφουμε το κείμενο που θέλουμε να
στείλουμε.

client.println(".");

if(!eRcv()) return 0;

Serial.println(F("Sending QUIT"));
```



```
client.println("QUIT");
if(!eRcv()) return 0;

client.stop();

Serial.println(F("disconnected"));

return 1;
}
// Συνάρτηση για την αποστολή email αν παραβιαστεί η πόρτα2.
byte sendEmail2()
{
  byte thisByte = 0;
  byte respCode;

  if(client.connect(Server,port) == 1) {
    Serial.println(F("connected"));
  } else {
    Serial.println(F("connection failed"));
    return 0;
  }

  if(!eRcv()) return 0;

  Serial.println(F("Sending hello"));
  // replace 1.2.3.4 with your Arduino's ip
  client.println("EHLO 192.168.1.177"); //Θέτουμε την διεύθυνση IP που
  //λειτουργεί η Ethernet Shield.
  if(!eRcv()) return 0;

  Serial.println(F("Sending auth login"));
  client.println("auth login");
  if(!eRcv()) return 0;

  Serial.println(F("Sending User"));
  // Change to your base64 encoded user
  client.println("bWl4b3Nwd25AZ21haWwuY29t"); //Γράφουμε το e-mail με το
  //οποίο κάναμε εγγραφή στην υπηρεσία SMTP2GO σε κωδικοποιημένη μορφή base64
  //(όπως αναφέρθηκε σε προηγούμενο κεφάλαιο).

  if(!eRcv()) return 0;

  Serial.println(F("Sending Password"));
  // change to your base64 encoded password
  client.println("V0pmNzZSWWlFVjNy"); //Γράφουμε το password με το οποίο
  //κάναμε εγγραφή στην υπηρεσία SMTP2GO σε κωδικοποιημένη μορφή base64 (όπως
  //αναφέρθηκε σε προηγούμενο κεφάλαιο).
```

```
if(!eRcv()) return 0;

// change to your email address (sender)
Serial.println(F("Sending From"));
client.println("MAIL From: <mixospwn@gmail.com>"); //Γράφουμε το e-mail
του αποστολέα.
if(!eRcv()) return 0;

// change to recipient address
Serial.println(F("Sending To"));
client.println("RCPT To: < mixospwn@gmail.com >"); //Γράφουμε το e-mail
του παραλήπτη.
if(!eRcv()) return 0;

Serial.println(F("Sending DATA"));
client.println("DATA");
if(!eRcv()) return 0;

Serial.println(F("Sending email"));

// change to recipient address
client.println("To: < mixospwn@gmail.com >"); //Γράφουμε το e-mail του
παραλήπτη.

// change to your address
client.println("From: <mixospwn@gmail.com>"); //Γράφουμε το e-mail του
αποστολέα.

client.println("Subject: Arduino Alarm System\r\n"); //Γράφουμε το θέμα του e-mail
που θέλουμε να στείλει ο μικροελεγκτής μέσω της Ethernet shield.

client.println("DOOR 2 OPENED !!!!"); //Γράφουμε το κείμενο που θέλουμε να
στείλουμε.

client.println(".");

if(!eRcv()) return 0;

Serial.println(F("Sending QUIT"));
client.println("QUIT");
if(!eRcv()) return 0;

client.stop();

Serial.println(F("disconnected"));
```

```
    return 1;
}

byte eRcv()
{
    byte respCode;
    byte thisByte;
    int loopCount = 0;

    while(!client.available()) {
        delay(1);
        loopCount++;

        // if nothing received for 10 seconds, timeout
        if(loopCount > 10000) {
            client.stop();
            Serial.println(F("\r\nTimeout"));
            return 0;
        }
    }

    respCode = client.peek();

    while(client.available())
    {
        thisByte = client.read();
        Serial.write(thisByte);
    }

    if(respCode >= '4')
    {
        efail();
        return 0;
    }

    return 1;
}

void efail()
{
    byte thisByte = 0;
    int loopCount = 0;

    client.println(F("QUIT"));
```

```
while(!client.available()) {
  delay(1);
  loopCount++;

  // if nothing received for 10 seconds, timeout
  if(loopCount > 10000) {
    client.stop();
    Serial.println(F("\r\nTimeout"));
    return;
  }
}

while(client.available())
{
  thisByte = client.read();
  Serial.write(thisByte);
}

client.stop();

Serial.println(F("disconnected"));
}
```

Κώδικας του 1^{ου} υποσυστήματος

- Ο κώδικας (sketch όπως ονομάζονται τα προγράμματα στο Arduino) όπως αναφέρθηκε ξεκινάει με την εισαγωγή των βιβλιοθηκών για κάθε ένα εξάρτημα ή λειτουργία που χρησιμοποιήσουμε.

```
#include <VirtualWire.h> //Δήλωση βιβλιοθήκης VirtualWire που
χρησιμοποιείται για την ασύρματη επικοινωνία των RF μονάδων.
```

- Στη συνέχεια γίνονται οι δηλώσεις των μεταβλητών
int j=0;
int i=0;
int led_pin = 13; //Δήλωση της μεταβλητής led_pin στο ψηφιακό pin 13.
int transmit_pin = 4; //Δήλωση της μεταβλητής transmit_pin στο ψηφιακό pin
4 η μεταβλητή αυτή χρησιμοποιείται για τα δεδομένα του πομπού.
int receive_pin=7; //Δήλωση της μεταβλητής receive_pin στο ψηφιακό pin
4. Η μεταβλητή αυτή χρησιμοποιείται για την λήψη των δεδομένων.
int pir_pin = 12; //Δήλωση της μεταβλητής pir_pin στο ψηφιακό pin 12. Η
μεταβλητή αυτή χρησιμοποιείται για την επικοινωνία του pir με το Arduino nano.
int val = 0;
int val2=0;
- Αφού τελείωσαν οι δηλώσεις στη συνέχεια εκτελείται η ρουτίνα setup(). Η συγκεκριμένη συνάρτηση εκτελείται μόνο μία φορά μέσα στο πρόγραμμα και

χρησιμοποιείται για την αρχικοποίηση των εξαρτημάτων και τη λειτουργία των ακίδων του Arduino

```
void setup() //Καλείται η συνάρτηση void setup().
{
  Serial.begin(9600); //Αρχικοποίηση σειριακής επικοινωνίας με το
  Arduino στα 9600 bps.
  vw_set_tx_pin(transmit_pin); //Καθορίζεται η ακίδα με τη μεταβλητή
  transmit_pin ως ακίδα μετάδοσης δεδομένων.
  vw_set_rx_pin(receive_pin); //Καθορίζεται η ακίδα με τη μεταβλητή
  receive_pin ως ακίδα λήψης δεδομένων.
  vw_setup(400); //Αρχικοποιεί τη βιβλιοθήκη VirtualWire και
  δηλώνει τη ταχύτητα επικοινωνίας των δεδομένων (4000 bps). Όλες οι ακίδες πρέπει να
  δηλωθούν πριν από αυτή την εντολή.
  vw_rx_start(); //Ενεργοποιεί τη διαδικασία λήψης. Πρέπει να
  καλεστεί αυτή η συνάρτηση πριν οποιαδήποτε λήψη δεδομένων.
  pinMode(led_pin, OUTPUT); //Καθορίζει το ψηφιακό ακροδέκτη led_pin ως
  έξοδο.
  pinMode(pir_pin, INPUT); //Καθορίζει τη μεταβλητή pir_pin ως είσοδο.
}
```

- Έπειτα ξεκινάει η ρουτίνα Void loop() η οποία καταλαμβάνει το μεγαλύτερο κομμάτι κώδικα και είναι ο βασικός κορμός του προγράμματος. Η συνάρτηση αυτή επαναλαμβάνεται συνεχώς.

```
void loop() //Καλείται η συνάρτηση void loop().
{
  char msg[1] = {'0'}; //Δημιουργία πίνακα msg με ένα χαρακτήρα
  μέσα.
```

- Δημιουργία ενός πίνακα που αντιγράφεται το μήνυμα που στέλνει ο πομπός και τύπο δεδομένων uint8_t ενώ buflen είναι το μέγιστο μέγεθος του πακέτου. Αυτός ο πίνακας χρησιμοποιείται από την VirtualWire για τον δέκτη.

```
uint8_t buflen = VW_MAX_MESSAGE_LEN;
uint8_t buf[buflen];
```

- Το φως της πόρτας1 δεν ελέγχεται μόνο από τον διακόπτη στον πίνακα ελέγχου, αλλά και από τα buttons της **web εφαρμογής**. Αν ο δέκτης λάβει πακέτο με το χαρακτήρα «5» σημαίνει ότι ο χρήστης απομακρυσμένα μέσω της web εφαρμογής ζητάει από τον μικροελεγκτή στο 1^ο υποσύστημα να ενεργοποιήσει τον φωτισμό. Αντίθετα αν λάβει το πακέτο «6» απενεργοποιεί τον φωτισμό.

```
if (buf[0]== '5') //Αν λάβει πακέτο με το χαρακτήρα «5».
{
  digitalWrite(led_pin,HIGH); //Ενεργοποιεί το led.
}
if (buf[0]== '6') //Αν λάβει πακέτο με το χαρακτήρα «6».
{
  digitalWrite(led_pin,LOW); //Απενεργοποιεί το led.
}
```

}

- Το παρακάτω τμήμα κώδικα είναι και το τελευταίο για το 1^ο υποσύστημα και αναπτύχθηκε για την λειτουργία του PIR αισθητήρα. Αν ο αισθητήρας ανιχνεύσει κίνηση στη πόρτα1 ο πομπός μεταδίδει πακέτο για να ειδοποιήσει τον πίνακα ελέγχου ότι η πόρτα παραβιάστηκε.

```
val2=digitalRead(pir_pin); //Διαβάζει τη κατάσταση του pir
αισθητήρα.
if (val2==HIGH&&val==0)
{
    val=1;
    msg[0] = '1'; //Θέτει στο πίνακα msg το χαρακτήρα
«1».
    vw_send((uint8_t *)msg, 1); //Μεταδίδει μέσω της VirtualWire το
πακέτο με το χαρακτήρα «1».
    vw_wait_tx(); //Αναμένει μέχρι να μεταδοθεί ολόκληρο
το πακέτο.
    digitalWrite(led_pin, HIGH); //Ενεργοποιεί το led.
    val2=LOW;
}
else if(val==1&&val2==LOW)
{
    msg[0] = '0'; //Θέτει στο πίνακα msg το χαρακτήρα
«0».
    vw_send((uint8_t *)msg, 1); //Μεταδίδει μέσω της VirtualWire το
πακέτο με το χαρακτήρα «0».
    vw_wait_tx(); //Αναμένει μέχρι να μεταδοθεί ολόκληρο
το πακέτο.
    digitalWrite(led_pin, LOW); //Απενεργοποιεί το led.
    val=0;
}
}
```

Κώδικας του 2^{ου} υποσυστήματος

- Ο κώδικας (sketch όπως ονομάζονται τα προγράμματα στο Arduino) όπως αναφέρθηκε ξεκινάει με την εισαγωγή των βιβλιοθηκών για κάθε ένα εξάρτημα ή λειτουργία που χρησιμοποιήσουμε.

```
#include <VirtualWire.h> //Δήλωση βιβλιοθήκης VirtualWire που
χρησιμοποιείται για την ασύρματη επικοινωνία των RF μονάδων.
```

- Στη συνέχεια γίνονται οι δηλώσεις των μεταβλητών

```
int i=0;
int led_pin = 11; //Δήλωση της μεταβλητής led_pin στο ψηφιακό
pin 11.
int transmit_pin = 4; //Δήλωση της μεταβλητής transmit_pin στο
ψηφιακό pin 4 η μεταβλητή αυτή χρησιμοποιείται για τα δεδομένα του πομπού.
int receive_pin=5; //Δήλωση της μεταβλητής receive_pin στο
ψηφιακό pin 5.Η μεταβλητή αυτή χρησιμοποιείται για την λήψη των δεδομένων.
```



```
int pir_pin = 12; //Δήλωση της μεταβλητής pir_pin στο ψηφιακό pin 12. Η μεταβλητή αυτή χρησιμοποιείται για την επικοινωνία του pir με το Arduino nano.
```

```
int val = 0, val2=0;
```

- Αφού τελείωσαν οι δηλώσεις στη συνέχεια εκτελείται η ρουτίνα setup(). Η συγκεκριμένη συνάρτηση εκτελείται μόνο μία φορά μέσα στο πρόγραμμα και χρησιμοποιείται για την αρχικοποίηση των εξαρτημάτων και τη λειτουργία των ακίδων του Arduino

```
void setup() //Καλείται η συνάρτηση void setup().  
{  
  Serial.begin(9600); //Αρχικοποίηση σειριακής επικοινωνίας με το Arduino στα 9600 bps.  
  vw_set_tx_pin(transmit_pin); //Καθορίζεται η ακίδα με τη μεταβλητή transmit_pin ως ακίδα μετάδοσης δεδομένων.  
  vw_set_rx_pin(receive_pin); //Καθορίζεται η ακίδα με τη μεταβλητή receive_pin ως ακίδα λήψης δεδομένων.  
  vw_setup(400); //Αρχικοποιεί τη βιβλιοθήκη VirtualWire και δηλώνει τη ταχύτητα επικοινωνίας των δεδομένων (4000 bps). Όλες οι ακίδες πρέπει να δηλωθούν πριν από αυτή την εντολή.  
  vw_rx_start(); //Ενεργοποιεί τη διαδικασία λήψης. Πρέπει να καλεστεί αυτή η συνάρτηση πριν οποιαδήποτε λήψη δεδομένων.  
  pinMode(led_pin, OUTPUT); //Καθορίζει το ψηφιακό ακροδέκτη led_pin ως έξοδο.  
  pinMode(pir_pin, INPUT); //Καθορίζει τη μεταβλητή pir_pin ως είσοδο.  
}
```

- Έπειτα ξεκινάει η ρουτίνα Void loop() η οποία καταλαμβάνει το μεγαλύτερο κομμάτι κώδικα και είναι ο βασικός κορμός του προγράμματος. Η συνάρτηση αυτή επαναλαμβάνεται συνεχώς.

```
void loop() //Καλείται η συνάρτηση void loop().  
{  
  char msg[1] = {'2'}; //Δημιουργία πίνακα χαρακτήρων msg με ένα χαρακτήρα μέσα.
```

- Δημιουργία ενός πίνακα που αντιγράφεται το μήνυμα που στέλνει ο πομπός και τύπο δεδομένων uint8_t ενώ buflen είναι το μέγιστο μέγεθος του πακέτου. Αυτός ο πίνακας χρησιμοποιείται από την VirtualWire για τον δέκτη

```
uint8_t buflen = VW_MAX_MESSAGE_LEN;  
uint8_t buf[buflen];
```

- Το φως της πόρτας2 δεν ελέγχεται μόνο από τον διακόπτη στον πίνακα ελέγχου, αλλά και από τα buttons της **web εφαρμογής**. Αν ο δέκτης λάβει πακέτο με το χαρακτήρα «a» σημαίνει ότι ο χρήστης απομακρυσμένα μέσω της web εφαρμογής ζητάει από τον μικροελεγκτή στο 2^ο υποσύστημα να ενεργοποιήσει τον φωτισμό. Αντίθετα αν λάβει το πακέτο «c» απενεργοποιεί τον φωτισμό.

```
if (buf[0]== 'a') //Αν λάβει πακέτο με το χαρακτήρα «a».  
{
```

```
digitalWrite(led_pin,HIGH);           //Ενεργοποιεί το led.
}
if (buf[0]== 'c')                     //Αν λάβει πακέτο με το χαρακτήρα «c».
{
digitalWrite(led_pin,LOW);           //Απενεργοποιεί το led.
}
}
```

- Το παρακάτω τμήμα κώδικα είναι και το τελευταίο για το 2^ο υποσύστημα και αναπτύχθηκε για την λειτουργία του PIR αισθητήρα. Αν ο αισθητήρας ανιχνεύσει κίνηση στη πόρτα2 ο πομπός μεταδίδει πακέτο για να ειδοποιήσει τον πίνακα ελέγχου ότι η πόρτα παραβιάστηκε.

```
val2=digitalRead(pir_pin);           //Διαβάζει τη κατάσταση του pir
αισθητήρα.
if (val2==HIGH&&val==0)
{
val=1;
msg[0] = '3';                         //Θέτει στο πίνακα msg το χαρακτήρα
«3».
vw_send((uint8_t *)msg, 1);           //Μεταδίδει μέσω της VirtualWire το
πακέτο με το χαρακτήρα «3».
vw_wait_tx();                         //Αναμένει μέχρι να μεταδοθεί ολόκληρο
το πακέτο.
digitalWrite(led_pin, HIGH);         //Ενεργοποιεί το led.
val2=LOW;
}
else if(val==1&&val2==LOW)
{
msg[0] = '4';                         //Θέτει στο πίνακα msg το χαρακτήρα
«4».
vw_send((uint8_t *)msg, 1);           //Μεταδίδει μέσω της VirtualWire το
πακέτο με το χαρακτήρα «4».
vw_wait_tx();                         //Αναμένει μέχρι να μεταδοθεί ολόκληρο
το πακέτο.
digitalWrite(led_pin, LOW);          //Απενεργοποιεί το led.
val=0;
}
}
[24] , [25] , [27] , [29] , [31]
```

Αφού γράψουμε τον κώδικα στο Arduino IDE το πρόγραμμα είναι έτοιμο για να τρέξει. Πρώτα όμως γίνεται η αποσφαλμάτωση (debug). Η λειτουργία του debug γίνεται μέσα στο λογισμικό Arduino IDE πατώντας το κουμπί της επικύρωσης αριστερά κάτω από το κύριο μενού.



```
#include <VirtualWire.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#include "RTClib.h"
#include <Ethernet.h>
#include <SPI.h>
#include <SD.h>

int pir_state=0;
int pir_statel=0;
char c = 0; // received data
char command[2] = "\0"; // command
int i=0;
boolean LED_status = 0; //LED status, 0 =off
int pinSpeaker = 8;
const int led_pin = 9;
const int ledpin = 5;
const int receive_pin = 6;
int transmit_pin = 7;
const int button1 = 23;
const int button2 = 22;
int val=0;
int val2=0;
int val3=0;
int val4=0;
```

Ολοκλήρωση μεταγλώττισης.

Το σχέδιο χρησιμοποιεί 23844 bytes (9%) του χώρου αποθήκευσης του προγράμματος. Το μέγιστο είναι 253952 bytes.
Οι καθολικές μεταβλητές χρησιμοποιούν 2929 bytes (35%) δυναμικής μνήμης, αφήνοντας 5263 bytes για τοπικές μεταβλητές.

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) στο COM11

Εικόνα 38 Στιγμιότυπο αποσφαλμάτωσης προγράμματος

Αν το πρόγραμμα επικυρώθηκε χωρίς σφάλματα στη συνέχεια πατάμε την επιλογή Ανέβασμα. Σε αυτό το σημείο μέσω της σειριακής επικοινωνίας USB το πρόγραμμα εγκαθίστανται στη πλακέτα Arduino και ενεργοποιείται η λειτουργία του προγράμματος.

7. ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

7.1. Συμπεράσματα

Όπως είδαμε το σύστημα σχεδιάστηκε και κατασκευάστηκε με γνώμονα τη μέγιστη ασφάλεια του χώρου. Μπορεί να τοποθετηθεί όπου εμείς επιθυμούμε (πχ. Επιχειρήσεις, εταιρείες, αγροτικές εγκαταστάσεις, οικιακές κατοικίες, εξωτερικούς χώρους κ.α.). Οι λειτουργίες του συστήματος είναι πολλές και ο χρήστης μπορεί είτε απομακρυσμένα είτε αν βρίσκεται στο σημείο του συστήματος, να παρακολουθεί αν τείνει να παραβιαστεί ο χώρος. Το πλεονέκτημα του είναι η αμφίδρομη ασύρματη επικοινωνία με τα υποσυστήματα χωρίς τη χρήση περιττών καλωδίων. Επιπλέον τα υποσυστήματα λειτουργούν με πλήρη ενεργειακή αυτονομία εξοικονομώντας πόρους.

7.2. Μελλοντικές Επεκτάσεις

Μετά τη περάτωση της εργασίας προτείνουμε ορισμένες βελτιώσεις. Μια μελλοντική ανάπτυξη του συστήματος θα μπορούσε να είναι η προσθήκη κάρτας microSD ώστε να καταγράφεται σε αρχείο η ημερομηνία, ώρα και ο συγκεκριμένος χώρος (πχ πόρτα) που έγινε η παραβίαση. Το αρχείο αυτό θα μπορεί να φαίνεται και στην web εφαρμογή κατά τη σύνδεση του χρήστη. Επιπλέον θα μπορούσε να γίνει μελλοντική μελέτη για την αύξηση της εμβέλειας της ασύρματης επικοινωνίας. Όσο μεγαλύτερη είναι η εμβέλεια του συστήματος τόσο πιο πολλές επιλογές έχει ο χρήστης για να ασφαλίσει τον χώρο του. Επίσης μια σημαντική επέκταση θα ήταν η μελέτη και κατασκευή μεγαλύτερου αριθμού υποσυστημάτων για την κάλυψη περισσότερων χώρων.

8. ΠΑΡΑΡΤΗΜΑΤΑ

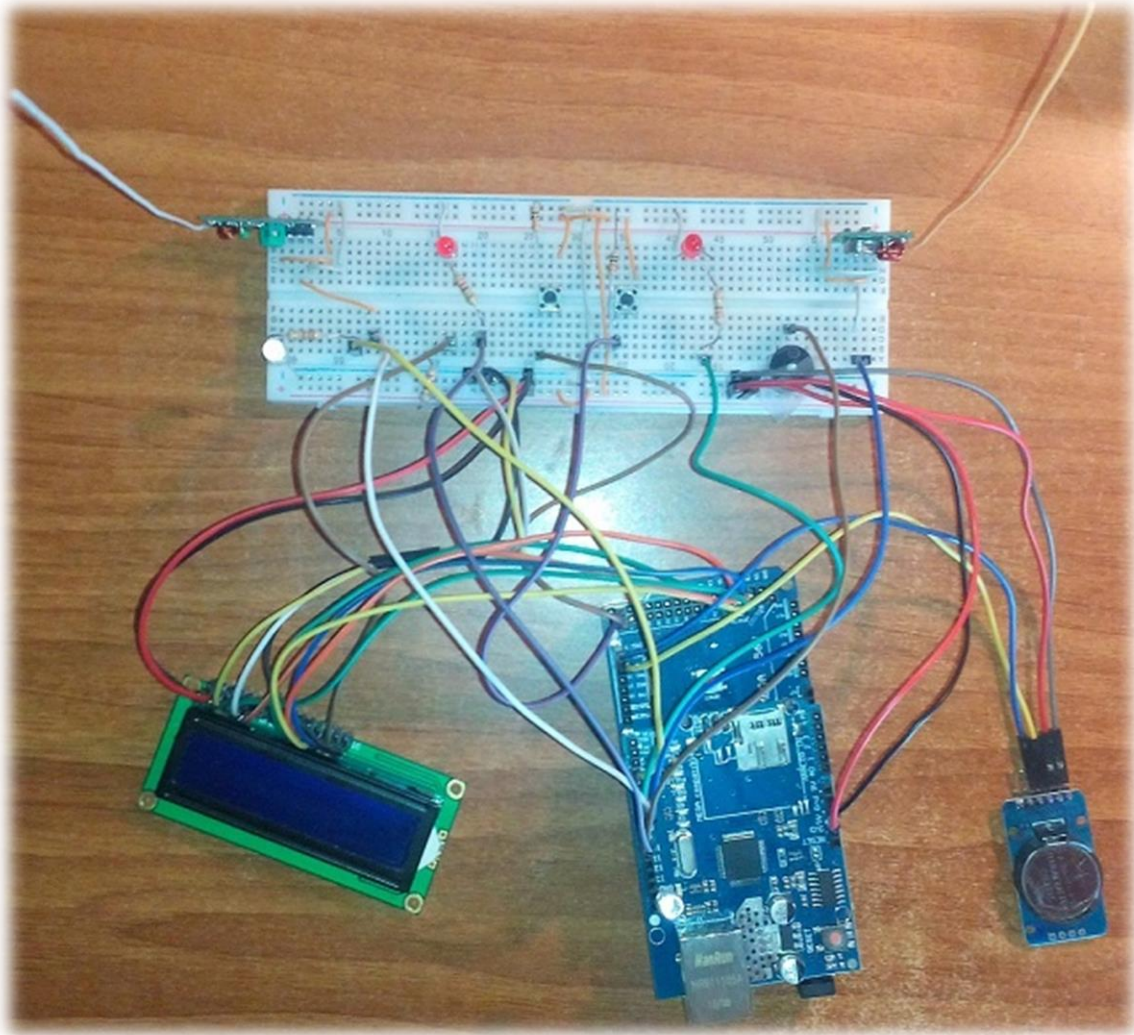
8.1. Παράρτημα κοστολόγησης εξαρτημάτων

Το κόστος αγοράς των εξαρτημάτων υπολογίστηκε με τιμές αγοράς πρώτου εξάμηνου του 2017 και προμηθεύτηκαν από εγχώρια καταστήματα.

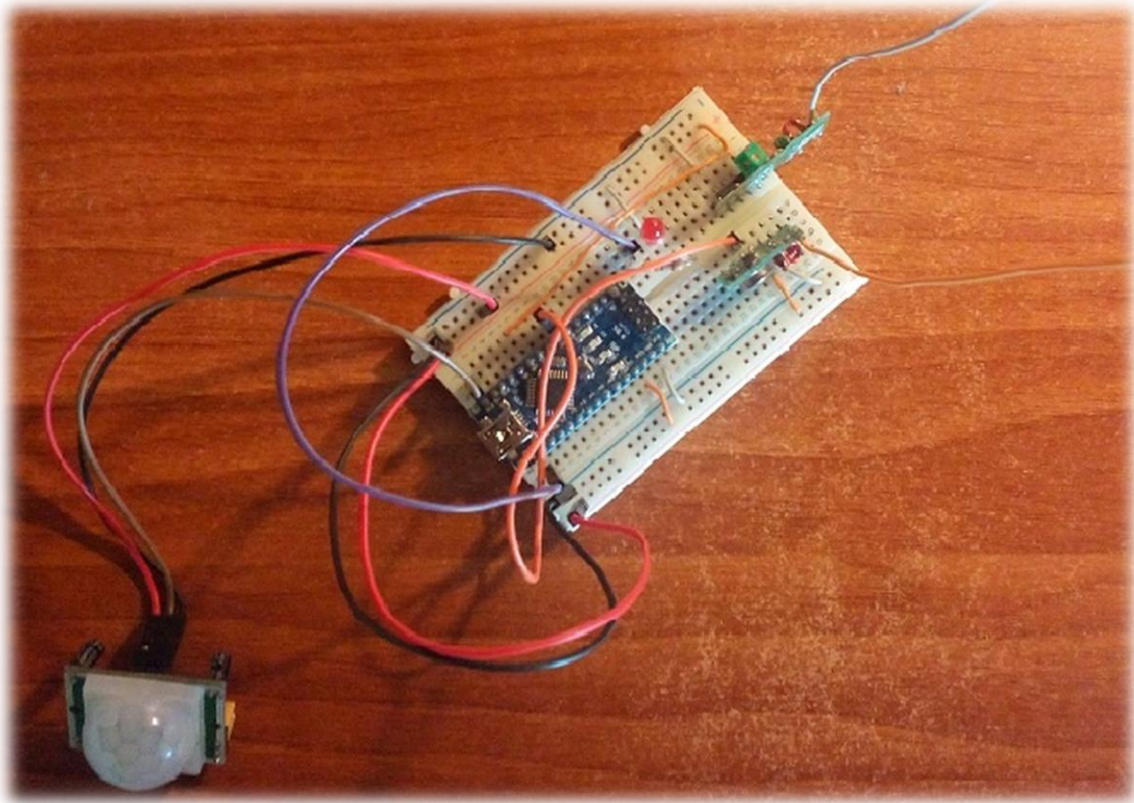
ΕΞΑΡΤΗΜΑΤΑ	ΤΕΜΑΧΙΑ	ΚΟΣΤΟΣ ΤΕΜΑΧΙΟΥ	ΣΥΝΟΛΙΚΟ ΚΟΣΤΟΣ
RF 433 MHz Transmitter & Receiver	3	1,97 €	5,91 €
Arduino mega 2560 R3 + USB CABLE	1	17,89 €	17,89 €
Ethernet shield W5100 Network Board	1	12,30 €	12,30 €
DS3231 AT24C32 IIC RTC Module	1	2,00 €	2,00 €
LCD HD44780 Display Module	1	3,39 €	3,39 €
Solderless MB – 102 Breadboard 830 Tie	1	4,50 €	4,50 €
12 x 12 x 10 mm Momentary Tactile Push Button	2	0,15 €	0,30 €
Active Buzzer	1	0,40 €	0,40 €
40 Male to Female Wires for Arduino (20 cm)	1	3,50 €	3,50 €
40 Male to Male Wires for Arduino (20 cm)	1	3,50 €	3,50 €
Leds Super Bright Red / Blue	10	0,40 €	0,40€
Resistors 220 Ohm / 10 KOhm	10	0,20 €	0,20 €
PIR Motion Detector Module HC-SR501	2	3,50 €	7,00 €
Arduino NANO v3.0 w/ ATmega328P with USB	2	4,98 €	9,96 €
Mini Solderless BreadBoard 400 - Contacts	1	3,00 €	3,00 €
SYB – 170 Mini Prototype Breadboard	2	1,00 €	2,00€
Solar Panel 6V 180mA 1,1W 112 x 84 mm	2	6,80 €	13,60 €
TP4056 Micro usb Charger Module 5V with battery protection	2	1,00 €	2,00 €
LM2577 DC-DC 2~24 V TO 5~28V 2A Step – up Boost Converter with Micro USB	2	1,49 €	2,98€
Single Slot 18650 Battery Case	2	0,79 €	1,58 €
Samsung INR 18650 – 30Q 3000mAh 3.6V	2	5,80 €	11,60 €
Συνολικό κόστος συστήματος: 108,00€			

Παράρτημα Α Συνολικό κόστος εξαρτημάτων όλου του συστήματος

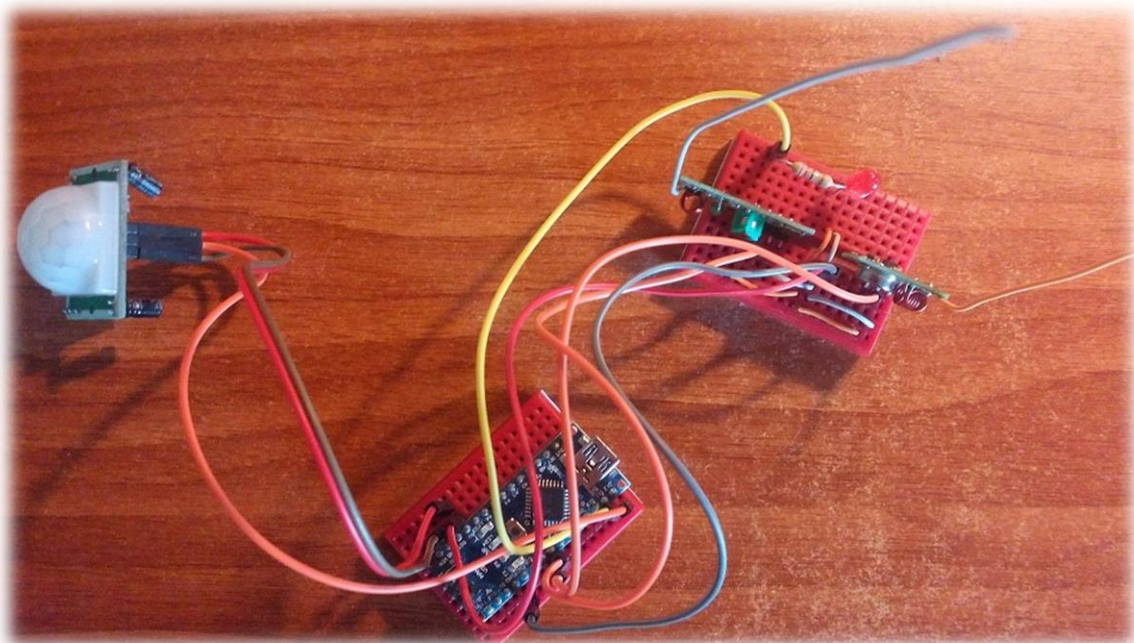
8.2. Παράρτημα φωτογραφιών



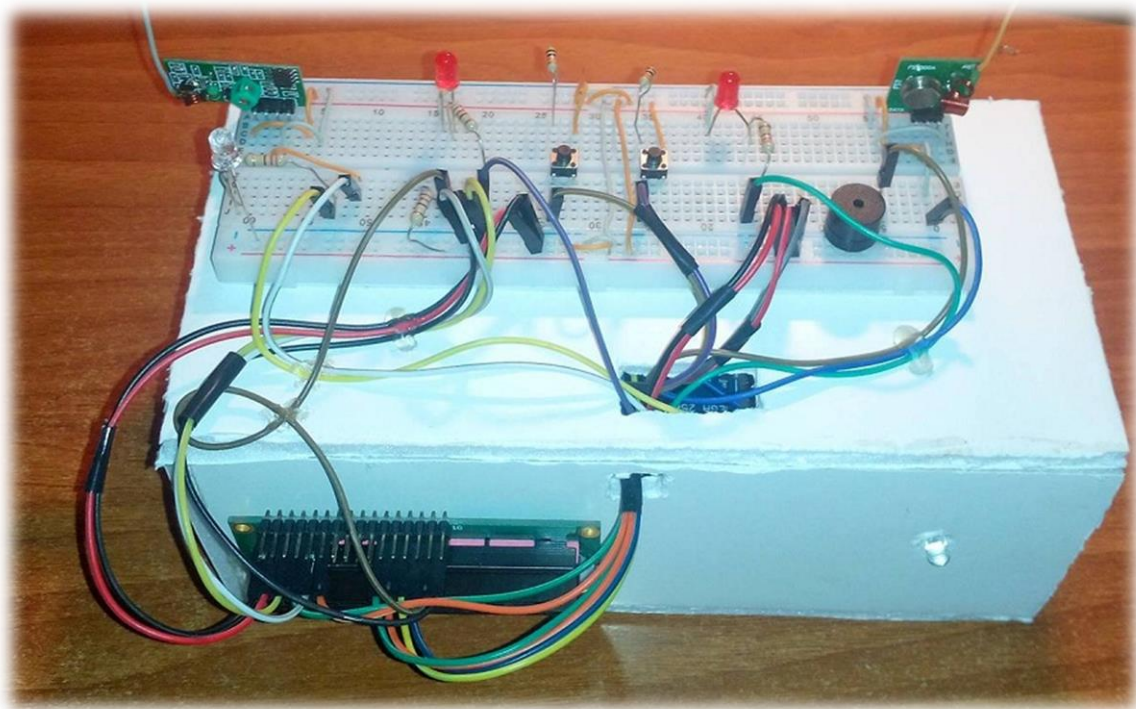
Παράρτημα Β Κάτοψη συστήματος κεντρικού πίνακα ελέγχου



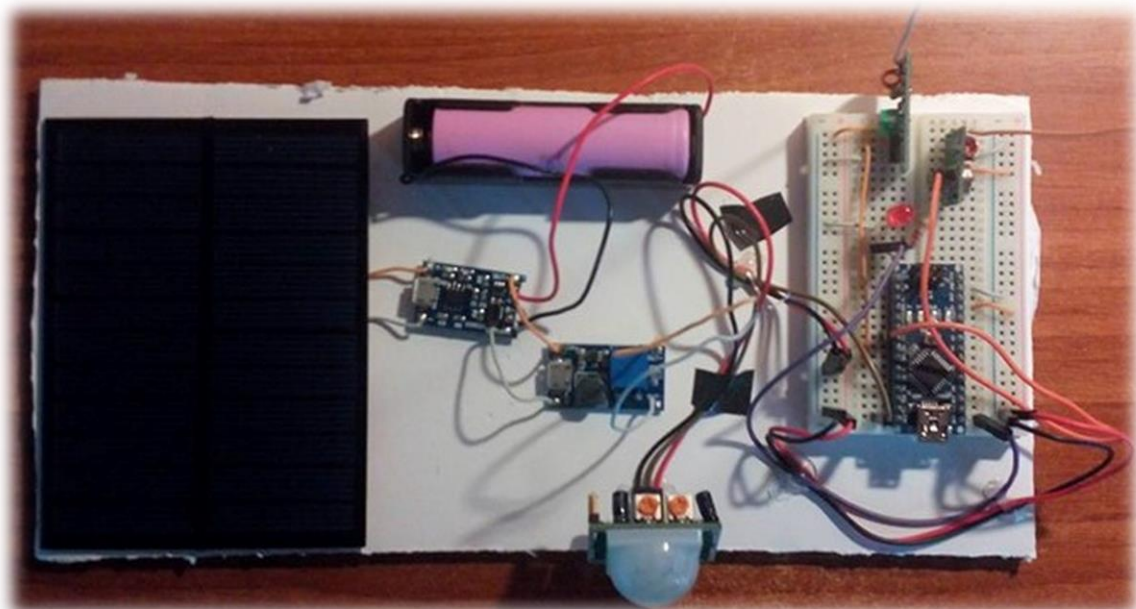
Παράρτημα C Κάτοψη 1^{ου} υποσυστήματος



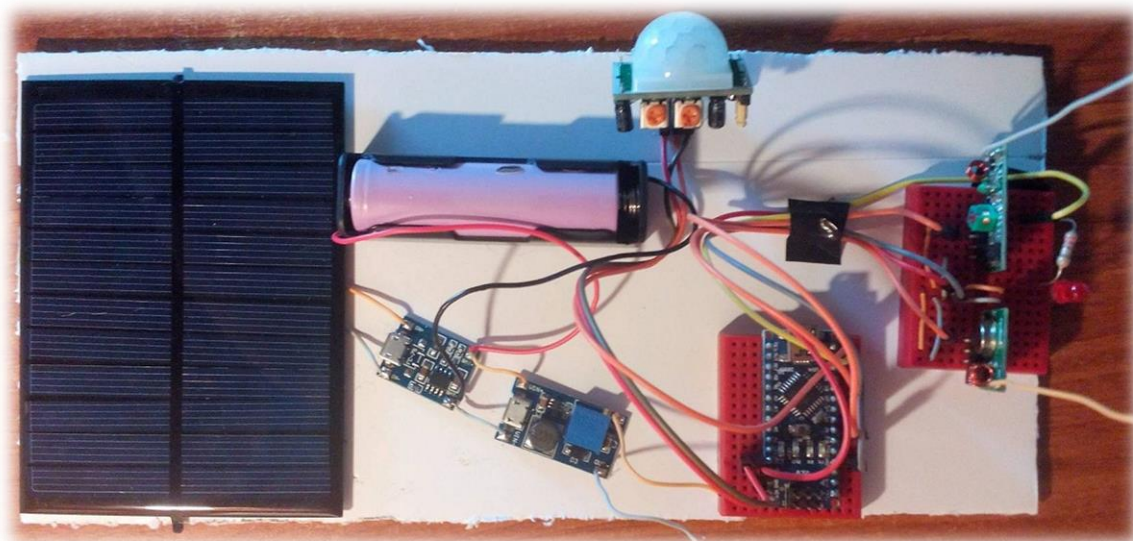
Παράρτημα D Κάτοψη 2^{ου} υποσυστήματος



Παράρτημα Ε Κάτοψη συστήματος κεντρικού πίνακα ελέγχου (τελικό αποτέλεσμα)



Παράρτημα F Κάτοψη 1^{ου} υποσυστήματος (τελικό αποτέλεσμα)



Παράρτημα G Κάτοψη 2^ο υποσυστήματος (τελικό αποτέλεσμα)

9. ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] http://repfiles.kallipos.gr/html_books/1285/bookES.html#embedded_in_daylife
- [2] https://el.wikipedia.org/wiki/%CE%91%CE%BD%CE%AF%CF%87%CE%BD%CE%B5%CF%85%CF%83%CE%B7_%CE%BA%CE%AF%CE%BD%CE%B7%CF%83%CE%B7%CF%82
- [3] Tanenbaum, A. , & Wetherall, D. (2012). *Δίκτυα Υπολογιστών*(5^η εκδ.). Κλειδάριθμος
- [4] <https://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82>
- [5] <http://brain.ee.auth.gr/dokuwiki/doku.php?id=zigbee:zigbee>
- [6] Χατζινικολάκης, Γ. (2016). *Προγραμματισμός με Python στο Raspberry Pi*. Ανακτήθηκε από το διαδικτυακό τόπο: [http://www.sepchou.gr/docs/raspberry/Python%20with%20Pi%20\[Cheat%20sheets%20&%20Activities\].pdf](http://www.sepchou.gr/docs/raspberry/Python%20with%20Pi%20[Cheat%20sheets%20&%20Activities].pdf)
- [7] Παπάζογλου, Π., & Λιώνης, Σ. (2015). *Ανάπτυξη εφαρμογών με το Arduino Ένας πλήρης οδηγός για αρχάριους και προχωρημένους*. Τζιόλα
- [8] <http://learning.grobotronics.com/getting-started/Arduino-uno/>
- [9] <http://brain.ee.auth.gr/dokuwiki/doku.php?id=Arduino:Arduino>
- [10] <https://deltahacker.gr/Arduino-intro/>
- [11] http://henrysbench.capnfatz.com/henrys-bench/Arduino-sensors-and-input/Arduino-hc-sr501-motion-sensor-tutorial/#attachment_wp-att-2133/0/
- [12] <https://www.mpja.com/download/31227sc.pdf>
- [13] http://www.mantech.co.za/Datasheets/Products/433Mhz_RF-TX&RX.pdf
- [14] <https://www.Arduino.cc/en/Main/ArduinoEthernetShieldV1>
- [15] <https://store.Arduino.cc/Arduino-nano>
- [16] <https://en.wikipedia.org/wiki/Buzzer>
- [17] <http://Arduinomodules.info/ky-012-active-buzzer-module/>
- [18] <http://www.haoyuelectronics.com/Attachment/TP4056-modules/TP4056.pdf>
- [19] <https://web.wpi.edu/Pubs/E-project/Available/E-project-010908-124414/unrestricted/DS3231-DS3231S.pdf>
- [20] http://195.130.93.18/pachidis//HLKY/004_HL%CE%9A%CE%A5_LAB_ORGANA_ERGASTHRIOU.pdf
- [21] <http://www.electroniccircuits.gr/antistaseis.html>
- [22] <http://www.electroniccircuits.gr/diodoi.html>
- [23] <https://www.sparkfun.com/datasheets/Components/LED/COM-09590-YSL-R531R3D-D2.pdf>
- [24] <http://www.instructables.com/id/ARDUINO-WIRELESS-HOME-SECURITY-SYSTEM/>

- [25] <https://github.com/adafruit/RTClib/blob/master/examples/ds3231/ds3231.ino>
- [26] <http://howtomechatronics.com/tutorials/Arduino/Arduino-ds3231-real-time-clock-tutorial/>
- [27] <https://ictronic.wordpress.com/2016/07/02/interfacing-16x2-lcd-with-Arduino-without-potentiometer/>
- [28] <http://tutorial.cytron.com.my/2011/07/27/an-Arduino-room-monitoring-web-server/>
- [29] https://github.com/CytronTechnologies/Arduino_Ethernet-Shield/blob/master/Arduino-Ethernet-Room-Monitoring-Source-code/ArduiServer/ArduiServer.ino
- [30] <https://www.smtp2go.com/>
- [31] <https://playground.Arduino.cc/Code/Email>
- [32] <https://www.base64encode.org/>
- [33] <http://www.instructables.com/id/SOLAR-POWERED-ARDUINO-WEATHER-STATION/>
- [34] <http://fritzing.org/home/>
- [35] <https://www.Arduino.cc/en/main/software>
- [36] <https://store.Arduino.cc/Arduino-mega-2560-rev3>
- [37] <https://www.olimex.com/Products/Breadboarding/BB-PWR-3608/resources/MT3608.pdf>