



Τ.Ε.Ι. ΚΡΗΤΗΣ

**Σχολή Εφαρμοσμένων Επιστημών
Τμήμα Ηλεκτρονικών Μηχανικών**

**Υλοποίηση αυτόνομου συστήματος βασισμένου
στην πλατφόρμα Arduino με δυνατότητα
παρακολούθησης και ελέγχου από απόσταση,
μέσω GPRS και ιστοσελίδας.**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

Κοντορινάκη Ευστράτιου

Επιβλέπων: **Δρ. Μηχ. Νικόλαος Σ. Πετράκης**
Καθηγητής Εφαρμογών

**Χανιά,
Οκτώβριος 2015**

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
Σχολή Εφαρμοσμένων Επιστημών
Τμήμα Ηλεκτρονικών Μηχανικών



ΠΑΡΟΥΣΙΑΣΗ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

με θέμα:

Υλοποίηση αυτόνομου συστήματος βασισμένου στην πλατφόρμα Arduino με δυνατότητα παρακολούθησης και ελέγχου από απόσταση, μέσω GPRS και ιστοσελίδας.

Κοντορινάκη Ευστράτιου

Δευτέρα 19 Οκτωβρίου 2015, ώρα 13:00, Αίθουσα 4

Εξεταστική Επιτροπή: Καθ. Εφαρμ. Νικόλαος Πετράκης (επιβλέπων)
Επικ. Καθ. Αντώνιος Κωνσταντάρης
Καθ. Εφαρμ. Ιωάννης Μπαρμπουνάκης

Εν' συντομία

Στόχος της παρούσας πτυχιακής εργασίας είναι η μελέτη και η σχεδίαση ενός συστήματος που θα επιτηρεί και θα ελέγχει διάφορες συσκευές εξ αποστάσεως. Θα λαμβάνει μετρήσεις από αισθητήρες (θερμοκρασίας, στάθμης υγρού κ.α.) παρακολουθώντας σε πραγματικό χρόνο την κατάσταση του συστήματος μέσα από το περιβάλλον μιας ιστοσελίδας που φιλοξενείται σε έναν απομακρυσμένο εξυπηρετητή και μέσω αυτής να δίνει εντολές στα διάφορα υποσυστήματα.

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων.....	i
Περίληψη.....	iii
Abstract.....	iii
1. Εισαγωγή.....	1
1.1 Διάρθρωση της πτυχιακής εργασίας.....	2
2. Γνωριμία με την πλατφόρμα Arduino.....	3
2.1 Η βασική ιδέα της πλατφόρμας Arduino.....	3
2.2 Τα μέρη μίας πλατφόρμας Arduino.....	4
2.3 Επιλογή πλατφόρμας Arduino έναντι άλλων.....	4
2.4 Γνωριμία με το Arduino Mega.....	6
2.4.1 Είσοδοι & Έξοδοι του Arduino Mega.....	6
2.4.3 Χαρακτηριστικά του Arduino Mega.....	8
2.4.4 Βασικές Μνήμες του Arduino Mega.....	9
2.4.5 Τροφοδοσία του Arduino Mega.....	9
2.4.6 Επικοινωνία του Arduino Mega.....	11
2.4.7 Προστασία από υπερένταση στην USB θύρα.....	12
2.5 Ασπίδα Arduino.....	12
2.5.1 Η βασική ιδέα της ασπίδας Arduino.....	12
2.5.2 Γνωριμία με την ασπίδα IComSat.....	14
2.5.3 Χαρακτηριστικά της ασπίδας IComSat.....	14
2.5.4 Τροφοδοσία της ασπίδας IComSat.....	14
2.5.5 Επικοινωνία της ασπίδας IComSat με το Arduino.....	15
2.5.6 LED και Κουμπιά της ασπίδας IComSat.....	15
3. Λογισμικά που χρησιμοποιήθηκαν στην Εργασία.....	17
3.1 Λογισμικό Arduino IDE.....	17
3.1.1 Περιγραφή της εγκατάστασης του Arduino IDE.....	17
3.1.2 Περιβάλλον ανάπτυξης του Arduino.....	18
3.1.3 Δημιουργία σεναρίου (sketch) στο περιβάλλον του Arduino.....	19
3.1.4 Ανέβασμα σεναρίου στο Arduino.....	19
3.1.5 Σειριακή Οθόνη.....	20
3.1.6 Συστατικά και λειτουργίες ενός σεναρίου.....	21
3.2 Λογισμικό Xampp Εξυπηρετητή.....	24
3.2.1 Εγκατάσταση του λογισμικού XAMPP.....	24
3.2.2 Φιλοξενία της σελίδα μας στον Εξυπηρετητή XAMPP.....	27
3.2.3 Δημιουργία Βάσης Δεδομένων στο XAMPP.....	27
3.3 Λογισμικό Notepad++.....	28
3.3.1 Περιβάλλον εργασίας.....	29
3.3.2 Χαρακτηριστικά και λειτουργίες του Notepad++.....	29
3.3.3 Πληροφορίες εγκατάστασης.....	30
4.Υλικό μέρος του συστήματος.....	31
4.1 Μέτρηση θερμοκρασίας με τον αισθητήρα DS18B20.....	31
4.1.1 Περιγραφή λειτουργίας του αισθητήρα DS18B20.....	32
4.1.2 Σύνδεση του αισθητήρα θερμοκρασίας με το Arduino.....	34
4.1.3 Κώδικας μέτρησης θερμοκρασίας.....	35
4.2 Μέτρηση στάθμης υγρού με τον αισθητήρα υπερήχων HC-SR04.....	36
4.2.1 Η μονάδα υπερήχων HCSR-04.....	38
4.2.2 Σύνδεση της μονάδας υπερήχων με το Arduino.....	39
4.2.3 Κώδικας μέτρησης στάθμης.....	40
4.3.1 Μονάδα ελέγχου τεσσάρων ηλεκτρονόμων με οδήγηση τάσης 5V.....	42
4.3.2 Σύνδεση μονάδας ηλεκτρονόμων με το Arduino.....	43
4.3.4 Κώδικας οδήγησης ηλεκτρονόμου.....	44
4.4 Επικοινωνία με την ασπίδα IComSat.....	45
4.4.1 Χρήση των AT εντολών.....	46
4.4.2 Λειτουργία της ασπίδας IComSat ως πελάτης (client).....	46

2.4.2 Λειτουργία του SIM900 ως εξυπηρετητή (Server).....	47
4.5 Ενημέρωση παρούσας κατάστασης με διόδους φωτός (LED)	48
4.5.1 Σύνδεση του LED με το Arduino.....	49
4.5.2 Κώδικας οδήγησης φωτοδιόδου(LED).....	49
4.6 Διάφορα εξαρτήματα που χρειάστηκαν για την κατασκευή του κουτιού.	50
5 Λογισμικό μέρος του συστήματος.	51
5.1 Mysql βάση δεδομένων.	51
5.1.1 Οι πίνακες της βάσης δεδομένων που χρησιμοποιήθηκαν στην εργασία μας.....	52
5.1.2 Σύνδεση με την βάση δεδομένων.....	53
5.2 Ασφάλεια Ιστοσελίδας.....	54
5.3 Περιβάλλον ιστοσελίδας του πελάτη.	55
5.3.1 Φόρμες επικοινωνίας με τον χρήστη.....	56
5.3.2 Ενημέρωση καταστάσεων της συσκευής.....	57
5.3.3 Μέτρηση των αισθητήρων στάθμης και θερμοκρασίας.	57
5.4 Επικοινωνία Πελάτη με Εξυπηρετητή.....	58
5.4.1 Έλεγχος εξόδων του Arduino μέσα από την ιστοσελίδα του πελάτη.....	59
5.4.2 Ενημερώσεις των καταστάσεων του Arduino μέσα από την ιστοσελίδα του πελάτη.....	61
5.5 Οι λειτουργίες του Arduino	63
5.5.1 Σύνδεση του Arduino με το δίκτυο GSM/GPRS.....	64
5.5.2 Λειτουργία της ασπίδας σαν εξυπηρετητής πολλαπλών συνδέσεων.	66
5.5.3 Αποστολή δεδομένων στον εξυπηρετητή με POST αιτήματα.....	67
5.5.4 Φιλτράρισμα μηνύματος εξυπηρετητή.....	68
6. Οδηγίες χρήσης της συσκευής.....	70
6.1 Περιγραφή συσκευής.....	70
6.2 Χαρακτηριστικά συσκευής.....	71
6.3 Τα υλικά μέρη της συσκευής.....	71
6.3.1 Κεραία.....	71
6.3.2 Βύσμα τροφοδοσίας.....	72
6.3.3 Φωτεινές ενδείξεις LED.....	72
6.3.4 Είσοδοι της συσκευής.....	73
6.3.5 Είσοδοι Αισθητήρων.	73
6.3.6 Τροφοδοσία άλλων συσκευών ή αισθητήρων.	73
6.3.7 Έξοδοι της συσκευής.....	75
6.4 Οδηγός έναρξης.....	75
6.5 Σύνδεση με την ιστοσελίδα για τον έλεγχο και επιτήρηση της συσκευής από απόσταση.	76
6.5.1 Απομακρυσμένος έλεγχος των εξόδων.	78
6.5.2 Ρυθμίσεις παραμέτρων Στάθμης.....	79
6.5.3 Ρυθμίσεις παραμέτρων Θερμοκρασίας.....	80
6.5.4 Ενημέρωση της κατάστασης της συσκευής.....	80
6.5.5 Μέτρηση Στάθμης Δεξαμενής.....	81
6.5.6 Μέτρηση Θερμοκρασίας Περιβάλλοντος.....	81
6.5.7 Έξοδος από την ιστοσελίδα του χρήστη.....	82
7. Συμπεράσματα και μελλοντικές επεκτάσεις	83
7.1 Προβλήματα που επιλύθηκαν κατά την υλοποίηση της πτυχιακής εργασίας.....	83
7.2 Μελλοντικές επεκτάσεις.....	84
Βιβλιογραφία.....	86

Περίληψη

Σκοπός αυτής της πτυχιακής εργασίας είναι η σχεδίαση και υλοποίηση ενός συστήματος το οποίο θα επιτηρεί και θα ελέγχει διάφορες συσκευές ισχύος μέσα από το περιβάλλον μίας ιστοσελίδας.

Για την επίτευξη του στόχου αυτού χωρίστηκε σε δύο υποσυστήματα στο υλικό και στο λογισμικό.

Το υλικό υποσύστημα βασίζεται στην υπολογιστική πλατφόρμα Arduino Mega και σε άλλα εξαρτήματα με τα οποία αλληλεπιδρά για τον έλεγχο συσκευών ισχύος μέσω ηλεκτρονόμων (ρελέ), την λήψη μετρήσεων στάθμης και θερμοκρασίας μέσω αισθητήρων και την διαχείριση καταστάσεων μέσω διακοπικών εισόδων. Επιπλέον συνδέεται στο Διαδίκτυο (Internet) χρησιμοποιώντας την ασπίδα GSM/GPRS μέσω ασύρματου δικτύου.

Παράλληλα ο χρήστης μπορεί μέσα από το περιβάλλον μιας ιστοσελίδας να διαβάσει την κατάσταση του συστήματος από μακριά, να λάβει τις μετρήσεις ακόμα και να αλλάξει τις καταστάσεις των συσκευών που επιτηρεί.

Το λογισμικό υποσύστημα βασίζεται σε ένα απομακρυσμένο εξυπηρετητή που φιλοξενεί την ιστοσελίδα του χρήστη και όλες τις τεχνολογίες διαδικτύου που κάνουν εφικτή την επικοινωνία με το Arduino.

Abstract

The aim of this final work is to design and implement a system that will monitor and control various power devices through the environment of a website. To achieve this goal, the system was divided in two subsystems the hardware and the software. The hardware subsystem is based on Arduino Mega computing platform and other components with which it interacts to control power devices via relay, in order to receive level and temperature measurements via sensors and management situations through switching inputs. Furthermore it is connected to the Internet using the GSM / GPRS shield via wireless network. At the same time, the user can read the system status remotely, through the environment of a web page, take measurements and even change the status of the supervised devices. Additionally, the user can through the environment of a web page to read the system status remotely, take measurements and even changing situations of the devices it controls. The software subsystem is based on a remote server that hosts the user's website and all the web technologies that make it possible to communicate with the Arduino.

1. Εισαγωγή

Η υπολογιστική πλατφόρμα Arduino τα τελευταία χρόνια καλύπτει όλο και περισσότερες ανάγκες σε διάφορους τομείς του αυτοματισμού, και γενικά γνωρίζει πλήθος εφαρμογών. Το Arduino είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα WIRING (ουσιαστικά πρόκειται για τη C++ με κάποιες μετατροπές). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε PROCESSING, MAX/MSP, PURE DATA, SUPER COLLIDER. [1]

Το Arduino χρησιμοποιείται σαν ένας αυτόματος, έξυπνος ελεγκτής, ο οποίος με την χρήση διάφορων αισθητήρων, παρακολουθεί το περιβάλλον και σύμφωνα με τον εκάστοτε προγραμματισμό του αντιδρά στις μεταβολές του. Παρουσιάζονται όμως συχνά περιπτώσεις που δεν είναι δυνατή η παρουσία του ανθρώπινου παράγοντα για τον άμεσο έλεγχο και επιτήρηση της συσκευής. Προκύπτει έτσι η ανάγκη τηλεπικοινωνίας και απομακρυσμένου ελέγχου του Arduino. Παράλληλα, το Διαδίκτυο (Internet) ως μέσο τηλεπικοινωνίας είναι πολύ βολικό, γιατί καλύπτει αποστάσεις μεγάλης κλίμακας και πλέον έχει εισχωρήσει στην καθημερινότητα των ανθρώπων. Στις μέρες μας το κινητό τηλέφωνο, ο ηλεκτρονικός υπολογιστής και παρόμοιες συσκευές τηλεπικοινωνίας που κυκλοφορούν στην αγορά, αποτελούν συσκευές που ο άνθρωπος έχει σχεδόν πάντα μαζί του και επιτρέπουν την πρόσβαση και την περιήγηση στο Διαδίκτυο. Αυτές οι συσκευές μπορούν να αποτελέσουν το ρόλο του “τηλεχειριστηρίου” για το Arduino και τον απομακρυσμένο έλεγχο. Η επίτευξη απομακρυσμένου ελέγχου ενός αυτοματισμού με την χρήση του Arduino, καλύπτει και βρίσκει εφαρμογή σε συστήματα όπως

- Μετεωρολογικούς σταθμούς
- Οικιακούς αυτοματισμούς “Έξυπνα σπίτια”
- Σε αντλιοστάσια
- Συστήματα φωτοβολταϊκών
- Συστήματα παροχής ενέργειας με Ανεμογεννήτριες
- Ιατρικά κέντρα
- Συστήματα ασφαλείας
- Συστήματα αυτοκινήτων

Ο αριθμός συστημάτων που μπορεί να καλύψει απομακρυσμένος έλεγχος Arduino είναι πολύ μεγάλος. Όλα τα παραπάνω συστήματα βασίζονται στην ίδια λογική, την εποπτεία και τον έλεγχο του Arduino από μακριά χρησιμοποιώντας ως μέσο τηλεπικοινωνίας το Διαδίκτυο. Για τον λόγο αυτό σκοπός της πτυχιακής εργασίας είναι η κατασκευή ενός συστήματος “γενικής χρήσης” το οποίο θα επιτρέπει τον

απομακρυσμένο έλεγχο και την εποπτεία του Arduino μέσα από το περιβάλλον μιας ιστοσελίδας. Για να το επιτύχουμε αυτό η σχεδίαση του συστήματος ελέγχου θα γίνει στην πιο απλή μορφή χρήσης του Arduino. Το Arduino θα δέχεται τιμές μέσω των ακροδεκτών του από αισθητήρες για την μέτρηση στάθμης και θερμοκρασίας και θα δίνει τιμές από τους ακροδέκτες του με σκοπό τον έλεγχο και την επιτήρηση άλλων συσκευών.

1.1 Διάρθρωση της πτυχιακής εργασίας

Στο πρώτο κεφάλαιο της εργασίας γίνεται μια σύντομη περιγραφή της πλατφόρμας Arduino και την ενσωμάτωσή της στον απομακρυσμένο έλεγχο χρησιμοποιώντας ως μέσο επικοινωνίας το Διαδίκτυο.

Στο δεύτερο κεφάλαιο γίνεται μια εισαγωγή στην πλατφόρμα Arduino και στα διάφορα μοντέλα που υπάρχουν μέχρι σήμερα. Στην συνέχεια περιγράφεται η πλακέτα Arduino Mega και αναλύονται τα τεχνικά χαρακτηριστικά, οι λειτουργίες και τα βασικά μέρη που την αποτελούν. Τέλος γίνεται αναφορά στην ασπίδα GSM, στα μέρη που την αποτελούν και στον τρόπο σύνδεσης με το Arduino.

Στο τρίτο κεφάλαιο γίνεται αναφορά στα λογισμικά προγράμματα που χρησιμοποιήθηκαν για την υλοποίηση της εργασίας. Αναλύεται το περιβάλλον ανάπτυξης της κάθε πλατφόρμας καθώς και βασικές λειτουργίες που προσφέρουν.

Στο τέταρτο κεφάλαιο γίνεται αναφορά στο υλικό μέρος του συστήματος. Δίνονται τα τεχνικά χαρακτηριστικά και η λειτουργία κάθε εξαρτήματος καθώς και ο τρόπος σύνδεσης τους στο Arduino. Επίσης δίνονται και παραδείγματα του κώδικα για την εκάστοτε λειτουργία του εξαρτήματος με τις απαραίτητες επεξηγήσεις.

Στο πέμπτο κεφάλαιο παρουσιάζεται το λογισμικό μέρος του συστήματος. Το λογισμικό μέρος αποτελείται από τέσσερα υποσυστήματα τα οποία επικοινωνούν μεταξύ τους για την υλοποίηση του έργου. Συγκεκριμένα αναλύονται ο σχεδιασμός της βάσης δεδομένων, η κατασκευή της ιστοσελίδας του χρήστη, η ανάπτυξη αρχείων του εξυπηρετητή που επικοινωνούν με το Arduino και τέλος ο προγραμματισμός και οι λειτουργίες του Arduino σε σύνδεση με την ασπίδα GSM για την επικοινωνία με τον εξυπηρετητή και την σύνδεση στο Διαδίκτυο.

Στο έκτο κεφάλαιο δημιουργείται ένας συνοπτικός οδηγός χρήσης της συσκευής. Γίνεται περιγραφή των λειτουργιών που διαθέτει, αναλύονται τα τεχνικά χαρακτηριστικά και τα μέρη της συσκευής. Στην συνέχεια δίνονται οδηγίες για την σωστή λειτουργία και χρήση της συσκευής καθώς και της ιστοσελίδας του χρήστη.

Στο έβδομο και τελευταίο κεφάλαιο συνοψίζονται τα συμπεράσματα, τα προβλήματα που πρόεκυψαν κατά την υλοποίηση της εργασίας και παρατίθενται μελλοντικές επεκτάσεις που μπορούν να προστεθούν για την βελτίωση του συστήματος.

2. Γνωριμία με την πλατφόρμα Arduino.

Στο κεφάλαιο που ακολουθεί αναλύουμε την πλατφόρμα Arduino, αναφέρουμε βασικές λειτουργίες της και κάνουμε αναφορά στα διάφορα μοντέλα που κυκλοφορούν μέχρι σήμερα στο εμπόριο. Επικεντρωνόμαστε στην πλακέτα Arduino Mega η οποία αποτελεί την βάση του συστήματος. Επιπλέον αναφέρουμε και την χρήση ασπίδων που βασικό στόχος τους είναι να επεκτείνουν τις δυνατότητες του Arduino όπως και η ασπίδα GSM με στόχο την επικοινωνία του Arduino με το Διαδίκτυο χρησιμοποιώντας την τεχνολογία GSM.

2.1 Η βασική ιδέα της πλατφόρμας Arduino

Το Arduino είναι μια υπολογιστική πλατφόρμα, ένα εργαλείο που μας βοηθάει να κατασκευάσουμε ένα υπολογιστικό σύστημα, έχει την δυνατότητα να ελέγχει συσκευές του φυσικού κόσμου, σε αντίθεση με έναν απλό ηλεκτρονικό Υπολογιστή. Βασίζεται σε μια αναπτυξιακή πλακέτα που ενσωματώνει επάνω έναν μικροελεγκτή. Ο Μικροελεγκτής προγραμματίζεται μέσα σε ένα περιβάλλον ανάπτυξης συνδεδεμένος με Η/Υ. Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων, να δεχτεί εισόδους από αισθητήρια, όργανα και διακόπτες, αλλά και να αντιδράσει ανάλογα με το πώς έχει προγραμματιστεί ελέγχοντας φώτα, κινητήρες και άλλες συσκευές εξόδου του φυσικού κόσμου. Γενικότερα, οι εργασίες στον Μικροελεγκτή μπορούν να είναι αυτόνομες σε επίπεδο υλικού (hardware) ή να επικοινωνούν με κάποιο πρόγραμμα στον Η/Υ του προγραμματιστή (προγράμματα όπως τα Flash, Processing, MaxMSP) σε επίπεδο λογισμικού (software).

Το Arduino βασίζεται σε τεχνολογίες ανοιχτού κώδικα δίνοντας έτσι την δυνατότητα στον καθένα να κατεβάσει δωρεάν το λογισμικό ή ακόμα και να κατασκευάσει την ίδια την πλατφόρμα, αφού το κύκλωμα της πλακέτας διατίθεται με την άδεια χρήσης CREATIVE COMMONS. Τα σχέδια κατασκευής της πλατφόρμας δίνονται ελεύθερα στην επίσημη σελίδα του [Arduino: www.arduino.cc](http://www.arduino.cc). Το περιβάλλον ανάπτυξης του λογισμικού βασίζεται στην γλώσσα προγραμματισμού Processing και την γλώσσα προγραμματισμού Wiring.



Εικόνα 2.1 : Λογότυπο πλατφόρμας Arduino.

2.2 Τα μέρη μίας πλατφόρμας Arduino

Μία πλατφόρμα Arduino αποτελείται από ένα μικροελεγκτή ATMEL AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) καθώς και συμπληρωματικά εξαρτήματα με στόχο ο χρήστης να μπορεί να ενσωματώσει και να προγραμματίσει την πλατφόρμα με ευκολία σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz. Ο μικροελεγκτής εξαρχής είναι προγραμματισμένος με έναν φορτωτή εκκίνησης (*bootloader*) και επομένως δεν χρειάζεται εξωτερικός προγραμματιστής [2]

Όλες οι πλατφόρμες προγραμματίζονται με μία RS-232_σειριακή σύνδεση, αλλά ο τρόπος που επιτυγχάνεται σε κάθε πλατφόρμα είναι διαφορετικός. Τα σημερινά Arduino έχουν συνήθως ενσωματωμένο έναν τσιπ μετατροπέα "USB-to-Serial" όπως το "FDTI FT232" και έτσι μπορούν να προγραμματίζονται μέσω USB σύνδεσης. Κάποιες παραλλαγές, όπως το "Arduino Mini" και το ανεπίσημο "Boarduino" χρησιμοποιούν τσιπ προσαρμογής "USB-to-Serial" σε μορφή πλακέτας ή καλωδίου. Η πλακέτα του Arduino, όπως προαναφέραμε, μπορεί να συνδεθεί και με άλλα κυκλώματα καθώς διαθέτει ως συμπληρωματικά εξαρτήματα εκτεθειμένους τους ακροδέκτες εισόδου/εξόδου. Το Arduino Mega, για παράδειγμα, διαθέτει 54 ψηφιακούς ακροδέκτες εισόδου/εξόδου, από τους οποίους οι 15 μπορούν και παράγουν σήματα PWM, και 16 αναλογικές εισόδους. Αυτοί οι ακροδέκτες είναι διαθέσιμοι στην κορυφή της πλακέτας μέσω θηλυκών συνδέσεων μεγέθους 0,1 ιντσών.

2.3 Επιλογή πλατφόρμας Arduino έναντι άλλων

Στο εμπόριο διατίθεται πλήθος μικροελεγκτών και αναπτυξιακών πλακετών για παραγωγή διάφορων εφαρμογών, που για την χρήση τους δεν απαιτούνται υψηλές γνώσεις, καθώς κρύβουν τις δύσκολες λεπτομέρειες της αρχιτεκτονικής και επιτρέπουν τον άμεσο προγραμματισμό του μικροελεγκτή ακόμα και όταν εκτελεί ήδη ένα πρόγραμμα. Μια τέτοια πλατφόρμα είναι και το Arduino με βασικά πλεονεκτήματα τα οποία παρατίθενται παρακάτω:

- **Χαμηλό κόστος:** Συγκριτικά με άλλες πλατφόρμες η πλατφόρμα Arduino είναι αρκετά πιο οικονομική. Μάλιστα σε περίπτωση που βρεθούν δωρεάν τα σχηματικά (schematics) διαγράμματα μπορεί να κατασκευαστεί εξαρχής από τον χρήστη/ενδιαφερόμενο.
- **Εύχρηστο προγραμματιστικό περιβάλλον:** Το περιβάλλον προγραμματισμού του Arduino ενδείκνυται για αρχάριους, αλλά είναι ταυτόχρονα και ευέλικτο και για πιο προχωρημένους χρήστες.
- **Επεκτάσιμη σε περισσότερα από ένα Λειτουργικά Συστήματα:** Σε σχέση με τις υπάρχουσες πλατφόρμες του εμπορίου η πλατφόρμα Arduino δεν περιορίζεται μόνο στα Λειτουργικά Windows αλλά επεκτείνεται στα "Machinstoh OSX" ακόμα και στα "Linux"

- **Ανοιχτού λογισμικού:** Το λογισμικό του Arduino διανέμεται με την μορφή εργαλείων ανοιχτού κώδικα και είναι διαθέσιμα προς επέκταση.
- **Ανοιχτού Υλικού:** Οι μικροελεγκτές στους οποίους βασίζεται το Arduino είναι ο ATMEGA 328P της Atmel. Όπως προαναφέραμε, το Arduino βασίζεται σε τεχνολογίες ανοιχτού υλικού άρα ο οποιοσδήποτε μπορεί ακόμα και να κατασκευάσει μία πλατφόρμα Arduino. Η άδεια των σχηματικών για τα αναπτυξιακά δίδεται από το Creative Commons και δίνει την δυνατότητα σε έμπειρους και μη σχεδιαστές, βασισμένοι στο ήδη υπάρχον αναπτυξιακό, να κατασκευάσουν το δικό τους χωρίς να έρθουν αντιμέτωποι με νομικά προβλήματα.

Η πλατφόρμα Arduino αναπτύσσεται ολοένα και περισσότερο με το πέρασμα του χρόνου κερδίζοντας συνεχώς νέους υποστηρικτές. Το Arduino έχει γνωρίσει αρκετές εκδόσεις που ενώ διαφέρουν σε χαρακτηριστικά βασίζονται στην ίδια λογική.

Παρακάτω βλέπουμε τα επίσημα μοντέλα Arduino που έχουν κατασκευαστεί μέχρι σήμερα.



Arduino Uno



Arduino Leonardo



Arduino Due



Arduino Yún



Arduino Micro



Arduino Esplora



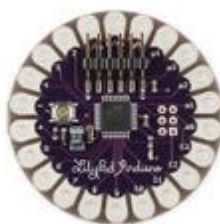
Arduino Mega ADK



Arduino Ethernet



Arduino Mega 2560

**Arduino Robot****Arduino Mini****Arduino Nano****LilyPad Arduino Simple****LilyPad Arduino****Arduino Pro Mini****Arduino Pro****LilyPad Arduino USB****Arduino Fio**

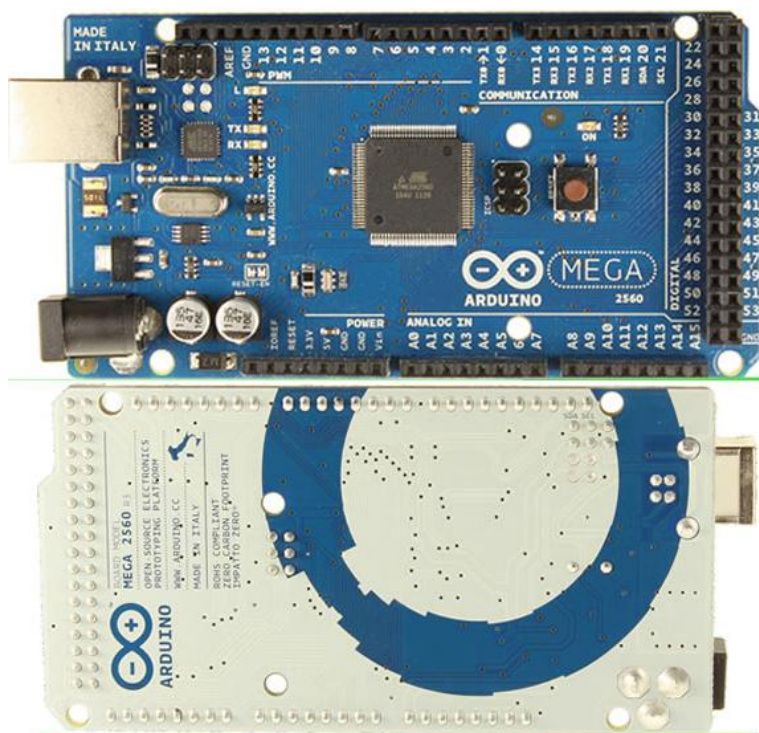
2.4 Γνωριμία με το Arduino Mega

Το Arduino Mega2560, όπως κάθε πλατφόρμα Arduino, σαν κύρια βάση έχει τον μικροελεγκτή ATmega2560. Πρόκειται ουσιαστικά για την εξέλιξη του προηγούμενου Arduino Mega που βασιζόταν στον μικροελεγκτή ATmega128 και πλέον τον έχει αντικαταστήσει με τον ATmega2560. Είναι προγραμματισμένος ώστε να ελέγχει τους 54 ψηφιακούς ακροδέκτες εισόδου ή εξόδου εκ των οποίων οι 15 μπορούν να χρησιμοποιηθούν και ως έξοδοι παλμικού σήματος PWM, 16 αναλογικές εισόδους, 4 σειριακές θύρες ("UARTS"), έναν κρύσταλλο ταλαντωτή στα 16MHz, μια σύνδεση USB ένα βύσμα τροφοδοσίας και ένα κουμπί επανεκκίνησης (Reset). Μας παρέχει όλα όσα χρειάζονται για την δημιουργία διαδραστικών εφαρμογών απλά συνδέοντάς το με ένα καλώδιο USB ή με εξωτερική τροφοδοσία.

2.4.1 Είσοδοι & Έξοδοι του Arduino Mega

Το Arduino Mega είναι μια επέκταση του Arduino Uno αυξάνοντας το σύνολο των ακροδεκτών (έχει 54 ψηφιακούς ακροδέκτες, ενώ το Arduino UNO 16). Αυτοί οι ακροδέκτες χρησιμοποιούν εντολές-συναρτήσεις και μπορούν να λειτουργήσουν ως

Είσοδοι ή Έξοδοι. Συγκεκριμένα με την συνάρτηση “pinMode()” ορίζουμε έναν ακροδέκτη σαν Είσοδο ή Έξοδο ενώ με τις συναρτήσεις “digitalRead()” και “digitalWrite()” μπορούν να διαβάσουν ή να γράψουν μια τιμή στον ακροδέκτη. Η τάση των ακροδεκτών που έχουν οριστεί σαν έξοδοι είναι 5V και μπορούν να αντέξουν μέχρι και 40mA ένταση ρεύματος. Παρόλο που η τιμή είναι χαμηλή, είναι αρκετή για να οδηγήσουμε ένα LED το οποίο έχει κατανάλωση 20mA. Για τους ακροδέκτες που ορίζονται σαν είσοδοι υπάρχει στο εσωτερικό του μικροελεγκτή μία αντίσταση “pull-up” στα 20-50KΩ έτσι προστατεύουμε τον ακροδέκτη αφήνοντας να περάσει ένα μικρό ρεύμα της τάξης των μΑ. Ακόμα έχει 16 αναλογικούς ακροδέκτες εισόδου, ακρίβειας 10 δυαδικών ψηφίων (10 μπιτ, δηλαδή 1.024 διαφορετικές τιμές), που μπορούν να διαβάσουν αναλογικές τιμές όπως μία τάση μπαταρίας, ένα ποτενσιόμετρο κ.α, επιστρέφοντας μία τιμή από το 0-1023 που αντιστοιχεί σε τάση 0V-5V αντίστοιχα. Από προεπιλογή μετρούν από το έδαφος έως 5 VOLT, αν και είναι δυνατόν να αλλάξει το άνω άκρο του εύρους τους, χρησιμοποιώντας τον ακροδέκτη AREF και με την βοήθεια της “analogReference ()” συνάρτησης.



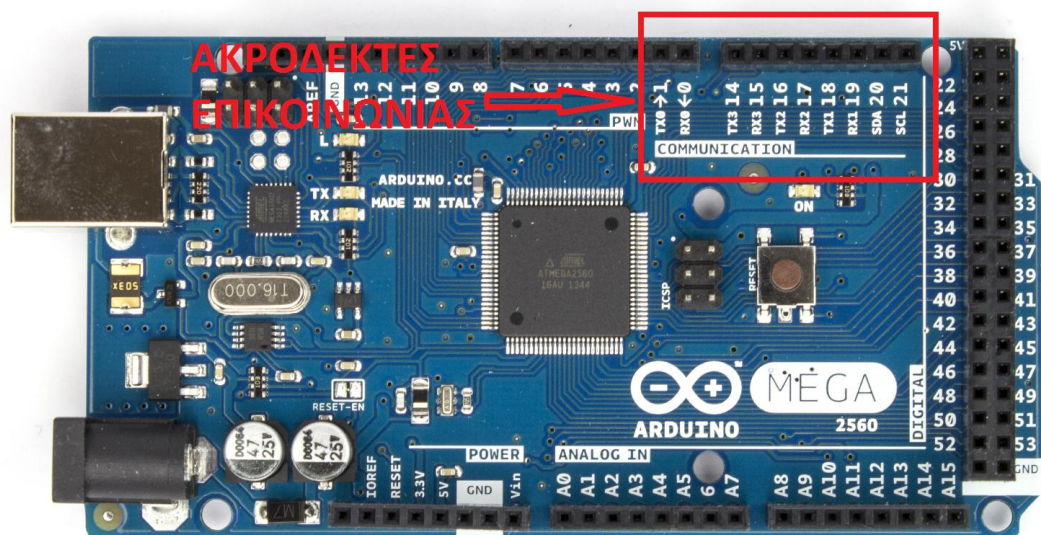
Εικόνα 2.2 : Δυο όψεις του Arduino Mega2560

Εκτός από τους ακροδέκτες που αναφέραμε υπάρχουν και άλλοι με συγκεκριμένες λειτουργίες όπως :

- **Σειριακή Επικοινωνία:** Σειριακή0: 0 (RX) και 1 (TX), Σειριακή 1: 19 (RX) και 18 (TX), Σειριακή 2: 17 (RX) και 16 (TX), Σειριακή 3: 15 (RX) και 14 (TX) .Ο ακροδέκτης με την σήμανση (RX) χρησιμοποιείται για τη λήψη ενώ με (TX) για την μετάδοση σειριακών δεδομένων TTL. Οι ακροδέκτες 0 και 1 συνδέονται

απευθείας με τους ακροδέκτες του σειριακού μετατροπέα (USB-to-TTL) ATmega16U2 για την επικοινωνία μέσω θύρας USB.

- **Εξωτερικές διακοπές (External Interrupts).** Οι ακροδέκτες 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3) και 21 (interrupt 2), μπορούν να προγραμματιστούν με την συνάρτηση `attachInterrupt()`, έτσι ώστε να διακόψουν την ροή του προγράμματος και να εκτελέσουν ένα άλλο κατά την οποιαδήποτε αλλαγή της τιμής, στην άνοδο ή την κάθοδο ακόμα και κατά το πέρασμα από το ένα στάδιο στο άλλο.
- **PWM:** Οι ακροδέκτες 2-13 και 44-46 μπορούν να παρέχουν Έξοδο υπό τη μορφή παλμών ακρίβειας 8-bit PWM με την συνάρτηση `analogWrite()`.
- **SPI:** Οι ακροδέκτες 51 (MOSI), 50 (MISO), 52 (SCK), 53 (SS) επιτρέπουν επικοινωνία SPI χρησιμοποιώντας την SPI βιβλιοθήκη.
- **TWI:** Για τους ακροδέκτες 20 (SDA) και 21 (SCL) είναι εφικτή η επικοινωνία μέσω TWI με τη χρήση της βιβλιοθήκης Wire.
- **LED:** Στον ακροδέκτη 13 υπάρχει ενσωματωμένο ένα LED. Όταν ο ακροδέκτης πάρει υψηλή τιμή (HIGH), το LED ανάβει ενώ όταν ο ακροδέκτης πάρει χαμηλή τιμή (LOW) το LED σβήνει.



Εικόνα 2.3 : Οι ακροδέκτες της σειριακής επικοινωνίας της πλατφόρμας Arduino.

2.4.3 Χαρακτηριστικά του Arduino Mega

- Μικροελεγκτής : ATmega2560
- Τάση λειτουργίας: 5V
- Τάση εισόδου: 7-12V

- Τάση εισόδου (όριο): 6-20V
- Ψηφιακοί Είσοδοι/Εξοδοι : 54 (εκ των οποίων 15 περιέχουν PWM εξόδους)
- Αναλογικές Εισόδους: 16
- DC ρεύματος I/O Pin: 40 mA
- Μέγιστο ρεύμα για τον ακροδέκτη 3.3V : 50 mA
- Μνήμη FLASH: 256 KB εκ των οποίων 8KB που χρησιμοποιούνται από τον φορτωτή εκκίνησης (bootloader)
- Μνήμη SRAM: 8 KB
- Μνήμη EPROM: 4 KB
- Ταχύτητα ρολογιού: 16 MHz

2.4.4 Βασικές Μνήμες του Arduino Mega

Η πλατφόρμα Arduino Mega διαθέτει τρεις βασικές μνήμες:

- **256 KB Μνήμη Flash:** Στην μνήμη αυτή αποθηκεύεται το πρόγραμμα μας που πρόκειται να εκτελεστεί. Ένα μέρος αυτής της μνήμης χρησιμοποιείται για την αποθήκευση του προγράμματος του φορτωτή εκκίνησης (bootloader) χωρητικότητας 8 KB.
- **8 KB Μνήμη SRAM** (στατική μνήμη τυχαίας προσπέλασης): Χρησιμοποιείται για την προσωρινή αποθήκευση των στατικών και των μεταβλητών δεδομένων του προγράμματος που εκτελείται.
- **4 KB Μνήμη EEPROM:** Σε αυτήν τη μνήμη αποθηκεύουμε τιμές που θέλουμε να παραμείνουν και μετά από μια διακοπή η επανεκκίνηση της πλακέτας σε αντίθεση με την μνήμη SRAM

2.4.5 Τροφοδοσία του Arduino Mega

Το Arduino Mega μπορεί να τροφοδοτηθεί με δύο τρόπους:

Ο πρώτος τρόπος είναι με εσωτερική τροφοδοσία παίρνοντας τροφοδοσία από το ίδιο το καλώδιο USB που συνδέουμε την πλατφόρμα μας με τον υπολογιστή. Αυτή η μέθοδος είναι πρακτική γιατί δεν χρειαζόμαστε εξωτερική τροφοδοσία αλλά μας περιορίζει στην ισχύ γιατί η σειριακή θύρα USB του υπολογιστή έχει μέγιστη τροφοδοσία ρεύματος 500 mA.

Ο δεύτερος τρόπος είναι με εξωτερική τροφοδοσία παίρνοντας τροφοδοσία από μια εξωτερική πηγή όπως ένα DC τροφοδοτικό 7-12 V ή έναν κοινό μετασχηματιστή και συνδέοντας την πηγή στην θύρα τροφοδοσίας του Arduino όπως φαίνεται στην εικόνα 2.4. Επιπλέον η εξωτερική τροφοδοσία μπορεί να συνδεθεί και απευθείας στους ακροδέκτες τροφοδοσίας του Arduino με την σήμανση Vin για τον θετικό πόλο και GND για την γείωση. Επειδή η τάση λειτουργίας της πλακέτας είναι 5V έχει ενσωματωμένο ένα ρυθμιστή τάσης που σταθεροποιεί την τάση στα 5V. Σε περίπτωση τροφοδοσίας κάτω από 7V ο σταθεροποιητής τάσης υπολειτουργεί βγάζοντας έξοδο κάτω από 5V και έτσι αντίστοιχα η έξοδος των ακροδεκτών να είναι χαμηλή. Επίσης αν

τροφοδοτήσουμε με τάση μεγαλύτερη από 12 V ο σταθεροποιητής υπερθερμαίνεται η κατανάλωση αυξάνει και υπάρχει κίνδυνος καταστροφής του.



Εικόνα 2.4 : Θύρες USB και τροφοδοσίας της πλατφόρμας Arduino.

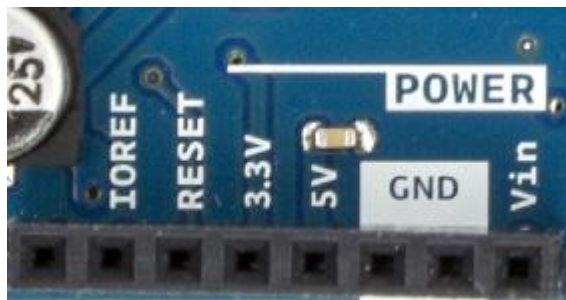
Υπάρχει μια συστοιχία με ακροδέκτες που έχουν την σήμανση “power” και είναι υπεύθυνοι για την παροχή τροφοδοσίας.

Οι ακροδέκτες τροφοδοσίας είναι οι εξής:

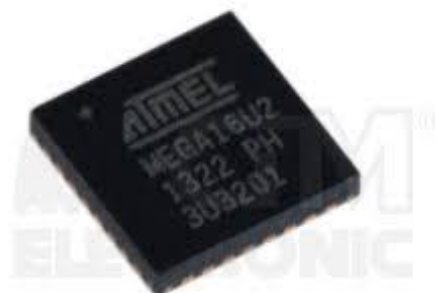
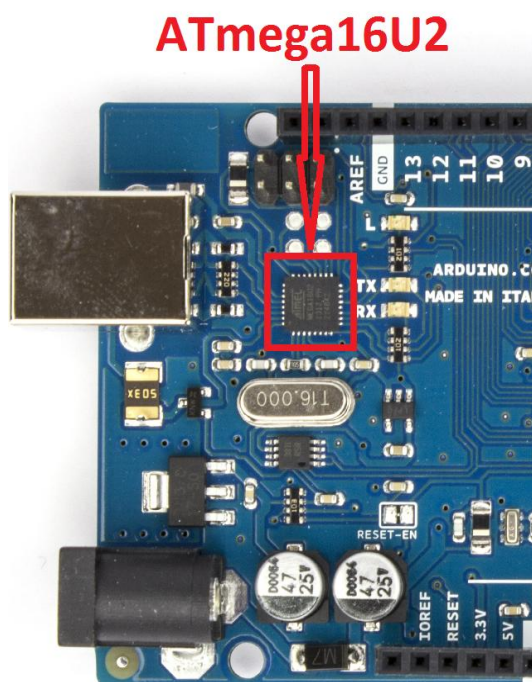
- **VIN:** Ο ακροδέκτης αυτός συνδέεται με την εξωτερική θύρα τροφοδοσίας του Arduino και έτσι μπορεί να τροφοδοτήσει κάποια άλλη συσκευή παρέχοντας τάση ίση με την τάση τροφοδοσίας του Arduino. Για παράδειγμα αν η τροφοδοσία της πλακέτας προέρχεται από έναν μετασχηματιστή στα 9V τότε και η τάση στον ακροδέκτη VIN θα είναι 9V.

Ακόμα μπορεί να λειτουργήσει και σαν εξωτερική τροφοδοσία για το ίδιο το Arduino όπως αναφέραμε παραπάνω συνδέοντας μια σταθεροποιημένη πηγή όπως μια μπαταρία, χωρίς όμως να τροφοδοτείται και από άλλη εξωτερική πηγή ταυτόχρονα.

- **5V:** Ακροδέκτης τάσης με την σήμανση 5V. Τροφοδοτεί με σταθερή τάσης 5V τον μικροελεγκτή ή άλλα ηλεκτρονικά στοιχεία που βρίσκονται πάνω στην πλακέτα του Arduino. Η σταθεροποιημένη τάση προέρχεται από το κύκλωμα ρύθμισης τάσης που έχει σαν είσοδο το ρεύμα της θύρας USB αν τροφοδοτείται από τον υπολογιστή, ή το ρεύμα της εξωτερικής τροφοδοσίας αν τροφοδοτείται η πλακέτα με εξωτερική τροφοδοσία όπως έχουμε αναφέρει και παραπάνω.
- **3V3:** Έξοδο τάσης 3,3V με μέγιστη κατανάλωση ρεύματος τα 50mA.
- **GND :** Ακροδέκτης γείωσης.



Εικόνα 2.5 : Ακροδέκτες τροφοδοσίας πλατφόρμας στο Arduino.



Εικόνα 2.6 : Μικροελεγκτής smd ATmega16U2 υπεύθυνος για την USB σύνδεση .

2.4.6 Επικοινωνία του Arduino Mega

Το Arduino MEGA επικοινωνεί με μικροελεγκτές, με ένα άλλο Arduino ή με τον Ηλεκτρονικό Υπολογιστή. Ο ATmega2560 διαθέτει 4 σειριακές θύρες UARTS (TTL 5V). Επιπλέον η αναπτυξιακή πλακέτα του Arduino παρέχει ένα ενσωματωμένο ολοκληρωμένο ATmega16U2 (αντί του τσιπ FTDI) που είναι προγραμματισμένο έτσι ώστε να παρέχει σειριακή επικοινωνία μεταξύ της πλακέτας και ενός Ηλεκτρονικού Υπολογιστή μέσα από την θύρα USB. Το ολοκληρωμένο ATmega16U2 έχει το πλεονέκτημα ότι επιτρέπει πιο γρήγορη μεταφορά και ταχύτερη σειριακή επικοινωνία μεταξύ Arduino και Υπολογιστή συγκριτικά με το FTDI. Το ATmega16U2 χρησιμοποιεί μόνο τους οδηγούς USB COM και έτσι δεν χρειάζεται εξωτερικούς οδηγούς. Οι RX και TX λυχνίες (LED) στην πλακέτα αναβοσβήνουν όταν γίνεται μετάδοση δεδομένων μέσω του "USB-to-RS232" σειριακή και USB σύνδεση με τον υπολογιστή (αλλά όχι για σειριακή επικοινωνία στις ακίδες 0 και 1).

2.4.7 Προστασία από υπερένταση στην USB θύρα

Το Arduino διαθέτει σύστημα επαναφοράς ασφάλειας προστατεύοντας την USB θύρα του υπολογιστή από την χαμηλή ή την υψηλή τάση. Σε περίπτωση που περισσότερα από 500 mA εφαρμοστούν στην θύρα USB τότε σταματάει αυτόματα η σύνδεση μέχρι να διορθωθεί το πρόβλημα όπου αποκαθίσταται αυτόματα και η σύνδεση. Αν και οι περισσότεροι υπολογιστές παρέχουν τη δική τους εσωτερική προστασία, το σύστημα επαναφοράς ασφάλειας παρέχει ένα επιπλέον επίπεδο προστασίας.

2.5 Ασπίδα Arduino

2.5.1 Η βασική ιδέα της ασπίδας Arduino.

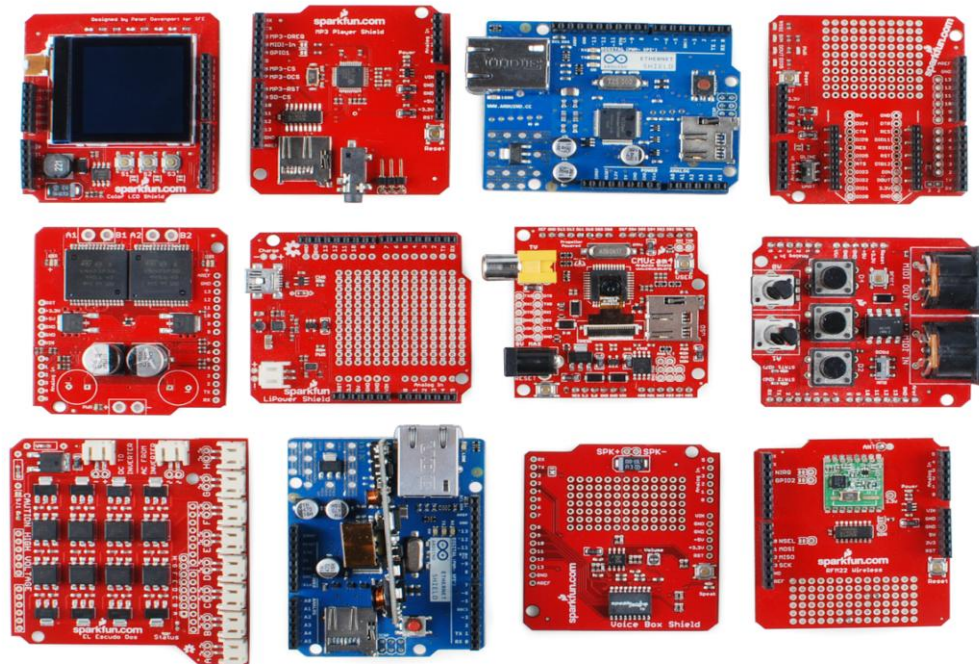
Οι λεγόμενες ασπίδες (Shield) είναι ολοκληρωμένες πλακέτες που είναι σχεδιασμένες ώστε να κουμπώνουν πάνω στο Arduino επεκτείνοντας τις δυνατότητές του. Υπάρχουν διαφορετικές ασπίδες που η καθεμία λειτουργεί για διαφορετικό σκοπό, όμως η φιλοσοφία τους είναι η ίδια, να κουμπώνουν εύκολα στην πλατφόρμα Arduino και να αποτελούν οικονομικές λύσεις.

Παρακάτω γίνεται αναφορά σε μερικές από τις δημοφιλέστερες ασπίδες του εμπορίου:

- **GSM/GPRS Shields:** Μας παρέχουν την δυνατότητα αποστολής σύντομων γραπτών μηνυμάτων SMS, τηλεφωνικής κλήσης σε κινητό καθώς και σύνδεσης στο Διαδίκτυο ακριβώς όπως και με ένα κινητό τηλέφωνο.
- **GPS Shield:** Χρησιμοποιεί τεχνολογία GPS προσθέτοντας GPS δυνατότητες στο Arduino (εντοπισμό στίγματος).
- **Ethernet Shield:** Συνδέει την συσκευή μας με το Internet μέσω ενός καλωδίου Ethernet.
- **WiFi Shield:** Μας επιτρέπει την σύνδεση με WIFI συσκευές. Υπάρχει και εδώ η δυνατότητα σύνδεσης με το Διαδίκτυο αλλά ασύρματα.
- **Shield οθόνης:** Μας επιτρέπουν την σύνδεση οθόνης στο Arduino, από πολύ απλές μέχρι και ιδιαίτερα υψηλής ανάλυσης οθόνες.
- **Wave Shield:** Αυτός ο τύπος ασπίδας χρησιμοποιείται κυρίως σε εφαρμογές που έχουν να κάνουν με ηχητικά σήματα/μουσική, συνήθως υποστηρίζουν και κάρτες SD.
- **Motor Shields:** Μας δίνει την δυνατότητα να ελέγξουμε μοτέρ διάφορων τύπων (DC, servo, stepper κ.α.).
- **Bluetooth Shield:** Μας επιτρέπει την ασύρματη επικοινωνία με bluetooth συσκευές.
- **SD Card Shield:** Επιτρέπει την χρήση εξωτερικής μνήμης SD Card στην συσκευή μας αυξάνοντας τον διαθέσιμο αποθηκευτικό χώρο.

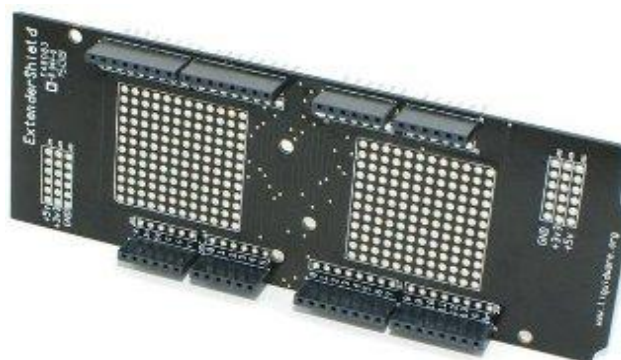
Οι ασπίδες σχεδιάζονται έτσι ώστε να μπορούν να κουμπώσουν πάνω στην πλακέτα Arduino αφήνοντας εκτεθειμένες τις υποδοχές τους ώστε να είναι δυνατή η σύνδεση και με άλλα εξαρτήματα ή και με άλλες ασπίδες. Είναι σημαντικό να αναφερθεί ότι κάθε ασπίδα για να συνδεθεί με το Arduino χρησιμοποιεί συγκεκριμένες ακίδες. Αυτό

έχει ως αποτέλεσμα πρώτον, ο αριθμός των ασπίδων που μπορούν να συνδεθούν ταυτόχρονα μεταξύ τους σε ένα Arduino είναι πεπερασμένος και δεύτερον, ότι οι ασπίδες που απαιτούν τις ίδιες ακίδες για να συνδεθούν με το Arduino δεν είναι συμβατές μεταξύ τους. Υπάρχουν όμως κάποιες ασπίδες που δεν προωθούν τις συνδέσεις του Arduino. Σε αυτές τις περιπτώσεις χρησιμοποιούνται ειδικές ασπίδες επέκτασης (extender shield) που κουμπώνουν επάνω στο Arduino και δίνουν την δυνατότητα σε άλλες ασπίδες να κουμπώσουν πάνω τους (βλέπε εικ. 2.8).



Εικόνα 2.7 : Μερικές από τις ασπίδες που είναι συμβατές με το Arduino

Στην αγορά κυκλοφορεί μεγάλη ποικιλία σε ασπίδες, ικανή να καλύψει κάθε απαίτηση. Σημαντικό πλεονέκτημα των ασπίδων είναι ότι κάθε ασπίδα, συνήθως, συνοδεύεται από δικές της βιβλιοθήκες καθιστώντας εύκολη τη χρήση χωρίς να απαιτούνται ιδιαίτερες γνώσεις στον τομέα των ηλεκτρονικών.



Εικόνα 2.8 : επέκταση ασπίδας δύο θέσεων

Συνοψίζοντας, οι ασπίδες αποτελούν τη σωστή επιλογή όταν θέλουμε να δημιουργήσουμε ολοκληρωμένα προγράμματα και γι' αυτό δεν συνιστάται η αγορά κάποιας έκδοσης του Arduino που δεν είναι απόλυτα συμβατή με τις ασπίδες.

2.5.2 Γνωριμία με την ασπίδα IComSat

Η ασπίδα της εταιρείας IComsat είναι μία GSM/GPRS ασπίδα συμβατή με το Arduino, έχοντας σαν βάση τον επεξεργαστή SIM900. Ελέγχεται μέσω AT εντολών και είναι πλήρως συμβατή με το Arduino Mega .

Με την ασπίδα IComSat είναι δυνατή η αποστολή και λήψη μηνυμάτων, η κλήση σε σταθερό ή κινητό τηλέφωνο, καθώς και η σύνδεση στο Διαδίκτυο δίνοντάς μας την δυνατότητα να ανοίξουμε μια ιστοσελίδα ή να χρησιμοποιήσουμε την ασπίδα ως εξυπηρετητή (Server).

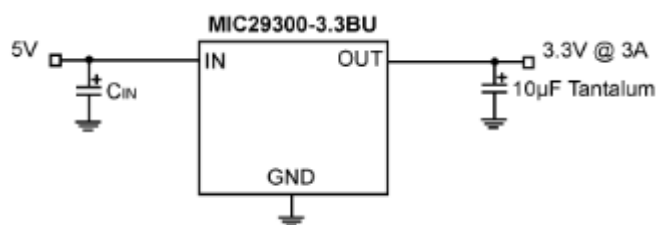
Η ασπίδα τροφοδοτείται από τον ακροδέκτη 5V του Arduino και Gnd αντίστοιχα και δεν χρειάζεται εξωτερική τροφοδοσία.

2.5.3 Χαρακτηριστικά της ασπίδας IComSat

- Υποστήριξη 4 ζωνών συχνοτήτων 8500/900/1800/1900Mhz
- Δυνατότητα πολλών συνδέσεων ταυτόχρονα.
- Υποστήριξη GPRS σύνδεση κλάσης B
- Κλάση 4 (2W@850/900MHz)
- Κλάση 1 (1W@1800/1900MHz)
- Ελέγχεται μέσα από AT εντολές
- Υποστήριξη επικοινωνίας μέσω σύντομων μηνυμάτων SMS (Short Message Service).
- Δυνατότητα επιλογής του σειριακού ακροδέκτη που συνδέεται με το Arduino
- Όλοι οι ακροδέκτες της ασπίδας IComsat είναι προσβάσιμοι .
- Διαθέτει ρολόι πραγματικής ώρας (RTC Real Time Clock) .
- Υποστηρίζει το αυτόματο άνοιγμα/κλείσιμο (Power On/Off) και την επανεκκίνηση της ασπίδας μέσα από το πρόγραμμα του Arduino.

2.5.4 Τροφοδοσία της ασπίδας IComSat

Όπως προαναφέραμε, η ασπίδα IComSat μπορεί να τροφοδοτηθεί απευθείας από τους ακροδέκτες τροφοδοσίας του Arduino (ακροδέκτης:5V και ακροδέκτης :Gnd) χωρίς να χρειάζεται ξεχωριστή τροφοδοσία.



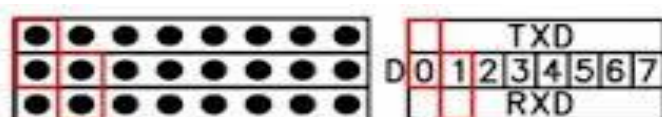
Εικόνα 2.9 : Απλό κύκλωμα σταθεροποιητής τάσης MIC29300 στα 3.3V

Επειδή η τάση λειτουργίας του επεξεργαστή SIM900 είναι από 2,5 - 4,5V η πλακέτα της ασπίδας έχει ενσωματωμένο ένα σταθεροποιητή τάσης (LDO linear regulator) που σταθεροποιεί την τάση εισόδου (7 - 20)V στην τάση εξόδου 3.3 V εξασφαλίζοντας έτσι τη σωστή λειτουργία του επεξεργαστή.

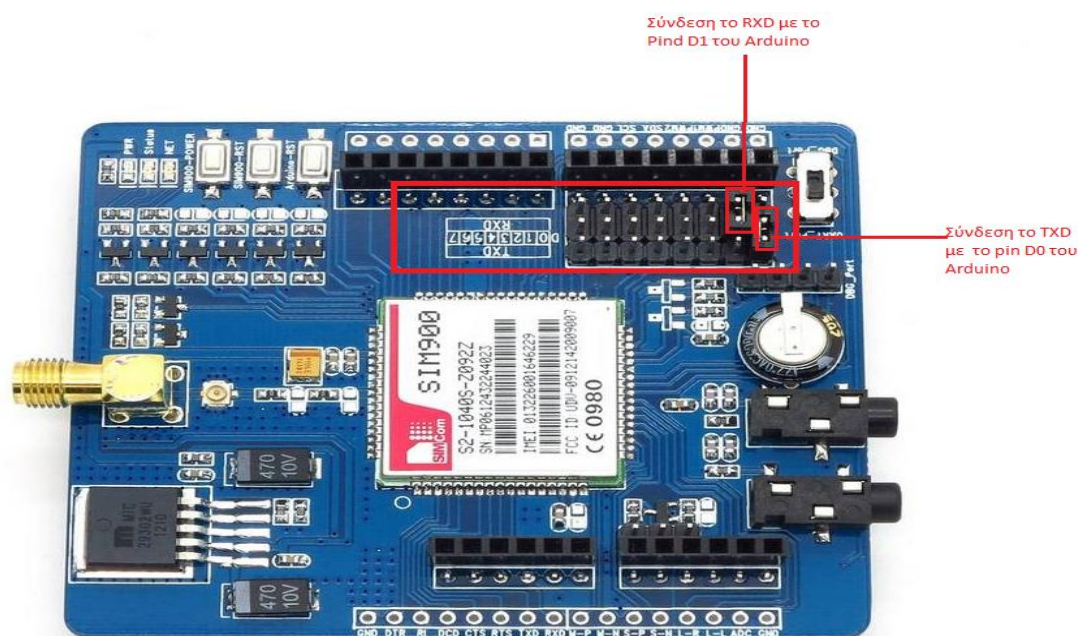
2.5.5 Επικοινωνία της ασπίδας IComSat με το Arduino

Η ασπίδα επικοινωνεί με το Arduino μέσω σειριακής σύνδεσης με ειδικούς ακροδέκτες, οι οποίοι αποτελούνται από τρεις σειρές: στην πρώτη έχουμε τους ακροδέκτες TXD για την αποστολή, στην δεύτερη σειρά κάθε ακροδέκτης αντιστοιχεί στους ακροδέκτες του Arduino από το D0-D7 και τέλος στην τρίτη σειρά έχουμε τους ακροδέκτες RXD για την λήψη. Με τον τρόπο αυτό γίνεται πολύ εύκολη η επιλογή της σειριακής σύνδεσης του Arduino με την ασπίδα.

Έτσι ένα παράδειγμα σύνδεσης της ασπίδας με το Arduino υπο φαίνεται στην παρακάτω εικόνα όπου χρησιμοποιούν οι ακροδέκτες 0 και 1 του Arduino τα αντίστοιχα Rx και Tx.



Εικόνα 2.10 : Σύνδεση Σειριακής με τις ακίδες του IComSat και του Arduino



Εικόνα 2.11: Ακίδες Επικοινωνίας του IComSat και τρόποι σύνδεσης

Τέλος πρέπει να σημειωθεί ότι ο ακροδέκτης TXD της ασπίδας συνδέεται με τον Rx ακροδέκτη του Arduino και αντίστοιχα ο RXD ακροδέκτης της ασπίδας με τον Tx ακροδέκτη του Arduino.

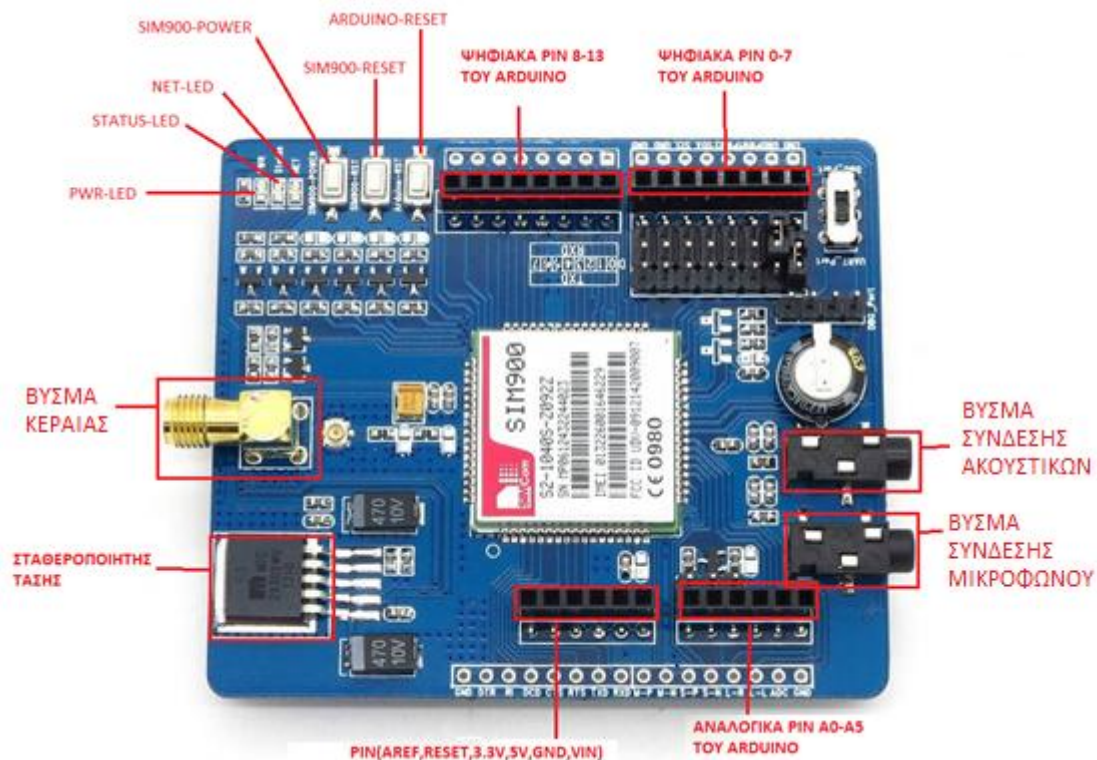
2.5.6 LED και Κουμπιά της ασπίδας IComSat.

Στον παρακάτω πίνακα παρατίθενται η κατάσταση του LED κατάστασης δικτύου (NETSTATUS) της ασπίδας IComSat.

- **STATUS:** Περιγράφει την κατάσταση λειτουργίας του επεξεργαστή SIM900
- **PWR:** Περιγράφει την κατάσταση τροφοδοσίας της ασπίδας IComSat. Ακόμα διαθέτει και τρία κουμπιά *SIM900-POWER*, *Sim900-RST*, *Arduino-RST*.

SIM900-POWER: Μετά την τροφοδοσία Του IComSat χρειάζεται να πιέσουμε το κουμπί SIM900-POWER για να το ενεργοποιήσουμε. Ο ακροδέκτης 9 του Arduino συνδέεται με το PWR-KEY του IComSat έτσι στέλνοντας έναν παλμό στον ακροδέκτη 9 (πλάτους 400μs) μπορεί το Arduino να θέσει σε λειτουργία ή να διακόψει την ασίδα. **Sim900-RST:** Κάνει επανεκκίνηση της ασίδας. Ο ακροδέκτης 8 του Arduino συνδέεται με το κουμπί επανεκκίνησης (RESET) της ασίδας και έτσι αν στείλουμε έναν παλμό διάρκειας 400μs κάνουμε επανεκκίνηση.

Κατάσταση του LED	Περιγραφή-Επεξήγηση
LED Σβηστό	Η ασίδα είναι εκτός λειτουργίας
64ms On/800ms Off	Η ασίδα δεν έχει συνδεθεί στο δίκτυο
64ms On/3000ms Off	Η ασίδα έχει συνδεθεί στο δίκτυο
64ms On/300ms Off	Η GPRS επικοινωνία έχει ολοκληρωθεί



Εικόνα 2.11 : Τα μέρη που αποτελείται μια ασίδα IComSat.

Τέλος με το κουμπί *Arduino-RST* μπορούμε να κάνουμε επανεκκίνηση στο πρόγραμμα που τρέχει στο Arduino.

3. Λογισμικά που χρησιμοποιήθηκαν στην Εργασία.

Στο κεφάλαιο αυτό αναλύουμε τα διάφορα προγράμματα που χρησιμοποιήθηκαν για την υλοποίηση της εργασίας. Το Arduino IDE είναι υπεύθυνο για την δημιουργία του κώδικα προγραμματισμού του Arduino Mega καθώς και για το πέρασμα του κώδικα από τον υπολογιστή στο Arduino. Το πρόγραμμα XAMPP φιλοξενεί όλα τα αρχεία του που αλληλεπιδρούν με το Διαδίκτυο και συμπεριφέρεται σαν ένας εξυπηρετητής ο οποίος δέχεται και στέλνει αιτήματα σύνδεσης. Εγκαθίσταται σε έναν απλό υπολογιστή και βοηθάει στον συχνό έλεγχο της ιστοσελίδας. Τέλος το Notepad++ είναι ένα πρόγραμμα επεξεργασίας κειμένου που υποστηρίζει πλήθος γλωσσών προγραμματισμού με πολύ μικρές απαιτήσεις.

3.1 Λογισμικό Arduino IDE.

3.1.1 Περιγραφή της εγκατάστασης του Arduino IDE.

Για να μπορέσουμε να συνδέσουμε το Arduino στον Υπολογιστή, να ανεβάσουμε και να τρέξουμε το πρόγραμμα χρειάζεται αρχικά να εγκαταστήσουμε το πρόγραμμα Arduino IDE στον υπολογιστή. Το λειτουργικό σύστημα που θα γίνει η εγκατάσταση είναι των Windows 7 και περιγράφεται παρακάτω:



- *Πλακέτα Arduino Mega και ένα καλώδιο USB*

Σε αυτήν την εργασία χρησιμοποιούμε την πλακέτα Arduino Mega και ένα καλώδιο USB τύπου από A σε B όπως φαίνεται στην εικόνα 3.1 για να είναι συμβατό με την πλακέτα μας.



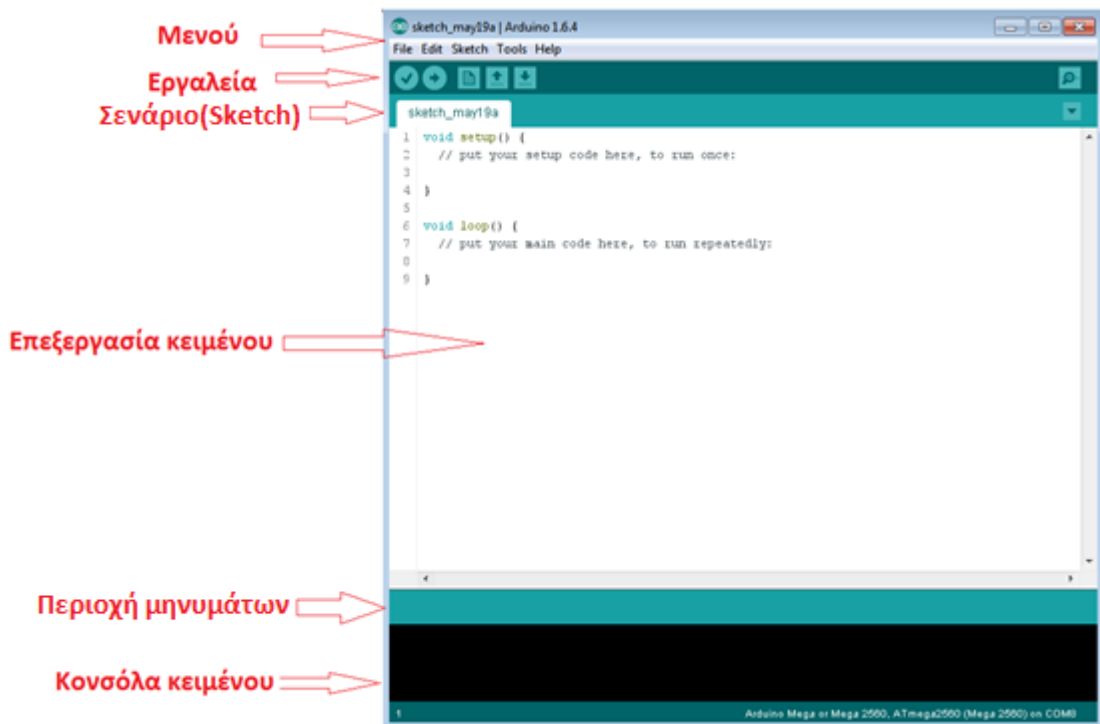
Εικόνα 3.1: Στα αριστερά η πλακέτα Arduino mega και στα δεξιά καλώδιο επικοινωνίας USB(A σε B).

- *Το πρόγραμμα του Arduino*
Κατεβάζουμε δωρεάν την τελευταία έκδοση το προγράμματος από τη επίσημη σελίδα του Arduino: <http://www.arduino.cc/en/Main/Software> .
- *Συνδέουμε την πλακέτα με τον Υπολογιστή*
Συνδέουμε το Arduino Mega με τον υπολογιστή χρησιμοποιώντας το καλώδιο USB. Η πράσινη φωτοδίοδος (LED) με την ένδειξη PWR αρχίζει να ανάβει για όσο διάστημα παραμένει συνδεδεμένη η πλακέτα με τον Η/Υ.

- *Κάνουμε εγκατάσταση στο πρόγραμμα*
 - ✓ Αφού συνδέσουμε την πλακέτα, τα Windows ξεκινάνε την διαδικασία αυτόματα
 - ✓ Κάνουμε κλικ στο μενού Έναρξης και ανοίγουμε τον πίνακα ελέγχου
 - ✓ Μεταβαίνουμε στο “Σύστημα” μετά στο “Υλικό” και επιλέγουμε “Διαχείριση Συσκευών”
 - ✓ Επιλέγοντας στις Θύρες θα πρέπει να δούμε το όνομα “Arduino Mega” .
 - ✓ Κάνουμε δεξί κλικ στην πλακέτα μας και επιλέγουμε “Ενημέρωση προγράμματος οδήγησης”.
 - ✓ Στην συνέχεια επιλέγουμε “Αναζήτηση στον Υπολογιστή μου “ και ψάχνουμε να βρούμε το αρχείο “arduino.inf” που βρίσκεται στον φάκελο “Drivers”.
 - ✓ Στην συνέχεια τα Windows ολοκληρώνουν την εγκατάσταση.
- Εκκίνηση της εφαρμογής του Arduino.
Ανοίγουμε την εφαρμογή κάνοντας διπλό κλικ στο εικονίδιο “arduino.exe” και έχουμε ολοκληρώσει την εγκατάσταση.

3.1.2 Περιβάλλον ανάπτυξης του Arduino

Το περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή πλατφόρμας γραμμένη σε γλώσσα προγραμματισμού Java και Processing. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κειμένου για τη σύνταξη κώδικα, μια περιοχή μηνυμάτων, μια κονσόλα κειμένου, μια γραμμή εργαλείων με κουμπιά για διάφορες λειτουργίες και μια σειρά από μενού.









Εικόνα 3.2 : Περιβάλλον ανάπτυξης Arduino IDE .

Θα πρέπει να συνδέσουμε την πλακέτα του Arduino με τον υπολογιστή μέσω της θύρας USB για να μπορέσουμε να φορτώσουμε τα προγράμματα που έχουμε δημιουργήσει στο περιβάλλον ανάπτυξης του Arduino.

3.1.3 Δημιουργία σεναρίου (sketch) στο περιβάλλον του Arduino.

Σενάρια καλούνται τα ολοκληρωμένα προγράμματα του Arduino με την κατάληξη .ino και γράφονται στον επεξεργαστή κειμένου. Μπορούμε να επιλέξουμε αντιγραφή/επικόλληση, αναζήτηση/αντικατάσταση κειμένου όπως και σε ένα κοινό επεξεργαστή κειμένου. Στην περιοχή μηνυμάτων απεικονίζονται μηνύματα κατά την αποθήκευση, κατά το ανέβασμα ενός σεναρίου καθώς επίσης και όταν υπάρξει κάποιο σφάλμα. Η κονσόλα κειμένου απεικονίζει το περιβάλλον γραφής κώδικα του Arduino συμπεριλαμβάνοντας και τα πλήρη μηνύματα σφαλμάτων και άλλες πληροφορίες. Στην κάτω δεξιά γωνία του παραθύρου μπορούμε να δούμε το μοντέλο της πλακέτας που τρέχουμε όπως και την θύρα του υπολογιστή που έχει γίνει η σύνδεση με την πλακέτα μας. Μια χρήσιμη και εύκολη στην χρήση είναι η γραμμή εργαλείων που περιέχει κουμπιά με τα οποία γίνεται έλεγχος για σφάλματα, ανεβάζουμε το πρόγραμμα στην πλακέτα, ανοίγουμε, αποθηκεύουμε σκίτσα και κάνουμε χρήση της σειριακής οθόνης. Στον παρακάτω πίνακα βλέπουμε τα εικονίδια των κουμπιών αυτών καθώς και χαρακτηριστικά τους.

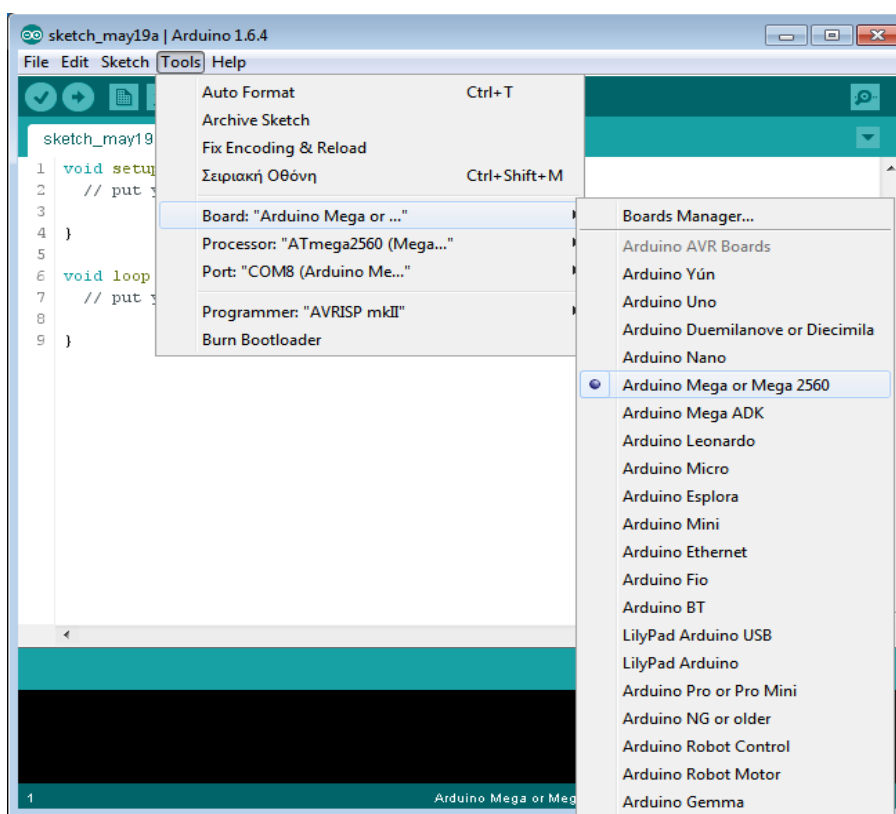
	Έλεγχος (Verify): Κάνει έλεγχο στον κώδικα για τυχόν λάθη
	Ανέβασμα (Upload): Συντάσσει και περνάει τον κώδικα στην πλακέτα.
	Νέο (New): Δημιουργεί ένα νέο σενάριο
	Άνοιγμα (Open): Επιλέγεις το σενάριο που θέλεις να επεξεργαστείς και να τρέξεις.
	Αποθήκευση (Save): Αποθηκεύει το σενάριο.
	Σειριακή Οθόνη (Serial Monitor): Ανοίγει την σειριακή οθόνη σε ξεχωριστό παράθυρο επιτρέποντας την απεικόνιση και την εισαγωγή δεδομένων

Εικόνα 3.3: Χαρακτηριστικά και λειτουργίες της γραμμής εργαλείων στο Arduino IDE

3.1.4 Ανέβασμα σεναρίου στο Arduino.

Πριν ανεβάσουμε το σενάριο με το πρόγραμμα θα πρέπει να κάνουμε κάποιες ρυθμίσεις. Επιλέγουμε την πλακέτα που χρησιμοποιούμε μέσα από το μενού

Tools>Board> π.χ για την πλακέτα που χρησιμοποιούμε επιλέγουμε το ArduinoMega. Στην συνέχεια να επιλέγουμε την σειριακή θύρα, μέσα από το μενού Tools>Serial Port. Από την στιγμή που επιλέγουμε την σωστή σειριακή θύρα και την πλακέτα πατώντας το κουμπί “Upload” που βρίσκεται στην γραμμή εργαλείων, ανεβάζουμε το σενάριο μας στο Arduino. Αφού ολοκληρωθεί το ανέβασμα θα γίνει αυτόματη επανεκκίνηση (Reset) της πλακέτας και στην συνέχεια το Arduino εκτελεί τον καινούριο κώδικα που ανεβάσαμε. Στις περισσότερες πλακέτες αναβοσβήνουν τα LED Rx και Tx του Arduino κατά την διάρκεια που ανεβάζουμε το πρόγραμμα. Το περιβάλλον του Arduino θα εμφανίσει ένα μήνυμα όταν ολοκληρωθεί το ανέβασμα, ή ένα σφάλμα και θα σταματήσει το ανέβασμα μέχρι να επιλυθούν όλα τα σφάλματα. Κατά το ανέβασμα ενός σεναρίου, το Arduino χρησιμοποιεί τον φορτωτή εκκίνησης (όπως έχουμε αναφέρει και σε άλλο κεφάλαιο) για μερικά δευτερόλεπτα. Ο φορτωτής εκκίνησης αναβοσβήνει το LED που επικοινωνεί με τον ακροδέκτη 13, την στιγμή της επανεκκίνησης.

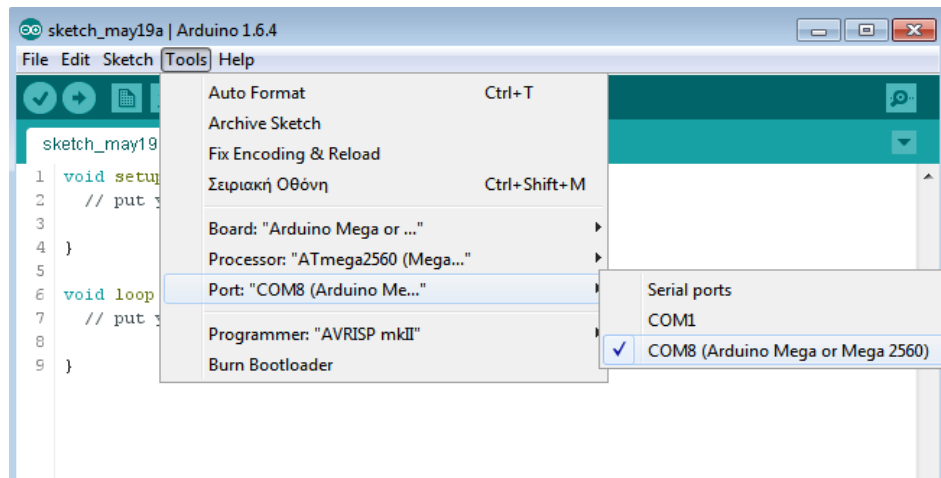


Εικόνα 3.4: Επιλογή της πλακέτας Arduino μέσα από το μενού του Arduino IDE

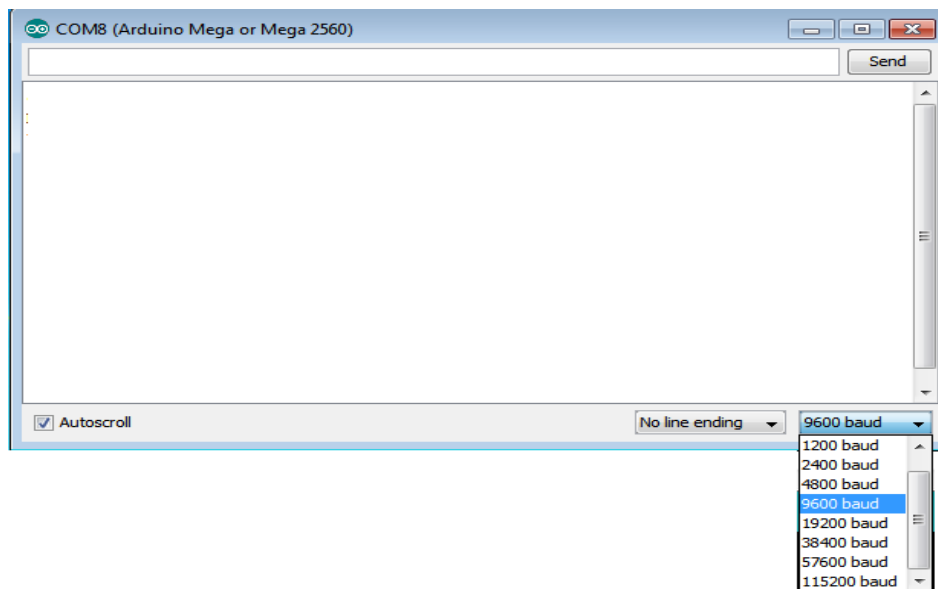
3.1.5 Σειριακή Οθόνη.

Η σειριακή οθόνη εμφανίζει τα σειριακά δεδομένα που στέλνει το Arduino (μέσω θύρας USB). Για να στείλουμε δεδομένα στο Arduino πληκτρολογούμε το κείμενο που θέλουμε να στείλουμε και στην συνέχεια το στέλνουμε πατώντας το κουμπί (Send) ή με το πλήκτρο Enter. Για να εμφανιστούν σωστά τα δεδομένα μας στην οθόνη της σειριακής θα πρέπει να επιλέξουμε τον ρυθμό μετάδοσης σύμφωνα με αυτόν που τρέχουμε στο πρόγραμμα. Αν έχουμε δηλώσει ρυθμό μετάδοσης στα 9600 με την εντολή Serial.begin(9600) τότε επιλέγουμε από την λίστα το 9600. Η απεικόνιση των

δεδομένων μπορεί να γίνει και με άλλα προγράμματα εκτός από το Arduino IDE όπως Processing, Flash, MaxMSP.



Εικόνα 3.5: Επιλογή της σειριακής πόρτας που επικοινωνεί με την πλακέτα Arduino μέσα από το μενού του Arduino IDE.



Εικόνα 3.6: Το περιβάλλον απεικόνισης της σειριακής οθόνης και η επιλογή του ρυθμού ταχύτητας δεδομένων.

3.1.6 Συστατικά και λειτουργίες ενός σεναρίου

Η γλώσσα που χρησιμοποιεί το Arduino Mega όπως και κάθε πλακέτα Arduino είναι η Wiring η οποία μοιάζει με την C++ με κάποιες διαφορές. Παρακάτω θα γίνει μια συνοπτική αναφορά στην βασική δομή, τις συναρτήσεις και τις μεταβλητές

- *Βασική δομή ενός σεναρίου*

Κάθε σενάριο θα πρέπει να περιλαμβάνει δύο ειδικές συναρτήσεις την loop() και την Setup(). Η συνάρτηση setup() καλείται μία φορά μόνο όταν ξεκινάει ένα σενάριο. Για τον λόγο αυτό χρησιμοποιείται συνήθως για τις αρχικοποιήσεις διάφορων καταστάσεων. Αντίθετα η συνάρτηση loop() καλείται ξανά και ξανά, είναι η βασική συνάρτηση στην οποία γράφουμε το κυρίως πρόγραμμα.

```
void setup()
```

```
{ //το πρόγραμμα μέσα στην setup() τρέχει μία μόνο φορά
}
```

```
void loop()
```

```
{ //το πρόγραμμα μέσα στην loop() επαναλαμβάνεται.
}
```

- *Γράφοντας το πρώτο μας σενάριο.*

Για να γίνει πιο εύκολη η κατανόηση του προγραμματισμού δημιουργήσαμε ένα απλό σενάριο που θα αναβοσβήνει ένα LED κάθε 2 δευτερόλεπτα και θα εξηγήσουμε κομμάτι-κομμάτι το πρόγραμμα. Η βασική δομή του προγράμματος δεν αλλάζει περιέχοντας σε κάθε πρόγραμμα τις δύο βασικές συναρτήσεις loop() και setup(). Το πρόγραμμα του παραδείγματος απεικονίζεται στην εικόνα. Το πρόγραμμα αποτελείται από τρία βασικά μέρη: τα σχόλια, τη δήλωση μεταβλητών και τις συναρτήσεις.

Σχόλια : Τα σχόλια είναι πρόγραμμα το οποίο δεν μεταγλωττίζεται και δεν εκτελείται. Σχόλια μπορούμε να εισάγουμε με δύο τρόπους:

α) Πρώτος τρόπος είναι να ξεκινήσουμε με “/*”, στο ενδιάμεσο γραφούμε το σχόλιο και κλείνουμε με “*/”. Σαν σχόλιο αναφέρουμε ένα κομμάτι προγράμματος το οποίο δεν θέλουμε να εκτελεστεί. Δεν μας περιορίζει στο μέγεθος που θα έχει ένα σχόλιο.

β) Δεύτερος τρόπος είναι να γράψουμε “//”, ορίζοντας σαν σχόλιο ολόκληρη την γραμμή. Οπότε ό,τι γράψαμε μετά “//” αποτελεί σχόλιο αλλά μόνο για αυτή την γραμμή. Έτσι και στο πρόγραμμα με την εισαγωγή “//” και “/* */” σχολιάζουμε την κάθε λειτουργία του εκτελέσιμου προγράμματος.

Δήλωση Μεταβλητών: Μεταβλητή είναι ένας αποθηκευτικός χώρος-θέση μνήμης που ακούει σε ένα όνομα και παίρνει τιμή συγκεκριμένου τύπου. Οι μεταβλητές παίζουν σημαντικό ρόλο στην σύνταξη του κώδικα, καθώς αλλάζοντας τις τιμές τους κατά την διάρκεια εκτέλεσης του προγράμματος, μπορούμε να επιτύχουμε διάφορες λειτουργίες. Η δήλωση μιας μεταβλητής γίνεται με τον εξής τρόπο: Τύπος Όνομα =Τιμή; .Παρακάτω αναφέρονται παραδείγματα βασικών τύπων μεταβλητών:

- int number: Ακέραιος αριθμός από το -32.768 μέχρι 32.767 π.χ: int val=34;
- float number: Δεκαδικός αριθμός π.χ: float num=10.238;
- char names: Χαρακτήρας (γράμμα ή ψηφίο) π.χ: char ch='a' ή '1'
- Boolean state: Δυαδικός παίρνει τιμές αληθής (True) ή ψευδής (False), 1 και 0

Στο παράδειγμά μας έχουμε δήλωση δύο μεταβλητών:

`int ledPin=13;` : Ακέραιος με τιμή “13” και όνομα “ledPin”. Επιλέξαμε την τιμή 13 γιατί η πλακέτα Arduino Mega έχει ενσωματωμένο LED το οποίο επικοινωνεί με τον ακροδέκτη 13 και έτσι δεν χρειαζόμαστε εξωτερικά εξαρτήματα και συνδέσεις για τον έλεγχο του LED.

`int time=2000;` : Ακέραιος με τιμή “2000” και όνομα “time”. Βάλαμε την τιμή 2000 γιατί θέλουμε να αναβοσβήνει κάθε 2 δευτερόλεπτα οπότε 2000 ms



```

1  /*
2  Ανάβει/σβήνει μια λυχνία κάθε δύο δευτερόλεπτα
3  */
4
5  int ledPin=13;
6  int time=2000;
7
8  // Συναρτηση αρχικοποίησης τρέχει μια φορά
9  void setup() {
10     //Ορίζουμε το ποδαράκι 13 του Arduino σαν Έξοδο.
11     pinMode(ledPin, OUTPUT);
12 }
13
14 // η loop() συναρτήσει επαναλαμβάνεται.
15 void loop() {
16     digitalWrite(ledPin, HIGH); // Δίνουμε στο ποδαράκι υψηλή έξοδο και η λυχνία ανάβει.
17     delay(time); // περιμένουμε για 2 δευτερόλεπτα
18     digitalWrite(ledPin, LOW); // Δίνουμε στο ποδαράκι 13 της λυχνια χαμηλή έξοδο, η λυχνία σβήνει
19     delay(time); // περιμένουμε 2 δευτερολεπτα

```

Εικόνα 3.7: Το περιβάλλον απεικόνισης της σειριακής οθόνης και η επιλογή του ρυθμού ταχύτητας δεδομένων.

Συναρτήσεις: Εκτός από τις βασικές συναρτήσεις `setup()` και `loop()` στο πρόγραμμα μας συναντάμε και άλλες. Αυτές είναι οι:

A) pinMode(pin, mode); Με την συνάρτηση `pinMode();` αρχικοποιούμε τους ακροδέκτες του Arduino σαν εισόδους ή εξόδους. Στην θέση “Pin” ορίζουμε τον ακροδέκτη του Arduino και στην θέση “mode ” την επιθυμητή κατάσταση που θέλουμε να έχει. Οι δυνατές καταστάσεις είναι δύο INPUTS για όρισμα ως είσοδος και OUTPUTS για όρισμα ως έξοδος. Στο παράδειγμα με την συνάρτηση `pinMode(LedPin,OUTPUT);` ορίζουμε το ποδαράκι 13 σαν έξοδο για να μπορεί να ελέγχει την φωτοδίοδο.

B) delay(value); Με την συνάρτηση `delay()` δημιουργούμε μία καθυστέρηση δευτερολέπτων στην ροή του προγράμματός μας ίση με την τιμή που δίνουμε στην

θέση "value". Στο παράδειγμά μας δημιουργούμε μια καθυστέρηση 2 δευτερολέπτων κάθε φορά που καλούμε την συνάρτηση delay(2000);

Γ) digitalWrite(pin, value); Με την συνάρτηση digitalWrite() καθορίζουμε την τιμή που θα έχει ένας ακροδέκτης. Αριθμούμε τον ακροδέκτη που θέλουμε να οδηγήσουμε στην θέση "pin" και στην τιμή "value" δηλώνουμε την τιμή με την οποία έχουμε ορίσει την έξοδό μας. Οι τιμές αυτές μπορεί να είναι HIGH για υψηλό δυναμικό 5 VOLT και LOW χαμηλό δυναμικό.

3.2 Λογισμικό Xampp Εξυπηρετητή

Το XAMPP είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού το οποίο περιλαμβάνει τις τελευταίες εκδόσεις του Apache, PHP, perl και MySQL, όπως προέρχεται και από την ετυμολογία του. Αποτελεί το πιο δημοφιλές περιβάλλον ανάπτυξης PHP σελίδων γιατί είναι πολύ εύκολο στην εγκατάστασή του, λογισμικό ανοιχτού κώδικα και δωρεάν. Το ακρωνύμιο του XAMPP φαίνεται παρακάτω:

- **X** (αναφέρεται στο «cross-platform» που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας)
- **A**pache HTTP εξυπηρετητής
- **M**ySQL
- **P**HP
- **P**erl

Επίσης οι σχεδιαστές του XAMPP προόριζαν το λογισμικό ως εργαλείο ανάπτυξης και δοκιμής ιστοσελίδων τοπικά στον υπολογιστή χωρίς να είναι απαραίτητη η σύνδεση στο Διαδίκτυο. Το XAMPP είναι ένα χρήσιμο βοήθημα για αυτούς που ασχολούνται με ιστοσελίδες και διαδικτυακές εφαρμογές γιατί δίνει την δυνατότητα να μετατρέψουμε τον υπολογιστή μας σε έναν ολοκληρωμένο διαδικτυακό εξυπηρετητή (Web Server) με λογισμικό ανοιχτού κώδικα. Από την στιγμή που το XAMPP εγκατασταθεί στον υπολογιστή διαχειρίζεται τον ίδιο τον υπολογιστή (localhost) ως ένα απομακρυσμένο κόμβο, ο οποίος συνδέεται με το πρωτόκολλο μεταφοράς αρχείων FTP. Το XAMPP υποστηρίζει την δημιουργία και διαχείριση βάσεων δεδομένων. Είναι λογισμικό ανεξάρτητο πλατφόρμας και τρέχει σε διάφορα λειτουργικά συστήματα Microsoft Windows, Linux, Solaris, και Mac OSX.

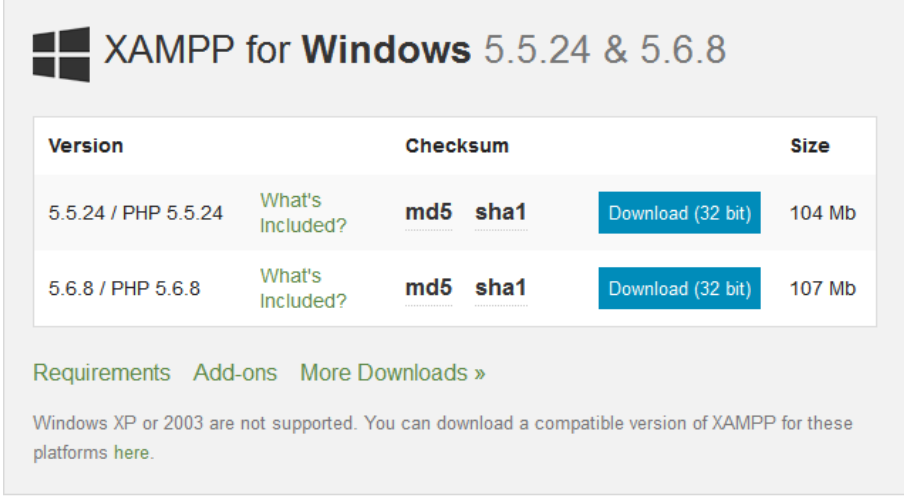
3.2.1 Εγκατάσταση του λογισμικού XAMPP.

Το λογισμικό του XAMPP όπως αναφέραμε παραπάνω είναι δωρεάν και θα το κατεβάσουμε από την επίσημη σελίδα:

<https://www.apachefriends.org/download.html>.

Η εγκατάσταση γίνεται σε υπολογιστή με λειτουργικό σύστημα Windows.

- ✓ Αφού επιλέξουμε την σωστή έκδοση για το λειτουργικό μας σύστημα. Windows 7 32bit.



XAMPP for Windows 5.5.24 & 5.6.8

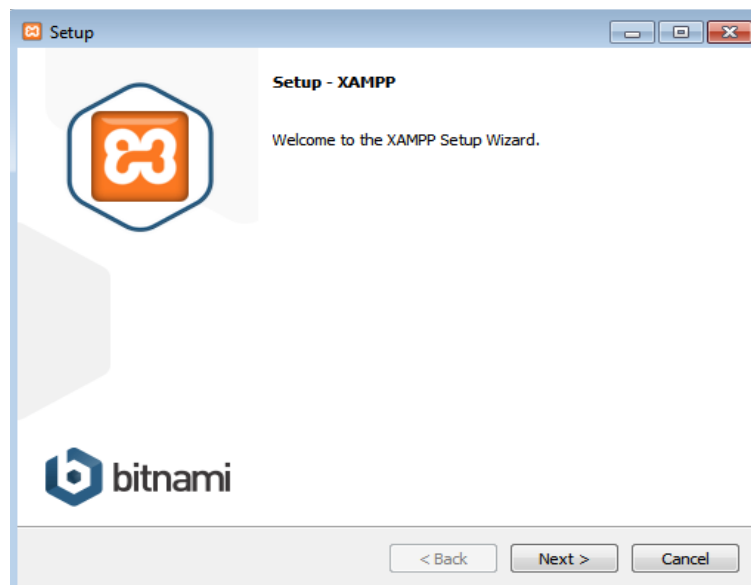
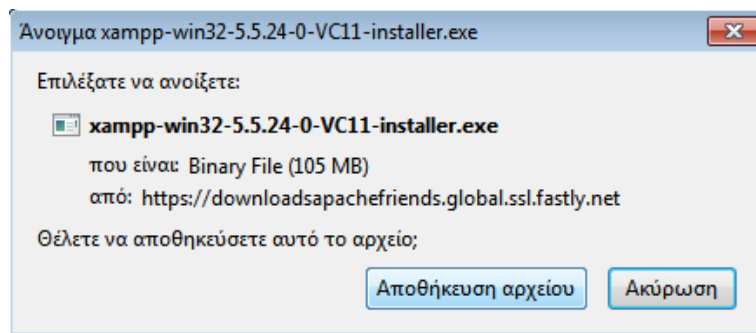
Version	Checksum	Size
5.5.24 / PHP 5.5.24 What's Included?	md5 sha1 Download (32 bit)	104 Mb
5.6.8 / PHP 5.6.8 What's Included?	md5 sha1 Download (32 bit)	107 Mb

[Requirements](#) [Add-ons](#) [More Downloads »](#)

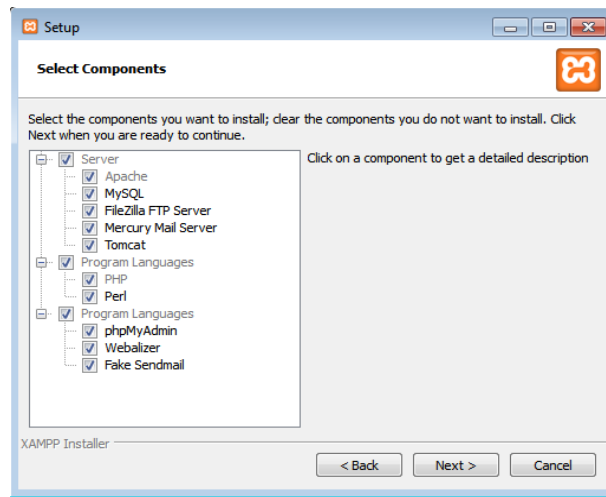
Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

Εικόνα 3.8: Επιλογή της έκδοσης windows για να κατεβάσουμε το πρόγραμμα XAMPP.

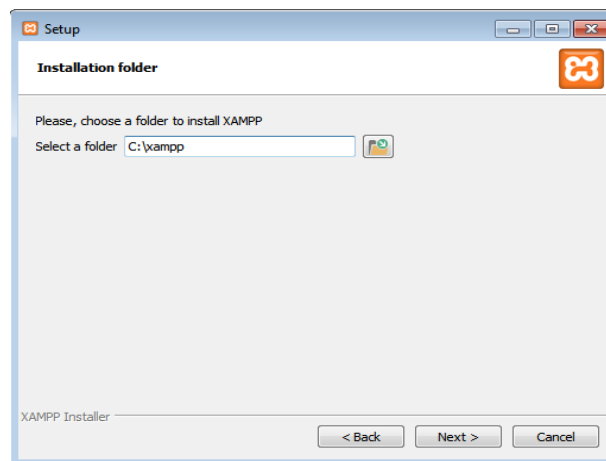
- ✓ Επιλέγουμε “αποθήκευση αρχείου”. Εκτελούμε το αρχείο και στην συνέχεια κάνουμε κλικ στο επόμενο (Next).



- ✓ Σημειώνουμε με τικ τα εργαλεία που θέλουμε να εγκαταστήσουμε και πατάμε Επόμενο (Next).



- ✓ Ορίζουμε τον φάκελο που θα γίνει η εγκατάσταση του προγράμματος μας και επιλέγουμε “Επόμενο” (Next).




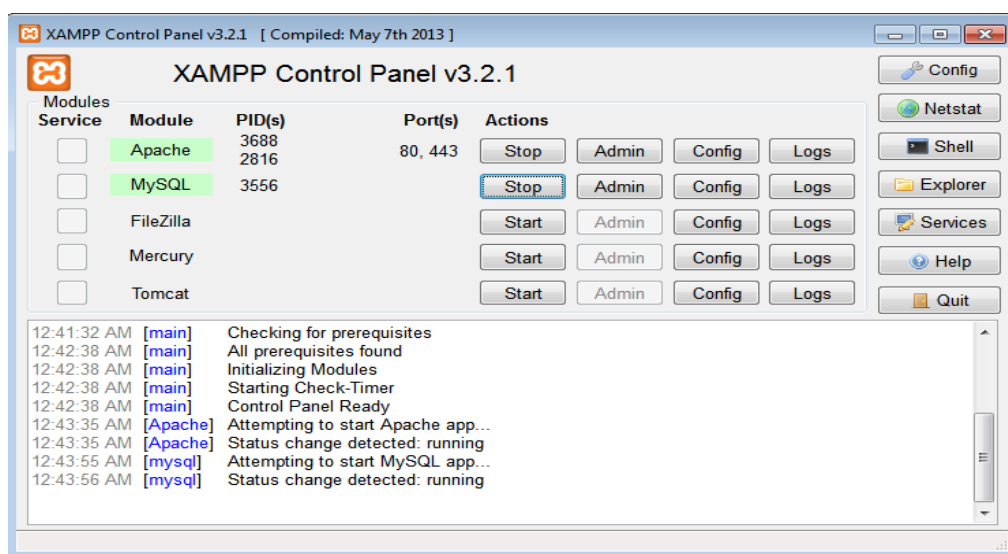
- ✓ Στην συνέχεια για τα επόμενα παράθυρα επιλέγουμε “Επόμενο” μέχρι να δούμε το παράθυρο που απεικονίζεται στην παρακάτω εικόνα. Όταν η πράσινη μπάρα γεμίσει επιλέγουμε το “Τέλος” που θα μας εμφανίσει και η εγκατάσταση έχει ολοκληρωθεί.



3.2.2 Φιλοξενία της σελίδα μας στον Εξυπηρετητή ΧΑΜΡΡ.

Για να μπορέσουμε να φιλοξενήσουμε την σελίδα μας στο πρόγραμμα ΧΑΜΡΡ πρέπει να βρούμε τον φάκελο “xampp” στην θέση που έχει οριστεί κατά την εγκατάσταση και μέσα στον φάκελο αυτό βρίσκουμε τον φάκελο “htdocs”. Στον “htdocs” φάκελο εισάγουμε όλα τα αρχεία που θέλουμε να φιλοξενήσει ο Εξυπηρετητής. Στην συνέχεια

ανοίγουμε το πρόγραμμα του ΧΑΜΡΡ  και ενεργοποιούμε πατώντας το κουμπί “Εναρξη” τα εργαλεία που θέλουμε να χρησιμοποιήσουμε στον εξυπηρετητή μας. Από την στιγμή που ενεργοποιηθούν οι πόρτες επικοινωνίας εμφανίζονται στην στήλη με την ονομασία “Ports” όπως φαίνεται και στην Εικ.3.5.



Εικόνα 3.9: Απεικόνιση του περιβάλλοντος Ανάπτυξης του ΧΑΜΡΡ

Ανοίγουμε τον φυλλομετρητή (browser) στον υπολογιστή μας, εισάγουμε την διεύθυνση “http://localhost/” και δίπλα το όνομα του αρχείου έχει η σελίδα. Για παράδειγμα για το αρχείο “test1.php” πρέπει να γράψουμε “localhost/test1.php” σε περίπτωση που είναι μέσα σε φάκελο η σελίδα μας βάζουμε στο ενδιάμεσο και το όνομα του φακέλου “localhost/folder/test1.php”.

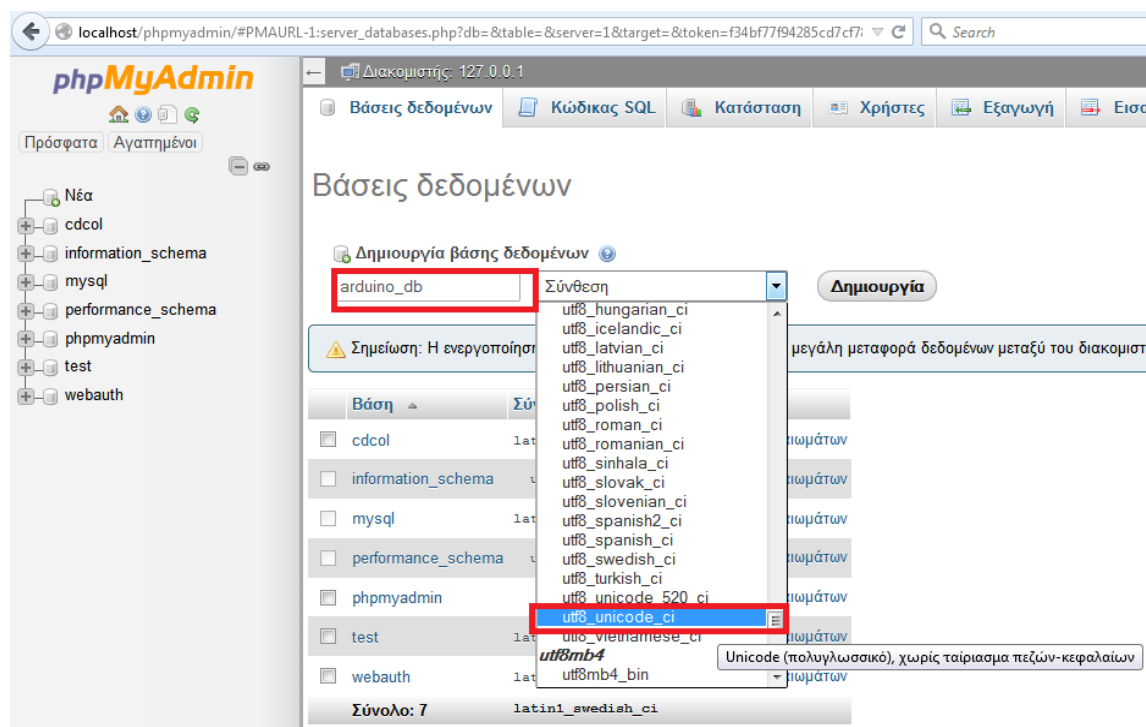
3.2.3 Δημιουργία Βάσης Δεδομένων στο ΧΑΜΡΡ.

Η MySQL είναι μια πολύ γρήγορη σε απόδοση και ισχυρή σε δυνατότητες υλοποίηση, ενός συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων. Τα δεδομένα στη MySQL αποθηκεύονται σε αντικείμενα βάσης δεδομένων, τα οποία ονομάζονται πίνακες. Ένας πίνακας, είναι μια συλλογή από σχετικές καταχωρήσεις δεδομένων και αποτελείται από στήλες και γραμμές. Οι βάσεις δεδομένων είναι χρήσιμες για την αποθήκευση πληροφοριών σε κατηγορίες.



Εικόνα 3.10: Το λογότυπο της σχεσιακής βάσης δεδομένων MySQL.

Υπάρχουν δύο τρόποι να δημιουργήσουμε μία βάση δεδομένων: Πρώτον δίνοντας την εντολή “CREATE databse” και “Το όνομα της βάσης ”μέσα από την γραμμή εντολών. Και δεύτερον μέσα από το εργαλείο phpMyAdmin. Για να δημιουργήσουμε βάση δεδομένων με την δεύτερη μέθοδο θα πρέπει να ανοίξουμε ένα φυλλομετρητή και να εισάγουμε την διεύθυνση <http://localhost/phpmyadmin>. Στη συνέχεια εισάγουμε το όνομα και τον τύπο στα σημεία της Εικ.3.11 που είναι υπογραμμισμένα με κόκκινο χρώμα και πατάμε το κουμπί “Δημιουργία”. Αν όλα έχουν γίνει σωστά θα μας εμφανίσει ένα μήνυμα “ Δημιουργήθηκε η βάση δεδομένων <<το όνομα της Βάσης>>”



Εικόνα 3.11: Δημιουργία Βάσης δεδομένων στο πρόγραμμα XAMPP.

3.3 Λογισμικό Notepad++

Πρόκειται για τον πιο γνωστό επεξεργαστή κειμένου (text editor). Είναι το αγαπημένο εργαλείο συγγραφής κώδικα των προγραμματιστών, καθώς αποτελεί ένα ελαφρύ πρόγραμμα το οποίο όμως έχει πάρα πολλές δυνατότητες.

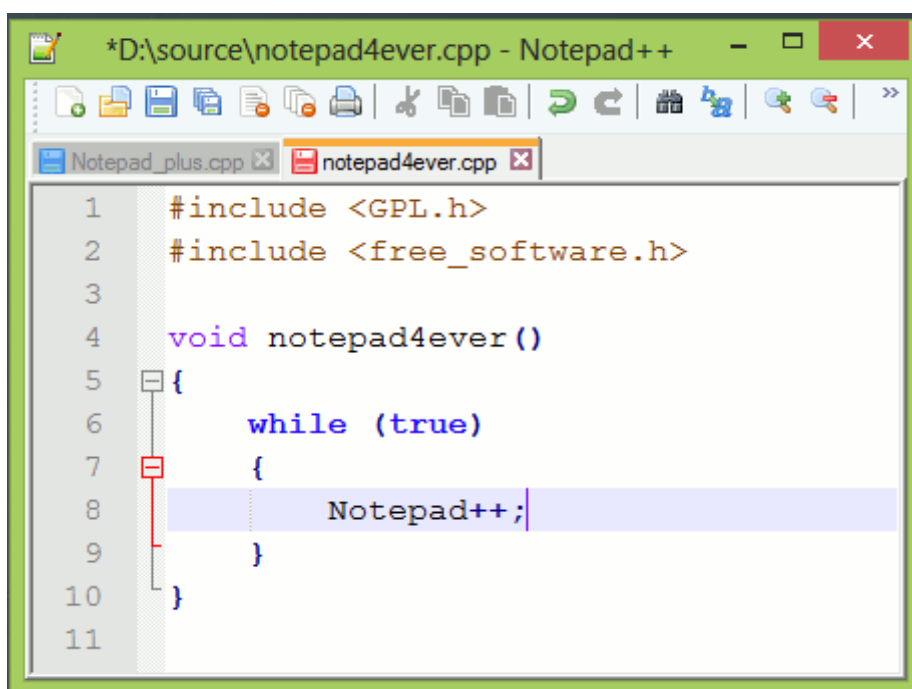


Εικόνα 3.12: Ένα από τα λογότυπα του προγράμματος Notepad++

Μπορούμε να το χρησιμοποιήσουμε και σαν επεξεργαστή πηγαίου κώδικα για να γράψουμε και να επεξεργαστούμε κώδικα σε PHP, HTML, C και πλήθος γλωσσών.

3.3.1 Περιβάλλον εργασίας.

Το περιβάλλον εργασίας είναι εξαιρετικά απλό και λειτουργικό. Στο επάνω μέρος του υπάρχει μία εργαλειοθήκη με τις κυριότερες λειτουργίες. Όλο το υπόλοιπο μέρος του καταλαμβάνεται από τον χώρο συγγραφής του κώδικα. Γύρω από τον χώρο αυτόν, μπορούν να εμφανιστούν διάφορα πλαίσια τα οποία παρουσιάζουν διάφορες πληροφορίες ανάλογα με τα πρόσθετα παράθυρα (plugins) από τα οποία προέρχονται.



```
1  #include <GPL.h>
2  #include <free_software.h>
3
4  void notepad4ever ()
5  {
6      while (true)
7      {
8          Notepad++;
9      }
10 }
11
```

Εικόνα 3.13: Το περιβάλλον ανάπτυξης του προγράμματος Notepad++.

3.3.2 Χαρακτηριστικά και λειτουργίες του Notepad++ .

Το σημαντικό του χαρακτηριστικό είναι ο φωτεινός χρωματισμός “ highlighting”, το οποίο χρωματίζει τον κώδικα ανάλογα με την δομή του και διευκολύνει την ανάπτυξή του. Για κάθε γλώσσα προγραμματισμού υπάρχει διαφορετικό χρωματικό στυλ. Το πρόγραμμα έχει έτοιμα στυλ για πολλές γλώσσες και δίνει την δυνατότητα να τα ρυθμίσουμε σύμφωνα με τις ανάγκες μας ή και να δημιουργήσουμε καινούρια.

Μπορούμε να επεξεργαστούμε πολλά έγγραφα συγχρόνως τα οποία ανοίγουν σε ξεχωριστές καρτέλες έτσι ώστε να επεξεργαζόμαστε τα αρχεία μια ολόκληρης εργασίας. Επιπλέον μπορούμε να βλέπουμε και να επεξεργαζόμαστε δύο έγγραφα συγχρόνως.

Πολύ δυνατό χαρακτηριστικό του προγράμματος είναι οι εκτεταμένες δυνατότητες εύρεσης κειμένου στο αρχείο που ήδη δουλεύετε ή στα αρχεία ενός ολόκληρου φακέλου. Το πρόγραμμα υποστηρίζει την πρόσθεση αυτόνομων προγραμμάτων, τα οποία επεκτείνουν τις δυνατότητές του. Υπάρχουν πολλά τέτοια προγράμματα που μπορούμε να κατεβάσουμε και να στήσουμε ένα πρόγραμμα σύμφωνα με τις ανάγκες

μας. Επιπλέον τις εργασίες που εκτελούνται συχνά μπορούμε να τις αυτοματοποιήσουμε με την χρήση μακροεντολών.

3.3.3 Πληροφορίες εγκατάστασης.

Το πρόγραμμα του Notepad++ είναι δωρεάν και στην σελίδα <https://notepad-plus-plus.org/download/v6.7.8.1.html> μπορούμε να κατεβάσουμε την τελευταία ενημερωμένη έκδοση. Το μέγεθός του είναι εξαιρετικά μικρό και οι απαιτήσεις του ελάχιστες. Για να το εγκαταστήσουμε απλά ακολουθούμε τα βήματα του αντίστοιχου οδηγού. Μπορούμε να το χρησιμοποιήσουμε και σαν φορητό πρόγραμμα, τοποθετώντας το σε ένα USB αποθηκευτικό χώρο και να το εκτελέσουμε σε οποιονδήποτε υπολογιστή. Για τον σκοπό αυτό είναι προτιμότερο το αντίστοιχο zip αρχείο. Τέλος, υποστηρίζεται η χρήση της ελληνικής γλώσσας στα μενού και στα πλαίσια διαλόγου του προγράμματος

4.Υλικό μέρος του συστήματος

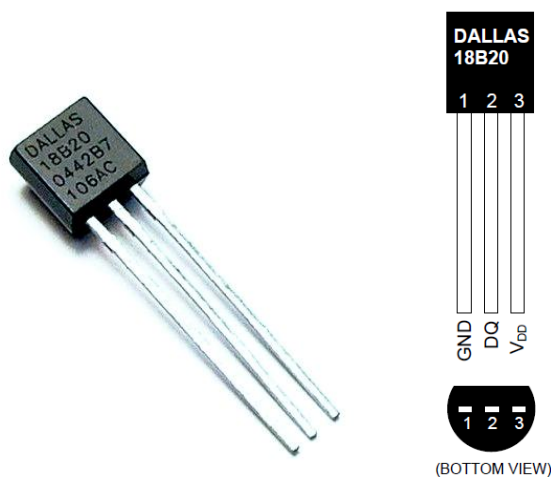
Στο κεφάλαιο αυτό θα αναλύσουμε τα υλικά μέρη του συστήματος. Βασικό μέρος του συστήματος είναι η πλατφόρμα Arduino στην οποία έχουμε αναφερθεί και στο κεφάλαιο 2. Η πλακέτα Arduino μας επιτρέπει την σύνδεση διάφορων εξαρτημάτων από τα οποία είτε λαμβάνει κάποια πληροφορία (αισθητήρες) είτε κάνει έλεγχο σε αυτά (συσκευές). Το Arduino οργανώνει τις λειτουργίες και τα εξαρτήματα υπακούουν σε αυτό. Τα υλικά μέρη που συνδέονται με το Arduino είναι:

- Αισθητήρας θερμοκρασίας
- Αισθητήρας υπερήχων
- Δίοδος φωτεινής ένδειξης τύπου LED
- Ηλεκτρονόμος (ρελέ)
- Ασπίδα GSM/GPRS.
- Λοιπά εξαρτήματα (κουμπιά, αντιστάσεις, σταθεροποιητής τάσης, κλέμες, καλώδια κ.α)

Το Arduino συλλέγει δεδομένα από τους αισθητήρες θερμοκρασίας, υπερήχων και από τις εισόδους και αλληλεπιδρά ανάλογα με αυτά τα δεδομένα στα LED και στις εξόδους των ηλεκτρονόμων. Επιπλέον με την βοήθεια της ασπίδας GSM/GPRS συνδέεται στο Διαδίκτυο και συμβάλει στην επικοινωνία με τον εξυπηρετητή (Server) για τον απομακρυσμένο έλεγχο και την επιτήρηση της κατάστασης του Arduino μέσω ιστοσελίδας.

4.1 Μέτρηση θερμοκρασίας με τον αισθητήρα DS18B20.

Για την μέτρηση της θερμοκρασίας χρησιμοποιούμε τον αισθητήρα θερμοκρασίας της εταιρείας Dallas DS18B20. Ο συγκεκριμένος τύπος αισθητήρα υπάγεται στους ψηφιακούς αισθητήρες και χρησιμοποιεί το πρωτόκολλο επικοινωνίας 1-wire με το οποίο απαιτείται μόνο ένα καλώδιο για την μεταφορά δεδομένων. Παρέχει μετρήσεις στην κλίμακα Κελσίου εύρους 9-12 δυαδικών ψηφίων (bit) και δυνατότητα λειτουργίας συναγερμού ορίζοντας μέγιστο και ελάχιστο σημείο πυροδότησης. Μετράει θερμοκρασίες από -55°C έως $+125^{\circ}\text{C}$ με ακρίβεια $\pm 0.5^{\circ}\text{C}$ σε εύρος -10°C έως $+85^{\circ}\text{C}$. Κάθε αισθητήρας έχει έναν μοναδικό σειριακό κωδικό 64 δυαδικών ψηφίων (64 bits) ο οποίος είναι ορισμένος από το εργοστάσιο κατασκευής του και έτσι μπορούμε στο ίδιο καλώδιο επικοινωνίας να έχουμε συνδεδεμένους και άλλους αισθητήρες. Με τον τρόπο αυτό, με την χρήση ενός επεξεργαστή μπορούμε να ελέγξουμε πολλούς αισθητήρες ταυτόχρονα, χρησιμοποιώντας μία μόνο γραμμή επικοινωνίας.



Εικόνα 4.1 : Αισθητήριο θερμοκρασίας DS18B20

Το ολοκληρωμένο DS18B20 έχει τρεις ακροδέκτες

- VDD: Τάση τροφοδοσίας 3 - 5.5V.
- DQ: Ακροδέκτης επικοινωνίας 1-wire .
- GND: Δυναμικό εδάφους.

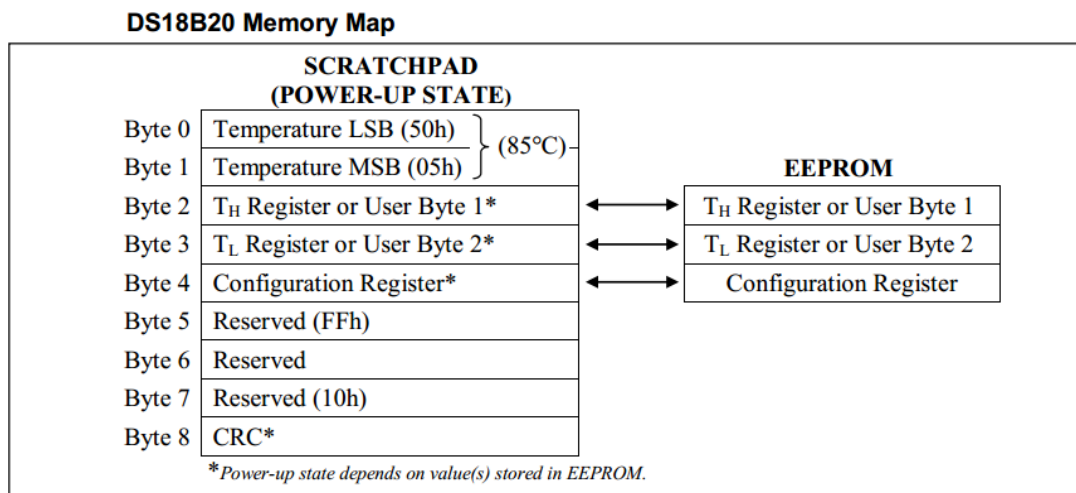
Εύρος μέτρησης θερμοκρασίας	-55°C έως +125°C
Τάση λειτουργίας	3 - 5.5V
Ακρίβεια	±0.5°C σε εύρος -10°C έως +85°C
Προτόκωλλο επικοινωνίας	1 –Wire (Μόνο ένας αγωγός επικοινωνίας).
Ειδική λειτουργία συναγερμού.	Ενεργοποίηση συναγερμού σε υψηλή και χαμηλή προγραμματιζόμενη θερμοκρασία.
Μέγιστος χρόνος μέτρησης θερμοκρασίας.	750 ms για ακρίβεια 12bit.
Ρυθμιζόμενη ακρίβεια	Από 9-12bit.

Πίνακας 4.1 Τεχνικά χαρακτηριστικά του αισθητήρα θερμοκρασίας DS18B20

4.1.1 Περιγραφή λειτουργίας του αισθητήρα DS18B20.

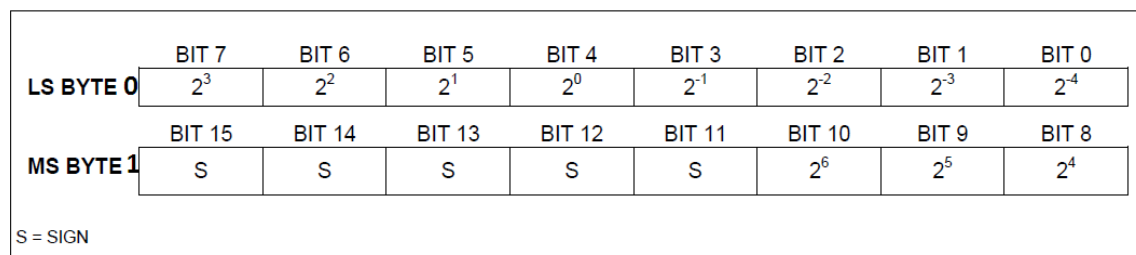
Η τεχνολογία 1-wire μας δίνει την δυνατότητα να συνδέσουμε πολλαπλούς αισθητήρες σε μία μόνο γραμμή επικοινωνίας για την μεταφορά δεδομένων. Μπορούμε να συνδέσουμε πάνω από 127 αισθητήρες σε μία μόνο γραμμή. Έχουμε μοντέλο επικοινωνίας αφέντη-σκλάβου (master-slave) όπου κάθε αισθητήριο που έχει τον ρόλο του σκλάβου (slave) υπακούει στις εντολές του αφέντη (master) ο οποίος

συνήθως είναι ένας μικροελεγκτής όπως το Arduino. Για να πάρουμε την μέτρηση θερμοκρασίας από έναν αισθητήρα θα πρέπει αρχικά να στείλει το Arduino έναν παλμό επαναφοράς “reset” και στην συνέχεια τα αισθητήρια που βρίσκονται στην ίδια γραμμή απαντάνε με έναν παλμό παρουσίας “presence”. Το Arduino ψάχνει για τον μοναδικό 64-μπιτο σειριακό αριθμό όπου χρησιμεύει και σαν διεύθυνση για τον εντοπισμό ενός αισθητήρα. Στην συνέχεια επιλέγει τον αισθητήρα που επιθυμεί για μέτρηση στέλνοντας την εντολή MATCH ROM [55h]. Για να πάρουμε την μέτρηση της θερμοκρασίας το Arduino στέλνει την εντολή μετατροπής CONVERT[44h] και ξεκινάει η διαδικασία μετατροπής θερμοκρασίας όταν η διαδικασία ολοκληρωθεί η θερμοκρασία αποθηκεύεται σε 2 ψηφιολέξεις (2 bytes) της μνήμης τύπου SRAM (scratchpad) του αισθητήρα. Η διαδικασία μετατροπής είναι αρκετά χρονοβόρα και επηρεάζεται άμεσα με την ακρίβεια της μετρήσης που έχουμε ορίσει, για 12 δυαδικά ψηφία είναι 750ms. Το αποτέλεσμα της μέτρησης αποθηκεύεται στις δυο πρώτες ψηφιολέξεις (byte0-byte1) της συνολικής μνήμης SRAM (scratchpad).



Σχήμα 4.1 : Διάγραμμα της μνήμης SRAM του αισθητήρα DS18B20.

Στην συνέχεια στέλνοντας την εντολή READ SCRATCHPAD[BEh] λαμβάνουμε από τον αισθητήρα και τα 9byte(byte0-byte8) της μνήμης SCRATCHPAD όπου και η μέτρηση της θερμοκρασίας βρίσκεται στα 2 πρώτα byte όπως φαίνεται και στο παρακάτω σχήμα.



Σχήμα 4.2 : Αναπαράσταση των 2 byte θερμοκρασίας.

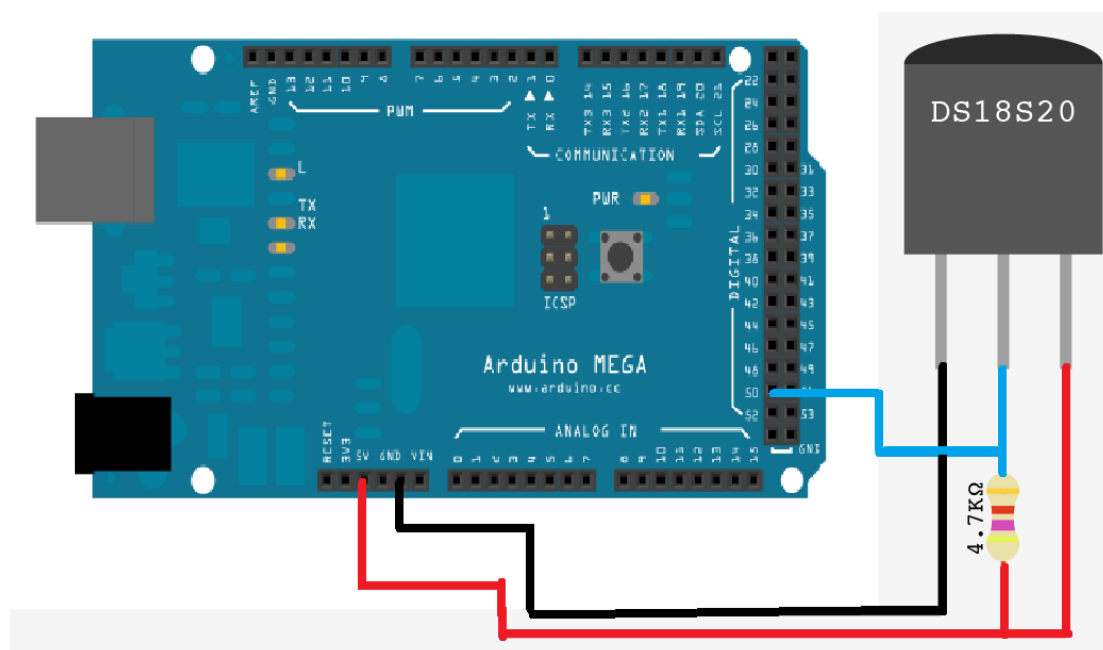
Για να υπολογίσουμε την θερμοκρασία πολλαπλασιάζουμε τον προσημασμένο 16 bit αριθμό με την ακρίβεια μέτρησης που έχουμε ορίσει. Η ακρίβεια για 9, 10, 11 ή 12bit είναι 0.5°C, 0.25°C, 0.125°C, ή 0.0625°C αντίστοιχα.

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

Πίνακας 4.2 Αντιστοίχιση θερμοκρασίας για ακρίβεια 12bit του αισθητήρα DS18B20.

4.1.2 Σύνδεση του αισθητήρα θερμοκρασίας με το Arduino

Συνδέουμε τον ακροδέκτη επικοινωνίας “DQ” του αισθητήρα θερμοκρασίας με ένα από τους ψηφιακούς ακροδέκτες που διαθέτει το Arduino Mega(Ακρ.50). Για την σωστή λειτουργία χρειάζεται επίσης να συνδέσουμε μία αντίσταση (pull-up) στην γραμμή επικοινωνίας 1-Wire με την τάση τροφοδοσίας, κρατώντας το επίπεδο τάσης σταθερό σε υψηλό δυναμικό όταν δεν ανταλλάζουμε πληροφορίες. Οι άλλοι δύο ακροδέκτες του αισθητήρα (VDD και GND) συνδέονται με τους ακροδέκτες τροφοδοσίας του Arduino 5V και GND αντίστοιχα βλέπε εικ.4.3.



Εικόνα 4.2 : Σύνδεση αισθητήρα θερμοκρασίας με το Arduino

4.1.3 Κώδικας μέτρησης θερμοκρασίας

Για την επικοινωνία του αισθητήρα DS18B20 με το Arduino Θα χρησιμοποιήσουμε την βιβλιοθήκη “OneWire.h” η οποία χρησιμοποιεί το πρωτόκολλο επικοινωνίας 1-wire και χρησιμοποιεί διάφορες συναρτήσεις όπως αναφέρονται παρακάτω:

OneWire oneWire(pin); : Δίνουμε ένα όνομα “oneWire” και στην συνέχεια ορίζουμε τον ακροδέκτη που θα χρησιμοποιηθεί σν δίαυλος επικοινωνίας.

oneWire.search(AddrArray); : Ψάχνουμε για την επόμενη συσκευή που βρίσκεται στον δίαυλο επικοινωνίας. Αν βρεθεί νέα συσκευή τότε αποθηκεύεται στον πίνακα AddrArray 8 θέσεων. Επιστρέφει ψευδής σε περίπτωση που δεν υπάρχει άλλη συσκευή.

oneWire.reset(); : Κάνει επανεκκίνηση στον δίαυλο επικοινωνίας. Συνήθως χρειάζεται πριν ξεκινήσει να επικοινωνεί με κάποια συσκευή.

oneWire.select(AddrArray); : Επιλέγει την συσκευή σύμφωνα με την διεύθυνση του που βρίσκεται στον πίνακα διεύθυνσης AddrArray 8 θέσεων.

oneWire.write(num); : Γράφει ένα byte στον δίαυλο επικοινωνίας.

oneWire.read(); : Διαβάζει ένα byte.

oneWire.crc8(dataArray,length); : Χρησιμοποιείται για την επιβεβαίωση αποστολής δεδομένων χωρίς σφάλμα.

Για την ευκολία στον προγραμματισμό θα δημιουργήσουμε μία συνάρτηση όπου θα υπολογίζει την θερμοκρασία και κάθε φορά που θέλουμε να πάρουμε την θερμοκρασία καλούμε την συνάρτηση getTemp();

```
#include <OneWire.h>
OneWire oneWire (50);
// Δηλώνουμε τον ακροδέκτη 50 σαν δίαυλο επικοινωνίας 1-wire του
// Arduino.
void setup(void) {
  Serial.begin(9600);
}
void getTemp (){
  byte i;
  byte present = 0;
  byte data[12]; //Αποθηκεύουμε τα δεδομένα της SRAM του αισθητήρα.
  byte addr[8]; //Αποθηκεύουμε τον 64bit κωδικό του αισθητήρα
  float celsius;
  oneWire.reset(); //Στέλνουμε ένα παλμό επαναφοράς.
  oneWire.write(0xcc); /*Στέλνουμε την εντολή SKIP ROM[CCh] με την οποία
  επικοινωνούμε απευθείας με τον αισθητήρα χωρίς να χρειάζεται να
  ξέρουμε τον 64bit κωδικό. Αυτή η εντολή χρησιμοποιείται στην
  περίπτωση που έχουμε μόνο ένα αισθητήρα συνδεδεμένο.*/
```



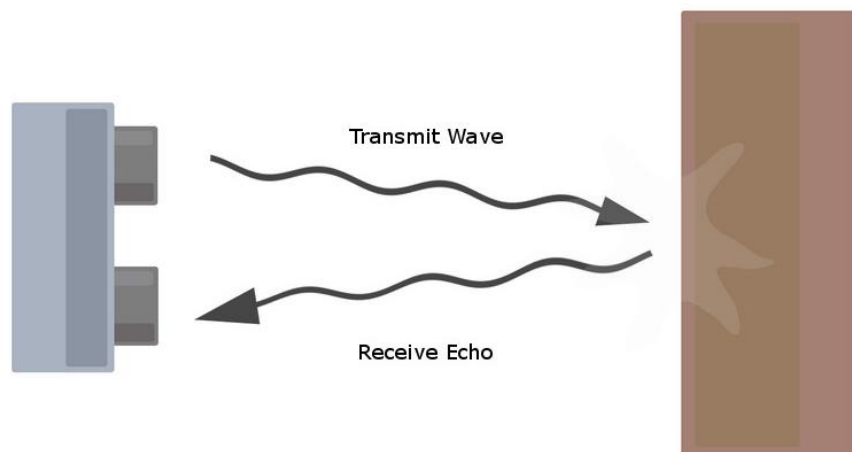
```

oneWire.write(0x44); // Ξεκινάμε διαδικασία μετατροπής σε
θερμοκρασία
delay(1000); //ελάχιστος χρόνος 750ms για ακρίβεια 12bit
present = oneWire.reset();
oneWire.write(0xcc);
oneWire.write(0xBE); //Ζητάμε να διαβάσουμε τα δεδομένα στις SRAM.
for ( i = 0; i < 9; i++) { // χρειαζόμαστε 9 Bytes πληροφορίας.
    data[i] = oneWire.read(); //Διαβάζουμε και αποθηκεύουμε κάθε Byte.
}
// μετατροπή των δεδομένων σε πραγματική θερμοκρασία.
unsigned int raw = (data[1] << 8) | data[0]; //Παίρνουμε την πληροφορία της
θερμοκρασίας.
byte cfg = (data[4] & 0x60); //Διάβαζουμε την ακρίβεια που έχουμε ορίσει.
if (cfg == 0x00){
    raw = raw >> 3; celsius = (float)raw / 2.0; Serial.println("conf:9bit"); } //
για ακρίβεια 9 bit με μέγιστο χρόνο 93.75 ms
else if (cfg == 0x20) {
    raw = raw >> 2; celsius = (float)raw / 4.0; Serial.println("conf:10bit"); }
//για ακρίβεια 10bit με μέγιστο χρόνο 187.5 ms.
else if (cfg == 0x40){
    raw = raw >> 1; celsius = (float)raw / 8.0; Serial.println("conf:11bit"); } //
για ακρίβεια 11bit με μέγιστο χρόνο 375 ms.
else celsius = (float)raw / 16.0;
// προκαθορισμένη ακρίβεια 12bit και μέγιστο χρόνο μετατροπής
750ms.
//Εμφάνιση της θερμοκρασίας στην σειριακή οθόνη.
Serial.print(" Temperature = ");
Serial.print(celsius);
Serial.print(" Celsius, ");
}

```

4.2 Μέτρηση στάθμης υγρού με τον αισθητήρα υπερήχων HC-SR04.

Για την μέτρηση της απόστασης χρησιμοποιούμε τον αισθητήρα υπερήχων. Επιλέξαμε τον αισθητήρα αυτό έναντι άλλων αισθητήρων γιατί μας επιτρέπει την μέτρηση χωρίς επαφή με το μετρούμενο αντικείμενο. Κι αυτό το κάνει ιδανικό αισθητήριο για διάφορες εφαρμογές όπως την μέτρηση υγρών σε δεξαμενές, τον εντοπισμό αντικειμένου σε ένα κινούμενο ρομποτικό σύστημα, σαν μετρητής αντικειμένων σε μία πόρτα καταστήματος και άλλες εφαρμογές.



Εικόνα 4.3 : Αναπαράσταση διάδοσης και ανάκλασης κύματος υπερήχων.

Υπέρηχος είναι ένα μηχανικό κύμα όπως ο ήχος, υψηλής συχνότητας την οποία δεν μπορεί να την αντιληφθεί το ανθρώπινο αυτί. Για να μεταδοθεί χρησιμοποιεί ένα μέσο μετάδοσης πχ το αέρα. Αυτή η τεχνολογία βρίσκει εφαρμογή στην μέτρηση της απόστασης χωρίς επαφή. Ο πομπός δημιουργεί και στέλνει ένα κύμα υπερήχων το οποίο ταξιδεύει με ταχύτητα ανάλογη με το μέσο διάδοσης του (π.χ : στον αέρα η ταχύτητα είναι 324 m/s). Το κύμα αυτό έχει τις φυσικές ιδιότητες του ήχου έτσι αν συναντήσει ένα εμπόδιο το κύμα ανακλάται και επιστρέφει πίσω. Στην θέση που είναι ο πομπός υπάρχει και ένα άλλο όργανο, ο δέκτης, ο οποίος αντιλαμβάνεται διάφορες πιέσεις και τις μεταφράζει σε μία τάση με την χρήση πιεζοηλεκτρικών στοιχείων. Ο δέκτης μπορεί και αντιλαμβάνεται ένα κύμα υπέρηχου. Η απόσταση του αντικειμένου που δημιούργησε την ανάκλαση του υπέρηχου μπορεί να υπολογιστεί, εφόσον γνωρίζουμε την ταχύτητα μετάδοσης, μετρώντας τον χρόνο που έκανε το σήμα από την στιγμή της μετάδοσης μέχρι και της λήψης, διαιρώντας το δια δύο και πολλαπλασιάζοντας το με την ταχύτητά του. Παίρνουμε το μισό του συνολικού χρόνου γιατί το σήμα διανύει την διπλάσια απόσταση από αυτή που θέλουμε να υπολογίσουμε μέχρι να επιστρέψει πίσω. Ο τύπος υπολογισμού απόστασης είναι:

$$\text{Απόσταση} = (\text{Συνολικός χρόνος} * \text{ταχύτητα διάδοσης ήχου})/2.$$

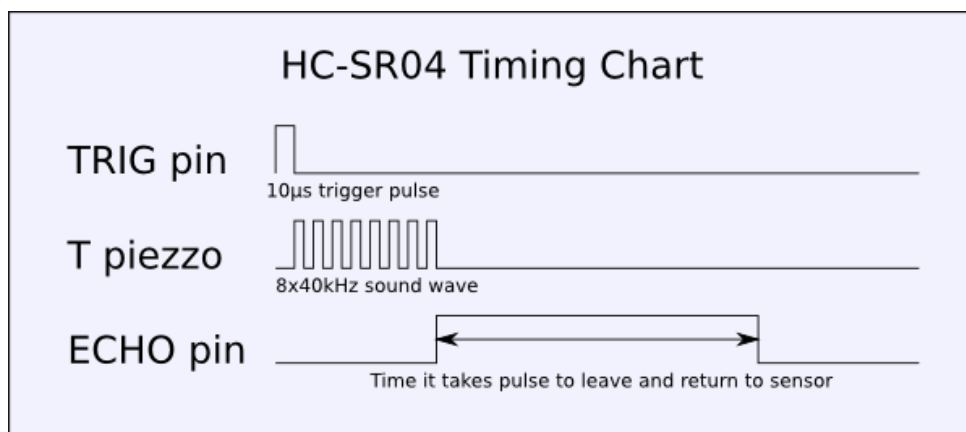


Εικόνα 4.4 : Οι δύο όψεις της μονάδας υπερήχων HCSR-04

4.2.1 Η μονάδα υπερήχων HCSR-04

Η μονάδα υπερήχων που χρησιμοποιήσαμε για την μέτρηση στάθμης σε υγρό περιλαμβάνει ένα πομπό υπερήχων ένα δέκτη και ένα κύκλωμα ελέγχου. Η αρχή λειτουργίας του είναι η εξής:

- 1) Στέλνουμε παλμό διάρκειας 10μs στον ακροδέκτη πυροδότησης (trigger) της μονάδας.
- 2) Η μονάδα από την στιγμή που λάβει τον παλμό εκπέμπει αυτόματα 8 παλμούς των 40Khz μέσω του πομπού και θέτει τον ακροδέκτη “echo” της μονάδας σε υψηλό επίπεδο “HIGH”.
- 3) Ο δέκτης περιμένει να επιστρέψει ο παλμός. Εάν το σήμα βρει κάποιο εμπόδιο και επιστρέψει, ο δέκτης αντιλαμβάνεται το σήμα που επέστρεψε. Την στιγμή που διαβάσει την επιστροφή του παλμού ο δέκτης της μονάδας, αυτόματα θέτει τον ακροδέκτη “echo” σε χαμηλό επίπεδο “LOW”. Ο χρόνος που παρέμεινε σε υψηλό επίπεδο ο ακροδέκτης “echo” αντιστοιχεί στον συνολικό χρόνο που έκανε για να μεταδοθεί και να επιστρέψει πίσω ένα κύμα υπερήχων.



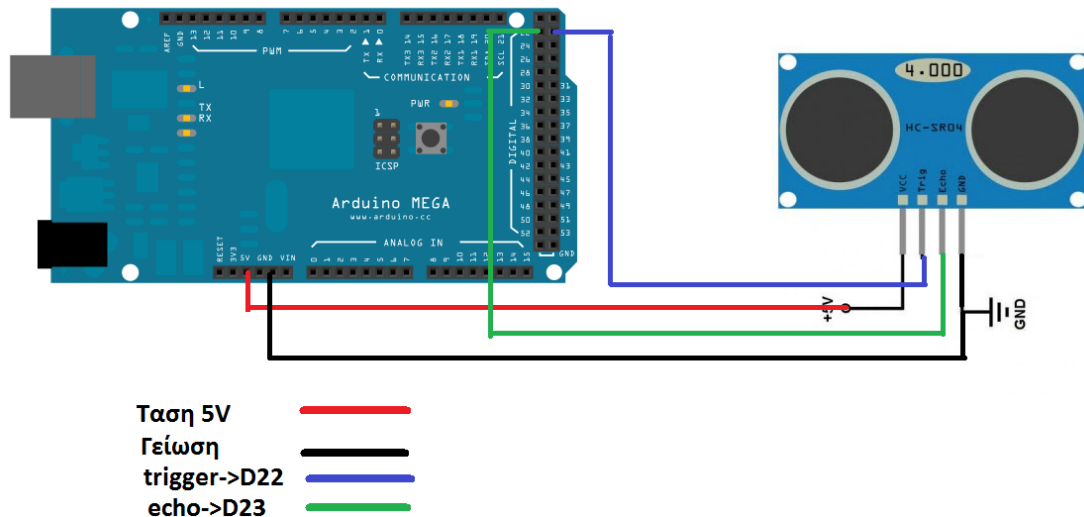
Εικόνα 4.5 : Χρονικό διάγραμμα για την μέτρηση του χρόνου μετάδοσης και λήψης

Τάση Λειτουργίας	DC 5V
Ρεύμα Λειτουργίας	15 mA
Συχνότητα Λειτουργίας	40 Khz
Μέγιστη Απόσταση	4m
Ελάχιστη Απόσταση	2cm
Γωνία Μέτρησης	15 μοίρες
Κανάλι Εισόδου "trigger"	Παλμός TTL 10µs
Κανάλι Εξόδου "echo"	Κανάλι εισόδου TTL
Διαστάσεις Πλακέτας	45x20x10

Πίνακας 4.3 Χαρακτηριστικά της μονάδας υπερήχων HC-SR04.

4.2.2 Σύνδεση της μονάδας υπερήχων με το Arduino

Η μονάδα υπερήχων διαθέτει τέσσερις ακροδέκτες οι οποίοι μπορούν να κουμπώσουν πάνω στους ακροδέκτες του Arduino. Συνδέουμε τους ακροδέκτες VCC και GND της μονάδας με τους ακροδέκτες 5V και GND του Arduino. Οι ακροδέκτες πυροδότησης "trigger" και ηχώ "echo" της μονάδας με τους ακροδέκτες 40 και 42 του Arduino. Η μονάδα έχει χαμηλές απαιτήσεις ρεύματος και έτσι το Arduino μπορεί να τροφοδοτήσει την μονάδα χωρίς κάποιο πρόβλημα.



Εικόνα 4.6 Σύνδεση του αισθητήρα υπερήχων με το Arduino.

4.2.3 Κώδικας μέτρησης στάθμης

Θα πρέπει να ορίσουμε τους ακροδέκτες 22 και 23 του Arduino ως είσοδο και έξοδο αντίστοιχα. Θα δημιουργήσουμε μία συνάρτηση όπου όταν την καλούμε θα επιστρέφει την απόσταση του μετρούμενου αντικειμένου σε εκατοστά του μέτρου (cm). Η συνάρτηση pulseIn(); Επιστρέφει τον χρόνο που παρέμεινε σε μία κατάσταση υψηλό-χαμηλό δυναμικό το σήμα στην είσοδο του ακροδέκτη. Ο χρόνος αυτός είναι σε ms και αντιστοιχεί στο συνολικό χρόνο που χρειάστηκε το κύμα υπερήχων μέχρι να επιστρέψει πίσω στην πηγή που το δημιούργησε. Η συνάρτηση pulseIn (echoPin,val) έχει δύο ορίσματα στο όρισμα echoPin: γράφουμε τον ακροδέκτη του Arduino που είναι συνδεδεμένος με τον ακροδέκτη "echo" και στο όρισμα val: δίνουμε την τιμή της κατάστασης που ενεργοποιεί τον μετρητή. Οι δυνατές τιμές είναι δύο HIGH για ενεργοποίηση στο υψηλό δυναμικό 5V και LOW στο χαμηλό δυναμικό 0V

```

#define echoPin 23;
#define triggerPin 22;
// Ορίζουμε τις μεταβλητές duration 4byte
long duration ,distance;

void setup(){
// Αρχικοποιούμε την σειριακή επικοινωνία σε ρυθμό μετάδοσης 9600 btu.
Serial.begin(9600);
pinMode(echoPin, INPUT);
pinMode(triggerPin,OUTPUT);
}
//Στην loop() δεν γράφουμε κώδικα απλά καλούμε την συνάρτηση που δημιουργήσαμε
void loop()
{
takeDistance();
delay(100);
}

```

```
long takeDistance()
{
  digitalWrite(pingPin,LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin,HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin,LOW);
  // Παραπάνω δημιουργούμε έναν παλμό διάρκειας 5 ms

  //Μετατροπή σήματος σε χρόνο.
  duration=pulseIn(echoPin,HIGH);
  distance=(duration/29)/2;

  //Απεικόνιση της απόστασης στην σειριακή οθόνη.
  serial.print(cm);
  serial.print("cm");
  serial.println();
  return distance;
}
```

4.3 Έλεγχος συσκευών με την χρήση ηλεκτρονόμων (Relay).



Εικόνα 4.7 Δύο όψεις του ηλεκτρονόμου.

Η συσκευή μας ελέγχει τρεις συσκευές: έναν ανεμιστήρα 220V 20W, μία αντλία νερού 220V 70W και ένα λαμπτήρα πυρακτώσεως 220V 60W με την χρήση ηλεκτρονόμων. Το Arduino στην εργασία μας οδηγεί ηλεκτρονόμους με σκοπό τον έλεγχο συσκευών ισχύος. Ο ηλεκτρονόμος, (relay) είναι ένας ηλεκτρικός διακόπτης που ανοίγει και κλείνει ένα ηλεκτρικό κύκλωμα κάτω από τον έλεγχο ενός άλλου ηλεκτρικού κυκλώματος. Όταν ηλεκτρικό ρεύμα διαρρέει το πηνίο του ηλεκτρονόμου, το παραγόμενο μαγνητικό πεδίο έλκει έναν σπλισμό που είναι μηχανικά συνδεδεμένος σε μια κινούμενη επαφή. Έτσι, η κινούμενη επαφή είτε συνδέεται με μια σταθερή επαφή είτε αποσυνδέεται από τη σταθερή επαφή. Μόλις το ηλεκτρικό ρεύμα στο πηνίο διακοπεί, ο σπλισμός επιστέφει στη θέση ηρεμίας του εξαιτίας μιας δύναμης επαναφοράς, που είναι ίση με το ήμισυ της μαγνητικής

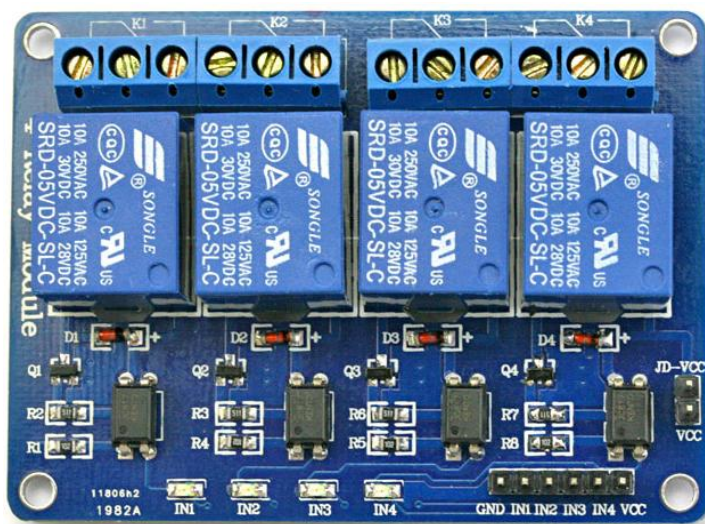
Κάθε επαφή ενός ηλεκτρονόμου μπορεί να είναι Κανονικά-Ανοικτή (Normally Open, NO), Κανονικά-Κλειστή (Normally Closed, NC) ή μεταγωγικός (change-over), ανάλογα με τον τύπο της.

- Μια επαφή **Κανονικά-Ανοικτή** συνδέει το κύκλωμα όταν ο ηλεκτρονόμος ενεργοποιείται το κύκλωμα αποσυνδέεται όταν ο ηλεκτρονόμος είναι ανενεργός. Μια τέτοια επαφή καλείται επίσης Επαφή Μορφής Α ή επαφή "make". Η επαφή μορφής Α είναι ιδανική για εφαρμογές που απαιτούν την ενεργοποίηση μιας πηγής υψηλής τάσης από απόσταση.
- Μια επαφή **Κανονικά-Κλειστή** αποσυνδέει το κύκλωμα όταν ο ηλεκτρονόμος ενεργοποιείται το κύκλωμα συνδέεται όταν ο ηλεκτρονόμος είναι ανενεργός. Μια τέτοια επαφή καλείται επίσης Επαφή Μορφής Β ή επαφή διακοπής "break". Η επαφή μορφής Β είναι ιδανική για εφαρμογές που απαιτούν το κύκλωμα να παραμένει κλειστό (ενεργό) μέχρι ο ηλεκτρονόμος να ενεργοποιηθεί.
- Μια επαφή **Μεταγωγική** μπορεί να ελέγχει δύο κυκλώματα. Ισοδυναμεί με μια επαφή κανονικά-ανοικτή και μια επαφή κανονικά-κλειστή που έχουν ένα κοινό ακροδέκτη. Μια τέτοια επαφή καλείται επίσης Επαφή Μορφής C.

4.3.1 Μονάδα ελέγχου τεσσάρων ηλεκτρονόμων με οδήγηση τάσης 5V.

Το μέγιστο Ρεύμα που μπορεί να περάσει από τον ακροδέκτη του Arduino είναι μόλις 40mA και η τάση του δεν ξεπερνάει τα 5V. Για τον λόγο αυτό είναι αδύνατο ένας μικροελεγκτής όπως και του Arduino να μπορέσει να ελέγξει συσκευές ισχύος χωρίς βοηθητικά εξαρτήματα. Ο ηλεκτρονόμος είναι ένα εξάρτημα που μας επιτρέπει με χαμηλή οδήγηση τάσης και ρεύματος να ελέγξουμε συσκευές ισχύος για παράδειγμα ένα φωτιστικό, μία γκαραζόπορτα, μία αντλία και πλήθος άλλων τέτοιων συσκευών. Παρόλο που απαιτήσεις οδήγησης ενός ηλεκτρονόμου είναι χαμηλές οι δυνατότητες τροφοδοσίας των ακροδεκτών του Arduino είναι ακόμα μικρότερες. Το Arduino αδυνατεί να οδηγήσει έναν ηλεκτρονόμο για τον λόγο αυτό χρησιμοποιούμε ακόμα ένα διακοπτικό κύκλωμα για την οδήγηση του ίδιου του ηλεκτρονόμου με ακόμα μικρότερες απαιτήσεις σε ρεύμα ικανές να οδηγηθούν από τον ακροδέκτη του Arduino. Τέτοια κυκλώματα είναι τρανζίστορ, MOSFET, FET, κ.α λειτουργώντας σαν αυτόματοι διακόπτες ή ενισχυτές ρεύματος και τάσης ανάλογα με την συνδεσμολογία τους.

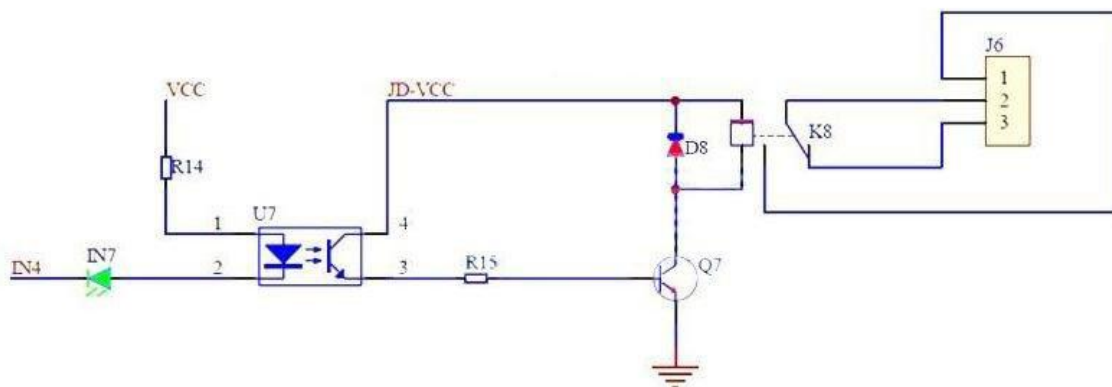
Χρησιμοποιώντας την μονάδα ηλεκτρονόμων μπορούμε να κάνουμε έλεγχο συσκευών ισχύος με ένα Arduino. Η μονάδα αυτή έχει ενσωματωμένα τέσσερα κυκλώματα οδήγησης ένα για το κάθε ηλεκτρονόμο και είναι τοποθετημένα πάνω σε μία πλακέτα.



Εικόνα 4.8 Πάνω όψη της μονάδας ηλεκτρονόμων της εργασίας.

Η πλακέτα έχει 6 ακροδέκτες με την σήμανση:

- Vcc : Εξωτερική τάση τροφοδοσίας 5V για την λειτουργία της πλακέτας
- IN1 : Ακροδέκτης οδήγησης του 1ου ηλεκτρονόμου .
- IN2 :Ακροδέκτης οδήγησης του 2ου ηλεκτρονόμου .
- IN3 :Ακροδέκτης οδήγησης του 3ου ηλεκτρονόμου .
- IN4 :Ακροδέκτης οδήγησης του 4ου ηλεκτρονόμου .
- Gnd : Ακροδέκτης γείωσης.



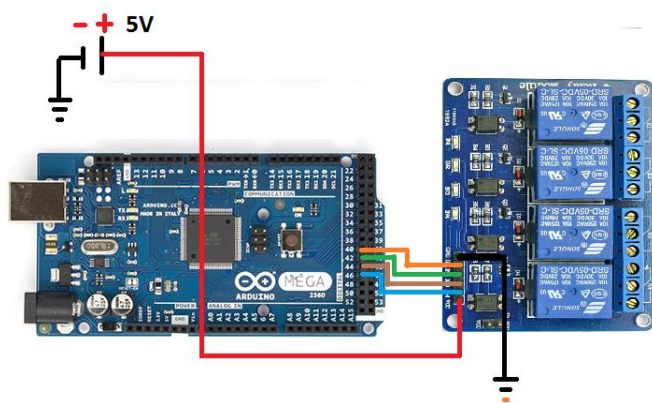
Εικόνα 4.9 Κύκλωμα οδήγησης ηλεκτρονόμου από τον ακροδέκτη του Arduino.

Η μονάδα ηλεκτρονόμων που χρησιμοποιήσαμε στην κατασκευή μας αποτελείται από τέσσερα ολοκληρωμένα κυκλώματα ηλεκτρονόμων τα οποία μπορούν να οδηγηθούν από χαμηλή τάση 5V και ρεύμα 25mA. Τα χαρακτηριστικά της μονάδας την κάνουν κατάλληλη για εφαρμογές με μικροελεγκτές και έλεγχο συσκευών υψηλών τάσεων και ρευμάτων ανάλογα με τον τύπο του ηλεκτρονόμου.

4.3.2 Σύνδεση μονάδας ηλεκτρονόμων με το Arduino..

Η μονάδα ηλεκτρονόμων κάνει εύκολη την σύνδεση με το Arduino συνδέοντας μόνο τους ακροδέκτες με την σήμανση IN και τον αριθμό του κάθε ηλεκτρονόμου. Οι

ακροδέκτες VCC και GND συνδέονται με εξωτερική τροφοδοσία 5V ή απευθείας από τους ακροδέκτες τροφοδοσίας του Arduino 5V και GND. Ο Έλεγχος των τεσσάρων ηλεκτρονόμων γίνεται από τους ακροδέκτες της μονάδας με την σήμανση IN1-IN4 και συνδέονται με τους ακροδέκτες του Arduino 44-48. Εφαρμόζοντας χαμηλό δυναμικό στον ακροδέκτη IN4 του κυκλώματος της εικόνας 4.10 επιτρέπουμε την ροή ρεύματος και μέσα από την αντίσταση R14 και στην συνέχεια το φωτοτρανζίστορ U7 οδηγεί ρεύμα στην βάση του τρανζίστορ Q7 με αποτέλεσμα να λειτουργεί σαν κλειστός διακόπτης και οδηγώντας με την σειρά του τον ηλεκτρονόμο σε κατάσταση ενεργοποίησης ενώνοντας τα δύο του άκρα. Αντίθετα αν οδηγήσουμε υψηλή τάση στην είσοδο της μονάδας τότε κρατάμε τον ηλεκτρονόμο απενεργοποιημένο.



Εικόνα 4.10 Σύνδεση μονάδας ηλεκτρονόμων στο Arduino.

4.3.4 Κώδικας οδήγησης ηλεκτρονόμου.

Για να μπορέσουμε να ενεργοποιήσουμε έναν ηλεκτρονόμο στέλνουμε μια χαμηλή τιμή στον ακροδέκτη του Arduino που είναι συνδεδεμένος ο ηλεκτρονόμος. Με την συνάρτηση `digitalWrite(2,LOW)` ενεργοποιούμε τον ηλεκτρονόμο που είναι συνδεδεμένος στον 2^ο ακροδέκτη του Arduino. Επιπλέον για τον εύκολο έλεγχο των ηλεκτρονόμων δημιουργήθηκε η συνάρτηση `driveRele(val)` με την οποία ανάλογα με την τιμή που δίνουμε στο όρισμα "val" ενεργοποιούμε και τους αντίστοιχους ηλεκτρονόμους. Η τιμή "val" μεταφράζεται σε δυαδική μορφή με το bit με την λιγότερη σημαντικότητα να αντιστοιχεί στον 1ο ηλεκτρονόμο το 2ο bit στον 2ο ηλεκτρονόμο και αντίστοιχα για τους άλλους δύο. Για παράδειγμα με την τιμή 3 η οποία μεταφράζεται σε δυαδική μορφή 0011 ενεργοποιεί τον 3ο και 4ο ηλεκτρονόμο με λογικό '0' και απενεργοποιεί τον 1^ο και 2ο ηλεκτρονόμο με το αντίστοιχο bit λογικό '1'.

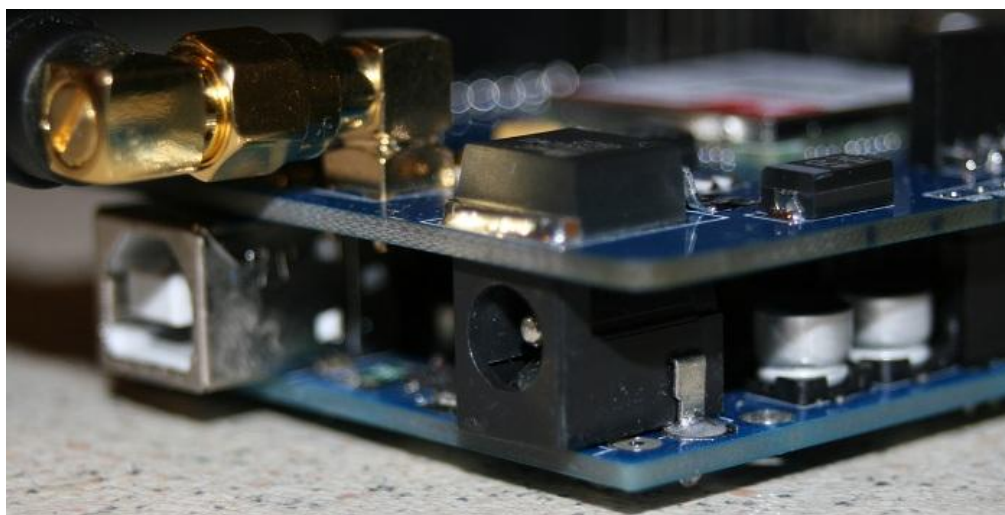
```
#define relePin1
#define relePin2
#define relePin3
#define relePin4
// Δηλώσεις ακροδεκτών των ρελέ 42,44,46,48

void setup()
```

```
{  
  Serial.begin(9600);  
  pinMode(relePin1,OUTPUT);  
  pinMode(relePin2,OUTPUT);  
  pinMode(relePin3,OUTPUT);  
  pinMode(relePin4,OUTPUT);  
  // Δηλώνουμε ως εξόδους τους ακροδέκτες του Arduino που επικοινωνούν  
  // με τις εισόδους της μονάδας ηέλιτρονόμων.  
}  
  
void loop()  
{  
  driveRele(2);  
  delay(500);  
  driveRele(4 );  
}  
  
void driveRele(byte rele)  
{  
  digitalWrite(relePin1,bitRead(0,rele));  
  digitalWrite(relePin2,bitRead(1,rele));  
  digitalWrite(relePin3,bitRead(2,rele));  
  digitalWrite(relePin4,bitRead(3,rele));  
}
```

4.4 Επικοινωνία με την ασπίδα IComSat.

Η ασπίδα IComSat μας δίνει την δυνατότητα επικοινωνίας του Arduino με το Διαδίκτυο χρησιμοποιώντας την ασύρματη τεχνολογία GSM/GPRS. Για την συγκεκριμένη ασπίδα έχουμε μιλήσει στο κεφάλαιο 2 και έτσι δεν θα αναφερθούμε στο κατασκευαστικό της κομμάτι “hardware”. Παρακάτω αναφέρουμε τον τρόπο επικοινωνίας της ασπίδας με το Arduino καθώς και μερικά παραδείγματα των AT εντολών.



Εικόνα 4.11 Η Ασπίδα GSM900 κουμπωμένη πάνω στο Arduino Mega.

Κουμπώνουμε την ασπίδα πάνω στην πλακέτα του Arduino. Για να μπορέσει η ασπίδα να επικοινωνήσει σειριακά με το Arduino συνδέουμε με ένα καλώδιο τον ακροδέκτη

Tx1 του Arduino με την σειρά ακροδεκτών RX της ασπίδας και τον ακροδέκτη Rx1 με την σειρά ακροδεκτών Tx της ασπίδας. Έχουμε επιλέξει την σειριακή θύρα 1 του Arduino για επικοινωνία με την ασπίδα και έχουμε αποφύγει την χρήση έτοιμων βιβλιοθηκών της ασπίδας για εξοικονόμηση χώρου μνήμης και για καλύτερο έλεγχο του κώδικα. Οποιαδήποτε ανάγκη επικοινωνίας δημιουργήθηκε έχει καλυφθεί με συναρτήσεις που έχουμε δημιουργήσει. Παρακάτω θα αναφερθούμε στις βασικότερες συναρτήσεις για την επικοινωνία του Arduino με την ασπίδα.

4.4.1 Χρήση των AT εντολών

Ο επεξεργαστής SIM900 της ασπίδας IComsat επικοινωνεί μέσω AT εντολών που στέλνονται σειριακά. Η εταιρεία κατασκευής του επεξεργαστή έχει έναν αναλυτικό οδηγό τέτοιων εντολών για τον πλήρη έλεγχο του επεξεργαστή, καλύπτοντας έτσι μια μεγάλη γκάμα λειτουργιών όπως τηλεφωνική κλήση, αποστολή μηνύματος, επικοινωνία με το Διαδίκτυο, έλεγχο της τροφοδοσίας κτλ.

Στη σελίδα <http://imall.iteadstudio.com/im120417009.html> μπορούμε να κατεβάσουμε δωρεάν σε μορφή pdf τον πλήρη οδηγό AT εντολών. Το Arduino για να στείλει σειριακά δεδομένα χρησιμοποιεί τις εντολές Serial.print() και Serial.send().

Για να στείλουμε μία AT εντολή γράφουμε Serial1.println(" εντολή ");. Μέσα στα εισαγωγικά βάζουμε την εντολή που θέλουμε να στείλουμε και αποθηκεύεται στο χώρο προσωρινής αποθήκευσης (buffer) της σειριακής του Arduino χωρητικότητας 64 byte. Το Arduino για να διαβάσει σειριακά δεδομένα στον ακροδέκτη Rx1 χρησιμοποιεί τις εξής εντολές: Serial1.read(); με την οποία διαβάζει την πληροφορία σε χαρακτήρες "ASCII" και την Serial.Byte(); με την οποία διαβάζει την πληροφορία σε μορφή "Byte".

4.4.2 Λειτουργία της ασπίδας IComSat ως πελάτης (client).

Για να μπορέσουμε να ανοίξουμε μία σελίδα στο Διαδίκτυο ή να στείλουμε δεδομένα σε έναν εξυπηρετητή χρειάζεται να πάρουμε την ιδιότητα του πελάτη. Πριν από οποιαδήποτε ενέργεια και για να μπορέσουμε να έχουμε μια επιτυχή σύνδεση πρέπει να συνδεθούμε στο GSM/GPRS δίκτυο μέσω μίας κάρτας SIM. Για να βοηθήσουμε στην κατανόηση της λειτουργίας του επεξεργαστή SIM900 μέσω AT εντολών με μπλε χρώμα είναι η απάντηση που περνούμε από τον επεξεργαστή SIM900 και με μαύρο οι εντολές που δίνουμε στον επεξεργαστή.

```
AT+CPIN? // Έλεγχος αριθμού της κάρτας SIM .
```

```
+CPIN: READY //το pin είναι σωστό.
```

```
OK
```

```
AT+CSQ // Διαβάζει την ένταση σήματος του δικτύου.
```

```
CSQ: 20,0 // το σήμα μας είναι καλό.
```

```
OK
```

```
AT+CREG? // Έλεγχος εγγραφής στο δίκτυο.
```

```
+CREG: 0,1//επιτυχή σύνδεση στο δίκτυο
```

```
OK
```

```
AT+CGATT?// Ελέγχει την σύνδεση στο GPRS δίκτυο.
```

```

+CGATT: 1 // επιτυχής σύνδεση στο GPRS δίκτυο.
OK
AT+CSTT="internet.open"," "," ";// Εισαγωγή των APN δεδομένων της Vodafone .1
OK
AT+CIICR // Αναφορά στην ασύρματη σύνδεση (GPRS ή CSD)
OK
AT+CIFSR // IP διεύθυνση της ασπίδας.
10.78.245.128//διεύθυνση αλλάζει σε κάθε καινούρια σύνδεση.
AT+CIPSTART="TCP ","192.168.2.3","9090";
// Ξεκινάει TCP σύνδεση με τον εξυπηρετητή.
OK
CONNECT OK // Σύνδεση επιτυχής.
AT+CIPSEND// Προετοιμασία για αποστολή δεδομένων.
> hello TCP server //Αποστολή δεδομένων στον απομακρυσμένο εξυπηρετητή.
SEND OK // Αποστολή δεδομένων επιτυχής
hello sim900 // Απάντηση του απομακρυσμένου εξυπηρετητή
CLOSED // Ο εξυπηρετητής έκλεισε την σύνδεση.

```

Η εντολή AT+CIPSTART="A ","B","Γ"; έχει τρία ορίσματα:

- A:Πρωτόκολλο επικοινωνίας που χρησιμοποιούμε για την σύνδεση (π.χ: TCP)
- B:IP διεύθυνση του εξυπηρετητή (π.χ: 192.168.2.3)
- Γ:Η πόρτα στην οποία ακούει ο εξυπηρετητής (π.χ: 9090)

2.4.2 Λειτουργία του SIM900 ως εξυπηρετητή (Server).

Για να μπορέσουμε να δεχθούμε ένα αίτημα από έναν πελάτη ή έναν άλλο εξυπηρετητή χρειάζεται να λειτουργήσουμε την ασπίδα στην πλευρά του εξυπηρετητή. Με αυτό τον τρόπο θέτουμε την ασπίδα σε κατάσταση αναμονής νέας σύνδεσης. Όταν λάβει το αίτημα σύνδεσης του πελάτη ανοίγει μία σύνδεση μαζί του και στέλνει τα δεδομένα που του ζήτησε και στην συνέχεια κλείνει την σύνδεση. Έτσι επιτυγχάνεται ο έλεγχος μέσα από μία ιστοσελίδα δίνοντας στην ιστοσελίδα το ρόλο του πελάτη. Παρακάτω αναφέρουμε τις AT εντολές με τις οποίες η ασπίδα λειτουργεί ως εξυπηρετητής.

```

AT+CPIN?
+CPIN: READY
OK
AT+CSQ
CSQ: 20,0
OK
AT+CREG?
+CREG: 0,1
OK

```

¹ APN(Access Point Name) Όνομα σημείου πρόσβασης χρησιμοποιείται από τον αρμόδιο πάροχο κινητής τηλεφωνίας και περιέχει δεδομένα σύνδεσης με το Διαδίκτυο. Τα δεδομένα αυτά είναι: APN NAME{ }, Username{ }, Password{ }.

```

AT+CGATT?
+CGATT: 1 // Επιτυχής σύνδεση στο GPRS δίκτυο
OK
AT+CIPSERVER=1,1234
//Ορισμός της ασπίδας ως εξυπηρετητή που ακούει στην πόρτα 1234.
OK
SERVER OK //Ορισμός επιτυχής
AT+CIFSR //Λήψη IP διεύθυνσης του εξυπηρετητή
10.78.33.227
AT+CIPSTATUS // Λήψη της κατάστασης σύνδεσης δικτύου της ασπίδας
OK
STATE: SERVER LISTENING // Αναμονή σύνδεσης από εξωτερικό πελάτη
REMOTE IP: 10.78.103.220 // Εμφάνιση της IP διεύθυνσης του πελάτη
hello server // Το μήνυμα του πελάτη.
AT+CIPSND// Διαδικασία έναρξης αποστολής δεδομένων στον πελάτη.
>hello client // Αποστολή δεδομένων στον πελάτη.
SEND OK // Αποστολή επιτυχής.
AT+CIPSERVER=0 //Σταματάει να βρίσκεται σε κατάσταση ακρόασης.
OK// Ενέργεια επιτυχής.
SERVER CLOSE // Διακοπή σύνδεσης ως εξυπηρετητής.
hello sim900 //Συνέχεια λήψης δεδομένων από τον πελάτη και μετά τη διακοπή.
AT+CIPCLOSE //Διακοπή σύνδεσης TCP
CLOSE OK .// Διακοπή επιτυχής

```

4.5 Ενημέρωση παρούσας κατάστασης με διόδους φωτός (LED) .



Εικόνα 4.12: Δίοδοι φωτός σε διάφορα χρώματα και μεγέθη

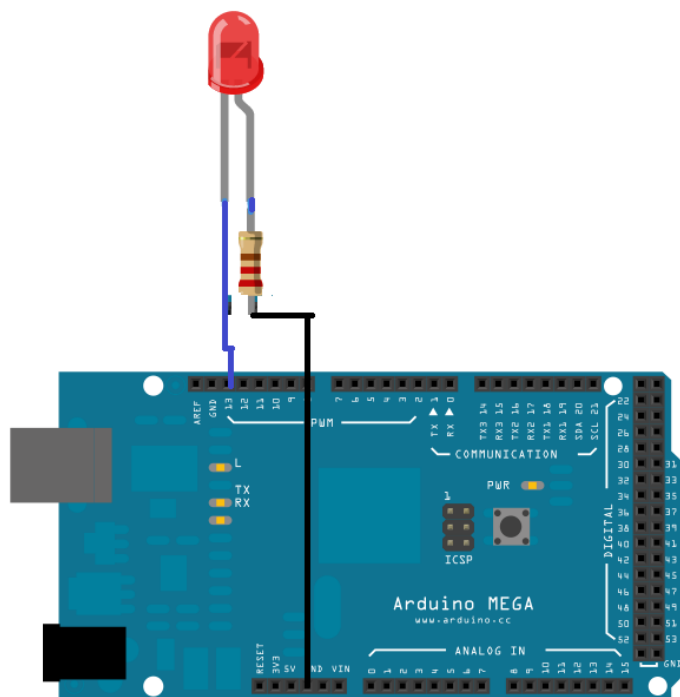
Για πιο εύκολη κατανόηση της λειτουργίας και για την ενημέρωση της κατάστασης που βρίσκεται η κατασκευή χρησιμοποιούμε 19 φωτοδιόδους (LED). *Δίοδος Εκπομπής Φωτός*, (LED, *Light Emitting Diode*), αποκαλείται ένας ημιαγωγός ο οποίος εκπέμπει φωτεινή ακτινοβολία στενού φάσματος όταν του παρέχεται μία ηλεκτρική τάση κατά τη φορά ορθής πόλωσης. Παρακάτω αναφέρονται οι λειτουργίες των φωτοδιόδων της κατασκευής:

- 4 LED για την ενημέρωση κατάστασης των εξόδων της συσκευής.
- 4 LED για την ενημέρωση κατάστασης εισόδων της συσκευής
- 3 LED για την κατάσταση τροφοδοσίας της συσκευής, για την κατάσταση σύνδεσης στο GPRS δίκτυο και για την ενημέρωση επικοινωνίας με τον εξυπηρετητή-πελάτη.

Τα LED που έχουμε χρησιμοποιήσει στην κατασκευή μας είναι οικονομικά με χαμηλές απαιτήσεις λειτουργίας που μπορούν να ελεγχθούν από το Arduino. Για να λειτουργήσει το LED χρειάζεται διαφορά δυναμικού 2V και ρεύμα 20 mA.

4.5.1 Σύνδεση του LED με το Arduino.

Για την προστασία του LED από καταστροφή βάζουμε σε σειρά μία αντίσταση για να περιοριστεί το ρεύμα που θα περάσει από τον ακροδέκτη του Arduino. Σύμφωνα με τον νόμο του Ohm ($R=V/I$) υπολογίζουμε την ελάχιστη τιμή της αντίστασης. $R=(V_s-V_L)/I_L$ ή $R=(5V-2V)/20mA=150\Omega$ με V_s =τάση τροφοδοσίας, V_L =τάση φωτοδιόδου και I_L =ρεύμα φωτοδιόδου. Χρησιμοποιώντας αντίσταση 220Ω δεν υπερβαίνουμε τα όρια ρεύματος. Συνδέουμε τον ένα ακροδέκτη της αντίστασης με τον ακροδέκτη 13 του Arduino. και τον άλλο ακροδέκτη της αντίστασης τον συνδέουμε με τον θετικό ακροδέκτη (άνοδο) της φωτοδιόδου. Τέλος συνδέουμε τον ακροδέκτη γείωσης της φωτοδιόδου με τον ακροδέκτη GND του Arduino.



Εικόνα 4.13 Συνδεσμολογία Arduino Mega με LED κόκκινου χρώματος.

4.5.2 Κώδικας οδήγησης φωτοδιόδου(LED).

Παρακάτω αναφέρουμε ένα παράδειγμα που θα αναβοσβήνει κάθε ένα δευτερόλεπτο ένα LED.

```
//Δίνουμε το όνομα led στον ακροδέκτη 13.
```

```
int led = 13;
```

```
//Ο κώδικας που είναι μέσα στην setup() τρέχει μόνο μία φορά κατά την  
//έναρξη.
```

```
void setup() {
```

```
  // Αρχικοποιούμε τον ακροδέκτη 13 σαν έξοδο .
```

```

pinMode (led, OUTPUT) ;
}
//Ο κώδικας που είναι μέσα στην loop() επαναλαμβάνεται.
void loop () {
digitalWrite (led, HIGH) ; // Ανοίγουμε το led(υψηλό δυναμικό στον
//ακροδέκτη 13).
delay (1000) ; // Περιμένουμε ένα δευτερόλεπτο.
digitalWrite (led, LOW) ; // Σβήνουμε το LED(χαμηλό δυναμικό στον
//ακροδέκτη 13).
delay (1000) ; // Περιμένουμε ένα δευτερόλεπτο.
}

```

4.6 Διάφορα εξαρτήματα που χρειάστηκαν για την κατασκευή του κουτιού.

Εκτός από τα εξαρτήματα που αναφέραμε παραπάνω χρειαστήκαμε επιπλέον και άλλα εξαρτήματα όπως :

- Κλέμες: τρεις κλέμες των 4 θέσεων για τις εισόδους.
- Τροφοδοτικό 9V 3A:
- Καλώδια διαφόρων μεγεθών:
- Πλαστικές βάσεις στήριξης της πλακέτας
- Πλαστικές θήκες για τα LED
- Διακόπτης τροφοδοσίας
- Διάτρητη πλακέτα
- Βύσμα τροφοδοσίας
- Πλαστικό κουτί

Εξαρτήματα	Τιμή σε ευρώ
3 x Κλέμες 4 θέσεων	3x0.95€=2.85€
19 x Φωτοдиодοι τύπου LED	19x0.20=3.80€
19 x Πλαστικές θήκες φωτοδιόδων	19x0.15=2.15€
1 x Διακόπτης τροφοδοσίας	1x0.50=0.50€
1 x Βύσμα τροφοδοσίας	1x1.00=1.00€
8 x Πλαστικές βάσεις στήριξης πλακέτας	8x0.20=1.60€
1 x Ασπίδα GSM/GPRS	1x45.00=45.00€
1 x Πλατφόρμα Arduino Mega	1x55.00=55.00€
1 x Μονάδα τεσσάρων ηλεκτρονόμων	1x8.00=8.00€
1 x Αισθητήρα υπερήχων	1x4.00=4.00€
1 x Αισθητήρα θερμοκρασίας	1x2.00=2.00€
1 x Διάτρητη πλακέτα 2x3 cm	1x1.20=1.20€
1 x Πλαστικό κουτί κατασκευής	1x4.00=4.00€
19 x Αντιστάσεις 220Ωμ	19x0.01=0.19€
Συνολικό κόστος υλικών κατασκευής	131,29€

5 Λογισμικό μέρος του συστήματος.

Σε αυτό το κεφάλαιο θα γίνει μια περιγραφή στην λειτουργία του συστήματος και τον τρόπο υλοποίησής του από την πλευρά του λογισμικού. Σκοπός είναι να δημιουργηθεί μία συσκευή που θα αντιδρά στις εντολές που δίνει ο χρήστης μέσα από το περιβάλλον μίας ιστοσελίδας. Η ιστοσελίδα θα ενημερώνει τον χρήστη για την παρούσα κατάσταση της συσκευής. Το σύστημα αυτό επιτρέπει τον απομακρυσμένο έλεγχο χρησιμοποιώντας τεχνολογίες ασύρματης επικοινωνίας GPRS και διαδικτυακών λειτουργιών. Η λειτουργία του συστήματος στηρίζεται στην σύνδεση τεσσάρων επιμέρους υποσυστημάτων:

- **Arduino:** Περιλαμβάνει όλα τα υλικά μέρη τα οποία συνδέονται με το Arduino και αλληλεπιδρά με αυτά παίρνοντας ή δίνοντας πληροφορίες. Αποτελεί τον εγκέφαλο του υλικού μέρους. Με την βοήθεια της ασπίδας GSM/GPRS μπορεί να επικοινωνεί με τον εξυπηρετητή και να του αναθέτει ή να δέχεται διάφορες λειτουργίες.
- **Πελάτης-Χρήστης:** Είναι το πρόσωπο όπου χειρίζεται την εφαρμογή. Ο πελάτης έχει την δυνατότητα να μιλάει με τον εξυπηρετητή και να ανταλλάσσει δεδομένα και αρχεία οποιαδήποτε στιγμή μέσα από την ιστοσελίδα. Μέσω της ιστοσελίδας πραγματοποιείται επικοινωνία πελάτη –εξυπηρετητή. Ο χρήστης αναθέτει κάποιες λειτουργίες στον εξυπηρετητή και αυτός επικοινωνεί με την βάση δεδομένων ή το Arduino. Ο πελάτης έχει την δυνατότητα να αλλάξει την κατάσταση των εξόδων της συσκευής και να ενημερωθεί για μετρήσεις και για την συνολική κατάσταση που βρίσκεται η συσκευή μας.
- **Εξυπηρετητής:** Είναι ο συνδεδετικός κρίκος ανάμεσα στον Πελάτη και στο Arduino. Ο πελάτης τού αναθέτει λειτουργίες και αυτός τις εκτελεί. Ο εξυπηρετητής είναι υπεύθυνος και για την ασφάλεια του συστήματος από την πλευρά του πελάτη και για τον έλεγχο και την ενημέρωση από την πλευρά Arduino.
- **Βάση Δεδομένων:** Επικοινωνεί μόνο με τον εξυπηρετητή. Είναι ένας αποθηκευτικός χώρος του εξυπηρετητή για να εκπληρώνει τις λειτουργίες που του αναθέτουν και για δικές του λειτουργίες.

Η περιγραφή των επόμενων κεφαλαίων θα γίνει σύμφωνα με την επικοινωνία μεταξύ δύο υποσυστημάτων και των λειτουργιών τους στο κυρίως σύστημα.

5.1 Mysql βάση δεδομένων.

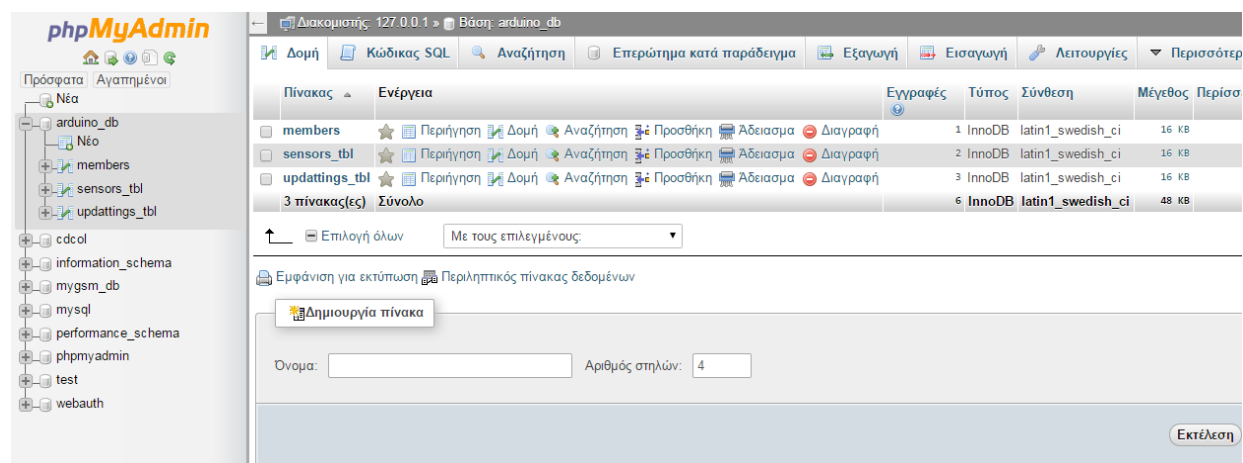
Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων ανοικτού κώδικα που χρησιμοποιεί την πιο γνωστή γλώσσα (SQL : Structured Query Language) για την προσθήκη και την επεξεργασία δεδομένων σε μια βάση δεδομένων. Η αποθήκευση των δεδομένων γίνεται μέσα σε αντικείμενα βάσης που έχουν την μορφή πινάκων. Ένας πίνακας, είναι μια συλλογή από σχετικές καταχωρήσεις δεδομένων και αποτελείται από στήλες και γραμμές. Οι βάσεις δεδομένων είναι χρήσιμες για την

αποθήκευση πληροφοριών σε κατηγορίες. Για την δημιουργία πίνακα υπάρχουν δύο μέθοδοι μέσω εντολών και μέσα από το περιβάλλον του XAMPP όπως περιγράφηκε στο κεφάλαιο 3.

Για την αποθήκευση των μετρήσεων των αισθητήρων και τις ρυθμίσεις διάφορων παραμέτρων, την IP του Arduino, την ασφάλεια της ιστοσελίδας και την πρόσβαση μόνο από εξουσιοδοτημένους χρήστες χρησιμοποιήθηκε η mysql βάση δεδομένων η οποία είναι ασφαλής γρήγορη και συμβατή με τον server.

5.1.1 Οι πίνακες της βάσης δεδομένων που χρησιμοποιήθηκαν στην εργασία μας.

Δημιουργήθηκε η βάση δεδομένων με το όνομα Arduino_db και τρεις πίνακες :



Εικόνα 5.1 Οι πίνακες της βάσης δεδομένων που χρειάστηκαν για την υλοποίηση του συστήματος.

Ο πίνακας με το όνομα members έχει τέσσερα στοιχεία :

id	username	password	ardu_ip
1	stratos	1234	192.168.2.3

Πίνακας 5.1 Τα στοιχεία του πίνακα members της βάσης δεδομένων.

- **Id:** αύξων μετρητής με κάθε νέα γραμμή. Κάθε χρήστης έχει μοναδικό id.
- **username:** Το όνομα του χρήστη που έχει πρόσβαση στο σύστημα
- **password:** Ο αντίστοιχος κωδικός του χρήστη που του επιτρέπει πρόσβαση στο σύστημα.
- **ardu_ip:** Η IP διεύθυνση της συσκευής Arduino η οποία είναι ελεγχόμενη από τον χρήστη

Ο πίνακας με το όνομα sensors_tbl έχει δύο γραμμές και 6 στήλες :

id	Name	Value	High	Low	Step
1	Level	17	31	5	4
2	Temp	21	12	11	4

Πίνακας 5.2 Τα στοιχεία του πίνακα Sensors_tbl της βάσης δεδομένων.

- *Name*: Ορίζει το είδος της μέτρησης, στάθμης και θερμοκρασίας.
- *Value*: Τρέχουσα τιμή του αισθητήρα μέτρησης
- *High*: Μέγιστη μετρούμενη τιμή που έχει ορίσει ο πελάτης
- *Low*: Ελάχιστη μετρούμενη τιμή που έχει ορίσει ο πελάτης
- *Step*: Το βήμα μέτρησης του αισθητήρα.

Ο πίνακας αυτός κρατάει τα στοιχεία που έχουν τα αισθητήρια που χρησιμοποιούμε. Στην πρώτη γραμμή αναφερόμαστε στον αισθητήρα υπερήχων ο οποίος υπολογίζει την στάθμη της δεξαμενής. Στην δεύτερη γραμμή αναφερόμαστε στο αισθητήριο θερμοκρασίας. Κάθε σειρά περιέχει την τρέχουσα τιμή του μετρούμενου στοιχείου, την μέγιστη τιμή, ελάχιστη και το βήμα μέτρησης. Η στήλη “Value” ανανεώνεται συνεχώς με νέες τιμές αισθητήρων από το Arduino. Ενώ οι τρεις τελευταίες στήλες ενημερώνονται από την φόρμα που συμπληρώνει ο πελάτης στον περιβάλλον των ρυθμίσεων.

Ο πίνακας με το όνομα `updattings_tbl` περιέχει τις ενημερώσεις της κατάστασης της συσκευής.

id	Name	Num1	Num2	Num3	Num4	Num5	Num6	Num7	Num8
1	Inputs	0	0	0	0	0	0	0	0
2	Outputs	1	1	0	0	0	0	0	0
3	Alarms	0	0	0	0	0	0	0	0

Πίνακας 5.3 Τα στοιχεία του πίνακα `updattings_tbl` της βάσης δεδομένων.

- *Inputs*: Περιέχει την αριθμημένη κατάσταση των εισόδων της συσκευής, με 0 όταν δεν υπάρχει είσοδος και ‘1’ όταν υπάρχει είσοδος.
- *Outputs*: Περιέχει τις αριθμημένες εξόδους της συσκευής, με ‘0’ εάν η έξοδος είναι απενεργοποιημένη, με ‘1’ εάν είναι ενεργοποιημένη και με ‘2’ αν βρίσκεται σε κατάσταση αναμονής μετά από μια εντολή αναμένοντας την απάντηση επιβεβαίωσης.
- *Alarms*: Περιέχει μία ακολουθία ενημερώσεων ανάλογα με τις ανάγκες του χρήστη. Αυτές μπορεί να είναι ενημέρωση για κατάσταση μέγιστης θερμοκρασίας, ελάχιστης, μέγιστης στάθμης κ.α. Υπάρχει δυνατότητα μέχρι και 8 ενημερώσεων.

Στην στήλη εξόδων “Outpouts” εκτός από τις καταστάσεις ‘0’ και ‘1’ έχουμε συμπεριλάβει ακόμα δύο: την ‘2’ η οποία μας ενημερώνει ότι ο πελάτης έχει στείλει μια εντολή ενεργοποίησης εξόδου και περιμένει την επιβεβαίωση και η κατάσταση ‘3’ που είναι εντολή απενεργοποίησης με αντίστοιχο τρόπο.

5.1.2 Σύνδεση με την βάση δεδομένων.

Πριν από οποιαδήποτε ενέργεια που σχετίζεται με την βάση δεδομένων πρέπει να υπάρξει μια σύνδεση. Για να μπορέσουμε να συνδεθούμε με την βάση δεδομένων χρησιμοποιούμε την εντολή `mysql_connect` γραμμένη σε γλώσσα PHP. Έχουμε

δημιουργήσει μία συνάρτηση με το όνομα Connection(); που βρίσκεται στο αρχείο “connect.php” στα αρχεία του εξυπηρετητή και κάθε φορά που χρειάζεται να συνδεθούμε με την βάση καλούμε την συνάρτηση αυτή.

Η συνάρτηση έχει την μορφή της εικόνας 5.2

```
<?php

function Connection(){
    $server="localhost";
    $user="root";
    $pass='';
    $db="arduino_db";
    $connection = mysql_connect($server, $user, $pass);
    if (!$connection) {
        die('MySQL ERROR: ' . mysql_error());
    }

    mysql_select_db($db) or die( 'MySQL ERROR: ' . mysql_error() );
    return $connection;
}
```

Εικόνα 5.2 Κώδικας της συνάρτησης Connection() γραμμένη σε PHP.

Η σύνταξη της εντολής mysql_connect(); παίρνει τρία ορίσματα. Πρώτον την διεύθυνση IP του εξυπηρετητή αν η βάση βρίσκεται στον ίδιο υπολογιστή με τον εξυπηρετητή δηλώνεται ως “localhost”. Δεύτερον το όνομα του χρήστη και τρίτον τον κωδικό. Για την ασφάλεια των δεδομένων οι βάσεις δεδομένων χρησιμοποιούν ένα όνομα και έναν κωδικό. Η εντολή επιστρέφει “false” εάν δεν μπορέσει να συνδεθεί αλλιώς επιστρέφει “true”.

5.2 Ασφάλεια Ιστοσελίδας

Για την ασφάλεια της ιστοσελίδας ώστε να επιτρέπεται η πρόσβαση μόνο σε εξουσιοδοτημένους χρήστες, δημιουργήθηκε ένα περιβάλλον επικοινωνίας με τον Πελάτη .

Δημιουργία φόρμας επικοινωνίας με τον πελάτη login.php

Κατά την είσοδο του πελάτη στην ιστοσελίδα που φιλοξενεί ο εξυπηρετητής, δημιουργήθηκε μια φόρμα επικοινωνίας. Σε αυτή την φόρμα ο πελάτης καλείται να εισάγει το όνομα και τον κωδικό που του έχει παραχωρηθεί από τον διαχειριστή.

Εισαγωγή στο σύστημα ελέγχου

Επιβεβαίωση μέλους

Όνομα :

Κωδ.Προσβασης :

Εικόνα 5.3 Φόρμα επικοινωνίας για την εισαγωγή του χρήστη στο σύστημα.

Στην συνέχεια όταν ο πελάτης εισάγει τα στοιχεία πατώντας το κουμπί “Είσοδος” γίνεται :

- Έλεγχος στοιχείων εισαγωγής του πελάτη `check_login.php`.

Σε αυτό το αρχείο γίνεται ο έλεγχος των στοιχείων που έχει δώσει ο πελάτης με τα στοιχεία που είναι αποθηκευμένα στον πίνακα “members” της βάσης δεδομένων. Αν τα στοιχεία που δίνει ο πελάτης μέσω της φόρμας επικοινωνίας είναι σωστά τότε μεταφέρεται στην κεντρική σελίδα. Εάν τα στοιχεία είναι λάθος δεν επιτρέπεται η πρόσβαση στην κεντρική σελίδα και επιστρέφει στην αρχική φόρμα εισόδου και ειδοποιώντας τον χρήστη ότι έδωσε λανθασμένα στοιχεία με το εξής μήνυμα “ Η Εισαγωγή δεδομένων ήταν λανθασμένη προσπαθήστε ξανά.”

- Επιτρεπόμενη είσοδος `success_login.php`

Το αρχείο αυτό τοποθετείται στην αρχή της κεντρικής σελίδας και επιτρέπει την πρόσβαση σε αυτή μόνο αν έχουν εισαχθεί τα σωστά στοιχεία συγκρίνοντας τα με αυτά που είναι αποθηκευμένα στον πίνακα “members” της βάσης δεδομένων.

- Διακοπή της εισόδου μας `logout.php`

Αυτό το αρχείο είναι υπεύθυνο για την έξοδο από την ιστοσελίδα μεταφέροντας τον πελάτη στην αρχική φόρμα εισόδου. Ο πελάτης βγαίνει από το σύστημα επιλέγοντας τον σύνδεσμο που εμφανίζεται στο πάνω αριστερό μέρος της σελίδας. Ο πελάτης αν θέλει να εισέλθει ξανά στο σύστημα χρειάζεται να εισάγει εκ νέου τα στοιχεία που του ζητάει η φόρμα επικοινωνίας.

5.3 Περιβάλλον ιστοσελίδας του πελάτη.

Το τμήμα της ιστοσελίδας που αλληλεπιδρά με τον πελάτη είναι γραμμένο σε γλώσσα HTML και χρησιμοποιεί τεχνολογίες JavaScript και Ajax για την αυτόματη ενημέρωση των αισθητήρων και την ενημέρωση των καταστάσεων.

Έφόσον ο πελάτης εισέλθει επιτυχώς στο σύστημα όπως πειγράψαμε παραπάνω ανοίγεται ένα παράθυρο ενημέρωσης και ελέγχου της συσκευής. Το παραθυρό αυτό έχει δημιουργηθεί με σκοπό να είναι απλό και εύχρηστο προς τον χρήστη παραλείποντας πολύπλοκες λειτουργίες χειρισμού. Οι λειτουργίες που επιτρέπει η σελίδα στον πελάτη είναι :

- Ενημέρωση για την κατασταση εισόδων της συσκευής με αναπαράσταση φωτεινών ενδείξεων
- Ενημέρωση για την κατάσταση εξόδων της συσκευής με αναπαράσταση φωτεινών ενδείξεων
- Ενημέρωση για τις ειδοποιήσεις που έχουν λάβει χωρα μέχρι τώρα με αναπαράσταση φωτεινών ενδείξεων

- Έλεγχο των εξόδων της συσκευής μέσα απο μία λίστα με κουτάκια.
- Ρυθμίσεις διαφόρων παραμέτρων των μετρήσεων στάθμης και θερμοκρασίας.

5.3.1 Φόρμες επικοινωνίας με τον χρήστη

Οι φόρμες επικοινωνίας χρησιμοποιούνται για την αποστολή δεδομένων του πελάτη στον εξυπηρετητή. Διαθέτει τρεις φόρμες επικοινωνίας με τις οποίες ο πελάτης μπορεί να εισάγει παραμέτρους των μετρήσεων της στάθμης και της θερμοκρασίας, όπως και να κάνει έλεγχο στις εξόδους της συσκευής. Οι φόρμες έχουν την παρακάτω μορφή.

Φόρμα Ελέγχου των εξόδων της συσκευής. Η φόρμα αυτή αποτελείται από επτά κουμπιά και είναι στοιχισμένα μέσα σε έναν πίνακα. Στην πρώτη γραμμή του πίνακα βρίσκονται τα 4 κουμπιά εναλλαγής κατάστασης εξόδων και έχουν τις ονομασίες «Out1-Out4» . Ενεργοποιώντας τα κουμπιά αυτά μπορούμε να αλλάξουμε την κατάσταση της αντίστοιχης εξόδου. Επίσης κατά την ενεργοποίηση των κουμπιών εναλλαγής τα κουμπιά κιτρινίζουν ενημερώνοντας μας ότι είναι ενεργοποιημένα και στην συνέχεια αν ξανά επιλέξουμε το ίδιο επανέρχεται στην αρχική του κατάσταση. Στην δεύτερη γραμμή του πίνακα ακολουθούν τα κουμπιά με τις ονομασίες :

Send: Στέλνει τις καταστάσεις των κουμπιών εναλλαγής στο Arduino. Κατά την διάρκεια που η αποστολή και μετάδοση των δεδομένων της φόρμας είναι σε εξέλιξη αυτόματα αλλάζει και η ονομασία του κουμπιού από «Send» σε «Loading» και απενεργοποιείται. Μετά από μια επιτυχή αποστολή επιστρέφει ξανά στην αρχική του κατάσταση.

Set_all: Ενεργοποιεί αυτόματα όλα τα κουμπιά εναλλαγής κατάστασης που βρίσκονται στην πρώτη γραμμή. Το κουμπί αυτό μπορούμε να αλλάξουμε όλες τις εξόδους μόνο με το πάτημα ενός κουμπιού.

Reset_all: Απενεργοποιεί αυτόματα όλα τα κουμπιά εναλλαγής κατάστασης που ήταν ενεργοποιημένα σε προηγούμενη κατάσταση..

Φόρμα εισαγωγής παραμέτρων για την μέτρηση θερμοκρασίας. Η φόρμα αυτή έχει τρία ορίσματα εισόδων. Η μονάδα μέτρησης της θερμοκρασίας είναι οι βαθμοί Κελσίου -°C .

- *Μέγιστη(Ειδοπ3)* : Ο πελάτης έχει την δυνατότητα να ορίσει τη μέγιστη επιτρεπόμενη θερμοκρασία λειτουργίας της συσκευής.
- *Ελάχιστη(Ειδοπ4):* Ο πελάτης έχει την δυνατότητα να ορίσει την ελάχιστη θερμοκρασία λειτουργίας της συσκευής.
- *Ακρίβεια:* Η ακρίβεια μέτρησης της θερμοκρασίας.

Η παρένθεση δίπλα από τα ορίσματα μας ενημερώνει ότι σε περίπτωση μέγιστης ή ελαχίστης θερμοκρασίας θα ενεργοποιηθεί και η αντίστοιχη φωτεινή ένδειξη των ειδοποιήσεων. Στην συνέχεια πατώντας το κουμπί «Ενημέρωση» που βρίσκεται κάτω

από την φόρμα στέλνονται τα δεδομένα που εισήγαγε ο πελάτης στον εξυπηρετητή για επεξεργασία.

Φόρμα εισαγωγής παραμέτρων μέτρησης της στάθμης. Η φόρμα αυτή έχει τέσσερα ορίσματα εισόδων. Η μονάδα μέτρησης της στάθμης είναι τα εκατοστά – cm.

- *Μέγιστη(Ειδοπ1):* Μέγιστη επιτρεπόμενη μέτρηση στάθμης.
- *Ελάχιστη(Ειδοπ2) η:* Ελάχιστη επιτρεπόμενη μέτρηση στάθμης.
- *Ακρίβεια:* Η ακρίβεια μέτρησης της στάθμης.
- *Αποστ.Αισθητήρα:* Η απόσταση που έχουμε τοποθετήσει τον αισθητήρα με το μετρούμενο αντικείμενο.

5.3.2 Ενημέρωση καταστάσεων της συσκευής.

Ο χρήστης μέσα από ένα εικονικό περιβάλλον από LEDS μπορεί να ενημερώνεται για την συνολική κατάσταση της συσκευής. Για την ενημέρωση αυτή έχει δημιουργηθεί ένα εικονικό περιβάλλον LED που τρέχει σε γλώσσα JavaScript και μας ενημερώνει για τις παρακάτω καταστάσεις.

- *Κατάσταση εισόδων:* Αναπαράσταση των εισόδων της συσκευής. Η γκρι ένδειξη υποδεικνύει ότι δεν έχουμε είσοδο και η πράσινη ότι έχουμε είσοδο
- *Κατάσταση εξόδων:* Αναπαράσταση των εξόδων της συσκευής. Η γκρι ένδειξη υποδεικνύει ότι η έξοδος είναι απενεργοποιημένη και η πράσινη ότι η έξοδος είναι ενεργοποιημένη.
- *Κατάσταση ειδοποιήσεων της συσκευής:* Αναπαράσταση των ειδοποιήσεων της συσκευής (μέγιστη θερμοκρασία, μέγιστη στάθμη κ.α.). Η γκρι φωτεινή ένδειξη υποδεικνύει ότι δεν έχουμε κάποια ειδοποίηση και η πορτοκαλί ότι έχουμε κατάσταση ειδοποίησης. Τις καταστάσεις ειδοποίησης ο χρήστης δεν μπορεί να τις αλλάξει.

Ο χρήστης επίσης μπορεί να πληροφορηθεί για την λειτουργία κάθε φωτεινής ένδειξης αφήνοντας το ποντίκι πάνω στο LED για μερικά δευτερόλεπτα. Στην συνέχεια εμφανίζεται ένα σύντομο μήνυμα με την περιγραφή λειτουργίας του LED.

5.3.3 Μέτρηση των αισθητήρων στάθμης και θερμοκρασίας.

Για την ενημέρωση της στάθμης του υγρού στην δεξαμενή καθώς και για την θερμοκρασία περιβάλλοντος που βρίσκεται η συσκευή μας, δημιουργήσαμε μία εικονική αναπαράσταση δεξαμενής και ένα εικονικό θερμόμετρο.

- **Εικονική αναπαράσταση δεξαμενής:** Μέσα από το περιβάλλον εικονικής δεξαμενής ο χρήστης ενημερώνεται για την τρέχουσα στάθμη του υγρού που μετράει η συσκευή μας χρησιμοποιώντας τον αισθητήρα υπερήχων όπως έχουμε αναφέρει σε παραπάνω κεφάλαιο. Το ύψος του της στάθμης της εικονικής δεξαμενής που είναι με μπλε φόντο είναι ανάλογο της στάθμης της πραγματικής δεξαμενής. Επιπλέον η δεξαμενή δέχεται τα ορίσματα της μέγιστης και ελάχιστης στάθμης, το ύψος του αισθητήρα καθώς και την

ακρίβεια μέτρησης στην βαθμονόμηση της δεξαμενής. Όλα αυτά τα ορίσματα δίνονται από τον χρήστη μέσω της φόρμας παραμέτρων της στάθμης.

- **Εικονική αναπαράσταση θερμομέτρου:** Μέσα από το περιβάλλον του εικονικού θερμομέτρου ο χρήστης ενημερώνεται την τρέχουσα θερμοκρασία περιβάλλοντος που μετράει η συσκευή με την χρήση του αισθητήρα θερμοκρασίας DS18B20. Το ύψος της πράσινης μπάρας αντιστοιχεί στην τρέχουσα θερμοκρασία και τα δύο ορίσματα αντιστοιχούν στην μέγιστη και ελάχιστη θερμοκρασία. Τα ορίσματα μπορούν να αλλάξουν από τον χρήστη μέσα από την φόρμα παραμέτρων της θερμοκρασίας. ,

Για τη συνεχή ενημέρωση των μετρήσεων χρησιμοποιούμε την συνάρτηση που έχουμε δημιουργήσει και είναι γραμμένη σε γλώσσα javascript. Κάθε 1 δευτερόλεπτο ανανεώνει ένα κομμάτι του κώδικα. Αυτές οι ενέργειες δεν γίνονται αντιληπτές από τον χρήστη γιατί τρέχουν στο πίσω μέρος της σελίδας του χρήστη.

```
<script type="text/javascript">
var auto_refresh = setInterval(
  function () { $('#load_updatings_wrap').load('ardu.php
#load_updatings',function(response,status,xhr){
  takeDatabaseVals();//παίρνει τα δεδομένα από την βάση δεδομένων.
  gettemp();//διαβάζει την στάθμη στην εικονική δεξαμενή.
  getlevel();//διαβάζει την θερμοκρασία στο εικονικό θερμόμετρο
});

}, 1000); // επαναλαμβάνει κάθε ένα δευτερόλεπτο.
</script>
```

Με την εντολή setInterval(); μπορούμε να καλούμε μία συνάρτηση ξανά και ξανά σε προκαθορισμένο χρονικό διάστημα.

Π.χ.: setInterval(function() {alert("Hello")}, 3000); εμφανίζει το μήνυμα hello στην οθόνη μας κάθε 3 δευτερόλεπτα. Με την εντολή load('index.php #load_updatings_wrap'); φορτώνουμε το πλαίσιο "div" με την ονομασία "load_updatings" που βρίσκεται στο αρχείο index.php.

5.4 Επικοινωνία Πελάτη με Εξυπηρετητή

Η επικοινωνία του πελάτη με τον εξυπηρετητή είναι αμφίδρομη. Ο πελάτης στέλνει δεδομένα στον εξυπηρετητή για τον έλεγχο της συσκευής και αντίθετα ο πελάτης λαμβάνει δεδομένα από τον εξυπηρετητή για την ενημέρωση κατάστασης της συσκευής. Οι επικοινωνία γίνεται με την ανταλλαγή αιτημάτων σύνδεσης και αποστολή δεδομένων.

5.4.1 Έλεγχος εξόδων του Arduino μέσα από την ιστοσελίδα του πελάτη.

Ο πελάτης μπορεί να κάνει έλεγχο στην συσκευή στέλνοντας δεδομένα στον εξυπηρετητή και στην συνέχεια ο εξυπηρετητής στο Arduino όπου είναι και ο τελικός προορισμός. Για να μπορέσει να στείλει ο πελάτης δεδομένα στον εξυπηρετητή χρησιμοποιεί τις φόρμες επικοινωνίας που βρίσκονται στο αρχείο "index.php" που προαναφέραμε. Για την ασύγχρονη αποστολή δεδομένων και για να συνεχίσουμε να τρέχουμε στην ίδια σελίδα χρησιμοποιούμε τεχνολογία jquery με την οποία μπορούμε όλες οι διαδικασίες να γίνονται πίσω από την κυρίως σελίδα. Ένα παράδειγμα αποστολής των δεδομένων των καταστάσεων εναλλαγής των εξόδων φαίνεται παρακάτω.

```
function sendOuts(){
    var msg="Outputs="+btnvalue;//δημιουργούμε το μήνυμα Outputs και την
    τιμή εξόδων για εναλλαγή της εξόδου των εξόδων.
    Στην συνέχεια αποστέλουμε με POST αίτημα το μήνυμα και περιμένουμε
    την απάντηση που βρίσκεται στα «data»
    $.post("Outputs.php",msg,
        function(data,status){
            var pos = data.search("Out=");
            ψάχνουμε για την απάντηση Out= και την τιμή.
            if(pos>=0){
                var len=data.length;
                var res=data.substring(4,len);
                var val=parseInt(res); φιλτράρουμε την τιμή σαν ακέραιο.
                InitialvalueLed(val); αλλάζουμε τις καταστάσεις των LEDs των εξόδων.
            }
            else{
                δημιουργία μηνύματος σφάλματος εφόσον υπήρξε πρόβλημα στην
                ολοκλήρωση της αποστολής.
                alert("Υπήρξε σφάλμα στην επικοινωνία με το arduino. "+data);
            }
            Ενεργοποιούμε τα κουμπιά αποστολής δεδομένων.
            enablebuttons();
            updatewaiting=0;
            αλλάζουμε το όνομα από «Loading» σε «Send» στο κουμπί
            $("#btntsend").text("Send");
        });
    }
}
```

Στην συνέχεια τα δεδομένα τα λαμβάνει το αρχείο "ToArduino.php" του εξυπηρετητή και αφού φιλτράρει την πληροφορία της φόρμας επεξεργάζεται τα δεδομένα δημιουργώντας ένα μήνυμα ελέγχου εξόδων για το Arduino της μορφής @Out=3#. Με αυτή την εντολή το Arduino εναλλάσσει κατάσταση στις εξόδους 1 και 2 της συσκευής. Οι δύο ακριανοί χαρακτήρες '@' και '#' δηλώνουν την αρχή και το τέλος του μηνύματος ελέγχου. Με την ονομασία Out αναφερόμαστε στις εξόδους (Outputs). Η

τιμή καθορίζεται από τα κουμπιά εναλλαγής των εξόδων που έχει επιλέξει ο πελάτης στην φόρμα ελέγχου. Π.χ.: αν μαρκάρει ο χρήστης τα κουτάκια OUT1 και OUT2 στέλνεται ο αριθμός "0011" σε δυαδική μορφή με το λιγότερο σημαντικό bit να είναι το OUT1. Εάν μετατρέψουμε τον αριθμό αυτό σε δεκαδικό μας κάνει τον αριθμό 3. Αφού δημιουργήσει το μήνυμα στην συνέχεια το στέλνει στο Arduino ανοίγοντας την επικοινωνία με την τεχνολογία socket. Με τις εντολές socket_write() στέλνουμε δεδομένα και με την εντολή socket_read() διαβάζουμε την απάντηση. Όταν λάβουμε την απάντηση φιλτράρουμε την τιμή και στην συνέχεια αποθηκεύουμε τις νέες καταστάσεις των εξόδων στην βάση δεδομένων. Παρακάτω παρατίθεται ένα παράδειγμα κώδικα που υλοποιεί επικοινωνία με το Arduino.

```
<?php ακολουθεί κώδικας γραμμένος σε php
include("connect.php");
$link=Connection(); καλούμε την συνάρτηση για να συνδεθούμε με την βάση
δεδομένων.
$result=mysql_query("SELECT ardu_ip FROM members",$link); κρατάμε την
διεύθυνση του Arduino.
if($result != FALSE)
{
    $row = mysql_fetch_array($result);
    //echo $row['ardu_ip'];
    mysql_free_result($result);
    $Outputs=mysql_query("SELECT * FROM updattings_tbl WHERE id='2'");
    $resOutputs=mysql_fetch_array($Outputs); διαβάζουμε τις τιμές των εξόδων
από την βάση δεδομένων
    $Outputs=($resOutputs['Num1']*1)+($resOutputs['Num2']*2)+($resOutputs['Num3'
]*4)+($resOutputs['Num4']*8); μετατρέπουμε την δυαδική σε δεκαδική.
    $val=0;
    $len=0;
    $msgControl="@";
    $response="";
    foreach($_POST as $x => $x_value)
    { if($x == "Outputs")
      { $State=1;
        $Outputs=$x_value^$Outputs;
        $msgControl="$Outputs";
        Φτιάχνουμε την επιθυμητή τιμή των εξόδων εναλλάσσοντας τις
εξόδους που έχει επιλέξει ο χρήστης
      }
    }
}

$service_port ='80';
$address =$row['ardu_ip'];
$socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
Δημιουργία socket με το Arduino στην πόρτα 80 που ακούει το Arduino.
$resultSocket = socket_connect($socket, $address, $service_port);
```

```

if($resultSocket)
{
    $msgControl="@Out=$Outputs#";
    //echo "$msgControl";
    socket_write($socket,$msgControl,strlen($msgControl)); στέλνουμε το
μηνυμα στο Arduino
    $msg_resp=socket_read($socket,60); διαβάζουμε την απάντηση
    parse_str($msg_resp);
    $response="connect";
    if($Out)
    {
        { Εάν η απάντηση ήταν σωστή ενημερώνουμε την βάση δεδομένων με
τις καινούριες καταστάσεις των εξόδων.
        $res=$Out & 1;
        $sql="UPDATE updattings_tbl SET Num1=$res WHERE id='2'";
        mysql_query($sql,$link);
        $res=($Out>>1) & 1;
        $sql="UPDATE updattings_tbl SET Num2=$res WHERE id='2'";
        mysql_query($sql,$link);
        $res=($Out>>2) & 1;
        $sql="UPDATE updattings_tbl SET Num3=$res WHERE id='2'";
        mysql_query($sql,$link);
        $res=($Out>>3) & 1;
        $sql="UPDATE updattings_tbl SET Num4=$res WHERE id='2'";
        mysql_query($sql,$link);
        echo "Out=$Out";
    }
}
socket_close($socket); Κλείνουμε την σύνδεση με το Arduino
mysql_close($link); κλείνουμε την σύνδεση με την βάση δεδομένων.
?>

```

5.4.2 Ενημερώσεις των καταστάσεων του Arduino μέσα από την ιστοσελίδα του πελάτη.

Η ιστοσελίδα που αλληλεπιδρά με τον πελάτη είναι το αρχείο 'index.php'. Για να μπορέσει ο πελάτης να ενημερωθεί για την συνολική κατάσταση της συσκευής και να πάρει τις μετρήσεις της θερμοκρασίας και της στάθμης, έχουμε ορίσει ένα πλαίσιο "div" το οποίο ανανεώνεται κάθε 1 δευτερόλεπτο και συλλέγει δεδομένα από τους πίνακες της βάσης δεδομένων. Αφού τα επεξεργαστεί τα χρησιμοποιεί για την ενημέρωση του πελάτη.

Ο πελάτης μπορεί να ενημερωθεί για τις παρακάτω καταστάσεις-μετρήσεις:

- Κατάσταση των εξόδων της συσκευής.
- Κατάσταση των εισόδων της συσκευής.
- Κατάσταση των ειδοποιήσεων της συσκευής.
- Μετρήσεις της θερμοκρασίας.
- Μετρήσεις της στάθμης.

Για τις τρεις πρώτες καταστάσεις έχει δημιουργηθεί ένα εικονικό περιβάλλον αναπαράστασης των διόδων LED που έχει η συσκευή. Για την αναπαράσταση της

εικονικής διόδου LED έχει δημιουργηθεί μια συνάρτηση σε γλώσσα javascript. Παρακάτω παρατίθεται ένα παράδειγμα με δύο συναρτήσεις που είναι υπεύθυνες για την εναλλαγή των φωτεινών ενδείξεων των ειδοποιήσεων που στοιχίζονται σε έναν πίνακα.

`<script>` ακολουθεί κώδικας γραμμένος σε *javascript-jquery*

```
function InitialvalueLed_Alarms(val){
```

Συνάρτηση η οποία εναλλάσσει όλα τα LEDs των ειδοποιήσεων ανάλογα με το όρισμα που θα της δώσουμε

```
  changeLed("#alarm1",val&1);
```

αλλάζει την κατάσταση φωτεινής ειδοποίησης 1σύμφωνα με την τιμή του πρώτου bit της τιμής «val»

```
  changeLed("#alarm2",((val>>1)&1));
```

```
  changeLed("#alarm3",((val>>2)&1));
```

```
  changeLed("#alarm4",((val>>3)&1));
```

```
  changeLed("#alarm5",((val>>4)&1));
```

```
  changeLed("#alarm6",((val>>5)&1));
```

```
  changeLed("#alarm7",((val>>6)&1));
```

```
  changeLed("#alarm8",((val>>7)&1));
```

```
}
```

```
function changeLed(idled,val)
```

{ συνάρτηση που μας επιτρέπει την ενεργοποίηση-απενεργοποίηση φωτεινής ένδειξης σύμφωνα με το id και την τιμή που δίνουμε 0:κλειστό 1:ανοιχτό.

```
  if(val==0)
```

```
  { $(idled).attr("src", "images/grey.jpg");}
```

```
  else if(val==1)
```

```
  { $(idled).attr("src", "images/green.jpg");}
```

```
  else if(val==2)
```

```
  { $(idled).attr("src", "images/orangeOFF.jpg");}
```

```
  else if(val==3)
```

```
  { $(idled).attr("src", "images/orangeON.jpg");}
```

```
}
```

`</script>`

Δημιουργήσαμε έναν χώρο όπου φιλοξενούμε τον πίνακα με τα μηνύματα και τα αντίστοιχα LEDs.

```
<div id="alarm_div">
```

```
  <table style="margin:5px auto;" border="1">
```

```
    <caption> ονομασία πίνακα
```

```
    Led ειδοποιήσεων,εισόδων και εξόδων της συσκευής.
```

```
  </caption>
```

```
  <tr> δήλωση γραμμής πίνακα
```

```
    <th scope="col">Ειδοπ.1</th> δήλωση επικεφαλίδας
```

```
    <th scope="col">Ειδοπ.2</th>
```

```
    <th scope="col">Ειδοπ.3</th>
```

```
    <th scope="col">Ειδοπ.4</th>
```

```
    <th scope="col">Ειδοπ.5</th>
```

```

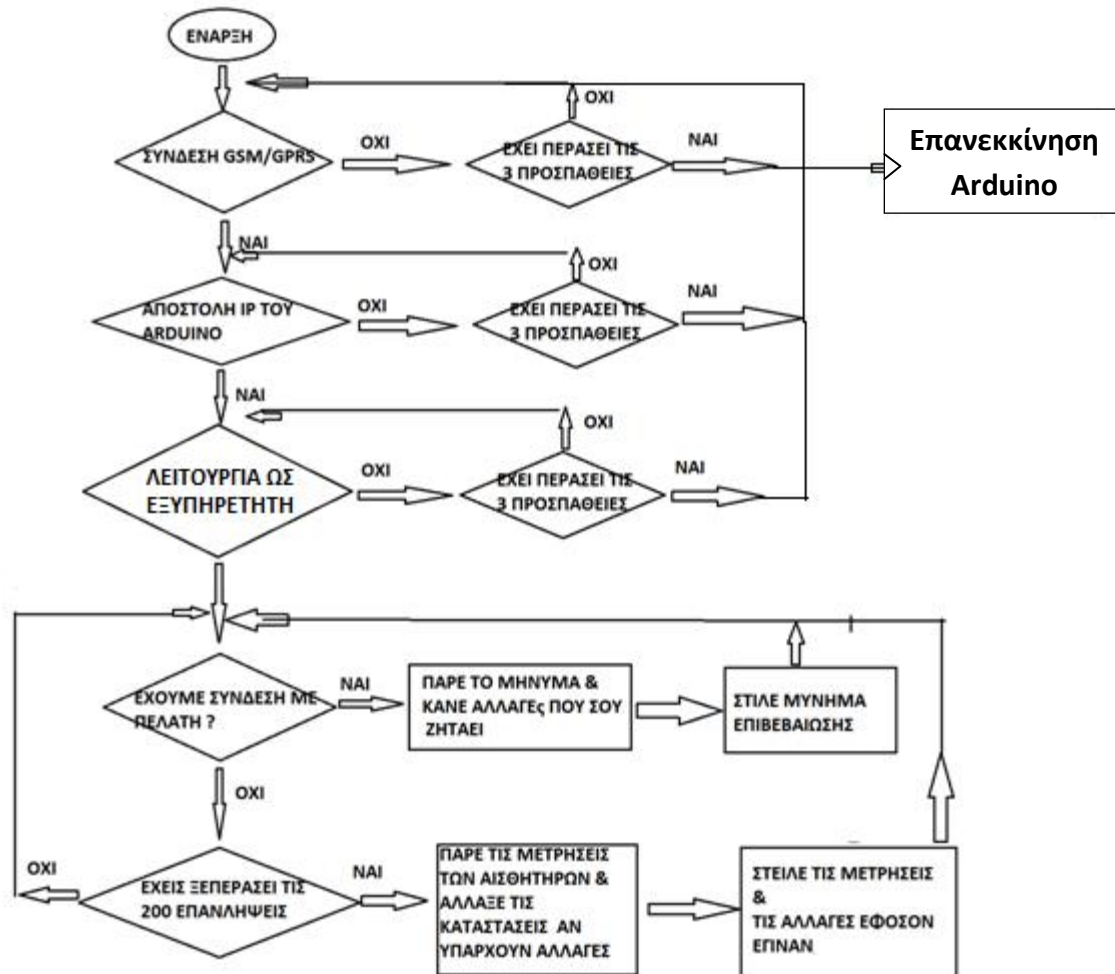
<th scope="col">Ειδοπ.6</th>
<th scope="col">Ειδοπ.7</th>
<th scope="col">Ειδοπ.8</th>
</tr>
<tr>  επόμενη γραμμή
  <td></td>
  Μέσα στο κάθε κελί έχουμε εισάγει ένα εικονικό LED με ύψος και
  πλάτος 42 πίξελ. Μέσα στο «title=" μήνυμα"» γράφουμε το μήνυμα
  επεξηγήσεις κάθε LED
  <td></td>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
</tr>
</table> τέλος του πίνακα
</div> τέλος του πλαισίου

```

5.5 Οι λειτουργίες του Arduino

Σε αυτό το κεφάλαιο θα αναλύσουμε την επικοινωνία του Arduino με τον εξυπηρετητή από την πλευρά του Arduino. Το Arduino με την βοήθεια της ασπίδας GSM/GPRS μπορεί να επικοινωνεί ασύρματα με τον εξυπηρετητή στέλνοντας του POST αιτήματα χρησιμοποιώντας τεχνολογία HTTP. Με την ονομασία Arduino αναφερόμαστε στην υλική σύνδεση του Arduino με την ασπίδα. Το Arduino λειτουργεί σαν εξυπηρετητής ο οποίος μπορεί να συνδεθεί και να εξυπηρετήσει μέχρι και επτά πελάτες διατηρώντας μέχρι και επτά ταυτόχρονες συνδέσεις. Στο διάγραμμα ροής του κώδικα του Arduino βλέπουμε τον καθοριστικό ρόλο που παίζει το Arduino σαν εξυπηρετητής για το κομμάτι του Διαδικτύου και για τις υπόλοιπες συνδέσεις των εξαρτημάτων.

Για την επικοινωνία του Arduino με την ασπίδα έχουμε αποφύγει να χρησιμοποιήσουμε έτοιμες βιβλιοθήκες. Για να μπορέσουμε να χειριστούμε την ασπίδα σύμφωνα με τις ανάγκες μας έχουμε δημιουργήσει συναρτήσεις για τις διάφορες λειτουργίες της ασπίδας. Μερικές από αυτές αναλύονται παρακάτω.



Εικόνα 5.4 Διάγραμμα ροής κώδικα του Arduino.

5.5.1 Σύνδεση του Arduino με το δίκτυο GSM/GPRS

Για να μπορέσει το Arduino να αλληλεπιδράσει με τον εξυπηρετητή ασύρματα χρησιμοποιούμε την ασύρματη τεχνολογία GPRS/GSM χρησιμοποιώντας την ασπίδα GSM/GPRS. Όπως έχουμε προαναφέρει η ασπίδα ελέγχεται μέσω AT εντολών. Για να συνδεθεί στο δίκτυο GSM ακολουθεί μια σειρά τέτοιων εντολών. Έτσι για να μπορέσουμε να συνδέσουμε το Arduino στο Διαδίκτυο έχουμε δημιουργήσει μια συνάρτηση με την οποία το Arduino στέλνει AT εντολές στην ασπίδα για να συνδεθεί με το Διαδίκτυο.

Από την στιγμή που η μονάδα τροφοδοτείται με ρεύμα, για να λειτουργήσει χρειάζεται έναν παλμό στο ακροδέκτη 8 του Arduino που επικοινωνεί με την ασπίδα. GSM_ON έχει την τιμή 8. Παρακάτω έχουμε την συνάρτηση powerON() η οποία θέτει την ασπίδα IComSat σε λειτουργία.

```

void powerON()
{
  digitalWrite(GSM_ON, HIGH);
  delay(1200);
  digitalWrite(GSM_ON, LOW);
  delay(10000);
}
  
```

```
Serial.println("power on");
}
```

Στην συνέχεια μία άλλη συνάρτηση που δημιουργήθηκε για την αποστολή των AT εντολών στην μονάδα είναι η `sendATwaitResp(τιμη1,τιμη2,τιμη3)`; Η συνάρτηση αυτή στέλνει μια AT εντολή στην “τιμή1”, περιμένει ένα χρονικό διάστημα δευτερολέπτων τόσων όσων είναι της “τιμή2”, και παίρνει την απάντηση από την TX της μονάδας και RX1 του Arduino. Η απάντηση που περιμένουμε είναι η “τιμή3”. Η συνάρτηση επιστρέφει αληθής αν έλαβε την απάντηση “τιμη3” στο χρονικό διάστημα “τιμη2”.

```
boolean sendATwaitResp(char at[],int maxcnt,char resp[])
{ int i=0;
  int j=0;
  char c;
  int len=strlen(resp);
  Serial1.println(at); // Αποστολή εντολής σειριακά
  while(i<maxcnt)
  { i++;
    delay(1);
    if((Serial1.available())>0) // Έλεγχος για δεδομένα στην σειριακή.
    { c=Serial1.read(); //Διάβασε τα δεδομένα
      i=0;
      Serial.print(c);
      if(resp[j]==c)//Εάν τα δεδομένα είναι ίδια με την απάντηση που
//περιμένουμε
      { j++;
        if(j==len){return true;} // Εάν βρέθηκε η απάντηση επέστρεψε αληθής
      }
      else j=0;
    }
  }

  return false; //Αλλιώς ψευδής
}
```

Τώρα είμαστε σε θέση να δούμε την συνάρτηση `startGsm()` σύνδεσης στο δίκτυο GSM/GPRS.

```
boolean startGsm()
{powerON(); //Στέλνει ένα παλμό έναρξης στην ασπίδα
delay(3000); //Καθυστίρηση 3 δευτερόλεπτα
sendATwaitResp("AT+CPIN?",3000,"READY"); // Στέλνει μια AT εντολή και
//περιμένει 3 δευτερολεπτα μέχρι να έρθει η απάντηση "READY"
delay(1000);
sendATwaitResp("AT+CSQ",3000,"OK");// Έλεγχος του σήματος
delay(1000);
```

```

    sendATwaitResp("AT+CREG?",3000,"OK"); //Έλεγχος για εγγραφή στο GSM
    //δίκτυο
    delay(1000);
    sendATwaitResp("AT+CGATT?",3000,"OK"); //Έλεγχος για σύνδεση στο GPRS
    //δίκτυο
    delay(1000);
    if(!sendATwaitResp("AT+CSTT=\"internet.open\",\",\",\",\",3000,\"OK\")){return 0;}
    //Εισαγωγή των APN στοιχείων της εταιρίας κινητής τηλεφωνίας. Για την
    //vodafone είναι "internet.open".
    delay(1000);
    sendATwaitResp("AT+CIICR",5000,"OK");//
    delay(3000);
    if (!sendATwaitResp("AT+CIFSR",5000,"ERROR"))//Λήψη διεύθυνσης IP της
    //ασπίδας
    { delay(5000);
    Serial.println();
    Serial.println("GPRS OK");//Εμφάνιση μηνύματος "GPRS OK" στη σειριακή οθόνη
    Serial.println();
    return 1;
    }
    else
    return 0;
    }

```

5.5.2 Λειτουργία της ασπίδας σαν εξυπηρετητής πολλαπλών συνδέσεων.

Για να μπορέσουμε να είμαστε σε θέση να λάβουμε ένα αίτημα από την σελίδα-εξυπηρετητή πρέπει να ορίσουμε την ασπίδα σε λειτουργία εξυπηρετητή πολλαπλών συνδέσεων. Αυτό μας δίνει δυνατότητα εκτέλεσης αμφίδρομης επικοινωνίας με τον "εξυπηρετητή" και πολλαπλές συνδέσεις από πελάτες. Η συνάρτηση που δημιουργήσαμε για αυτή την λειτουργία είναι η openServer()

```

boolean openServer()
{ int i=0;

    while(! sendATwaitResp("AT+CIPSERVER=1,80",3000,"SERVER OK") &&(i<3))
    { i++;
    }
    if(i<3)
    { digitalWrite(PIN_Led_Net,LED_ON); // Ενεργοποίηση της διόδου φωτός για
    //την κατάσταση σύνδεσης με το δίκτυο με την σήμανση "CON"
    return true;
    }
    return false;
}

```

Με την AT εντολή AT+CIPSERVER=1,80 ορίζουμε την μονάδα μας σε κατάσταση πολλαπλών συνδέσεων με το '1' και με το '80' ορίζουμε την πόρτα που θα ακούει ο εξυπηρετητής-Arduino. Αν μετά από τρεις προσπάθειες δεν καταφέρουμε να

λειτουργήσει η μονάδα σαν εξυπηρετητής, επιστρέφει η συνάρτηση ψευδής. Αντίθετα ενεργοποιούμε την δίοδο φωτός με την ένδειξη "CON" της συσκευή μας και επιστρέφει η συνάρτηση αληθής.

5.5.3 Αποστολή δεδομένων στον εξυπηρετητή με POST αιτήματα.

Ο τρόπος που στέλνει το Arduino δεδομένα στον εξυπηρετητή είναι μέσω POST αιτημάτων χρησιμοποιώντας το πρωτόκολλο επικοινωνίας HTTP. Ένα POST αίτημα έχει μία συγκεκριμένη δομή. Αποτελείται από :

- Γραμμή αίτησης
- Κεφαλίδες
- Κενή γραμμή
- Το μήνυμα

Ένα παράδειγμα POST αιτήματος φαίνεται παρακάτω:

```
POST /index.html HTTP/1.1
Host:192.168.2.3
User-Agent: Arduino/1.0
Content-Type : application/x-www-form-urlencoded
Connection: close
Content-Length: 256
```

Η πρώτη γραμμή είναι η γραμμή αίτησης ενώ οι επόμενες είναι οι κεφαλίδες στην συνέχεια ακολουθεί μια κενή γραμμή και τέλος γράφουμε το μήνυμα που θέλουμε να στείλουμε. Η γραμμή αίτησης περιέχει τρία πεδία: το πεδίο μεθόδου, το πεδίο URL και το πεδίο HTTP έκδοσης. Με την ίδια λογική έχουμε δημιουργήσει και την συνάρτηση `sendPost()` η οποία ανταλλάζει δεδομένα με τον εξυπηρετητή μέσω POST αιτημάτων.

```
boolean sendPost(const char server[],const char path[],char msg[])
{
  Serial1.print("AT+CIPSTART="); //Εναρξη σύνδεσης με τον εξυπηρετητή

  Serial1.print(charclient);
  Serial1.print(",\TCP\","\"); //Τύπου σύνδεσης TCP
  Serial1.print(server);
  Serial1.print("\","\");
  Serial1.println("9090"); // Πόρτα που ακούει ο εξυπηρετητής
  clientsNumber++;
  delay(500);
  if (waitResp(10000,"CONNECT OK"))
  { Serial1.print("AT+CIPSEND="); //Προετοιμασία για αποστολή δεδομένων
    Serial1.println(charclient);
    if(waitResp(5000,">"))//
    { digitalWrite(PIN_Led_Connect,LED_ON); //Ενεργοποίησε το LED της
      //συσκευής για κατάσταση αποστολής δεδομένων.
      Serial1.print("POST "); // Έναρξη αποστολής αιτήματος
      Serial1.print(path);
```



```

Serial1.print(" HTTP/1.0\nHost: ");
Serial1.println(server);
Serial1.println("User-Agent: Arduino/1.0");
Serial1.println("Content-Type: application/x-www-form-urlencoded");
Serial1.println("Connection: close");
Serial1.print("Content-Length: ");
itoa(strlen(msg),itoaBuffer,10); //Υπολογίζει το μέγεθος του μηνύματος
//που στέλνουμε
Serial1.println(itoaBuffer);
Serial1.println();//εισάγει κενή γραμμή μετά τις κεφαλίδες
Serial1.println(msg); //Γράφουμε το μήνυμα που θέλουμε να στείλουμε
Serial1.println();
Serial1.print(end_c[0]);
Serial1.print(end_c[1]); // Ολοκλήρωση μηνύματος αποστολή
if(waitResp(10000,"SEND OK")) //είχαμε επιτυχή αποστολή
{ if(whileHTTPResp(10000,"@OK#")) //Περιμένουμε να διαβάσουμε το
//μήνυμα @OK# το οποίο γραφούμε στο τέλος κάθε σελίδας του εξυπηρετητή
//για επιβεβαίωση
    { Serial.println("received msg");
      sended=1;
    }
}
digitalWrite(PIN_Led_Connect,LED_OFF); //Σβήνουμε το LED της συσκευής
//με την ένδειξη "send"
}
}
return sended; // Επιστρέφει το αποτέλεσμα αληθής για επιτυχημένη
αποστολή και λήψη η αλλιώς επιστρέφει ψευδής.
}

```

5.5.4 Φιλτράρισμα μηνύματος εξυπηρετητή.

Μετά από μία σύνδεση πελάτη, αρχίζει η μεταφορά δεδομένων. Στα δεδομένα που στέλνει ο πελάτης-εξυπηρετητής δεν είναι όλα χρήσιμη πληροφορία. Η πληροφορία που χρειάζεται το Arduino περικλείεται στους χαρακτήρες έναρξης '@' και τέλους '#', όπως έχουμε αναφέρει σε προηγούμενο κεφάλαιο. Για το λεγόμενο φιλτράρισμα έχουμε δημιουργήσει μια συνάρτηση με το όνομα readMsgFromServer(). Η συνάρτηση αυτή μας επιστρέφει την χρήσιμη πληροφορία σε έναν αλφαριθμητικό πίνακα στον κώδικα μας result[]. Να σημειωθεί ότι η συνάρτηση καλείται από την στιγμή που ανιχνεύσουμε το γράμμα '@'.

```

void readMsgFromServer(char result[])
{ int i=0; // μετρητής
  Serial.print("msg is:");
  do
  { c=Serial1.read();
    if((c>0)&& (i<60)) //όριο 60 χαρακτήρων
    { result[i]=c; //αποθηκεύει τον χαρακτήρα που διάβασε στον πίνακα

```

```
    Serial.print(result[i]); // εμφάνιση της χρήσιμης πληροφορία στην
//σειριακή οθόνη
    i++;
}
delay(10); //περιμένει 0,01 δευτερόλεπτα για τον επόμενο σειριακό
//χαρακτήρα

} while(c!='#'); //μέχρι να διαβάσει τον τερματικό χαρακτήρα δήλωσης
//τέλους.
result[i-1]='\0';
}
```

Η παραπάνω συνάρτηση διαβάζει χαρακτήρες από την σειριακή1 και τους αποθηκεύει στον πίνακα χαρακτήρων result[] μέχρι να συναντήσει τον χαρακτήρα τέλους '#' όπου σταματάει και η χρήσιμη πληροφορία. Επίσης έχουμε ορίσει και ένα όριο των 60 χαρακτήρων που μπορεί να αποθηκεύσει για προστασία από υπερχείλιση του πίνακα σε περίπτωση που δεν ανιχνεύσει τον τερματικό χαρακτήρα ή η πληροφορία μας είναι πολύ μεγάλη. Το όριο των 60 χαρακτήρων είναι επαρκές εφόσον δεν υπάρχει μήνυμα που να ξεπερνάει τους 60 χαρακτήρες χρήσιμης πληροφορίας.

6. Οδηγίες χρήσης της συσκευής.

6.1 Περιγραφή συσκευής.

Η συσκευή που υλοποιήθηκε στην παρούσα εργασία βασίζεται στην πλατφόρμα Arduino Mega συνδεδεμένη με άλλα περιφερειακά κυκλώματα, λαμβάνει μετρήσεις στάθμης υγρών μιας δεξαμενής, και θερμοκρασίας με την χρήση αισθητήρων. Αλληλεπιδρά ελέγχοντας διάφορες συσκευές χρησιμοποιώντας ηλεκτρονόμους. Ακόμα συνδέεται στο GSM/GPRS δίκτυο χρησιμοποιώντας την ασπίδα GSM IComSat η οποία είναι κουμπωμένη πάνω στην πλατφόρμα και μας επιτρέπει τον απομακρυσμένο έλεγχο και την ενημέρωση της συσκευής από απόσταση. Ο απομακρυσμένος έλεγχος και η επιτήρηση της συσκευής γίνεται μέσω μιας απλής ιστοσελίδας η οποία φιλοξενείται σε έναν απομακρυσμένο εξυπηρετητή. Με απλό σχεδιασμό και εύκολη στην χρήση μας παρέχει ένα περιβάλλον που δίνει την δυνατότητα στον πελάτη για την ενημέρωση των μετρήσεων σε πραγματικό χρόνο όπως επίσης την επιτήρηση και τον έλεγχο της συσκευής.



Εικόνα 6.1 Η πάνω όψη της συσκευής.

Είναι μία εύχρηστη συσκευή η οποία μπορεί να βρει εφαρμογή σε διάφορα συστήματα όπως:

- Οικιακοί αυτοματισμοί.
- Συναγερμοί.
- Αυτόματο πότισμα.

- Έλεγχος υγρών σε δεξαμενή.
- Επιτήρηση χώρων .
- Επιτήρηση και έλεγχος γεννήτριας.
- Φωτοβολταϊκά συστήματα.

6.2 Χαρακτηριστικά συσκευής

- Προστασία συσκευής από μη εξουσιοδοτημένους χρήστες με κωδικό πρόσβασης στην ιστοσελίδα.
- 2 αναλογικές εισόδους τάσης 5V (χρήση αισθητήρων).
- 2 ψηφιακές εισόδους τάσης 5V (χρήση αισθητήρων).
- 4 εξόδους με ηλεκτρονόμους (rele) δυνατότητας ανοικτής και κλειστής επαφής (NO-NC) μέχρι 10 A.
- Τάση τροφοδοσίας από 9-12V DC.
- Φωτεινές ενδείξεις LED για κάθε είσοδο-έξοδο.
- Απομακρυσμένο έλεγχο μέσα από ιστοσελίδα.
- Εικονικό περιβάλλον ειδοποιήσεων ανάλογα με τις ανάγκες του χρήστη.
- Διαθέτει θύρα USB για τον προγραμματισμό ή σύνδεση με υπολογιστή
- Υποστηρίζει μετρήσεις στάθμης υγρού και θερμοκρασίας.
- Ελάχιστη μεταφορά δεδομένων λόγω εξελιγμένου προγραμματισμού.
- Αυτόνομη γιατί χρησιμοποιεί την ασύρματη σύνδεση στο Διαδίκτυο και δεν περιορίζεται στον χώρο που θα εγκατασταθεί .

6.3 Τα υλικά μέρη της συσκευής.

6.3.1Κεραία

Η συσκευή χρησιμοποιεί κεραία, με βύσμα SMA, κατάλληλη για δίκτυα GSM/GPRS με συχνότητες ανάλογα με την περιοχή του δικτύου κινητής τηλεφωνίας. Στην Ευρώπη οι συχνότητες είναι 900 MHz και 1800MHz .



Εικόνα 6.2 Κεραία GSM/GPRS με βύσμα SMA για συχνότητες 890-960 MHz.

6.3.2 Βύσμα τροφοδοσίας.

Το βύσμα τροφοδοσίας της συσκευής βρίσκεται στην μπροστινή πλευρά με τον κεντρικό ακροδέκτη να είναι ο θετικός πόλος και το εξωτερικό περίβλημα η γείωση.



Εικόνα 6.3 Η μπροστινή όψη της συσκευής και η θέση του βύσματος τροφοδοσίας.

Η συσκευή τροφοδοτείται με πηγή συνεχούς τάσης από 9-12V και μέγιστο ρεύμα 3 A. Σε κατάσταση ηρεμίας η κατανάλωση είναι μικρότερη των 200mA.

6.3.3 Φωτεινές ενδείξεις LED

Για την οπτική αναπαράσταση της κατάστασης των εισόδων και εξόδων η συσκευή είναι τροφοδοτημένη με φωτεινές ενδείξεις τύπου LED. Όταν είναι ενεργοποιημένη μία είσοδος-έξοδος το αντίστοιχο LED είναι ανοιχτό και όταν είναι απενεργοποιημένη μία είσοδος-έξοδος το LED είναι κλειστό. Η συσκευή διαθέτει 19 LED για ενημερώσεις διαφόρων καταστάσεων όπως παρατίθενται παρακάτω:

4x LED για την κατάσταση των εισόδων IN1-IN4.

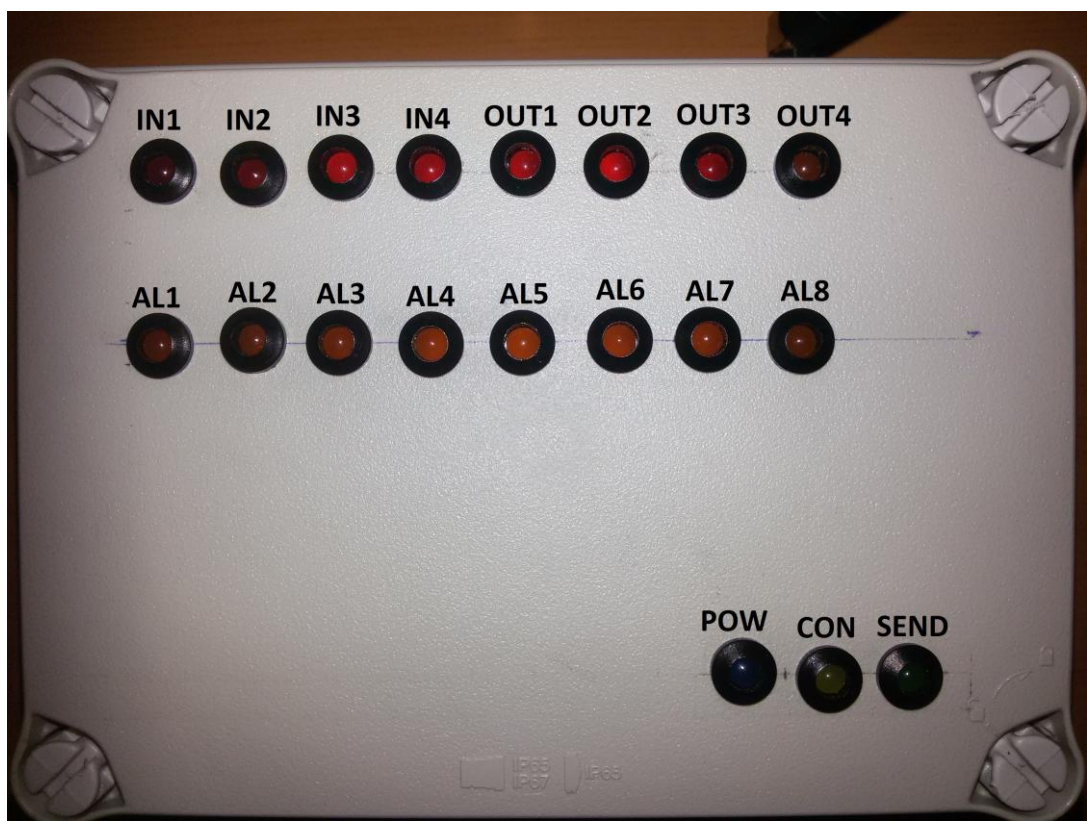
4x LED για την κατάσταση των εξόδων OUT1-OUT4.

8x LED για τις ειδοποιήσεις της συσκευής όπως AL1-AL8

- Μέγιστη ελάχιστη θερμοκρασία.
- Μέγιστη ελάχιστη Στάθμη.
- 4 ενδείξεις για κάλυψη επιπλέον αναγκών.

3x LED για την κατάσταση της συσκευής όπως

- *POW*: τροφοδοσία συσκευής
- *CON*: Σύνδεση με τον εξυπηρετητή και λήψη δεδομένων.
- *SEND*: Αποστολή δεδομένων στον εξυπηρετητή



Εικόνα 6.4 Τα LED ενημέρωσης των καταστάσεων της συσκευής (είσοδοι, έξοδοι, ειδοποιήσεις, κατάσταση συσκευής).

6.3.4 Είσοδοι της συσκευής.

Έχει τέσσερις ψηφιακές εισόδους με την ονομασία “INPUTS” αριθμώντας από το 1-4 με μέγιστη τάση εισόδου τα 5V και μέγιστο ρεύμα τα 40mA. Σε διαφορά δυναμικού από 2-5V λογικό ‘1’ και διαφορά δυναμικού από 2-0V λογικό ‘0’.

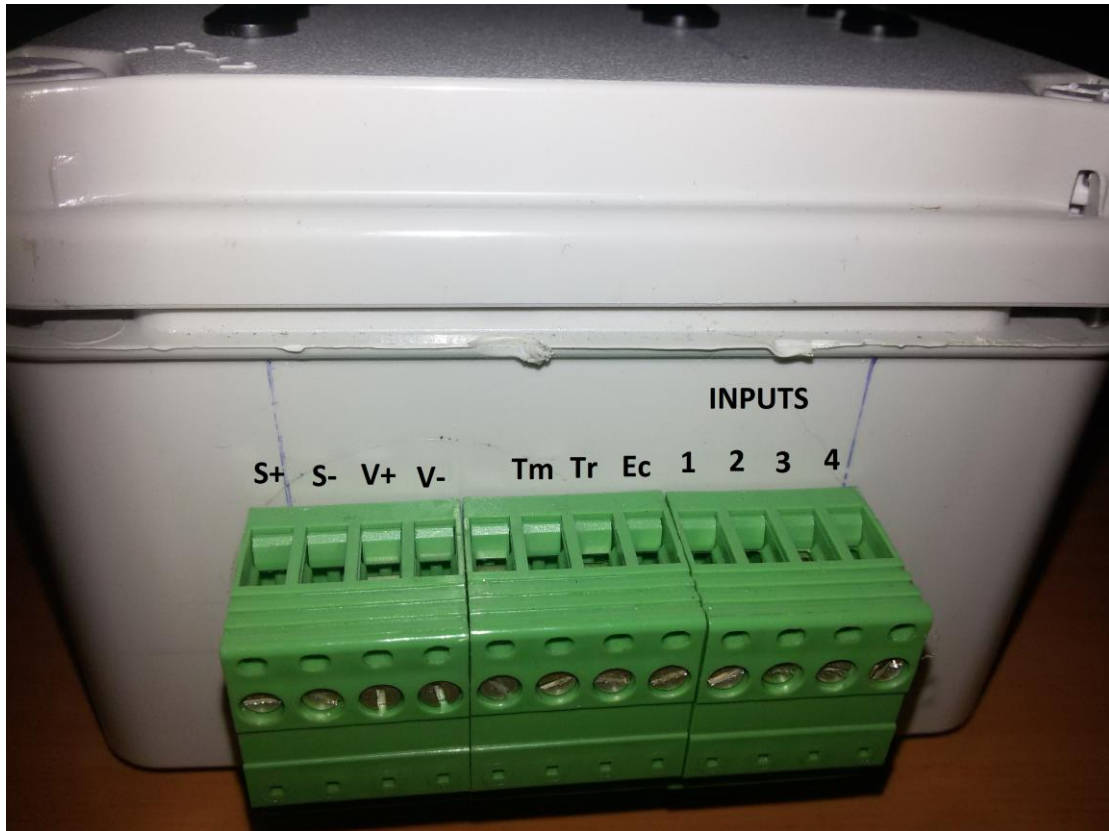
6.3.5 Είσοδοι Αισθητήρων.

Υποστηρίζει την σύνδεση δύο αισθητήρων:

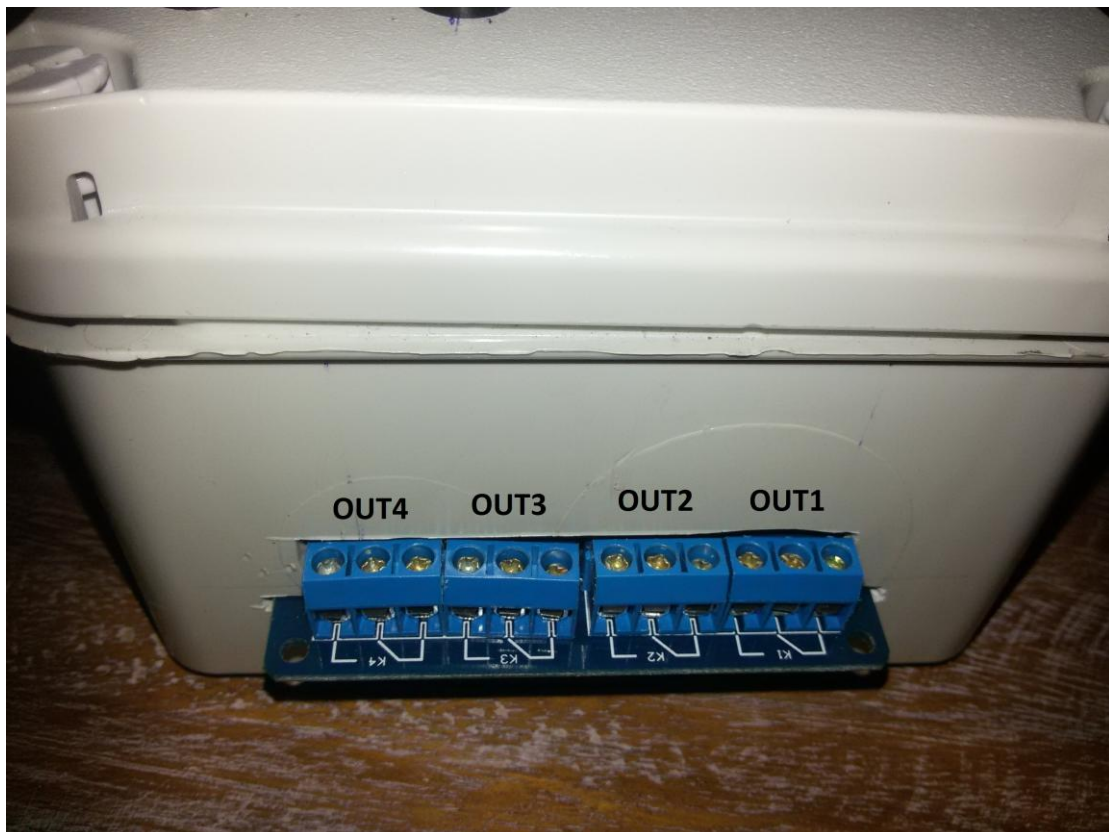
- Ψηφιακό αισθητήρα θερμοκρασίας DS18B20. Μετράει θερμοκρασίες από 0-100. Συνδέουμε το μεσαίο ακροδέκτη του αισθητήρα στην είσοδο με την σήμανση Tm και τους άλλους δύο ακροδέκτες στην τροφοδοσία.
- Ψηφιακό αισθητήρα Υπερήχων HC-SR04. Μετράει αποστάσεις από 1-200cm. Συνδέουμε τους ακροδέκτες της μονάδας υπερήχων «Trigger» και «Echo» στις εισόδους της συσκευής με την σήμανση «Tr» και «Ec».

6.3.6 Τροφοδοσία άλλων συσκευών ή αισθητήρων.

Οι κλέμες με την σήμανση “S+”-“S-” παρέχουν τροφοδοσία τάσης ανάλογης με την τάση τροφοδοσίας της συσκευής. Οι θέσεις “S” της κλέμα τροφοδοτούνται απευθείας από το βύσμα τροφοδοσίας της συσκευής ενώ οι κλέμες με την σήμανση “V+”, “V-” παρέχουν σταθεροποιημένη τροφοδοσία 5V και μέγιστο ρεύμα 1A. Οι κύρια εφαρμογή αυτών των ακροδεκτών είναι να ενεργοποιούν τις εισόδους IN1-IN4 και την τροφοδοσία των αισθητήρων.



Εικόνα 6.5 Η πλάγια δεξιά όψη της συσκευής με τις κλέμες εισόδων, αισθητήρων, τροφοδοσίας.



Εικόνα 6.6 Η πλάγια αριστερή όψη της συσκευής με τις εξόδους των ηλεκτρονόμων.

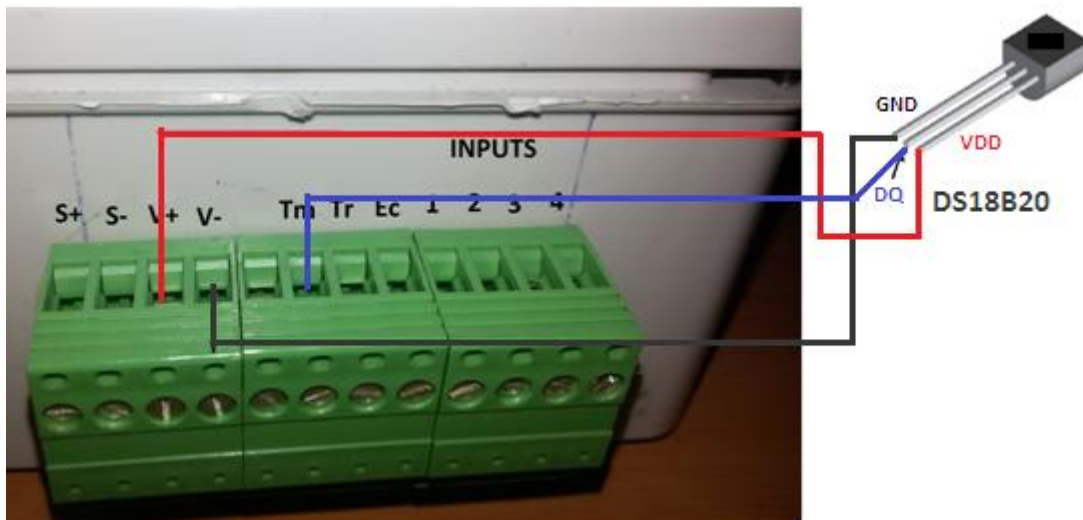
6.3.7 Έξοδοι της συσκευής.

Έχει τέσσερις εξόδους OUT1, OUT2, OUT3, OUT4 οδηγούμενες από ηλεκτρονόμους 10A στα 220 V AC και 10 A στα 30V DC. Μπορούν να λειτουργήσουν και στις δύο καταστάσεις Κανονικά-Ανοικτή (NO) και Κανονικά-Κλειστή(NC).

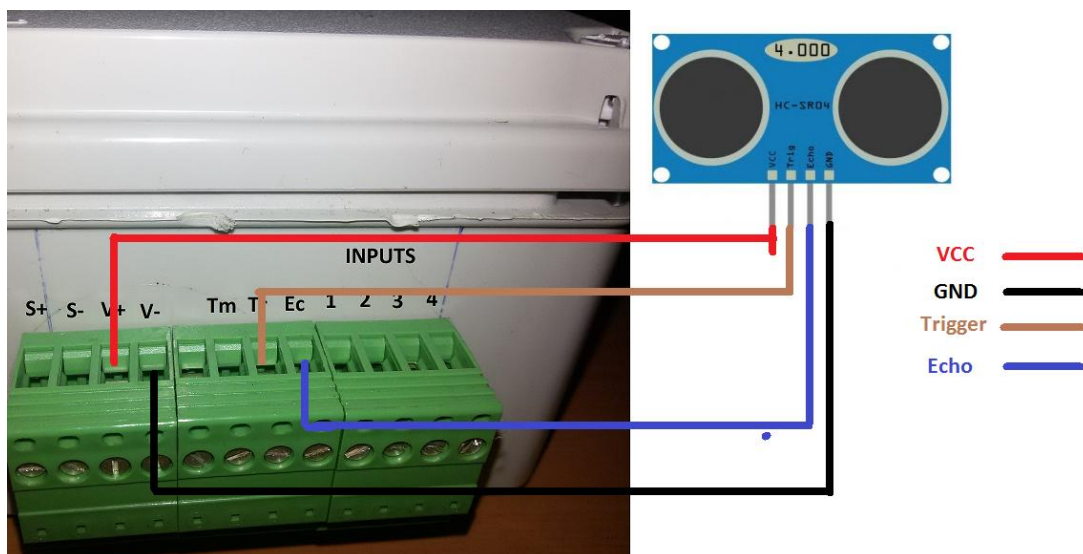
6.4 Οδηγός έναρξης.

Συνδέουμε τον αισθητήρα θερμοκρασίας DS18B20 όπως δείχνει η εικόνα 6.7 με τον μεσαίο ακροδέκτη συνδεδεμένο στην κλέμα με την ονομασία «Tm» και τους άλλους δύο τον θετικό ακροδέκτη του αισθητήρα στην κλέμα με την ονομασία «V+» και τον αρνητικό στην κλέμα «V-» όπως στην εικόνα 6.7.

Συνδέουμε τον αισθητήρα υπερήχων HC-SR04 με τους ακροδέκτες της μονάδας «Trig», «Ech», «Vcc» και «GND» να συνδέονται τις κλέμες «Tr», «Ec», «V+» και «V-» όπως φαίνεται στην εικόνα 6.8.



Εικόνα 6.7 Αναπαράσταση της σύνδεση αισθητήρα θερμοκρασίας στην συσκευή.



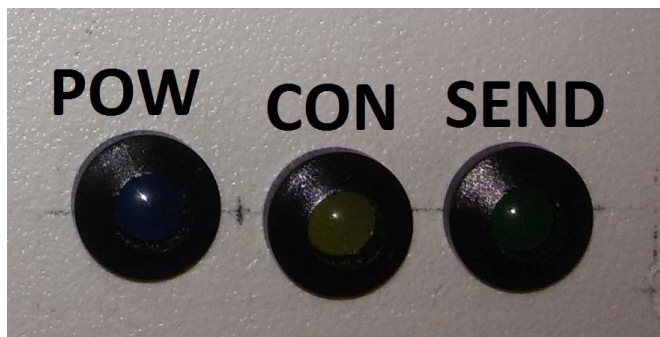
Εικόνα 6.8 Αναπαράσταση της σύνδεση αισθητήρα υπερήχων στην συσκευή.

Τις συσκευές που θέλουμε να ελέγξουμε τις συνδέουμε στους ακροδέκτες εξόδων με την σήμανση OUT1-OUT4. Υποστηρίζοντας τον χειρισμό μέχρι και 4 εξόδων.

Η ενεργοποίηση των εισόδων γίνεται είτε με εξωτερική τροφοδοσία εφαρμόζοντας υψηλό δυναμικό τάσης 5V στους ακροδέκτες με την σήμανση IN1-IN4. Είτε εφαρμόζοντας τάση απευθείας από τον ακροδέκτη με την σήμανση V+ από την κλέμα εξόδου της συσκευής, στις εισόδους IN1-IN4. Είσοδος ενεργοποιείται σε υψηλό δυναμικό .

Βιδώνουμε την κεραία στην υποδοχή SMA που βρίσκεται στην πίσω όψη της συσκευής. Τροφοδοτούμε με τάση 9-12V DC την συσκευή συνδέοντας την πηγή τροφοδοσίας στο βύσμα τροφοδοσίας με τον θετικό πόλο στο κέντρο, ή απευθείας στις κλέμες με την σήμανση S+ θετικός πόλος και S- η Γείωση.

Πατώντας το κουμπί έναρξης ανάβει το LED με την ένδειξη POW και μας ενημερώνει ότι η συσκευή μας είναι τροφοδοτημένη με τάση και είναι σε λειτουργία .



Εικόνα 6.9 Τα τρία LED ενημέρωσης κατάστασης της συσκευής.

Στην συνέχεια τα LED της δεύτερης στήλης των ειδοποιήσεων ενεργοποιούνται το ένα μετά το άλλο. Από την στιγμή που θα ενεργοποιήσουμε την συσκευή μας μέχρι και να συνδεθεί στο Διαδίκτυο και στην συνέχεια στον απομακρυσμένο εξυπηρετητή μεσολαβούν κάποιες διαδικασίες που τρέχει η συσκευή. Κατά την εκτέλεση αυτών των διαδικασιών ενεργοποιείται και το αντίστοιχο LED όπως παρατίθενται στον πίνακα 6.1. Σκοπός της λειτουργίας είναι τα LED να ενημερώνουν τον χρήστη για την εξέλιξη των διαδικασιών μέχρι να ολοκληρωθεί η σύνδεση στο Διαδίκτυο αλλά και να ενημερωθεί σε περίπτωση μια αστοχίας ή ενός σφάλματος σε ποια διαδικασία υπήρξε το πρόβλημα.

Μετά από μια επιτυχή σύνδεση της συσκευής στο διαδίκτυο τα LED των ειδοποιήσεων απενεργοποιούνται και χρησιμοποιούνται πλέον για την αναπαράσταση των ειδοποιήσεων της συσκευής. Το LED με την ένδειξη CON ενεργοποιείται κάθε φορά που λαμβάνει δεδομένα από την σελίδα-εξυπηρετητή και απενεργοποιείται όταν ολοκληρωθεί η επικοινωνία.

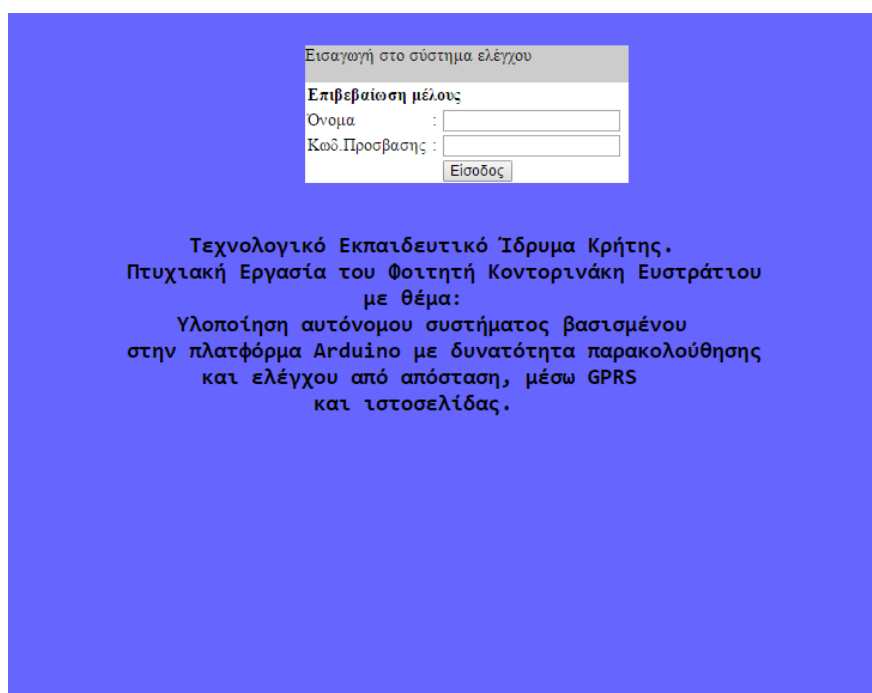
6.5 Σύνδεση με την ιστοσελίδα για τον έλεγχο και επιτήρηση της συσκευής από απόσταση.

Για τον απομακρυσμένο έλεγχο και επιτήρηση της συσκευής έχει δημιουργηθεί ένα περιβάλλον ιστοσελίδας του πελάτη, εύκολη και απλή στην χρήση από τον

εξουσιοδοτημένο χρήστη. Για την εισαγωγή μας στο σύστημα πληκτρολογούμε την διεύθυνση IP που βρίσκεται ο εξυπηρετητής και φιλοξενεί την σελίδα του χρήστη. Μετά την εισαγωγή θα σας εμφανιστεί ένα παράθυρο επικοινωνίας όπως της εικόνα 6.10 που θα ζητάει να δώσουμε το όνομα χρήστη και τον κωδικό στα πλαίσια με την ονομασία “Όνομα” και “Κωδ.Πρόσβασης”.

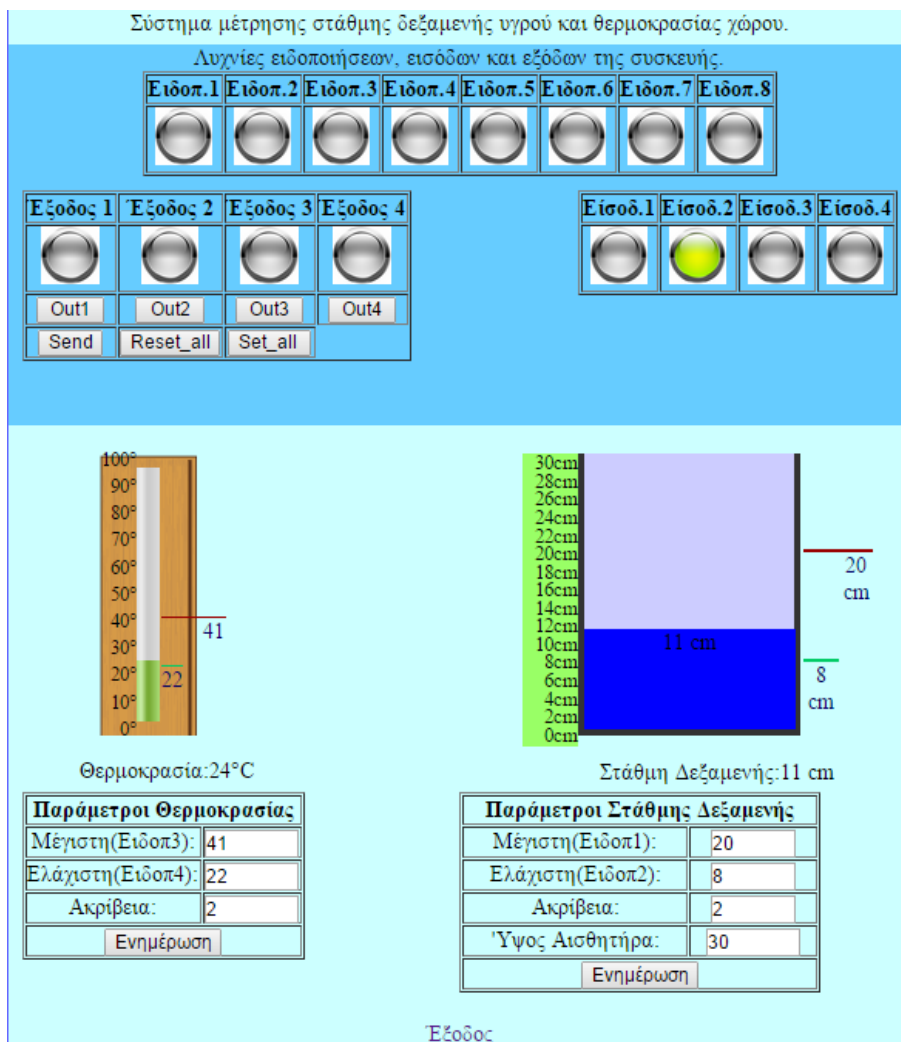
Ειδοποιήσεις (Alarms)	Εκτελέσιμες λειτουργίες.
AL1	Έλεγχος του σήματος σύνδεσης.
AL2	Σύνδεση στο GSM δίκτυο.
AL3	Ενεργοποίηση του GPRS.
AL4	Λειτουργία συσκευής ως εξυπηρετητής πολλαπλών συνδέσεων.
AL5	Εισαγωγή APN δεδομένων.
AL6	Διάβασμα διεύθυνσης IP της συσκευής.
AL7	Αποστολή της IP της συσκευής στον εξυπηρετητή.
AL8	Σύνδεση με τον απομακρυσμένο εξυπηρετητή.

Πίνακας 6.1 Αντιστοίχιση των LED ειδοποιήσεων με τις εκτελέσιμες λειτουργίες κατά την εκκίνηση της συσκευής.



Εικόνα 6.10 Το περιβάλλον σύνδεσης του χρήστη με το σύστημα.

Εάν τα δεδομένα που δώσαμε στην φόρμα επιβεβαίωσης μέλους αντιστοιχούν σε κάποιο εξουσιοδοτημένο μέλος πατώντας το κουμπί «Είσοδος» μεταβαίνουμε στην κύρια σελίδα διαχείρισης του συστήματος μέτρησης στάθμης και θερμοκρασίας δεξαμενής υγρού. Στην εικόνα 6.11 βλέπουμε το γραφικό περιβάλλον της σελίδας του χρήστη.

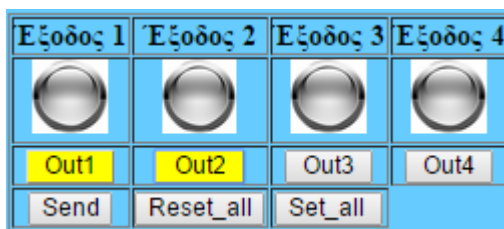


Εικόνα 6.11 Το περιβάλλον της κεντρικής σελίδας που αλληλεπιδρά ο χρήστης με το σύστημα.

6.5.1 Απομακρυσμένος έλεγχος των εξόδων.

Για τον απομακρυσμένο έλεγχο των εξόδων της συσκευής ενεργοποιούμε τα αντίστοιχα κουμπιά των εξόδων που θέλουμε να αλλάξουμε την τρέχουσα κατάσταση τους. Τα κουμπιά αυτά έχουν τις ονομασίες «Out1-Out4» και από την στιγμή που ενεργοποιηθούν κιτρινίζουν ειδοποιώντας μας ότι είναι ενεργοποιημένα ενώ αν τα ξανά επιλέξουμε απενεργοποιούνται επιστρέφοντας στην αρχική τους κατάσταση. Επιπλέον με την ενεργοποίηση του κουμπιού «Set_all» ενεργοποιούμε ταυτόχρονα και τα τέσσερα κουμπιά εναλλαγής εξόδου ενώ με το κουμπί «Reset_all» απενεργοποιούμε ταυτόχρονα και τα τέσσερα κουμπιά. Τέλος το κουμπί με την ονομασία «Send» στέλνει τις αλλαγές των εξόδων στο Arduino. Μετά από την αποστολή των εξόδων το κουμπί «Send» απενεργοποιείται χωρίς να μπορεί να

επιλέγει από τον χρήστη και αλλάζει ονομασία σε «Loading» δηλώνοντας ότι είναι σε κατάσταση αναμονής περιμένοντας να ολοκληρωθεί η επικοινωνία επιτυχώς. Εφόσον τα δεδομένα σταλούν ξανά επιστρέφει στην αρχική του κατάσταση ενώ σε περίπτωση που υπάρξει κάποιο πρόβλημα στην επικοινωνία εμφανίζει ένα μήνυμα σφάλματος «Υπήρξε σφάλμα στην επικοινωνία με το Arduino.» και τον λόγο που υπήρξε σφάλμα. Ο χρήστης πληροφορείται για την λειτουργία κάθε κουμπιού αν αφήσει το ποντίκι πάνω από το κουμπί για μερικά δευτερόλεπτα στην συνέχεια εμφανίζεται ένα σύντομο μήνυμα που εξηγεί την λειτουργία του κουμπιού.



Εικόνα 6.12 Ενεργοποίηση των κουμπιών «Out1-Out2» για εναλλαγή των καταστάσεων εξόδων.

6.5.2 Ρυθμίσεις παραμέτρων Στάθμης

Στο πλαίσιο κάτω αριστερά με την ονομασία “Παράμετροι Στάθμης Δεξαμενής” μπορούμε να δηλώσουμε παραμέτρους για την μέτρηση στάθμης.

- **Μέγιστη(Ειδοπ1):** Ορίζουμε την μέγιστη επιτρεπτή στάθμη σε cm. Αν η στάθμη του υγρού ξεπεράσει την τιμή της μέγιστη στάθμης που έχουμε ορίσει, τότε ενεργοποιείται η φωτεινή ένδειξη μέγιστης στάθμης ανάβοντας το LED με την ονομασία Ειδοπ.1.
- **Ελάχιστη(Ειδοπ2):** Ορίζουμε την ελάχιστη επιτρεπτή στάθμη σε cm. Αν η στάθμη του υγρού πέσει κάτω την τιμή της ελάχιστης στάθμης που έχουμε ορίσει, τότε ενεργοποιείται η φωτεινή ένδειξη ελάχιστης στάθμης ανάβοντας το LED με την ονομασία Ειδοπ.2.
- **Ακρίβεια:** Η ακρίβεια μέτρησης της στάθμης ορίζει την ακρίβεια βαθμονόμησης της εικονικής δεξαμενής .για παράδειγμα με ακρίβεια=2cm παίρνουμε τις μετρήσεις στάθμης 20, 22, 24 κ.α.
- **Ύψος Αισθητήρα:** Η απόσταση που βρίσκεται το αισθητήριο από τον πυθμένα της δεξαμενής. Επίσης ορίζει την μέγιστη τιμή βαθμονόμησης της εικονικής δεξαμενής.

Οι τιμές της στάθμης είναι από 1-200cm.

Παράμετροι Στάθμης Δεξαμενής	
Μέγιστη(Ειδοπ1):	19
Ελάχιστη(Ειδοπ2):	7
Ακρίβεια:	1
Ύψος Αισθητήρα:	30
Ενημέρωση	

Εικόνα 6.13 Η φόρμα επικοινωνίας για την ρύθμιση των παραμέτρων της στάθμης δεξαμενής.

6.5.3 Ρυθμίσεις παραμέτρων Θερμοκρασίας

Κάτω από το εικονικό θερμόμετρο βρίσκεται το πλαίσιο θερμοκρασίας στο οποίο μπορούμε να δηλώσουμε τις παραμέτρους για την θερμοκρασία. Αυτές οι παράμετροι είναι μέγιστη, ελάχιστη θερμοκρασία, ακρίβεια μέτρησεις θερμοκρασίας.

- *Μέγιστη(Ειδοπ3):* Εισάγουμε την μέγιστη επιτρεπτή θερμοκρασία. Αν η θερμοκρασία της συσκευής ξεπεράσει την μέγιστη θερμοκρασία τότε ενεργοποιείται η φωτεινή ένδειξη μέγιστης θερμοκρασίας ανάβοντας το LED με την ονομασία Ειδοπ.3.
- *Ελάχιστη(Ειδοπ4):* Εισάγουμε την ελάχιστη επιτρεπτή θερμοκρασία. Αν η θερμοκρασία της συσκευής ξεπεράσει την μέγιστη θερμοκρασία τότε ενεργοποιείται η φωτεινή ένδειξη μέγιστης θερμοκρασίας ανάβοντας το LED με την ονομασία Ειδοπ.4.
- *Ακρίβεια:* Η ακρίβεια μέτρησης που ανιχνεύει την αλλαγή θερμοκρασίας. Οι τιμές της θερμοκρασίας είναι από 0-100 C

Παράμετροι Θερμοκρασίας	
Μέγιστη(Ειδοπ3):	41
Ελάχιστη(Ειδοπ4):	25
Ακρίβεια:	2
Ενημέρωση	

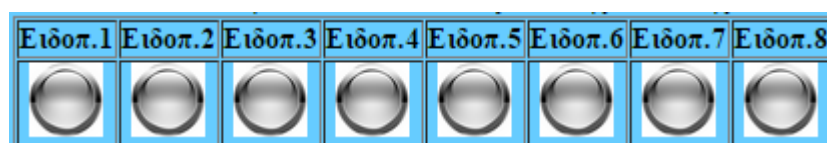
Εικόνα 6.14 Η φόρμα επικοινωνίας για την ρύθμιση των παραμέτρων της θερμοκρασίας.

6.5.4 Ενημέρωση της κατάστασης της συσκευής

Εκτός από τον απομακρυσμένο έλεγχο της συσκευής η ιστοσελίδα του πελάτη μάς δίνει την δυνατότητα να επιτηρήσουμε την κατάσταση στην οποία βρίσκεται η συσκευή. Για την εύκολη ενημέρωση του πελάτη για την κατάσταση της συσκευής έχουν δημιουργηθεί εικονικές αναπαραστάσεις LED προσομοιάζοντας τα LED της συσκευής .

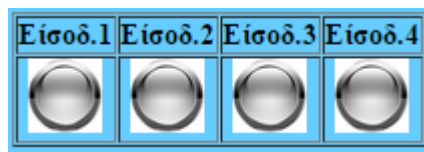
Έχουμε τρεις σειρές από εικονικές αναπαραστάσεις LED όπως αναφέρονται παρακάτω:

- *Ειδοποιήσεων:* Οι καταστάσεις των ειδοποιήσεων της συσκευής αναπαρίστανται στην σειρά με τα εικονικά LED με την σήμανση “Ειδοπ.1-Ειδοπ.8” για τις 8 ειδοποιήσεις.



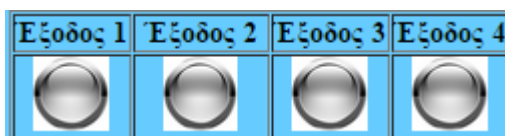
Εικόνα 6.15 Εικονική αναπαράσταση των LED των ειδοποιήσεων της ιστοσελίδας.

- *Εισόδων:* Οι καταστάσεις των εισόδων της συσκευής αναπαρίστανται στην σειρά με τα εικονικά LED με την σήμανση “Είσοδ.1- Είσοδ.4” για τις 4 εισόδους.



Εικόνα 6.16 Εικονική αναπαράσταση των LED των εισόδων της ιστοσελίδας.

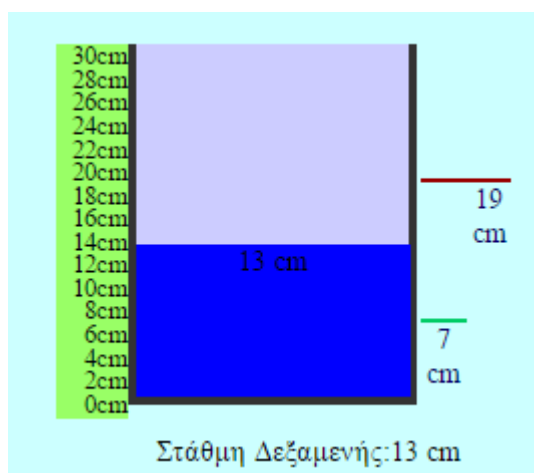
- *Εξόδων:* Οι καταστάσεις των εξόδων της συσκευής αναπαρίστανται στην σειρά με τα εικονικά LED με την σήμανση “Εξοδ.1- Έξοδ.4” για τις 4 εξόδους.



Εικόνα 6.17 Εικονική αναπαράσταση των LED των εξόδων της ιστοσελίδας.

6.5.5 Μέτρηση Στάθμης Δεξαμενής.

Ο χρήστης ενημερώνεται για την στάθμη υγρού της μετρούμενης δεξαμενής μέσα από το εικονικό περιβάλλον μιας εικονικής δεξαμενής. Η δεξαμενή αυτή προσομοιάζει την πραγματική δεξαμενή μεταβάλλοντας την στάθμη ανάλογα με την μέτρηση στάθμης της πραγματικής δεξαμενής. Επίσης ο χρήστης μπορεί να ορίσει ένα μέγιστο και ένα ελάχιστο επίπεδο στάθμης, την ακρίβεια μέτρησης-βαθμονόμησης καθώς και το μέγιστο ύψος. Και τα τέσσερα ορίσματα μπορούν να αλλαχτούν από τον χρήστη μέσα από την φόρμα αλλαγής παραμέτρων στάθμης δεξαμενής. Στην εικόνα 6.18 αναπαρίστανται η εικονική δεξαμενή.

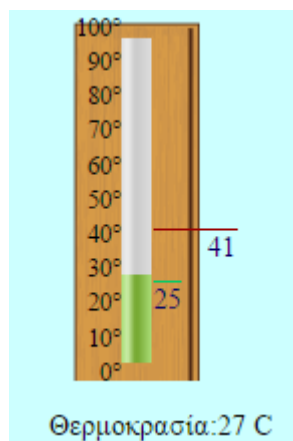


Εικόνα 6.18 Εικονική αναπαράσταση της δεξαμενής της σελίδα..

6.5.6 Μέτρηση Θερμοκρασίας Περιβάλλοντος.

Ο χρήστης ενημερώνεται για την θερμοκρασία περιβάλλοντος που βρίσκεται η συσκευή μέσα από το εικονικό περιβάλλον ενός θερμομέτρου. Η πράσινη μπάρα του θερμομέτρου αλλάζει ύψος σύμφωνα με ισχύουσα θερμοκρασία που μετράει ο

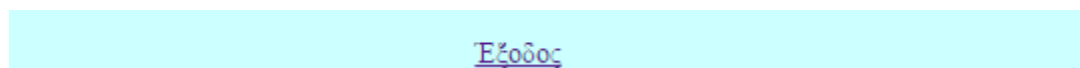
αισθητήρας. Επίσης όπως και στην περίπτωση της εικονικής δεξαμενής ο χρήστης μπορεί να ορίσει τα επίπεδα μέγιστης και ελάχιστης θερμοκρασίας καθώς και την ακρίβεια μέτρησης της θερμοκρασίας, μέσα από την φόρμα αλλαγής παραμέτρων της θερμοκρασίας. Στην εικόνα 6.19 αναπαρίσταται το εικονικό θερμόμετρο της σελίδας του χρήστη.



Εικόνα 6.19 Εικονική αναπαράσταση του θερμόμετρου της σελίδας..

6.5.7 Έξοδος από την ιστοσελίδα του χρήστη.

Στην κάτω θέση βρίσκεται ένα σύνδεσμος με το όνομα “Έξοδος” πατώντας πάνω στον σύνδεσμο κάνουμε έξοδο από την σελίδα του πελάτη. Για να μπορούμε ξανά στη σελίδα του χρήστη πρέπει εκ νέου να δώσουμε τα στοιχεία που μας ζητάει η φόρμα επικοινωνίας στην αρχική σελίδα εισαγωγής .



Εικόνα 6.19 Σύνδεσμος για την έξοδο από το σύστημα.

7. Συμπεράσματα και μελλοντικές επεκτάσεις.

Η πτυχιακή εργασία που υλοποιήθηκε επιτρέπει στον χρήστη μέσα από μια ιστοσελίδα να επιτηρεί την κατάσταση του συστήματος και των συσκευών που ελέγχονται από αυτό. Επιπλέον ο χρήστης μπορεί να αλλάξει την κατάσταση των διαφόρων συσκευών και να ενημερωθεί σε πραγματικό χρόνο για τις μετρήσεις της στάθμης και της θερμοκρασίας.

Το σύστημα της εργασίας πλεονεκτεί σε πολλά σημεία έναντι άλλων παρόμοιων συστημάτων του εμπορίου. Το σύστημα στηρίζεται στην τεχνολογία GSM/GPRS επιτρέποντας την ασύρματη επικοινωνία από οποιαδήποτε σημείο έχει κάλυψη δικτύου. Έτσι δεν περιορίζεται στο σημείο που θα εγκατασταθεί βρίσκοντας εφαρμογή σε συστήματα που δεν παρέχουν εγκαταστημένη σύνδεση δικτύου όπως σε βουνά απόμερα μέρη και περιοχές που η πρόσβαση δεν είναι εύκολη.

Αποτελεί ένα αυτόνομο σύστημα το οποίο μετά από απρόσμενη διακοπή ή επανεκκίνηση του συστήματος δεν χάνει τις καταστάσεις που είχε πριν το συμβάν αλλά επανέρχεται αυτόματα στην προηγούμενη κατάσταση. Αυτή η λειτουργία του δίνει αξιοπιστία και αντοχή στα διάφορα απρόσμενα συμβάντα που θα δημιουργηθούν.

Έχει τροφοδοτηθεί με λειτουργία αυτόματης επανεκκίνησης μετά από 3 ανεπιτυχείς προσπάθειες σύνδεσης με το Διαδίκτυο. Διατηρώντας την συσκευή λειτουργική και μετά από την παρουσία σφάλματος στη σύνδεση ακόμα και στο ίδιο το δίκτυο.

Ο προγραμματισμός της ιστοσελίδας έχει βασιστεί σε τεχνολογίες διαδικτύου επιτρέποντας τον απομακρυσμένο έλεγχο και την εποπτεία του συστήματος από οποιαδήποτε συσκευή συνδέεται στο Διαδίκτυο και μπορεί να ανοίξει μία ιστοσελίδα. Για παράδειγμα κινητό τηλέφωνο, φορητό υπολογιστή και επιτραπέζια συσκευή (tablet).

Τέλος το κόστος κατασκευής είναι αρκετά χαμηλό σε σύγκριση με άλλα συστήματα και βασίζεται σε πλατφόρμα ανοιχτού κώδικα έτσι μπορεί να επεκταθεί και παραπάνω.

7.1 Προβλήματα που επιλύθηκαν κατά την υλοποίηση της πτυχιακής εργασίας

Κατά την διάρκεια εκπόνησης της πτυχιακής εργασίας παρουσιάστηκε πλήθος προβλημάτων και δυσλειτουργιών με τα οποία ήρθαμε αντιμέτωποι άλλα μετά από σκέψη και περεταίρω μελέτη βρήκαμε τρόπους επίλυσής τους.

- Στην αρχή σαν πρώτη σκέψη ήταν η χρήση του Arduino Uno αντί του Arduino Mega λόγω μικρής χωρητικότητας σε μνήμη SRAM και σε αριθμό ακροδεκτών επιλέχθηκε το Arduino Mega καλύπτοντας αυτές τις ανάγκες.
- Μετά από επανεκκίνηση ή επανέναρξη του συστήματος έχανε τις προηγούμενες καταστάσεις των εισόδων και των εξόδων με αποτέλεσμα να χρειάζεται να επικοινωνήσει με τον εξυπηρετητή για να λάβει εκ νέου τις προηγούμενες τιμές και να επανέλθει στην προηγούμενη κατάσταση που βρισκόταν πριν. Το πρόβλημα αυτό λύθηκε με την αποθήκευση των τιμών αυτών σε μνήμη EEPROM που

διαθέτει το Arduino και σε κάθε αλλαγή κατάστασης ενημερώνονται και οι τιμές στην EEPROM.

- Η ασπίδα GSM τροφοδοτούταν από τον ακροδέκτη τροφοδοσίας “5V” του Arduino, ο οποίος δεν μπορούσε να ξεπεράσει το 1.5 Αμπρ ρεύμα τροφοδοσίας. Για τον λόγο αυτό το ρεύμα που κατανάλωνε η ασπίδα κατά την αποστολή δεδομένων ξεπέρασε στιγμιαία τα 1,5Α και έτσι η ασπίδα ξεκινούσε διαδικασία επανεκκίνησης. Κατά την επανεκκίνηση η ασπίδα έχανε την σύνδεση με το Διαδίκτυο και έτσι δεν μπορούσε να λειτουργήσει ο απομακρυσμένος έλεγχος. Λύση στο πρόβλημα αυτό έδωσε η κατασκευή εξωτερικού τροφοδοτικού που τροφοδοτεί ξεχωριστά την μονάδα GSM με μέγιστο ρεύμα μέχρι και 3Α χωρίς να επιβαρύνει την τροφοδοσία του Arduino.
- Για να μπορέσουμε να μειώσουμε στο ελάχιστο την μετάδοση δεδομένων του Arduino με τον εξυπηρετητή αποφεύχθηκε η χρήση έτοιμης βιβλιοθήκης της ασπίδας GSM. Με την χρήση ειδικών εντολών και δημιουργία νέων συναρτήσεων καλύψαμε τις λειτουργίες που χρειαστήκαμε με ελάχιστες απαιτήσεις σε δεδομένα.
- Κατά την έναρξη του συστήματος μέχρι και την σύνδεση με του εξυπηρετητή περνάει διάφορα στάδια με αποτέλεσμα την χρονική καθυστέρηση και πόσο μάλλον αν υπάρξει και δυσλειτουργία σε κάποιο από τα στάδια ο χρόνος αυτός παρατείνεται. Για αυτόν τον λόγο επινοήθηκε η λειτουργία έλεγχος κατάστασης έναρξης με την οποία σε κάθε μετάβαση από το ένα στάδιο στο άλλο ενεργοποιείται και ένα LED των καταστάσεων της ειδοποίησης, παρέχοντας έτσι μια οπτική παρακολούθηση των σταδίων έναρξης όπως και συγκεκριμενοποίηση σε περίπτωση βλάβης.
- Η εταιρεία κινητής τηλεφωνίας με την χρήση απλής σύνδεσης εμπόδιζε την σύνδεση της ασπίδας ως εξυπηρετητή επιτρέποντας μόνο την επικοινωνία ως πελάτης. Έτσι για να μπορέσουμε να ελέγξουμε το σύστημα μας μέσα από την ιστοσελίδα χρειαζόταν να στέλνει το Arduino επανειλημμένα αιτήματα σύνδεσης με τον εξυπηρετητή για να μπορέσει να ανιχνεύσει μια αλλαγή από τον χρήστη. Αυτό είχε ως αποτέλεσμα την άσκοπη κατανάλωση σε δεδομένα. Η λύση δόθηκε με την χρήση κάρτας SIM machine to machine όπου μας επιτρέπει την σύνδεση της ασπίδας ως εξυπηρετητή.

7.2 Μελλοντικές επεκτάσεις

Κατά την πορεία της μελέτης και της υλοποίησης της πτυχιακής εργασίας προέκυψαν σκέψεις για την αναβάθμιση και επέκταση του συστήματος.

Μια μελλοντική επέκταση θα ήταν να δημιουργηθεί ένας ιστοχώρος με ιεραρχία πελατών θα υπάρχει ο διαχειριστής και ο απλός πελάτης με τον διαχειριστή να έχει την εποπτεία των πελατών ακόμα να μπορεί να εισάγει ένα πελάτη στο σύστημα ή και να διαγράψει. Η βάση με το υπάρχον σύστημα έχει είδη δημιουργηθεί και λόγω χρήσης βάσης δεδομένων και η χρήση εργαλείων που υλοποιούνται για την δημιουργία δυναμικής ιστοσελίδας κάνει πιο εύκολη την επίτευξη αυτού του στόχου.

Μια άλλη μελλοντική επέκταση είναι να δημιουργηθεί λογισμικό το οποίο θα τρέχει σε συσκευές Android πίσω από άλλα προγράμματα παρέχοντας την συνεχή εποπτεία του συστήματος ή ακόμα και εκτός σύνδεσης με την χρήση τεχνολογίας SMS. Η ασπίδα υποστηρίζει την αποστολή και λήψη γραπτών μηνυμάτων τεχνολογίας SMS υλοποιώντας μόνο το κομμάτι του λογισμικού είναι δυνατόν να υλοποιηθεί αυτός ο στόχος. Ένα πρόγραμμα το οποίο δεν χρειάζεται ιδιαίτερες γνώσεις την τεχνολογία Android είναι το APP Inventor το οποίο σχεδιασμένο για αρχάριους στον προγραμματισμό και με εύκολο στην χρήση και απλό περιβάλλον μπορούμε να δημιουργήσουμε μια Android εφαρμογή.

Μια ακόμα μελλοντική επέκταση θα ήταν η προσθήκη εξωτερικής κάμερας στο υπάρχον σύστημα δίνοντάς μας και μια εικόνα για τον χώρο επίβλεψης του συστήματος. Αυτή η επέκταση θα είχε μεγάλο ποσοστό εφαρμογής κυρίως σε συστήματα ασφαλείας .

Ανάλογα με τις απαιτήσεις του συστήματος μια άλλη μελλοντική επέκταση και εύκολη στην υλοποίηση είναι η αύξηση του αριθμού των αισθητήρων που διαχειρίζεται το σύστημα. Αισθητήρες που θα μετράνε την βαρομετρική πίεση, την θερμοκρασία υγρού, της υγρασίας, την ροή των υγρών θα αύξανε κατά πολύ το εύρος των συστημάτων που θα εύρισκε εφαρμογή το σύστημα της παρούσας εργασίας.

Βιβλιογραφία.

- [1] <http://www.ardumotive.com/about-arduino.html>
- [2] el.wikipedia.org/wiki/Arduino
- [3] <http://www.arduino.cc/en/Main/Products>
- [4] <http://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- [5] <http://www.arduino.cc/en/Main/ArduinoGSMShield>
- [6] <http://imall.iteadstudio.com/im120417009.html>
- [7] <ftp://imall.iteadstudio.com/IM120417009 IComSat/DS IM120417009 IComSat.pdf>
- [8] <http://www.arduino.cc/en/Guide/Windows>
- [9] <http://www.arduino.cc/en/Guide/Environment>
- [10] <http://www.arduino.cc/en/Tutorial/Sketch>
- [11] <https://el.wikipedia.org/wiki/XAMPP>
- [12] <https://www.apachefriends.org/about.html>
- [13] <https://el.wikipedia.org/wiki/MySQL>
- [14] <https://notepad-plus-plus.org/download/v6.7.9.html>
- [15] <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [16] <http://playground.arduino.cc/Learning/OneWire>
- [17] <https://el.wikipedia.org/wiki/%CE%A5%CF%80%CE%AD%CF%81%CE%B7%CF%87%CE%BF%CF%82>
- [18] <http://www.micropik.com/PDF/HCSR04.pdf>
- [19] <http://arduinobasics.blogspot.gr/2012/11/arduinobasics-hc-sr04-ultrasonic-sensor.html>
- [20] <https://el.wikipedia.org/wiki/Ηλεκτρονόμος>
- [21] <http://www.sainsmart.com/4-channel-5v-relay-module-for-pic-arm-avr-dsp-arduino-msp430-ttl-logic.html>
- [22] <http://wiki.iteadstudio.com/IComSat>
- [23] <http://elecfreaks.com/store/download/datasheet/rf/SIM900/SIM900 AT%20Command%20Manual V1.03.pdf>
- [24] http://mtsystem.ru/sites/default/files/documents/sim900_tcpip_application_note_v1.02.pdf
- [25] https://el.wikipedia.org/wiki/Δίοδος_Εκπομπής_Φωτός
- [26] <http://www.arduino.cc/en/Tutorial/Blink?from=Tutorial.BlinkingLED>
- [27] <http://www.w3schools.com/html/>
- [28] <http://www.w3schools.com/js/default.asp>
- [29] <http://www.w3schools.com/php/default.asp>
- [30] <http://php.net/>