



Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής
Π.Μ.Σ "Πληροφορική"

Μεταπτυχιακή Διατριβή

| | |
|-----------------------|--|
| Τίτλος Διατριβής | Κατασκευή συστήματος καταγραφής, μελέτης και παρουσίασης θερμοκρασιών και πυρανίχνευσης με χρήση αισθητήρων |
| Όνοματεπώνυμο φοιτητή | Κατσάς Χρήστος |
| Πατρώνυμο | Στέφανος |
| Αριθμός Μητρώου | ΜΗΠΛ 10006 |
| Επιβλέπων | καθηγήτρια κ. Κατερίνα Καμπάση |

Διπλωματική Εργασία υποβληθείσα στο Τμήμα Πληροφορικής του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση Μεταπτυχιακού Διπλώματος Ειδίκευσης στην Πληροφορική

Σεπτέμβριος 2013

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

| | |
|---|-----------|
| ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ | 1 |
| ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ | 2 |
| Περίληψη | 4 |
| Abstract | 4 |
| ΚΕΦΑΛΑΙΟ 1 | 5 |
| Εισαγωγή | 5 |
| ΚΕΦΑΛΑΙΟ 2 | 6 |
| Ανασκόπηση πεδίου | 6 |
| 2.1 Projects από Ελλάδα: | 6 |
| 2.2 Projects από Εξωτερικό: | 8 |
| ΚΕΦΑΛΑΙΟ 3 | 9 |
| Παρουσίαση απαιτούμενων Hardware και Software | 9 |
| 3.1 Τι είναι οι μικροελεγκτές Arduino; | 9 |
| Τι είναι ο Arduino; | 9 |
| Γιατί να επιλέξω τον Arduino; | 10 |
| Το Υλικό (Hardware) του Arduino | 10 |
| Χαρακτηριστικά..... | 12 |
| Μνήμη..... | 12 |
| Τροφοδοσία | 12 |
| Επικοινωνία | 13 |
| 3.2 Hardware που χρησιμοποιήθηκε για την υλοποίηση της εργασίας | 13 |
| 3.3 Software που χρησιμοποιήθηκε στην υλοποίηση | 22 |
| ΚΕΦΑΛΑΙΟ 4 | 23 |
| Οδηγός Συνδεσμολογίας των Arduino parts και σύνδεση με Software | 23 |
| 4.1 Σύνδεση step by step των Arduino parts (hardware) | 23 |
| 4.2 Σύνδεση Πλακέτας Arduino με Windows (Step by Step διαδικασία) | 32 |
| Εγκατάσταση IDE | 32 |
| Εγκατάσταση οδηγού (driver) | 32 |
| 4.3 Arduino Software IDE - Processing IDE (Ανάλυση Περιβάλλοντος) | 33 |
| 4.4 Τρόπος αποθήκευσης των δεδομένων στην βάση-μεταφορά και ανάλυση περιβάλλοντος phpmyadmin | 35 |
| ΚΕΦΑΛΑΙΟ 5 | 40 |
| Παρουσίαση Συστήματος (User's manual)..... | 40 |
| 5.1 Έναρξη του προγράμματος (step-by-step διαδικασία) | 40 |
| 5.2 Περιγραφή και ανάλυση του interface του website | 43 |
| 5.3 Περιγραφή και ανάλυση του interface της Delphi εφαρμογής | 56 |
| ΚΕΦΑΛΑΙΟ 6 | 63 |
| Αρχιτεκτονική Συστήματος | 63 |
| 6.1 Αρχιτεκτονική προγράμματος για χρήση στην πλατφόρμα Arduino | 63 |
| 6.2 Μεταφορά δεδομένων και αποθήκευση στη βάση | 76 |
| 6.3 Διαδικασία, χαρακτηριστικά και τρόπους κατασκευής του website | 77 |
| ΚΕΦΑΛΑΙΟ 7 | 79 |
| Κατασκευή Delphi εφαρμογής | 79 |
| ΚΕΦΑΛΑΙΟ 8 | 83 |
| Συμπεράσματα | 83 |
| ΚΕΦΑΛΑΙΟ 9 | 84 |
| Μελλοντικές Επεκτάσεις | 84 |
| ΚΕΦΑΛΑΙΟ 10 | 85 |
| Βιβλιογραφία | 85 |
| Δικτυογραφία..... | 85 |

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

| | |
|---|----|
| Εικόνα 3-3 Πλακέτα Arduino Mega 2560..... | 14 |
| Εικόνα 3-4 Πλακέτα Arduino Ethernet για σύνδεση στο Internet..... | 14 |
| Εικόνα 3-5 Breadboard για την σύνδεση των αισθητήρων μας..... | 15 |
| Εικόνα 3-6 Ενδεικτικά καλώδια συνδέσεων | 16 |
| Εικόνα 3-7 Αισθητήρας υγρασίας DHT11 | 16 |
| Εικόνα 3-8 Αισθητήριο Θερμοκρασίας DS18B20 | 17 |
| Εικόνα 3-9 Αισθητήριο GAS και συνδεσμολογία | 17 |
| Εικόνα 3-10 Ενδεικτικές Αντιστάσεις..... | 18 |
| Εικόνα 3-11 LED για χρήση στον αισθητήρα στην χρήση ON/OFF | 18 |
| Εικόνα 3-12 Καλώδιο σύνδεσης Arduino Ethernet Shield με PC | 18 |
| Εικόνα 3-13 Καλώδιο Σύνδεσης Arduino με το PC..... | 19 |
| Εικόνα 3-14 Chip Xbee | 19 |
| Εικόνα 3-15 Regulators | 20 |
| Εικόνα 3-16 Αντιστάσεις..... | 20 |
| Εικόνα 3-17 Κεραμικοί πυκνωτές | 20 |
| Εικόνα 3-18 Μικροεπεξεργαστής AtMega 328..... | 21 |
| Εικόνα 3-19 Κρύσταλλος χρονισμού | 21 |
| Εικόνα 3-20 Ολοκληρωμένος ελεγκτής | 21 |
| Εικόνα 4-1 Σύνδεση του Ethernet Shield στην μητρική πλακέτα | 23 |
| Εικόνα 4-2 / 4-3 Schematic συνδεσμολογίας αισθητήρα DS18B20..... | 24 |
| Εικόνα 4-4 / 4-5 Εικόνες της συνδεσμολογίας του αισθητήρα DS18B20..... | 25 |
| Εικόνα 4-6 / 4-7 Schematic συνδεσμολογίας αισθητηρίου υγρασίας DHT11 | 26 |
| Εικόνα 4-8 Schematic συνδεσμολογίας αισθητηρίου GAS MQ-2 | 27 |
| Εικόνα 4-9 Εικόνα της συνδεσμολογίας αισθητηρίου GAS MQ-2 με επεξήγηση | 27 |
| Εικόνα 4-10 Συνδεσμολογία του LED ένδειξης ON/OFF..... | 28 |
| Εικόνα 4-11 Τελική μορφή του κυκλώματος του συστήματος αισθητήρων | 29 |
| Εικόνα 4-12 Schematic της standalone πλακέτας ΑΣΥΡΜΑΤΗΣ μεταφοράς δεδομένων..... | 30 |
| Εικόνα 4-13 Schematic της πλακέτας ΑΣΥΡΜΑΤΗΣ μεταφοράς δεδομένων που είναι ενσωματωμένη στο κεντρικό σύστημα αισθητήρων..... | 31 |
| Εικόνα 4-14 Καλώδιο Τροφοδοσίας μέσω θύρας USB στο PC | 32 |
| Εικόνα 4-14 Περιβάλλον Arduino IDE | 33 |
| Εικόνα 4-15 Περιβάλλον Processing IDE | 35 |
| Εικόνα 4-16: Στην εικόνα βλέπουμε την αρχική σελίδα της πλατφόρμας διαχείρισης και επιλέγουμε την βάση δεδομένων μας με τον τίτλο arduino_db | 37 |
| Εικόνα 4-17: Παρατηρούμε ότι η database αποτελείται από 1 table με την ονομασία data και επιλέγουμε περιήγηση για να δούμε τα επιμέρους στοιχεία του. | 37 |
| Εικόνα 4-18: Στην εικόνα αυτή βλέπουμε ένα screenshot από την βάση δεδομένων, το table της οποίας αποτελείται από τα εξής χαρακτηριστικά τα οποία ανανεώνονται αυτόματα όταν ο αισθητήρας αποστέλλει data | 38 |
| Εικόνα 4-19: Στην εικόνα βλέπουμε την δομή του table της βάσης μας με στοιχεία επιπλέον αυτών που αναφέραμε παραπάνω και αφορούν τα χαρακτηριστικά του table | 39 |
| Εικόνα 5-1 Διεύθυνση που μας ενδιαφέρει μέσα από το Command Prompt | 40 |
| Εικόνα 5-2 Συμπλήρωση ip και έυρεση ip επικοινωνίας Arduino με Internet | 41 |
| Εικόνα 5-3 Συμπλήρωση ip επικοινωνίας με Internet και έναρξη προγράμματος | 42 |
| Εικόνα 5-4 Έναρξη προγράμματος Processing για προγραμματισμό του LED | 43 |
| Εικόνα 5-5 Η διεύθυνση του Website της εφαρμογής..... | 43 |
| Εικόνα 5-6 Κεντρική σελίδα (HOME) του site | 44 |
| Εικόνα 5-7 Γραμμικό Διάγραμμα..... | 45 |
| Εικόνα 5-8 Γραμμικό Διάγραμμα αναλυτικό | 46 |
| Εικόνα 5-9 / 5-10 Γραμμικό Διάγραμμα από data ασύρματου αισθητήρα..... | 47 |
| Εικόνα 5-11 / 5-12 Ραβδόγραμμα από data ενσύρματου αισθητήρα..... | 48 |
| Εικόνα 5-13 / 5-14 Ραβδόγραμμα από data Ασύρματου αισθητήρα | 49 |
| Εικόνα 5-15 Οθόνη με Settings του site | 50 |

| | |
|---|----|
| Εικόνα 5-16 Παρουσίαση των 25 τελευταίων εγγραφών | 51 |
| Εικόνα 5-17 Παρουσίαση των τελευταίων 50 εγγραφών | 52 |
| Εικόνα 5-18 / 5-19 Οθόνη με μενού βοήθειας του site | 53 |
| Εικόνα 5-20 / 5-21 / 5-22 Εικόνες από συμπλήρωση φόρμας επικοινωνίας στο site | 55 |
| Εικόνα 5-23 Οθόνη με χρήσιμα Links μέσω της ιστοσελίδας..... | 55 |
| Εικόνα 5-24 Εικονίδιο Delphi εφαρμογής..... | 56 |
| Εικόνα 5-25 Αρχική σελίδα της Delphi εφαρμογής μας | 56 |
| Εικόνα 5-28 Οθόνη Κεντρικού Συστήματος | 58 |
| Εικόνα 5-29 Προειδοποιητικό μήνυμα ανόδου θερμοκρασίας..... | 59 |
| Εικόνα 5-30 Προειδοποιητικό μήνυμα για άνοδο θερμοκρασίας και ύπαρξης καπνού | 59 |
| Εικόνα 5-31 Φόρμα Επικοινωνίας της Delphi εφαρμογής | 60 |
| Εικόνα 5-32 Συμπλήρωση στοιχείων για SOS mail | 61 |
| Εικόνα 5-33 Διαδικασία αναμονής για αποστολή mail | 61 |
| Εικόνα 5-34 Επιτυχημένη αποστολή SOS mail | 62 |
| Εικόνα 5-35 Επιβεβαίωση λήψης του Emergency mail | 62 |
| Εικόνα 6-1 Πίνακας εντολών γλώσσας Arduino..... | 63 |
| Εικόνα 6-2 Διάγραμμα Use Case σε UML για την λειτουργία του website..... | 78 |
| Εικόνα 7-1 Το unit1 κατά την κατασκευή της Delphi εφαρμογής | 79 |
| Εικόνα 7-2 Το unit2 κατά την κατασκευή της Delphi εφαρμογής | 80 |
| Εικόνα 7-3 Το unit3 κατά την κατασκευή της Delphi εφαρμογής | 81 |
| Εικόνα 7-4 Τα units που αντιστοιχούν στα προειδοποιητικά μηνύματα κατά την διαδικασία κατασκευής της Delphi εφαρμογής | 82 |
| Εικόνα 7-5 Το unit9 κατά την κατασκευή της Delphi εφαρμογής | 83 |

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι, σε πρώτη φάση, η δημιουργία ενός οικιακού συστήματος καταγραφής, ανάλυσης και παρουσίασης θερμοκρασιών και υγρασίας καθώς και η δημιουργία ενός οικιακού συστήματος πυρασφάλειας το οποίο θα ενημερώνει τον χρήστη και τον server για την ανίχνευση καπνού και απότομης αύξησης θερμοκρασίας, στην ευρύτερη περιοχή στην οποία έχουμε τοποθετήσει τους αισθητήρες μας.

Οι μετρήσεις και οι καταγραφές των δεδομένων, εν προκειμένω των θερμοκρασιών και της υγρασίας, πραγματοποιήθηκαν σε διαφορετικούς χώρους με κριτήριο την ποικιλία εσωτερικών και εξωτερικών συνθηκών έτσι ώστε η δειγματοληψία να καλύπτει όσο το δυνατόν μεγαλύτερο φάσμα καταστάσεων.

Για την υλοποίηση της διπλωματικής εργασίας χρειάστηκαν εκτός των απαραίτητων γνώσεων πληροφορικής σε γλώσσα C++ για τον προγραμματισμό των αισθητήρων, βασικές γνώσεις PHP, HTML και SQL για την κατασκευή της ιστοσελίδας παρουσίασης των δεδομένων που καταγράψαμε και Delphi για την κατασκευή της desktop εφαρμογής.

Κλείνοντας θέλω να ευχαριστήσω όσους με βοήθησαν ανιδιοτελώς και μου μετέφεραν την επιπλέον γνώση κάποιων βασικών αρχών ηλεκτρολογίας και ηλεκτρονικών.

Ξεχωριστά θα ήθελα να πω όμως ένα ευχαριστώ στον καθηγητή κ.κ Αλέπη για την επίβλεψη, καθοδήγηση και τις πολύ χρήσιμες συμβουλές του όπου και όποτε χρειαζόμουν.

Abstract

The purpose of the current thesis is, primarily, the creation of a home security system which measures, records, analyses and presents temperature and humidity data, as well as the creation of a residential fire protection system which will notify the user and the server when it detects smoke and sudden temperature rise in the region in which we place the temperature sensors and smoke .

Measuring and recording of data, in this case the temperature and humidity data collection was made at different places in terms of the variety of internal and external conditions so that the sampling to cover the widest possible range of situations .

For the realization of this thesis, besides the basics in C + + language in order to program the sensors ,we also needed PHP, HTML and SQL to build the pages presenting the data recorded, and Delphi to build the desktop application .

Finally, I want to thank those who helped me selflessly and assisted me with the extra knowledge of some basic principles on the field of electronics.

Separately, I would like to thank dr. Alepis for his supervising, guidance and very helpful advices wherever needed.

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Οι επιθυμητές θερμοκρασιακές συνθήκες στον χώρο εξαρτώνται κατά κύριο λόγο από τις προσωπικές προτιμήσεις κάθε ατόμου.

Παρ' όλα αυτά, οι αποκλίσεις από αυτές είναι πάντα ένας αστάθμητος παράγοντας και τις περισσότερες φορές δυσάρεστος. Το σύστημα καταγραφής θερμοκρασιών κυρίως σε οικιακούς χώρους όπως και σε «κλειστούς» επαγγελματικούς χώρους αποκτούν μείζονα σημασία καθώς οι απότομες μεταβολές θερμοκρασίας μπορούν να προκαλέσουν σοβαρά προβλήματα. Η εξέλιξη της τεχνολογίας έχει αναδείξει την χρησιμότητα των ηλεκτρονικών συσκευών καταγραφής θερμοκρασιών και σε συνδυασμό με την εισαγωγή πρόσθετων παραμέτρων μας βοηθούν στην αποφυγή δυσάρεστων καταστάσεων.

Η διπλωματική αυτή εργασία ασχολείται με τη δημιουργία, βήμα προς βήμα, ενός οικιακού συστήματος μέτρησης, καταγραφής, ανάλυσης και παρουσίασης θερμοκρασιών και υγρασίας. Οι μετρήσεις μπορούν να γίνουν ιδανικά σε κλειστό οικιακό χώρο αλλά και στο εξωτερικό περιβάλλον.

Παράλληλα με τις απαραίτητες τροποποιήσεις και ρυθμίσεις και με τη χρήση της desktop εφαρμογής η εργασία παρέχει τη δυνατότητα πολλαπλών ειδοποιήσεων σε ειδικές περιπτώσεις όπως : απότομη αύξηση ή μείωση της θερμοκρασίας και όταν αυτό είναι σε πολύ χαμηλό ή υψηλό επίπεδο εμφανίζει στον χρήστη της εφαρμογής ενημερωτικό μήνυμα.

Επιπλέον μέσω του αισθητήρα καπνού με τον οποίο επικοινωνεί η εφαρμογή λαμβάνοντας τις μετρήσεις από την database, ειδοποιούμαστε σε περίπτωση ταυτόχρονης ανίχνευσης καπνού και αύξησης της θερμοκρασίας, για πιθανή φωτιά, και μας δίνεται η δυνατότητα αποστολής επείγοντος μηνύματος στον server.

Βασικό χαρακτηριστικό της εργασίας και της προσπάθειας δημιουργίας ενός οικιακού συστήματος καταγραφής θερμοκρασιών και πυρανίχνευσης είναι η δυναμική καταγραφή των δεδομένων.

Οι αισθητήρες την κατασκευή των οποίων θα αναλύσουμε παρακάτω, λειτουργούν και αποστέλλουν δεδομένα στην βάση σε πραγματικό χρόνο μέσω της σύνδεσης της πλακέτας με τους αισθητήρες στη θύρα Ethernet του H/Y.

Όσον αφορά την ανάλυση της διατριβής στα κεφάλαια που ακολουθούν γίνεται λεπτομερής περιγραφή:

- του hardware και software που χρειάστηκε για την ανάπτυξη της εργασίας
- του τρόπου κατασκευής της πλακέτας
- της συνδεσμολογίας και του user's manual
- της αρχιτεκτονικής του συστήματος
- καθώς και μιας σύντομης αναφοράς σε παρόμοια projects

Συγκεκριμένα στο κεφάλαιο 2 πραγματοποιείται μια ανασκόπηση πεδίου με αναφορές και παρουσίαση αντίστοιχων εφαρμογών με συναφές αντικείμενο στην Ελλάδα ή στο εξωτερικό.

Στο κεφάλαιο 3 γίνεται παρουσίαση του hardware και software που χρησιμοποιήθηκε καθώς επίσης και μια γενική αναφορά στο τι είναι οι μικροελεγκτές Arduino που χρησιμοποιήθηκαν στην κατασκευή των αισθητήρων και στις πλακέτες.

Ακολουθώντας στο κεφάλαιο 4 έχουμε τον οδηγό σύνδεσης και κατασκευής της πλακέτας Arduino (που ενσωματώνει τους αισθητήρες), ποια η διαδικασία σύνδεσης της με τον H/Y, το software που χρησιμοποιείται, καθώς και πως μεταφέρω τα data στην βάση δεδομένων που χρησιμοποιούμε.

Στο κεφάλαιο 5, γίνεται η ανάλυση βήμα προς βήμα των εξής διαδικασιών: έναρξη του project, καταγραφή, καταχώρηση και αποστολή δεδομένων. Επιπλέον παρουσιάζεται η ιστοσελίδα απεικόνισης και ανάλυσης των δεδομένων με όλες τις λειτουργίες και λεπτομέρειες καθώς και η desktop εφαρμογή σε γλώσσα DELPHI.

Στο κεφάλαιο 6 παρουσιάζεται η αρχιτεκτονική του συστήματος και ειδικότερα:

- ο τρόπος κατασκευής του προγράμματος με το οποίο λειτουργούν οι αισθητήρες
- κομμάτια του κώδικα (τι περιλαμβάνει και ποιές οι διαδικασίες του)
- αναφορά και περιγραφή της βάσης δεδομένων με ανάλυση στον τρόπο που μεταφέρονται μέσω του κώδικα τα data στη βάση και από τη βάση στο site.

Στο κεφάλαιο 7 γίνεται λεπτομερής ανάλυση και παρουσίαση:

- της desktop εφαρμογής κατασκευασμένης σε DELPHI
- στον τρόπο λειτουργίας
- στην δημιουργία και την αρχιτεκτονική
- στον τρόπο σύνδεσης – συγχρονισμού με την βάση δεδομένων

Στο 8^ο κεφάλαιο παρουσιάζονται κάποια τελικά συμπεράσματα της εργασίας, στο κεφάλαιο 9, οι μελλοντικές επεκτάσεις τις οποίες μπορούμε να κάνουμε στην εφαρμογή και τον τρόπο λειτουργίας της, και τέλος στο κεφάλαιο 10 γίνεται αναφορά στην βιβλιογραφία και δικτυγραφία που χρησιμοποιήσαμε.

ΚΕΦΑΛΑΙΟ 2

Ανασκόπηση πεδίου

Στο κεφάλαιο αυτό θα κάνουμε μία συνοπτική παρουσίαση παρόμοιων project σε Ελλάδα και εξωτερικό μετά από έρευνα στο διαδίκτυο:

2.1 Projects από Ελλάδα:

(i) Πανεπιστήμιο Πατρών, Πολυτεχνική Σχολή,
Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής

Τίτλος: Ανάπτυξη Δικτύου Αισθητήρων και Πληροφοριακού Συστήματος για τη διαχείριση του "Το Έξυπνο Σπίτι"

Επιμέλεια: Χουλιάρopoulos Αναστάσιος Α.Μ. : 3859

διεύθυνση url: <http://nemertes.lis.upatras.gr/jspui/bitstream/10889/4940/1/Thesis.pdf>

Σύντομη περιγραφή:

Η παρούσα διπλωματική έχει ως στόχο την παρουσίαση και ανάπτυξη ενός πλήρους πληροφοριακού συστήματος που είναι δυνατόν η εφαρμογή του στο πραγματικό κόσμο να καθιστά ένα σπίτι ως «έξυπνο». Ο σκοπός της δημιουργίας του συστήματος αυτού είναι η μέτρηση και η καταχώρηση των συνθηκών που επικρατούν σε ένα χώρο και ο εντοπισμός κινήσεων που γίνονται μέσα σε αυτόν ώστε να εκτελεστούν κάποιες λειτουργίες αυτόματα.

Το εν λόγω πληροφοριακό σύστημα αποτελείται από έναν κεντρικό υπολογιστή ο οποίος συνδέεται με ένα δίκτυο από διάφορους αισθητήρες, με μια βάση δεδομένων και έχει δυνατότητα επικοινωνίας με κινητό τηλέφωνο μέσω 3G δικτύου ώστε να υπάρχει απομακρυσμένη πρόσβαση σε όλες τις λειτουργίες.

Έτσι η ανάπτυξη και η παρουσίαση αυτού του συστήματος δίνει τη δυνατότητα δημιουργίας ενός έξυπνου σπιτιού με εξατομικευμένες προδιαγραφές και δυνατότητα επέκτασης του συστήματος.

Εν κατακλείδι, η παρούσα διπλωματική εργασία, θέλει να αναδείξει την ευκολία, την απλότητα, την ευελιξία, αλλά και την χρησιμότητα που έχει ένα έξυπνο σπίτι.

Παρουσιάζει ουσιαστικά την καρδιά του έξυπνου σπιτιού, ποιες είναι οι βασικές του μονάδες, πώς λειτουργούν και πώς αλληλεπιδρούν μεταξύ τους.

(ii) Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Αγρονόμων και Τοπογράφων Μηχανικών, Τομέας Τοπογραφίας

Διπλωματική Εργασία

Τίτλος: Κατασκευή Συστήματος Καταγραφής Ψηφιακών Δεδομένων (Data Logger) με χρήση της πλατφόρμας Arduino - Εφαρμογή με Μετεωρολογικά Δεδομένα

Επιμέλεια: Άννα Παπαναστασίου Α.Μ. : 06103046

διεύθυνση url: http://dspace.lib.ntua.gr/bitstream/123456789/3186/3/papanastasioua_arduino.pdf

Σύντομη περιγραφή:

Αντικείμενο της διπλωματικής εργασίας είναι η διερεύνηση και αξιοποίηση των δυνατοτήτων χαμηλού επιπέδου της ηλεκτρονικής πλατφόρμας Arduino για την δημιουργία ενός αυτόματου μετεωρολογικού σταθμού.

(iii) Πανεπιστήμιο Δυτικής Μακεδονίας, Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών
Διπλωματική Εργασία

Τίτλος: Σχεδιασμός και κατασκευή ολοκληρωμένου συστήματος καταγραφής και παρακολούθησης περιβαλλοντολογικών συνθηκών σε θερμοκήπιο

Επιμέλεια: Γιαννόπουλος Νικόλαος

διεύθυνση url: http://arch.icte.uowm.gr/mdasvg/pdfs/giannopoulos_thesis.pdf

Σύντομη περιγραφή:

Η διπλωματική αυτή εργασία ασχολείται με την δημιουργία ενός ενσωματωμένου συστήματος το οποίο συνδέεται μέσω διαδίκτυο με τον αντίστοιχο ιστοχώρο, με σκοπό την παρακολούθηση και την καταγραφή των περιβαλλοντικών συνθηκών ενός θερμοκηπίου, καθώς και τον έλεγχο συσκευών δράση (ανεμιστήρα, ποτίσματος κλπ). Η υλοποίηση του συστήματος έγινε με βάση την αρχιτεκτονική Arduino.

(iv) Τίτλος: Κατασκευή Καταγραφικού Θερμοκρασίας (Temperature Datalogger) διεύθυνση url:
<http://roscovnicoff.blogspot.com/2012/07/temperature-datalogger.html>

Σύντομη περιγραφή:

Στο project αυτό, δημιουργείται ένα ψηφιακό καταγραφικό θερμοκρασίας “αποξενωμένο” από το έτοιμο κύκλωμα του Arduino, χρησιμοποιώντας τον μικροελεγκτή μόνο του πάνω σε κύκλωμα κατασκευασμένο από τον δημιουργό. Προϋποθέτει τον προγραμματισμό του ATmega μικροεπεξεργαστή, τόσο με τις θέσεις των εξαρτημάτων, με το είδος της πλακέτας, όσο και με την κατασκευή του κουτιού που θα στεγαζόταν το ψηφιακό θερμόμετρο. Το θερμόμετρο, πέρα από την LCD οθόνη που δείχνει την ένδειξη της θερμοκρασίας και το thermistor που την μεταφράζει σε βαθμούς κελσίου, περιλαμβάνει μία κάρτα

Κατασκευή συστήματος καταγραφής, μελέτης και παρουσίασης θερμοκρασιών και πυρανίχνευσης με χρήση αισθητήρων

SD για την καταγραφή της θερμοκρασίας, ένα ρολόι και δύο κουμπάκια για τον χειρισμό του μενού της οθόνης.

2.2 Projects από Εξωτερικό:

(i) Τίτλος: Arduino + Temperature + Humidity

διεύθυνση url: <http://www.instructables.com/id/Arduino-Temperature-Humidity/>

Σύντομη περιγραφή:

Πρόκειται για την κατασκευή ενός απλού αισθητήρα θερμοκρασίας ο οποίος χρησιμοποιεί το αισθητήριο LM35 Precision Temperature Sensor, έναν humidity sensor και την πλακέτα Arduino. Το κύκλωμα θα αποστέλλει σειριακά τις πληροφορίες σχετικές με την θερμοκρασία και την υγρασία έτσι ώστε να μπορούν να αποθηκευτούν στον υπολογιστή και να χρησιμοποιηθούν αργότερα.

Τα υλικά που χρησιμοποιήθηκαν για την κατασκευή του project ήταν τα εξής:

- Arduino
- LM35 Precision Centigrade Temperature Sensor
- Breadboard
- Humidity Sensor
- Wires

(ii) Τίτλος: Temperature and Humidity Datalogger + Webserver

διεύθυνση url: <http://www.instructables.com/id/Temperature-and-Humidity-Datalogger-Webserver/>

Σύντομη περιγραφή:

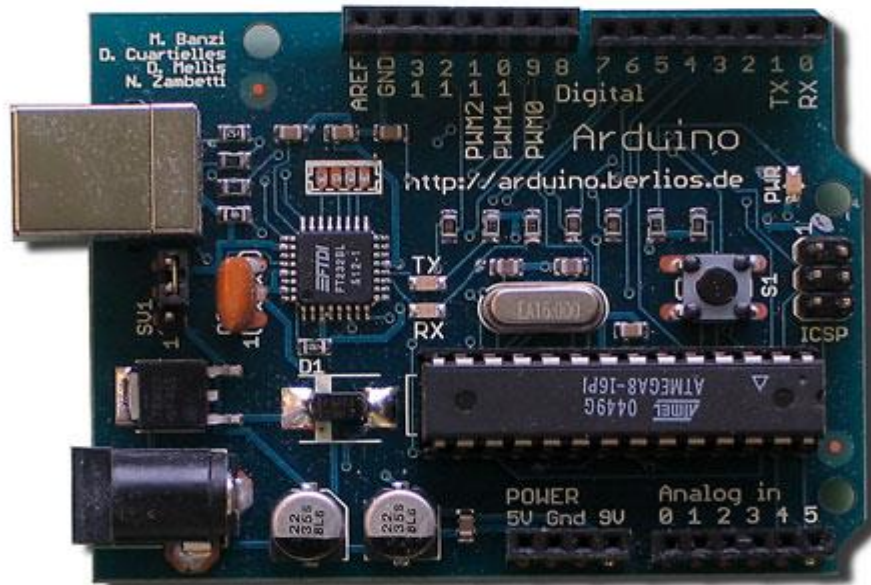
Στην εφαρμογή αυτή δημιουργείται ένας «συλλέκτης» δεδομένων θερμοκρασίας και υγρασίας με επιπρόσθετη την δυνατότητα να έχουμε πρόσβαση στα data μέσω του διαδικτύου. Με την χρήση του Arduino Ethernet η συγκεκριμένη ενέργεια γίνεται αρκετά εύκολη προσθέτοντας πάνω στην πλακέτα τα παρακάτω: ATMega 328, ethernet connectivity, sd card. Στο project ουσιαστικά χρησιμοποιούνται 3 αισθητήρες θερμοκρασίας και υγρασίας DHT11, που αποστέλλουν δεδομένα τα οποία αποθηκεύονται στην sd card και επιπλέον υπάρχει η δυνατότητα μέσω webserver να κάνουμε upload τα data στο διαδίκτυο

ΚΕΦΑΛΑΙΟ 3

Παρουσίαση απαιτούμενων Hardware και Software

3.1 Τι είναι οι μικροελεγκτές Arduino;

Τι είναι ο Arduino:



Εικόνα 3-1 Πλακέτα Arduino

Ο Arduino θα λέγαμε ότι είναι ένα εργαλείο για να κατασκευάσουμε ένα υπολογιστικό σύστημα με την έννοια ότι αυτό θα ελέγχει συσκευές του φυσικού κόσμου, σε αντίθεση με τον κοινό μας Ηλεκτρονικό Υπολογιστή. Είναι ανοιχτού υλικού και λογισμικού και βασίζεται σε μια αναπτυξιακή πλακέτα που ενσωματώνει επάνω έναν μικροελεγκτή και συνδέεται με τον Η/Υ για να τον προγραμματίσουμε μέσα από ένα απλό περιβάλλον ανάπτυξης. Ένας Arduino μπορεί να χρησιμοποιηθεί για να αναπτύξουμε διαδραστικά αντικείμενα, να δεχτούμε εισόδους από πληθώρα αισθητηρίων οργάνων και διακόπτες, αλλά και να ελέγχουμε διάφορα φώτα, κινητήρες και άλλες συσκευές εξόδου του φυσικού κόσμου. Τα Projects στον εν λόγω Μικροελεγκτή μπορούν να είναι αυτόνομα (σε επίπεδο hardware) ή να επικοινωνούν με κάποιο software στον Η/Υ του προγραμματιστή (προγράμματα όπως τα Flash, Processing, MaxMSP). Οι πλακέτες μπορούν εύκολα να συναρμολογηθούν ακόμη και από έναν αρχάριο ή να αγοραστούν μονταρισμένες. Το περιβάλλον ανάπτυξης του λογισμικού βασίζεται στην γλώσσα προγραμματισμού Processing και την γλώσσα προγραμματισμού Wiring, οι οποίες είναι ανοιχτού κώδικα (open source) και μπορεί κάποιος να τις "κατεβάσει δωρεάν". Η Γλώσσα προγραμματισμού του Arduino αποτελεί μια εφαρμογή σε software επίπεδο της καλωδίωσης. Εξομοιώνει θα λέγαμε απόλυτα το φυσικό περιβάλλον του μικροελεγκτή.

Γιατί να επιλέξω τον Arduino:

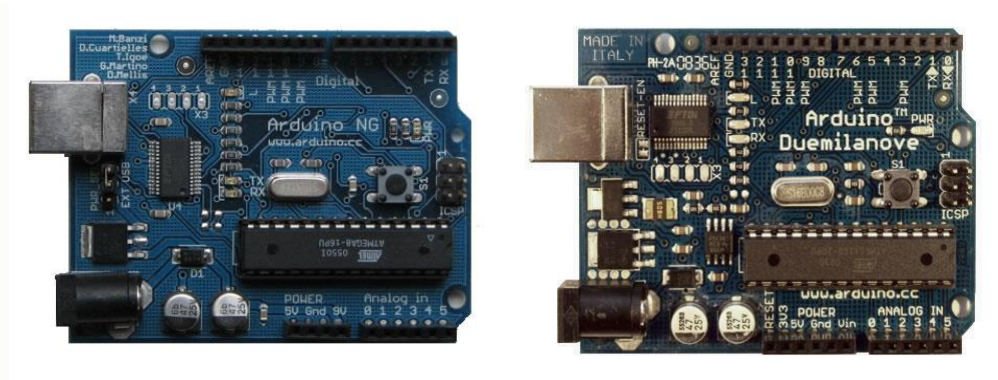
Υπάρχει πληθώρα άλλων μικροελεγκτών και αναπτυξιακών στο εμπόριο για να ασχοληθεί κάποιος εκεί έξω. Ο Basic Stamp της Parallax, ο BX-24 της Netmedia, το Handyboard του MIT και πολύ άλλη όμοιας λειτουργικότητας. Όλα αυτά τα εργαλεία που προαναφέραμε είναι απλά και για τον αρχάριο χρήστη καθώς "κρύβουν" τις δύσκολες λεπτομέρειες της αρχιτεκτονικής και επιτρέπουν τον άμεσο προγραμματισμό του μικροελεγκτή, προσφέροντας τα πάντα σε ένα και μόνο "πακέτο" έτοιμο για χρήση.

Ο Arduino διαφέρει από τους προηγούμενους γιατί απλοποιεί την διαδικασία να δουλεύει κάποιος με μικροελεγκτές, αλλά κάποια πλεονεκτήματα που προσφέρει σε σχέση με άλλους μικροελεγκτές για χρήση από δασκάλους, μαθητές και άλλους hobbίστες είναι τα παρακάτω:

- Φθηνός - Οι πλακέτες του Arduino είναι εξαιρετικά φθηνές σε σχέση με άλλες πλατφόρμες μικροελεγκτών. Ειδικά δε μπορεί με τα σχηματικά που κυκλοφορούν στο Internet να κατασκευάσει κάποιος την φθηνότερη έκδοση ενός Arduino. Ωστόσο ακόμα και αν προμηθευτεί την έτοιμη (μονταρισμένη πλακέτα) αυτή θα κοστίσει το μέγιστο 50 Euro.
- Τρέχει σε διάφορα Λειτουργικά Συστήματα. Οι μηχανικοί λογισμικού, ανέπτυξαν το περιβάλλον προγραμματισμού του Arduino για Windows, Machinstoh OSX και για λειτουργικά συστήματα Linux. Τα περισσότερα συστήματα ανάπτυξης Μικροελεγκτών περιορίζονται στα Windows.
- Απλό, ξεκάθαρο προγραμματιστικό περιβάλλον. Το περιβάλλον προγραμματισμού ενός Arduino ενδείκνυται για αρχάριους, αλλά είναι ταυτόχρονα και ευέλικτο και για πιο προχωρημένους χρήστες.
- Ανοιχτού λογισμικού και λογισμικού που επεκτείνεται και παραμετροποιείται. Το software του Arduino διανέμεται με την μορφή εργαλείων ανοιχτού λογισμικού και είναι διαθέσιμο προς επέκταση για έμπειρους προγραμματιστές. Η γλώσσα προγραμματισμού του μπορεί να επεκταθεί διαμέσου των βιβλιοθηκών την C++ και οι άνθρωποι που θέλουν να ασχοληθούν περισσότερο με τους μικροελεγκτές μπορούν να μεταβούν από τον Arduino στην AVR C που είναι για προγραμματισμό των Atmel Μικροελεγκτών και η γλώσσα στην οποία βασίστηκε το λογισμικό του Arduino. Ομοίως μπορεί κάποιος να προσθέσει κώδικα της AVR-C στο πρόγραμμα που έχει γράψει για τον Arduino του.
- Ανοιχτού Υλικού το οποίο μπορεί να επεκταθεί. Ο Arduino βασίζεται στους μικροελεγκτές της Atmel ATMEGA8 και ATMEGA168. Τα σχηματικά για τα αναπτυξιακά είναι κάτω από την άδεια της Creative Commons, επιτρέποντας σε έμπειρους σχεδιαστές να κατασκευάσουν το δικό τους αναπτυξιακό, εξελίσσοντας το ήδη υπάρχον χωρίς να έχουν νομικά προβλήματα. Η ακόμη καλύτερα όχι τόσο έμπειροι χρήστες μπορούν να επιδιώξουν την αντιγραφή και κατασκευή της πλακέτας σε ράστερ για να καταλάβουν την λειτουργία ενός Arduino.

Το Υλικό (Hardware) του Arduino

Ο μικροελεγκτής Arduino είναι στην ουσία μια αναπτυξιακή πλακέτα που ενσωματώνει έναν ATMEL ATMEGA168. Υπάρχουν πολλές εκδόσεις του μικροελεγκτή. Στα παρακάτω σχήματα βλέπετε την έκδοση NG και την έκδοση Duemilano. Η τελευταία έκδοση (2009) είναι αυτή του Arduino Duemilano. Οι κατασκευαστές του Arduino έχουν τοποθετήσει στις πλακέτες όλα τα απαραίτητα εξαρτήματα για την τροφοδοσία και την διασύνδεση των μικροελεγκτών με τον Ηλεκτρονικό Υπολογιστή. Η πλακέτα μπορεί να τροφοδοτηθεί είτε με τροφοδοτικό των 9Volt, είτε απευθείας από την USB θύρα του Υπολογιστή.



Εικόνα 3-2 Arduino NG

Ardino Duemilanove

Ακροδέκτες του μικροελεγκτή Arduino

Ο Arduino έχει 14 ψηφιακούς ακροδέκτες Εισόδου/Εξόδου. Αυτοί μπορούν να τεθούν ως είσοδοι ή ως έξοδοι με τις εντολές-συναρτήσεις `pinMode()`, `digitalWrite()`, and `digitalRead()`. Λειτουργούν στα 5 Volts και έχουν την δυνατότητα να παρέχουν ή να καταβυθίζουν ένταση της τάξεως των 40mA. Σε κάθε Pin υπάρχει εσωτερικά ένας Pull-up αντιστάτης στα 20-50KΩ. Επιπλέον έχει 5 Αναλογικούς ακροδέκτες Εισόδου. Αυτοί μπορούν να διαβάσουν αναλογικές τιμές όπως η τάση μιας μπαταρίας κτλ και να τις μετατρέψουν σε έναν αριθμό από 0-1023. Η μέτρηση της τάσης γίνεται από προκαθορισμένα από 0 έως 5 volts. Εκτός αυτού 6 εκ των 14 ψηφιακών ακροδεκτών οι P3, P5, P6, P9, P10 και P11 έχουν την δυνατότητα να προγραμματιστούν ώστε να λειτουργούν ως Αναλογικές Έξοδοι.

Κάποιοι ακροδέκτες έχουν συγκεκριμένες λειτουργίες:

- Σειριακή Λειτουργία: 0 (RX) and 1 (TX). Χρησιμοποιούνται για λήψη (RX) και εκπομπή (TX) TTL σειριακών δεδομένων. Αυτοί οι ακροδέκτες είναι συνδεδεμένοι με τους αντίστοιχους του ολοκληρωμένου FTDI USB-to-TTL Serial.
- Εξωτερικές Διακοπές: 2 και 3. Αυτοί οι ακροδέκτες μπορούν να ενεργοποιούν διακοπές αν ανιχνευθεί παλμός χαμηλής τάσης. Με την συνάρτηση `attachInterrupt()`.
- PWM: 3, 5, 6, 9, 10, and 11. Παρέχουν Έξοδο 8-bit PWM με την συνάρτηση `analogWrite()`.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Αυτοί οι ακροδέκτες επιτρέπουν επικοινωνία SPI, η οποία αν και παρέχεται από το hardware δεν είναι ακόμα διαθέσιμη στην γλώσσα προγραμματισμού του Arduino.
- LED: 13. Στον ακροδέκτη 13 υπάρχει ένα ενσωματωμένο LED. Όταν ο ακροδέκτης έχει τιμή HIGH, το LED φωτοβολεί..

Επιπλέον υπάρχουν ακροδέκτες για ειδικές λειτουργίες όπως:

- I2C: 4 (SDA) and 5 (SCL). Υποστηρίζει το πρωτόκολλο I2C (TWI) χρησιμοποιώντας βιβλιοθήκες τις Γλώσσας προγραμματισμού Wiring

Χαρακτηριστικά

| | |
|-----------------------------|---|
| Μικροελεγκτής | ATmega168 |
| Τάση Λειτουργίας | 5V |
| Τάση Εισόδου | 7-12V |
| Όρια Τάσης | 6-20V |
| Ψηφιακοί Ακροδέκτες I/O | 14 (of which 6 provide PWM output) |
| Ψηφιακοί Ακροδέκτες Εισόδου | 6 |
| DC ρεύμα ανά I/O Ακροδέκτη | 40 mA |
| DC ρεύμα για 3.3V Ακροδέκτη | 50 mA |
| Μνήμη Flash | 16 KB (2 KB χρησιμοποιούνται από τον bootloader) |
| SRAM | 1 KB |
| EEPROM | 512 bytes |
| Ταχύτητα Ρολογιού | 16 MHz |

Μνήμη

Το ολοκληρωμένο ATmega168 έχει 16KB μνήμης flash για την αποθήκευση κώδικα (2 KB εκ των οποίων χρησιμοποιούνται από τον bootloader). Έχει επίσης 1 KB SRAM και 512 bytes μνήμης EEPROM (τα οποία μπορούν να διαφαστούν και να γραφτούν με την βιβλιοθήκη EEPROM).

Τροφοδοσία

Το αναπτυξιακό Arduino Duemilanove τροφοδοτείται είτε από εξωτερική τροφοδοσία είτε απευθείας από την θύρα USB. Η επιλογή της πηγής γίνεται αυτόματα από το αναπτυξιακό. Ως εξωτερική τροφοδοσία ορίζεται είτε μια μπαταρία, είτε μετασχηματιστής των 9Volt από 220V. Η μπαταρία μπορεί να συνδεθεί στις υποδοχές του Arduino Vin και GND όπου τοποθετούνται ο θετικός πόλος και ο αρνητικός αντίστοιχα. Από την άλλη αν τροφοδοτήσουμε με μετασχηματιστή απλά τοποθετούμε το βύσμα στην υποδοχή που υπάρχει με τον θετικό πόλο στο κέντρο.

Η πλακέτα μπορεί να λειτουργήσει με εξωτερική πηγή από 6 έως 20 Volts. Αν ωστόσο τροφοδοτηθεί με λιγότερα από 7 Volt τα pin εξόδου 5Volt δεν θα καταφέρουν να εξάγουν τάση 5 Volts. Αν από την άλλη δώσουμε πάνω από 12 Volts θα υπερθερμανθεί ο σταθεροποιητής τάσης στην πλακέτα και ενδεχόμενος να καταστραφεί. Συνεπώς μια ιδανική τάση είναι τα 9 Volts.

Οι ακροδέκτες τροφοδοσίας είναι οι εξής:

- **VIN.** Ακροδέκτης για μη σταθεροποιημένη τάση. Συνήθως εδώ συνδέεται μια εξωτερική πηγή τροφοδοσίας.
- **5V.** Ακροδέκτης σταθεροποιημένης τάσης 5Volt. Χρησιμοποιείται για την τροφοδοσία του μικροελεγκτή ή άλλων ηλεκτρονικών στοιχείων.
- **3V3.** Το ολοκληρωμένο FTDI που βρίσκεται στην πλακέτα του Arduino παράγει τάση των 3.3V με μέγιστο ρεύμα 50mA.
- **GND.** Ακροδέκτες Γείωσης

Επικοινωνία

Ο Arduino Duemilavone έχει την δυνατότητα να επικοινωνεί με τον Ηλεκτρονικό Υπολογιστή, έναν άλλον Arduino ή άλλους μικροελεγκτές. Το ολοκληρωμένο ATmega 168 παρέχει σειριακή επικοινωνία TTL 5Volt UART, η οποία είναι διαθέσιμη από τους ακροδέκτες (λήψη RX) 0 και (εκπομπή TX) 1 του ολοκληρωμένου. Επιπλέον στην αναπτυξιακή πλακέτα του Arduino είναι ενσωματωμένο ένα ολοκληρωμένο το FTDI FT232RL το οποίο παρέχει σειριακή επικοινωνία με τον Ηλεκτρονικό Υπολογιστή για προγραμματισμό, πάνω από την θύρα USB με την βοήθεια των ανάλογων FTDI drivers. Οι drivers αυτοί περιλαμβάνονται στο software για τον Arduino και παρέχουν μια ιδεατή θύρα επικοινωνίας στον Ηλεκτρονικό Υπολογιστή για τους σκοπούς της επικοινωνίας.

3.2 Hardware που χρησιμοποιήθηκε για την υλοποίηση της εργασίας

Η μελέτη των δεδομένων θερμοκρασίας και υγρασίας καθώς και καπνού τα οποία συλλέγουμε για την ανάπτυξη της διατριβής, απαιτεί την χρήση δύο πλακετών οι οποίες κατασκευάστηκαν βήμα-βήμα και ενσωματώνουν τους αισθητήρες που είναι απαραίτητοι. Πιο συγκεκριμένα, χρησιμοποιήθηκαν πλακέτες οι οποίες πωλούνται στο εμπόριο και σκοπό έχουν να συνδεθούν στον υπολογιστή και να μεταφέρουν δεδομένα ενσύρματα μέσω καλωδίων, πλακέτα η οποία βοηθά στην μεταφορά δεδομένων στο internet και στην βάση μας μέσω Ethernet καλωδίου και πλακέτες οι οποίες κατασκευάστηκαν χειροποίητα προκειμένου να εξυπηρετήσουν τον δεύτερο τρόπο μεταφοράς δεδομένων, τον ασύρματο.

Η ανάλυση της συνδεσμολογίας, κατασκευής και λειτουργίας θα γίνει σε επόμενο κεφάλαιο. Στο παρόν κεφάλαιο θα γίνει παρουσίαση των hardware που χρησιμοποιήθηκαν για την κατασκευή των αισθητήρων και των πλακετών.

- Συσκευή μέτρησης ΕΝΣΥΡΜΑΤΗΣ μεταφοράς δεδομένων:

1. Arduino Mega 2560 Rev3



Εικόνα 3-3 Πλακέτα Arduino Mega 2560

Περιγραφή: Διαθέτει 54 ψηφιακές υποδοχές εισόδου/εξόδου (από τις οποίες 14 μπορούν να χρησιμοποιηθούν ως PWM έξοδοι), 16 αναλογικές εισόδους, 4 UARTs (hardware serial ports), 1 κρυσταλλικό ταλαντωτή 16 MHz, υποδοχή USB, υποδοχή τροφοδοσίας ρεύματος, 1 ICSP header και ένα κουμπί reset. Περιέχει ό,τι χρειάζεται για την λειτουργία του microcontroller, συνδέεται απλά με τον υπολογιστή μέσω USB ή εναλλακτικά μπορεί να τροφοδοτηθεί με μετασχηματιστή AC-to-DC ή μπαταρία. Το Mega είναι συμβατό με τα περισσότερα Arduino Shields.

2. Arduino Ethernet Shield Rev3



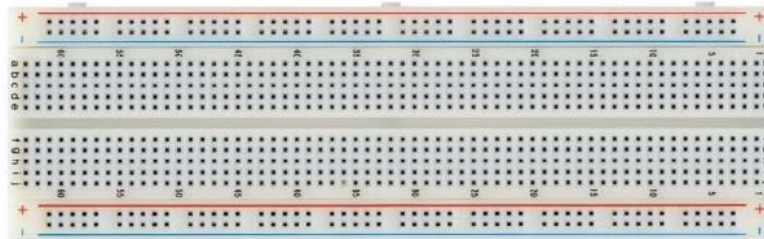
Εικόνα 3-4 Πλακέτα Arduino Ethernet για σύνδεση στο Internet

Περιγραφή: Το Arduino Ethernet Shield συνδέει Arduino σας στο διαδίκτυο σε λίγα λεπτά. Απλά συνδέστε την μονάδα πάνω πλακέτα Arduino σας, συνδέστε το δίκτυό σας με ένα καλώδιο RJ45 (δεν συμπεριλαμβάνεται) και ακολουθήστε μερικές απλές οδηγίες για να ξεκινήσετε να ελέγχετε τον κόσμο
Κατασκευή συστήματος καταγραφής, μελέτης και παρουσίασης θερμοκρασιών και πυραίνιχνευσης με χρήση αισθητήρων

σας μέσω του διαδικτύου. Όπως πάντα με το Arduino, κάθε στοιχείο της πλατφόρμας - hardware, λογισμικό και τεκμηρίωση - είναι ελεύθερα διαθέσιμο και open-source. Αυτό σημαίνει ότι μπορείτε να μάθετε πώς ακριβώς έχει κατασκευαστεί και να χρησιμοποιήσετε το σχεδιασμό του σαν σημείο εκκίνησης για τα δικά σας κυκλώματα.

- Requires an Arduino board (not included)
- Operating voltage 5V (supplied from the Arduino Board)
- Ethernet Controller: W5100 with internal 16K buffer
- Connection speed: 10/100Mb
- Connection with Arduino on SPI port

3. Breadboard



Εικόνα 3-5 Breadboard για την σύνδεση των αισθητήρων μας

Περιγραφή: Το Breadboard σας διευκολύνει να δοκιμάσετε το κύκλωμα, χωρίς την χρήση κολλητηριού. Σας προσφέρει γρήγορη και αξιόπιστη λύση για την δημιουργία πρωτοτύπων.

Οι ηλεκτρονικές συνδέσεις γίνονται πάνω στο Breadboard με την βοήθεια καλωδίων, εισάγωντάς τα στις ειδικές οπές που απέχουν η κάθε μία μεταξύ τους 0,1" (2,54mm) και ενώνονται οριζόντια ώστε να σας διευκολύνουν στην σύνδεση πολλών εξαρτημάτων χωρίς καλώδια. Οι οπές στο εσωτερικό τους περιέχουν μικρά ελατήρια για την άψογη επαφή τους με το καλώδια.

Οι γραμμές και οι στήλες του Breadboard είναι χαρακτηρισμένες από Γράμματα και Αριθμούς αντίστοιχα για να μπορείτε εύκολα να βρείτε το κύκλωμα σας. Χρησιμοποιείται εξαρτήματα με pins όπου το βήμα τους είναι 2,54mm (0.1") για να εφαρμόζουν στα Breadboard.

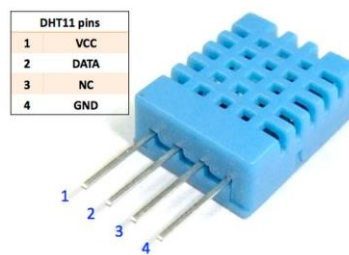
4. Breadboard Jumper Cable



Εικόνα 3-6 Ενδεικτικά καλώδια συνδέσεων

Περιγραφή: Για να πραγματοποιήσετε τις συνδέσεις πάνω σε ένα Breadboard μπορείτε να χρησιμοποιήσετε αυτά τα ειδικά καλώδια. Τα Jumper Cables είναι καλώδια όπου και στις δύο άκρες είναι αρσενικά και εφαρμόζουν άψογα στις οπές του Breadboard.

5. Humidity Sensor DHT11



Εικόνα 3-7 Αισθητήρας υγρασίας DHT11

Περιγραφή: Ο ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας DHT11 είναι ένας σύνθετος αισθητήρας περιέχει μια βαθμολογημένη παραγωγή ψηφιακών σημάτων της θερμοκρασίας και της υγρασίας. Εφαρμογή μιας αφιερωμένης ψηφιακής τεχνολογίας συλλογής ενοτήτων και της τεχνολογίας αντίληψης θερμοκρασίας και υγρασίας, ώστε να εξασφαλιστεί ότι το προϊόν έχει την υψηλή αξιοπιστία και την άριστη μακροπρόθεσμη σταθερότητα.

Ο αισθητήρας περιλαμβάνει μια ανθεκτική αίσθηση των υγρών συστατικών και των συσκευών μιας NTC θερμοκρασίας μέτρησης, και συνδεδεμένος με έναν υψηλής απόδοσης οκτάμπιτο μικροελεγκτή. Το προϊόν έχει την άριστη ποιότητα, τη γρήγορη απάντηση, την ισχυρή αποσυμφορητική ικανότητα, και το υψηλό κόστος.

Ενιαία τμηματική διεπαφή καλωδίων, ολοκλήρωση συστημάτων γρήγορη και εύκολη. Μικρό μέγεθος, χαμηλής ισχύος κατανάλωση, απόσταση μετάδοσης σημάτων μέχρι 20 μέτρα, που κάνουν το την καλύτερη επιλογή όλων των ειδών εφαρμογών και ακόμη και των πιο απαιτητικών εφαρμογών. Η κατάλληλη σύνδεση μπορεί να συνδεθεί άμεσα με τον πίνακα επέκτασης αισθητήρων Arduino.

Παράμετροι προϊόντων:

- Τάση παροχής ηλεκτρικού ρεύματος: +5 B
- Σειρά θερμοκρασίας: 0-50 °C λάθος ± 2 °C

- Σειρά υγρασίας: 20-90% λάθος $\pm 5\%$ RH RH
- Ακολουθία γραμμών διεπαφών: VCC, GND, S
- Μέγεθος ενότητας: 30 X 20mm

6. Αισθητήρας θερμοκρασίας DS18B20



Εικόνα 3-8 Αισθητήριο Θερμοκρασίας DS18B20

Περιγραφή: Ψηφιακός αισθητήρας θερμοκρασίας DS18B20 1-Wire από την Maxim-IC. Υπολογίζει βαθμούς Κελσίου ($^{\circ}\text{C}$) από -55°C έως 125°C ($\pm 0.5^{\circ}\text{C}$) με ακρίβεια 9 - 12bit. Κάθε αισθητήρας χαρακτηρίζεται από μία διεύθυνση 16-Bit. Λόγω των διεθύνσεων που έχουν οι αισθητήρες αυτοί μπορούν να συνδεθούν πολλοί μαζί σε ένα μόνο καλώδιο (1-wire protocol).

7. General Gas Sensor MQ-2 +Gas Sensor Breakout Board



Εικόνα 3-9 Αισθητήριο GAS και συνδεσμολογία

Περιγραφή: Αυτός ο αισθητήρας αερίου και καπνού ανιχνεύει τις συγκεντρώσεις εύφλεκτων αερίων στον αέρα και παράγει ανάγνωση της ως αναλογική τάση. Ο αισθητήρας μπορεί να μετρήσει συγκεντρώσεις εύφλεκτων αερίων από 300 έως 10.000 ppm. Μπορεί να λειτουργεί σε θερμοκρασίες από -20 έως 50°C και καταναλώνει λιγότερο από 150 mA στα 5 V.

Χρησιμοποιείται σε Εξοπλισμό ανίχνευσης διαρροής αερίου σε σπίτια και βιομηχανίες, και είναι κατάλληλος για την ανίχνευση LPG, ισο-βουτάνιου, προπάνιου, μεθάνιου, αλκοόλ, υδρογόνου και καπνού.

8. Αντιστάσεις



Εικόνα 3-10 Ενδεικτικές Αντιστάσεις

Περιγραφή:

Χαρακτηριστικά:

- Power (Watts) : 0.25 (1/4 Watts)
- Tolerance $\pm 5\%$
- Resistor Type Carbon Film

9. Led (green)



Εικόνα 3-11 LED για χρήση στον αισθητήρα στην χρήση ON/OFF

10. Καλώδιο Ethernet RJ45



Εικόνα 3-12 Καλώδιο σύνδεσης Arduino Ethernet Shield με PC

Περιγραφή: Το RJ45 καλώδιο που χρειάζεται για τη σύνδεση αυτή, είναι το ίδιο με το ethernet καλώδιο δικτύου.

11. Καλώδιο σύνδεσης Arduino με PC

**Εικόνα 3-13 Καλώδιο Σύνδεσης Arduino με το PC**

Περιγραφή: Πρόκειται για το καλώδιο που συνδέει την πλακέτα Arduino με το PC. Είναι το ίδιο με το καλώδιο που χρησιμοποιείται για την σύνδεση του εκτυπωτή στον υπολογιστή.

- Συσκευή μέτρησης ΑΣΥΡΜΑΤΗΣ μεταφοράς δεδομένων:
 1. Xbee 1mw

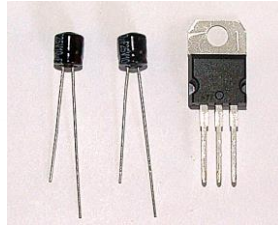
**Εικόνα 3-14 Chip Xbee**

Περιγραφή: Αυτό είναι το πολύ δημοφιλές module Xbee 2.4GHz από την Digi. Οι μονάδες αυτές λαμβάνουν το stack 802.15.4 (η βάση για Zigbee) και το μετατρέπουν σε ένα απλό στη χρήση σύνολο εντολών. Οι μονάδες αυτές επιτρέπουν μια πολύ αξιόπιστη και απλή επικοινωνία μεταξύ μικροελεγκτών, υπολογιστών, συστημάτων, πραγματικά οτιδήποτε με σειριακή θύρα! Υποστηρίζονται δίκτυα point to point και multi-point.

Χαρακτηριστικά:

- 3.3V @ 50mA
- 250kbps Max data rate
- 1mW output (+0dBm)
- 300ft (100m) range
- Wire antenna
- Fully FCC certified
- 6 10-bit ADC input pins
- 8 digital IO pins
- 128-bit encryption
- Local or over-air configuration
- AT or API command set

2. Regulators

**Εικόνα 3-15 Regulators**

Περιγραφή: Απαραίτητος ρυθμιστής για τάση 5V

3. Voltage Regulators (3) 780S

4. Αντίστασεις 220 ohm

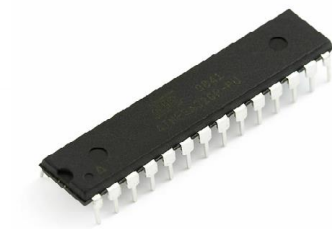
**Εικόνα 3-16 Αντιστάσεις**

5. Κεραμικός πυκνωτής 50V (2,2pf-100nf)

**Εικόνα 3-17 Κεραμικοί πυκνωτές**

Περιγραφή: Ένας κοινός πυκνωτής 0.1 μF . Χρησιμοποιείται σε όλα τα είδη των εφαρμογών για να αποθηκεύει ηλεκτρικό φορτίο, επομένως ηλεκτρική ενέργεια. Όταν ένας πυκνωτής είναι φορτισμένος, οι οπλισμοί του έχουν ηλεκτρικά φορτία κατά μέτρο ίσα και αντίθετα. Λόγω της δυνατότητάς τους να αποθηκεύουν ηλεκτρικό φορτίο και να το αποδίδουν κατόπιν αποφορτιζόμενοι σε ένα κύκλωμα (δρώντας έτσι ουσιαστικά ως πηγές ρεύματος), οι πυκνωτές αποτελούν βασικά στοιχεία κάθε σύγχρονου ηλεκτρονικού κυκλώματος

6. Μικροεπεξεργαστής AtMega 328 με Optiboot Arduino

**Εικόνα 3-18 Μικροεπεξεργαστής AtMega 328**

Περιγραφή: Ένας μικροεπεξεργαστής ATmega328 σε πακέτο DIP, ο οποίος περιέχει Arduino Optiboot (Uno 16MHz) Bootloader. Αυτό θα σας επιτρέψει να χρησιμοποιήσετε κώδικα Arduino στο project σας χωρίς να χρησιμοποιήσετε Arduino board.

Για να λειτουργήσει αυτό το chip με το Arduino IDE, θα χρειαστείτε εξωτερικό 16MHz crystal ή resonator, 5V παροχή και σειριακή σύνδεση. Εάν δεν θέλετε να κάνετε αυτήν την διαδικασία μπορείτε να αγοράσετε ένα έτοιμο arduino UNO.

Ο επεξεργαστής της Atmel ATmega328 8-Bit σε πακέτο 28 pin DIP. Είναι σαν το ATmega168, με διπλάσιο χώρο για flash. 32K χωρητικότητα για πρόγραμμα, 23 I/O lines, 6 απο τις οποίες είναι κανάλια για το 10-bit ADC. Λειτουργεί μέχρι και σε 20MHz με εξωτερικό crystal. Το πακέτο προγραμματίζεται σε κύκλωμα. 1.8V με 5V τάση λειτουργίας.

7. Κρύσταλλος 16MHz για χρονισμό (Crystal HC-49/S Low Profile 3,57 Mhz- 20 Mhz)

**Εικόνα 3-19 Κρύσταλλος χρονισμού**

Περιγραφή: Κρύσταλλος σταθερής συχνότητας. Χρησιμοποιείται για να παρέχει ωρολογιακή συχνότητα στον μικροεπεξεργαστή.

8. Ολοκληρωμένος ελεγκτής

**Εικόνα 3-20 Ολοκληρωμένος ελεγκτής**

Περιγραφή: Λογικός αναστροφέας των 3,3V από 5V για να δουλέψει το Xbee.

Έχει 4 εξόδους:

- τροφοδοσία
- γη
- TX (transfer data)
- RX (receive data)

3.3 Software που χρησιμοποιήθηκε στην υλοποίηση

- Λογισμικό της Arduino (1.0.1) στο οποίο γράψαμε τον κώδικα. Το λογισμικό είναι ανοιχτού κώδικα και βοηθά τον χρήστη στο να γράψει και να κάνει upload τον κώδικά του. Μπορεί να λειτουργήσει σε Windows, MAC και Linux. Το περιβάλλον είναι γραμμένο σε Java και βασισμένο στην γλώσσα Processing. Το λογισμικό το βρίσκουμε στην ιστοσελίδα www.arduino.cc. Πάμε στην ενότητα download και στη συνέχεια κατεβάζουμε το λογισμικό (open source πλατφόρμα)
- Processing: Είναι μία open source πλατφόρμα η οποία σκοπό έχει την δημιουργία των on/off buttons τα οποία έχουν ενσωματωθεί στο site. Στην συγκεκριμένη πλατφόρμα γράφουμε τον κώδικα που αφορά αποκλειστικά την χρήση των buttons on/off. Η δομή της πλατφόρμας είναι παρόμοια με εκείνη της πλατφόρμας Arduino. Το πρόγραμμα ης πλατφόρμας Processing λειτουργεί ταυτόχρονα με το πρόγραμμα της πλατφόρμας Arduino με την διαφορά ότι πρώτο σε λειτουργία μπαίνει το πρόγραμμα Arduino.
- XAMPP ή WAMPP : Το XAMPP είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl. Το XAMPP είναι ακρωνύμιο και αναφέρεται στα παρακάτω αρχικά:
 - X (αναφέρεται στο «cross-platform» που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας)
 - Apache HTTP εξυπηρετητής
 - MySQL
 - PHP
 - Perl

Το XAMPP είναι ένα ελεύθερο λογισμικό το οποίο περιέχει ένα εξυπηρετητή ιστοσελίδων το οποίο μπορεί να εξυπηρετεί και δυναμικές ιστοσελίδες τεχνολογίας PHP/MySQL. Είναι ανεξάρτητο πλατφόρμας και τρέχει σε Microsoft Windows, Linux, Solaris, and Mac OS X και χρησιμοποιείται ως πλατφόρμα για την σχεδίαση και ανάπτυξη ιστοσελίδων με την τεχνολογίες όπως PHP, JSP και Servlets.
- Notepad++: Με το Notepad++ μπορούμε να αντικαταστήσουμε το κλασικό Notepad των Windows. Πρόκειται ουσιαστικά για έναν text editor. Μπορούμε να επεξεργαστούμε τα αρχεία του κώδικά μας σε πολλές γλώσσες (πχ. HTML, php, C, C++, C#, pascal κ.τ.λ.) με δυνατότητα Syntax highlighting.
- DELPHI 7

ΚΕΦΑΛΑΙΟ 4

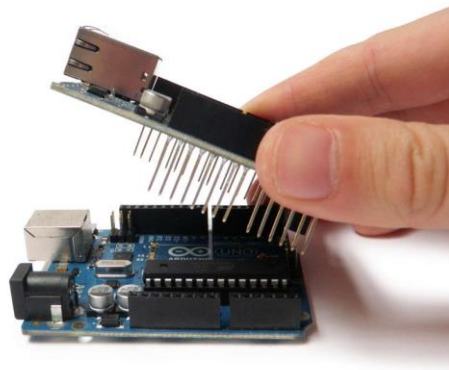
Οδηγός Συνδεσμολογίας των Arduino parts και σύνδεση με Software

4.1 Σύνδεση step by step των Arduino parts (hardware)

- Συνδεσμολογία για την κατασκευή συσκευής μέτρησης και ΕΝΣΥΡΜΑΤΗΣ μεταφοράς δεδομένων.

Η συσκευή ενσύρματης μεταφοράς δεδομένων αποτελεί τον βασικό αισθητήρα μέτρησης, καταγραφής και αποστολής δεδομένων. Η κατασκευή του προϋποθέτει την σύνθεση των επιμερούς συσκευών και αισθητήρων που περιγράφηκε στο προηγούμενο κεφάλαιο με τρόπο που θα αναλύσουμε παρακάτω.

Πρώτο βήμα είναι να συνδέσουμε την πλακέτα Ethernet Shield πάνω στην πλακέτα Arduino Mega όπως στην παρακάτω φωτογραφία:



Εικόνα 4-1 Σύνδεση του Ethernet Shield στην μητρική πλακέτα

Παρατηρούμε ότι τα pins της πάνω πλακέτας (Ethernet Shield) αντιστοιχούν ιδανικά στους υποδοχείς της πλακέτας Arduino Mega.

ΠΑΡΑΤΗΡΗΣΗ: Στην τελική μορφή της συσκευής, ανάμεσα στις δυο προαναφερθείσες πλακέτες, μεσολαβεί η πλακέτα που κατασκευάστηκε για την ασύρματη μεταφορά δεδομένων. (Κομμάτι που θα αναλυθεί παρακάτω)

Εν συνεχεία, συνδέουμε το σύνολο των πλακετών που δημιουργήσαμε προηγουμένως με το breadboard το οποίο περιλαμβάνει:

- i. τον αισθητήρα θερμοκρασίας DS18B20
- ii. τον αισθητήρα θερμοκρασίας- υγρασίας DHT11
- iii. τον αισθητήρα καπνού MQ-2 και
- iv. ένα LED το οποίο ανάβει και σβήνει ανάλογα με την λειτουργία ON/OFF της συσκευής

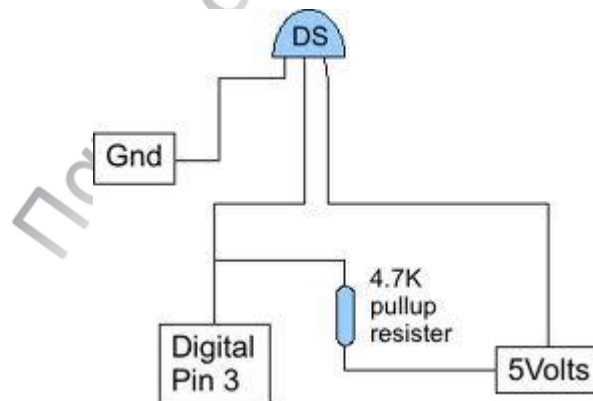
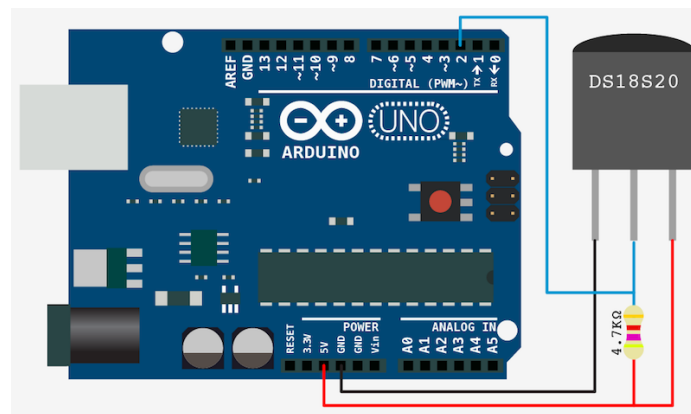
Παρακάτω γίνεται παρουσίαση της σύνδεσης των παραπάνω αισθητήρων πάνω στο breadboard και στις πλακέτες.

i. Αισθητήρας Θερμοκρασίας DS18B20

Στο προηγούμενο κεφάλαιο αναλύσαμε τα χαρακτηριστικά και τον τρόπο λειτουργίας του αισθητήρα θερμοκρασίας DS18B20. Στην ενότητα αυτή θα παρουσιάσουμε διαγραμματικά και απεικονιστικά τον τρόπο σύνδεσης του αισθητήρα πάνω στο breadboard και εν συνεχεία το ίδιο θα γίνει και για τις πλακέτες.

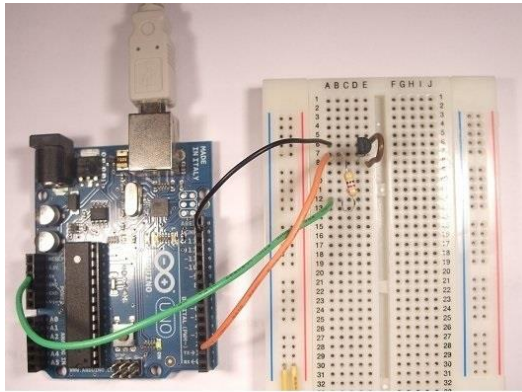
Αναλυτικότερα, ο αισθητήρας DS18B20 έχει 3 pins απο τα οποία τα αριστερό είναι που πρέπει να συνδεθεί με την υποδοχή GND για την γείωση του ρεύματος, το μεσαίο pin σε συνδιασμό με μια μικρή αντίσταση των 4,7 kΩ συνδέεται με το digital pin πάνω στην πλακέτα (στην περίπτωση μας το digital pin2) και τέλος το δεξί ακραίο pin του αισθητήρα θερμοκρασίας συνδέεται με την υποδοχή της τροφοδοσίας των 5V πάνω στην πλακέτα μας.

Ενδεικτικό Schematic της συνδεσμολογίας:

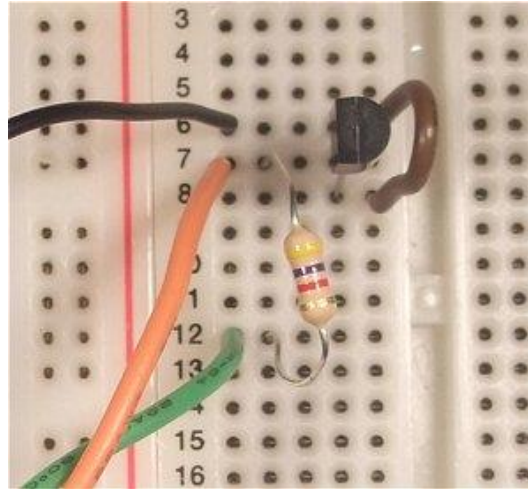


Εικόνα 4-2 / 4-3 Schematic συνδεσμολογίας αισθητήρα DS18B20

Εικόνες της συνδεσμολογίας μας:



- **Πράσινο Καλώδιο:** Σύνδεση του αισθητήριου DS18B20 με τροφοδοσία 5 V
- **Πορτοκαλί Καλώδιο:** Σύνδεση του pin του αισθητήριου με digital pin πάνω στο σύστημα με τις πλακέτες μας
- **Μαύρο Καλώδιο:** Σύνδεση με υποδοχή GROUND (GND) σε πλακέτα για την γείωση του ρεύματος
- **Καφέ Καλώδιο:** Συνδέεται στο breadboard και ενώνει το 1ο και το 3ο pin του αισθητήριου θερμοκρασίας



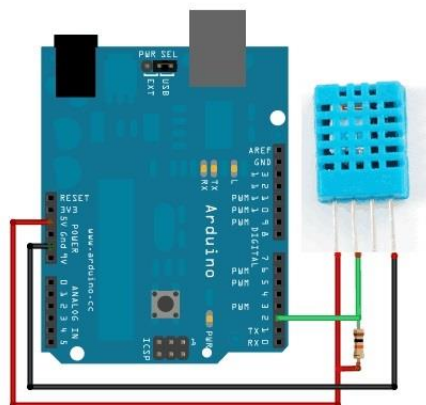
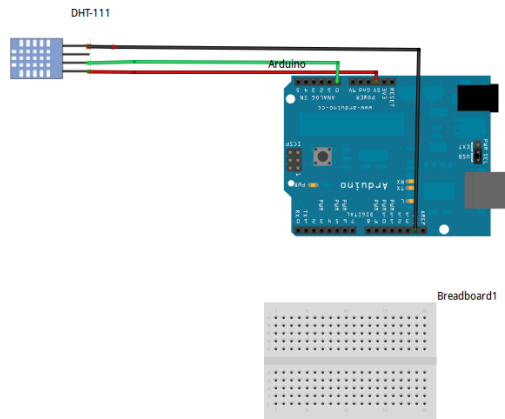
Εικόνα 4-4 / 4-5 Εικόνες της συνδεσμολογίας του αισθητήρα DS18B20

ii. Αισθητήρας θερμοκρασίας /Υγρασίας DHT11

Ο αισθητήρας DHT11 αποτελεί βασικό κομμάτι του project, διότι κάνει καταγραφή θερμοκρασιών αλλά κυρίως του ποσοστού υγρασίας στον χώρο που έχει τοποθετηθεί.

Η σημασία του είναι ακόμα μεγαλύτερη αν συνυπολογίσουμε ότι αποτελεί τον βασικό αισθητήρα στην standalone πλακέτα που μεταφέρει ασύρματα τα δεδομένα. Στο σύστημα πλακετών ενσύρματης μεταφοράς το οποίο αναλύουμε ο DHT11 χρησιμοποιείται για καταγραφή του ποσοστού υγρασίας στην ατμόσφαιρα. Όσον αφορά την συνδεσμολογία ο DHT11 έχει 4 pins εκ των οποίων τα 3 αντιστοιχούν σε υποδοχή πάνω στις πλακέτες μας. Το αριστερό pin αντιστοιχεί στην τροφοδοσία των 5V και συνδέεται στην αντίστοιχη υποδοχή, το αμέσως επόμενο το συνδέουμε με 2ο digital pin της πλακέτας το οποίο θα είναι η ταυτότητα του αισθητήρα στη πλακέτα (στην περίπτωση μας η σύνδεση γίνεται στο digital pin 5). Το επόμενο pin δεν έχει αντίστοιχη υποδοχή στην πλακέτα και τέλος το δεξί ακριανό pin αντιστοιχεί στη θύρα GND. Μεταξύ 2ου και 3ου pin στο breadboard του αισθητήρα, βάζουμε μια αντίσταση των 4,7 ΚΩ.

Ενδεικτικό Schematic συνδεσμολογίας:



- **Πράσινο Καλώδιο:** Σύνδεση με digital pin πάνω στην πλακέτα σε συνδιασμό με αντίσταση 4,7 κΩ
- **Κόκκινο Καλώδιο:** Τροφοδοσία συστήματος με 5V ρεύμα
- **Μαύρο Καλώδιο:** Σύνδεση με GROUND (GND)

Εικόνα 4-6 / 4-7 Schematic συνδεσμολογίας αισθητηρίου υγρασίας DHT11

iii. Αισθητήριο Gas MQ-2

Το αισθητήριο MQ-2 συνδέεται με το σύστημα πλακετών με την χρήση ειδικής βάσης πάνω στην οποία πρώτα συνδέεται με τα pins του και εν συνεχεία όλο μαζί συνδέεται στην πλακέτα μας. Ο MQ-2 έχει 9 pins τα οποία μαζί συνδέονται στη ειδική βάση, τα χρήσιμα pins γίνονται 5 εκ των οποίων θα χρησιμοποιήσουμε για την συνδεσμολογία τα 4.

Αναλυτικότερα, το καλώδιο που συνδέεται με το pin GND θα αντιστοιχεί στην υποδοχή GND στην πλακέτα. Το καλώδιο του pin B1 του αισθητηρίου αντιστοιχεί στην υποδοχή Analog Pin A0 το οποίο και συνδέει, κατά τον προγραμματισμό του αισθητήρα, το αισθητήριο με την πλακέτα. Τέλος τα pins H1 και A1 ενώνονται μεταξύ τους και αντιστοιχούν στην τροφοδοσία των 5V. Να σημειώσουμε ότι μεταξύ των καλωδίων των pins GND και B1 και ενώνουμε μια αντίσταση 4,7KΩ.

Ενδεικτικό Schematics συνδεσμολογίας



Εικόνα 4-8 Schematic συνδεσμολογίας αισθητηρίου GAS MQ-2

Εικόνα της συνδεσμολογίας



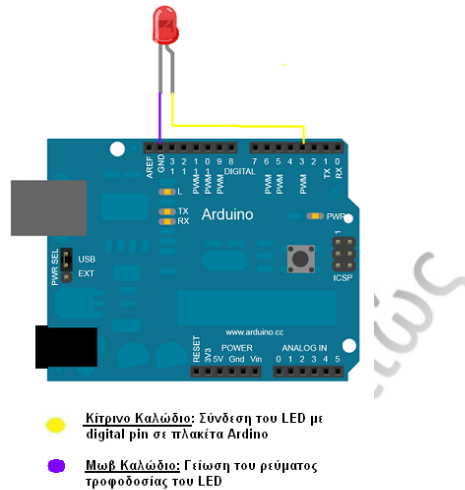
- **Κίτρινο Καλώδιο:** Συνδέεται με την αναλογική θύρα της πλακέτας μας και αποτελεί την "ταυτότητα" του αισθητηρίου στην πλακέτα
- **Κόκκινο Καλώδιο:** Καλώδιο που σε ένωση με το καλώδιο της H1 αντιστοιχούν στην τροφοδοσία των 5V
- **Μαύρο Καλώδιο:** Ενώνεται με την θύρα GND για γείωση του ρεύματος
- **Καφέ Καλώδιο:** Ομοίως με το κόκκινο καλώδιο αντιστοιχεί στην τροφοδοσία

Εικόνα 4-9 Εικόνα της συνδεσμολογίας αισθητηρίου GAS MQ-2 με επεξήγηση

iv. LED Ένδειξης ON/OFF

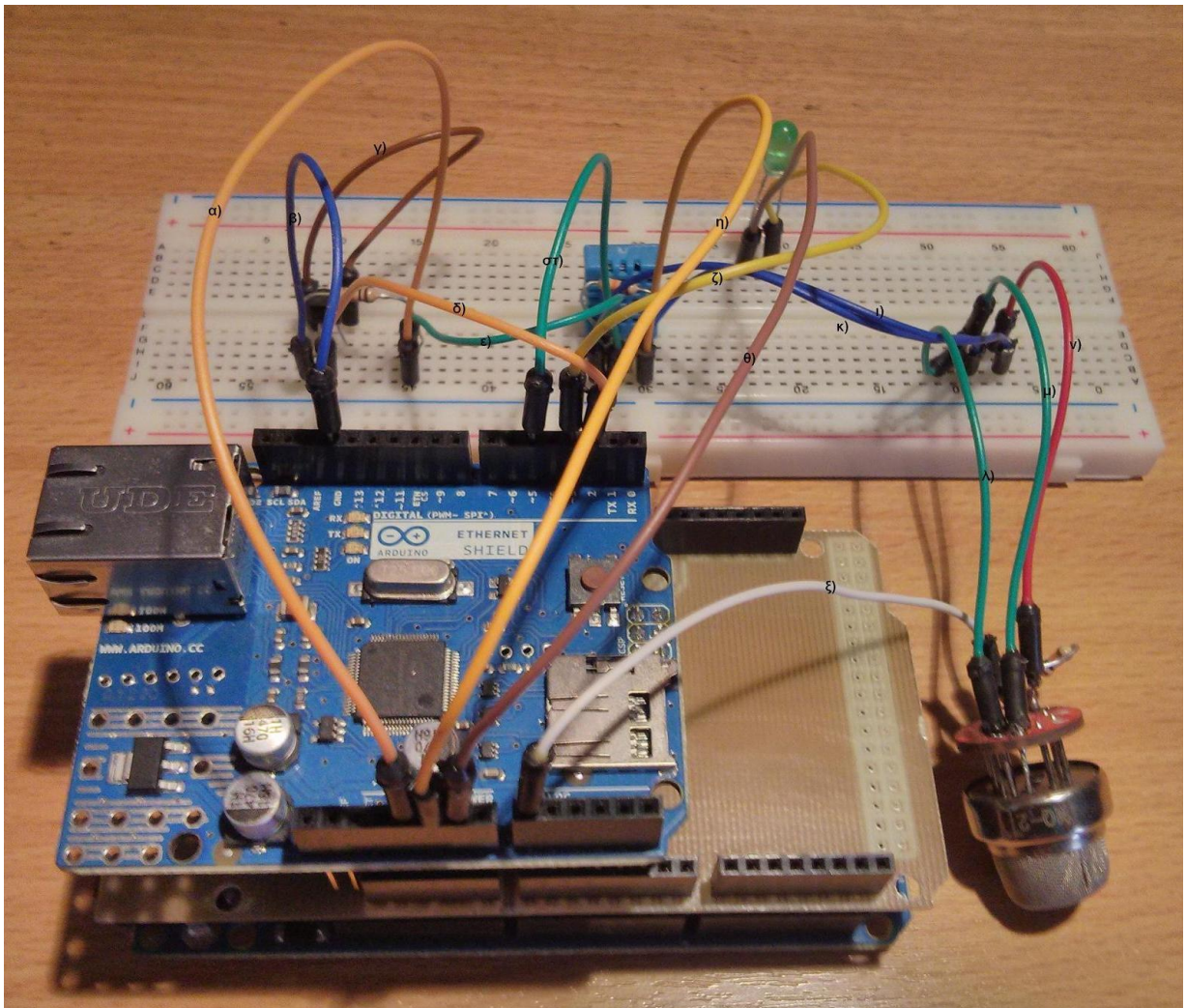
Το Led χρησιμοποιείται όπως αναφέρθηκε, για την ένδειξη ON/OFF στο σύστημα πλακετών. Η συνδεσμολογία είναι απλή διότι αποτελείται από δύο καλώδια εκ των οποίων το ένα συνδέεται με την αναλογική θύρα στην πλακέτα για τον μετέπειτα προγραμματισμό και το άλλο με την θύρα Ground για γείωση του ρεύματος.

Εικόνα της Συνδεσμολογίας:



Εικόνα 4-10 Συνδεσμολογία του LED ένδειξης ON/OFF

Τελική μορφή συνδεσμολογίας συσκευής ενσύρματης μεταφοράς δεδομένων:



Εικόνα 4-11 Τελική μορφή του κυκλώματος του συστήματος αισθητήρων

Υπόμνημα:

- α) Καλώδιο τροφοδοσίας 5V αισθητηρίου θερμοκρασίας DS18B20
- β) Καλώδιο για γείωση ρεύματος αισθητηρίου θερμοκρασίας DS18B20
- γ) Καλώδιο ένωσης 1^{ου} και 3^{ου} pin αισθητηρίου DS18B20
- δ) Καλώδιο σύνδεσης DS18B20 με digital pin 2 συστήματος πλακέτας.
- ε) Καλώδιο μεταφοράς ρεύματος τροφοδοσίας από DS18B20 στο αισθητήριο θερμοκρασίας-υγρασίας DHT11.
- στ) Καλώδιο σύνδεσης αισθητηρίου DHT11 με digital pin 5 συστήματος πλακετών
- ζ) Καλώδιο σύνδεσης LED με digital pin 3 συστήματος πλακετών.
- η) Καλώδιο ένωσης αισθητηρίου DHT11 με GND για γείωση του ρεύματος.
- θ) Καλώδιο γείωσης LED
- ι) Καλώδιο μεταφοράς ρεύματος τροφοδοσίας από DHT11 αισθητήρα σε Gas Sensor MQ-2

- κ) Καλώδιο μεταφοράς γείωσης από DHT11 σε MQ-2
- λ-μ) Καλώδιο τροφοδοσίας αισθητήρων καπνού MQ-2
- ν) Καλώδιο γείωσης αισθητήρων καπνού MQ-2
- ξ) Καλώδιο σύνδεσης αισθητήριου καπνού MQ-2 με Analog pin ΑΟ συστήματος πλακετών.

- Συνδέσεις για κατασκευή συσκευών μέτρησης και ΑΣΥΡΜΑΤΗΣ μεταφοράς δεδομένων.

Η συσκευή ασύρματης μεταφοράς δεδομένων είναι ταυτόχρονα συσκευή καταγραφής και αποθήκευσης θερμοκρασιών – ποσοστού υγρασίας. Σκοπός της ανωτέρω συσκευής, είναι η καταγραφή να γίνεται από την ασύρματη πλακέτα, που θα βρίσκεται σε απόσταση από το κύριο, ενσύρματο σύστημα πλακετών.

Βασικό εξάρτημα του συστήματος ασύρματης μεταφοράς αποτελεί το chip Xbee που υπάρχει στην πλακέτα «πομπό» αλλά και στην πλακέτα «δέκτη». Σκοπό έχει να μεταφέρει το σήμα της καταγραφής από την ασύρματη πλακέτα, με βάση πρωτόκολλα για point-to-point επικοινωνία.

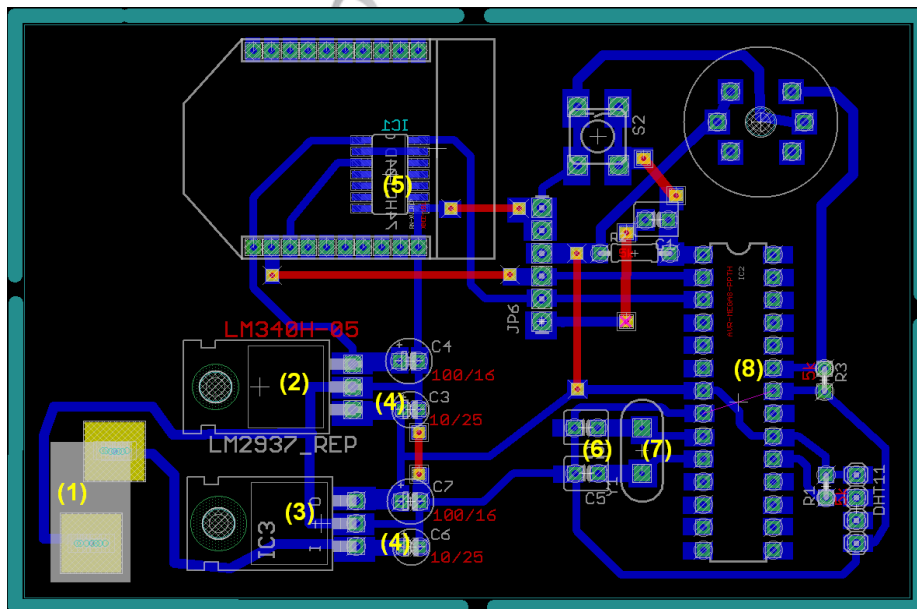
Αναφορικά με την συνδεσμολογία για την κατασκευή της πλακέτας, τα εξαρτήματα της οποίας παρουσιάστηκαν σε προηγούμενο κεφάλαιο, η σύνθεσή τους απαιτούσε την χρήση ηλεκτρολογικών εξαρτημάτων.

Τα κύρια εξαρτήματα της ασύρματης πλακέτας είναι ο μικροεπεξεργαστής ATmega 328 με Arduino Optiboot, όπως επίσης και οι regulators (ισοσταθμιστές) που μεταβάλλουν τα Volt της τροφοδοσίας ανάλογα με τα Volt που θα χρειαστεί τόσο το Xbee όσο και για την μητρική πλακέτα Arduino.

Στη συνέχεια, γίνεται διαγραμματική παρουσίαση της συνδεσμολογίας τόσο για την ασύρματη πλακέτα, όσο και για την πλακέτα «δέκτη» που ενσωματώνεται στο σύνολο- σύστημα των πλακετών ενσύρματης μεταφοράς (κύρια πλακέτα).

Schematic Συνδεσμολογίας

- i. Ασύρματη (απομακρυσμένη) πλακέτα



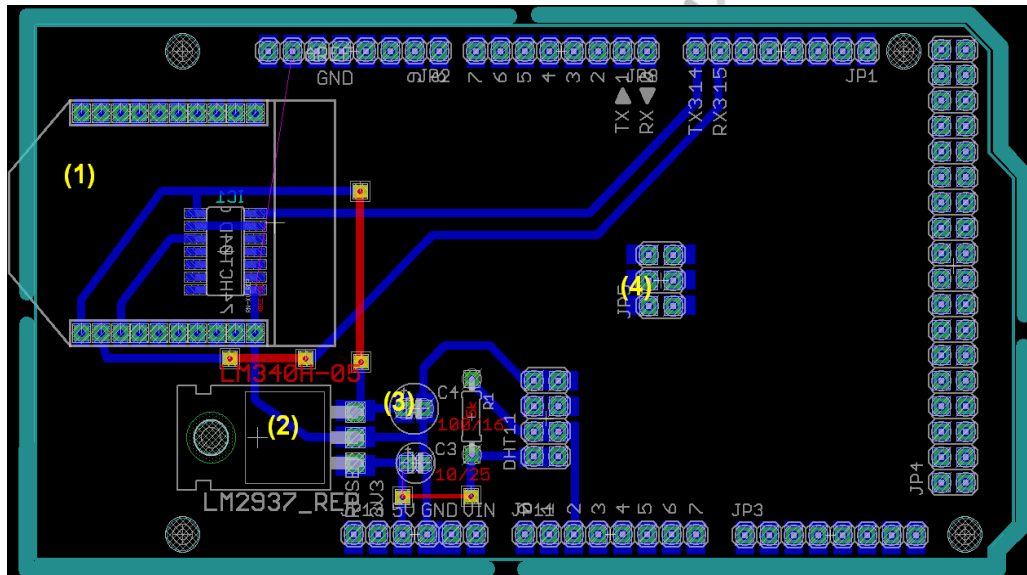
Εικόνα 4-12 Schematic της standalone πλακέτας ΑΣΥΡΜΑΤΗΣ μεταφοράς δεδομένων

Κατασκευή συστήματος καταγραφής, μελέτης και παρουσίασης θερμοκρασιών και πυρανίχνευσης με χρήση αισθητήρων

Υπόμνημα

1. Είσοδος για 9V ρεύμα
2. Regulator που μετατρέπει το ρεύμα των 9V σε 3,3V για τον Xbee
3. Regulator που μετατρέπει το ρεύμα των 9V σε 5V για το Arduino
4. Πυκνωτές (μαζί με Regulators)
5. Ολοκληρωμένος Ελεγκτής (Λογικός Αναστροφέας για μετατροπή 3,3V από 5V)
6. Πυκνωτές (κεραμικοί)
7. Crystal HC-49/S Low Profile (3,57MHz - 20MHz) 16 μεγάλου κύκλου που χρησιμοποιείται για τον χρονισμό του κυκλώματος
8. AtMega 328 με Arduino Optiboot (μικροεπεξεργαστής)

ii. Πλακέτα ενσωματωμένη στο σύνολο πλακετών



Εικόνα 4-13 Schematic της πλακέτας ΑΣΥΡΜΑΤΗΣ μεταφοράς δεδομένων που είναι ενσωματωμένη στο κεντρικό σύστημα αισθητήρων

Υπόμνημα

1. Chip Xbee με 4 εξόδους για:
 - Τροφοδοσία
 - Γείωση
 - RX (Receive data)
 - TX (Transfer data)
 2. Regulator που μετατρέπει το ρεύμα των 9V σε 3,3V για τον Xbee
 3. Πυκνωτές (μαζί με Regulators)
 4. Υποδοχή για ένωση με άλλη πλακέτα
- Τροφοδοσία Πλακετών

Τροφοδοσία πλακέτας ΕΝΣΥΡΜΑΤΗΣ μεταφοράς δεδομένων:

Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V. Η τροφοδοσία να γίνεται είτε απευθείας σε πρίζα με τάση 9V είτε με ένα απλό USB Cable όμοιο με αυτό που συνδέουμε και τον εκτυπωτή.



Εικόνα 4-14 Καλώδιο Τροφοδοσίας μέσω θύρας USB στο PC

Τροφοδοσία πλακέτας ΑΣΥΡΜΑΤΗΣ μεταφοράς δεδομένων:

Η standalone πλακέτα τροφοδοτείται μέσω της θύρας POWER (τροφοδοσίας), απευθείας στην πρίζα με ρεύμα 9V το οποίο με τη χρήση Regulators μεταβάλλει την τάση για την λειτουργία του Arduino και του Xbee

4.2 Σύνδεση Πλακέτας Arduino με Windows (Step by Step διαδικασία)

Εγκατάσταση IDE

Πρώτα κατεβάζει κανείς το PDE το Arduino από το www.arduino.cc. Υποστηρίζει Windows, Mac OS X, και Linux. Ο κώδικας του είναι επίσης ανοικτός (open source).

Το αρχείο εγκατάστασης αποτελείται από ένα αρχείο ZIP, το περιεχόμενο του οποίου μεταφέρεται σε οποιαδήποτε περιοχή του σκληρού δίσκου, χωρίς να απαιτείται setup.

Εγκατάσταση οδηγού (driver)

Με την εγκατάσταση του IDE, συνδέοντας τον μικροελεγκτή στον υπολογιστή με σύρμα USB γίνεται προσπάθεια από το λειτουργικό σύστημα να εγκατασταθεί ο οδηγός. Τα Windows Vista εγκαθιστούν αυτόματα τον οδηγό. Τα Windows 7 αποτυγχάνουν (το οποίο είναι φυσιολογικό). Από το Control Panel/System/Device Manager, βρίσκουμε το Arduino στα Com ports και επιλέγουμε εγκατάσταση

Κατασκευή συστήματος καταγραφής, μελέτης και παρουσίασης θερμοκρασιών και πυρανίχνευσης με χρήση αισθητήρων

οδηγών από τον σκληρό δίσκο. Το αρχείο εγκατάστασης βρίσκεται στο φάκελο Drivers που μεταφέραμε στο δίσκο μας από το Zip.

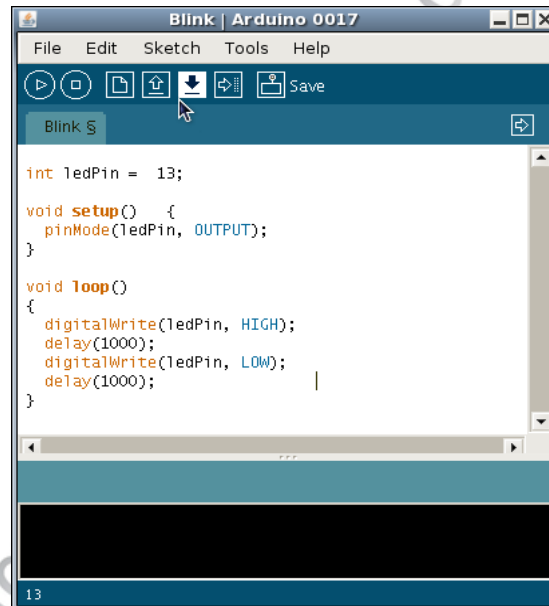
Περιηγούμαστε και βρίσκουμε τον driver με ονομασία "arduino.inf" που βρίσκεται στον φάκελο των drivers. Από εκεί και μετά τα Windows θα συνεχίσουν την εγκατάσταση.

Ανοίγοντας το IDE του Arduino, από το Tools/Board επιλέγουμε τον μικροελεγκτή που έχουμε συνδέσει (πχ Arduino Uno) και από το Tools/Port τον αριθμό θύρας USB με το οποίο είναι συνδεδεμένος. Ακολούθως η πλακέτα είναι έτοιμη για χρήση. Στη σελίδα του Arduino υπάρχει αγγλικός οδηγός εγκατάστασης με εικόνες για Windows, Mac και Linux








4.3 Arduino Software IDE - Processing IDE (Ανάλυση Περιβάλλοντος ανάπτυξης)

α. Το περιβάλλον ανάπτυξης του Arduino Software και του Processing Software

Το περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία πολυπλατφορμική εφαρμογή γραμμένη σε Java και βασίζεται στο περιβάλλον της γλώσσας προγραμματισμού Processing (<http://processing.org/>). Το περιβάλλον ανάπτυξης της πλατφόρμας Processing έχει ακριβώς τις ίδιες λειτουργίες με αυτήν του Arduino.



Εικόνα 4-14 Περιβάλλον Arduino IDE

| | |
|---|--|
|  | Έλεγχος του κώδικα για λάθη. |
|  | Τερματισμός της σειριακής κονσόλας. |
|  | Δημιουργία νέου έργου (sketch). |
|  | Παρουσίαση μενού με όλα τα αποθηκευμένα έργα. Πατώντας σε ένα από αυτά ανοίγει για επεξεργασία. |
|  | Αποθήκευση του έργου. |
|  | Μεταγλώττιση του κώδικα και ανέβασμα του στο Arduino. |
|  | Εμφάνιση της σειριακής κονσόλας. Αποστολή και λήψη δεδομένων που στάλθηκαν μέσω της σειριακής θύρας. |

Ρυθμίσεις του περιβάλλοντος ανάπτυξης:

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

1. Επιλογή πλακέτας. Από το μενού Tools -> Board επιλέγουμε την πλακέτα που έχουμε. Στο συγκεκριμένο οδηγό θα χρησιμοποιήσουμε το Arduino Diecimila, οπότε επιλέγουμε το “Arduino Diecimila, Duemilanove or Nano w/ Atmega168”.

2. Επιλογή σειριακής θύρας. Από το μενού Tools -> Serial Port επιλέγουμε την σειριακή θύρα ή θύρα USB που έχουμε συνδεδεμένο το Arduino (πχ. /dev/ttyUSB0 σε ΛΣ Linux).

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

β. Το περιβάλλον ανάπτυξης του Processing Software

Τι είναι η Processing:

Η Processing είναι μία ανοικτού κώδικα γλώσσα προγραμματισμού και περιβάλλον ανάπτυξης για άτομα που θέλουν να δημιουργήσουν εικόνες, κινούμενα σχέδια και διάφορες αλληλεπιδράσεις. Αρχικά αναπτύχθηκε σαν ένα σχεδιαστικό πρόγραμμα και για να διδάξει βασικές αρχές προγραμματισμού μέσα σε ένα οπτικό πλαίσιο, αλλά στη συνέχεια εξελίχθηκε σε ένα εργαλείο δημιουργίας ολοκληρωμένων επαγγελματικών εργασιών. Αυτή την στιγμή υπάρχουν δεκάδες χιλιάδες σπουδαστές, καλλιτέχνες, σχεδιαστές, ερευνητές και για χόμπι που χρησιμοποιούν τη Processing για διδασκαλία, προτυποποίηση και παραγωγή.

Μερικά από τα χαρακτηριστικά της Processing είναι:

-Είναι ελεύθερο/ανοικτό λογισμικό με άδεια χρήσης GPL/LGPL.

-Είναι πολυπλατφορμική, μπορεί να τρέξει σε λειτουργικά συστήματα GNU/Linux, Mac OS X και Windows.

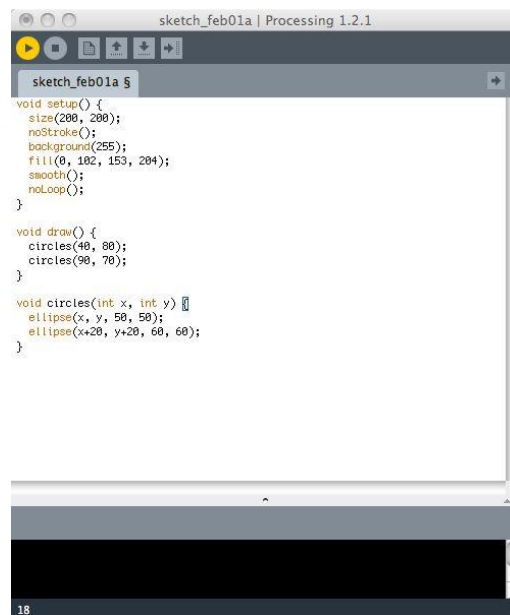
-Δημιουργεί διαδραστικά προγράμματα χρησιμοποιώντας δισδιάστατα (2D) ή τρισδιάστατα (3D) γραφικά.

-Ενσωμάτωση της OpenGL για επιτάχυνση 3D.

-Δημιουργία stand-alone desktop εφαρμογών και web-based εφαρμογών (applets).

-Υπάρχουν αρκετές βιβλιοθήκες επέκτασης της γλώσσας, για εφαρμογές ήχου, βίντεο, τεχνητής όρασης, κ.α

-Η Processing βασίστηκε στη δυνατότητες γραφικών της γλώσσας προγραμματισμού Java, απλοποιώντας την χρήση και δημιουργώντας νέα χαρακτηριστικά.



Εικόνα 4-15 Περιβάλλον Processing IDE

Όπως παρατηρούμε, η δομή του περιβάλλοντος ανάπτυξης της γλώσσας Processing είναι ακριβώς η ίδια με αυτή του περιβάλλοντος της Arduino.

4.4 Τρόπος αποθήκευσης των δεδομένων στην βάση-μεταφορά και ανάλυση περιβάλλοντος phpmyadmin

Ένα από τα πλέον σημαντικά κομμάτια αφορούσε την μεταφορά των δεδομένων που αποστέλλονται από τους αισθητήρες μας, στην βάση δεδομένων μας για περαιτέρω ανάλυση, επεξεργασία και παρουσίαση. Σε πρώτη φάση, θα γίνει μία παρουσίαση και ανάλυση της πλατφόρμας phpMyAdmin μέσω της οποίας διαχειριζόμαστε την βάση δεδομένων, και στην συνέχεια θα αναφέρουμε τον κώδικα που είναι υπεύθυνος για την παραπάνω μεταφορά.

Τί είναι το phpMyAdmin:

Το phpMyAdmin είναι μια εφαρμογή LAMP γραμμένη ειδικά για τη διαχείριση εξυπηρετητών MySQL. Γραμμένο σε PHP και προσβάσιμο μέσω ενός περιηγητή ιστοσελίδων, το phpMyAdmin προσφέρει ένα γραφικό περιβάλλον για εργασίες διαχείρισης βάσεων δεδομένων.

Το phpmyadmin είναι ένα σύνολο από php scripts με το οποίο διαχειριζόμαστε τις βάσεις δεδομένων που έχουμε μέσω web. Το phpmyadmin μπορεί να διαχειριστεί ένα ολόκληρο mysql server ή ακόμα και απλές βάσεις δεδομένων όπου ο κάθε χρήστης έχει ένα λογαριασμό και μπορεί να δημιουργήσει και να διαχειριστεί τις δικές του βάσεις δεδομένων.

Δυνατότητες του phpMyAdmin

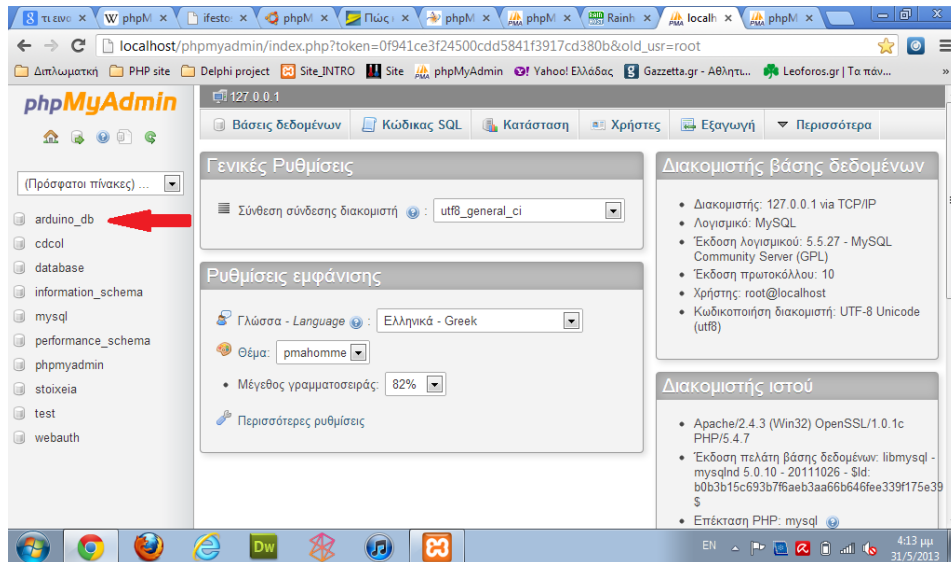
- Δημιουργεί και να διαγράφει βάσεις δεδομένων
- Δημιουργεί, τροποποιεί, διαγράφει, αντιγράφει και μετονομάζει πίνακες
- Κάνει συντήρηση της βάσης
- Προσθέτει, διαγράφει και τροποποιεί πεδία πινάκων
- Εκτελεί ερωτήματα SQL ακόμα και ομαδικά (batch)
- Διαχειρίζεται κλειδιά σε πεδία
- Φορτώνει αρχεία κειμένου σε πίνακες
- Δημιουργεί και διαβάζει πίνακες(που προέρχονται από dump βάσης)
- Εξάγει δεδομένα σε μορφή CVS, XML και LATEX
- Διαχειρίζεται πολλούς διακομιστές
- Διαχειρίζεται τους χρήστες MySQL και τα δικαιώματα τους
- Ελέγχει την αναφορική δραστηριότητα των δεδομένων των MyISAM πινάκων
- Δημιουργεί PDF γραφικών του layout της βάσης δεδομένων
- Εκτελεί αναζητήσεις σε όλη τη βάση δεδομένων ή μέρος αυτής
- Υποστηρίζει πίνακες InnoDB και ξένα κλειδιά
- Υποστηρίζει MySQLi, μια βελτιωμένη επέκταση του MySQLPhpMyAdmin

Διαδικασία εγκατάστασης

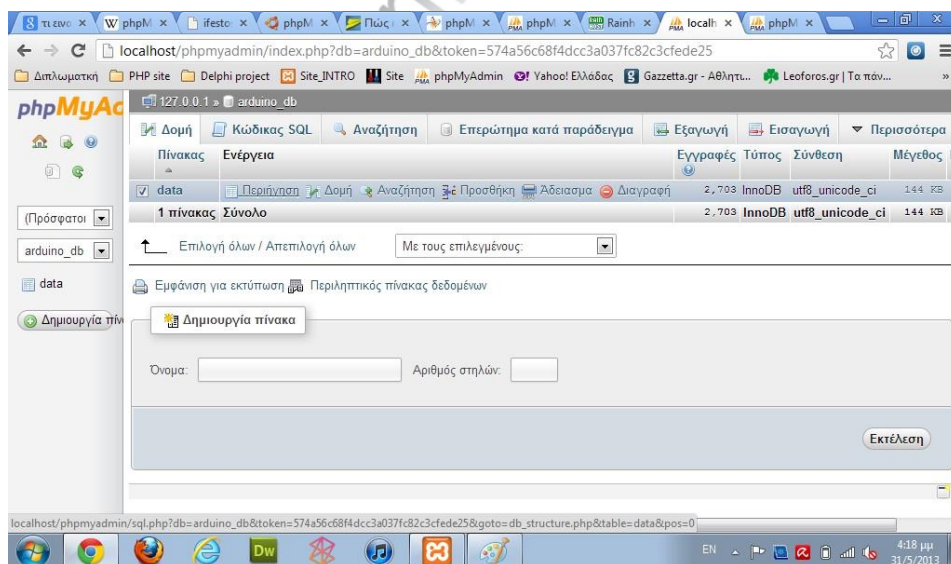
Η πλατφόρμα διαχείρισης της βάσης δεδομένων μας phpMyAdmin εγκαθίσταται αυτόματα μαζί με την εγκατάσταση του προγράμματος ελεύθερου λογισμικού XAMPP .Σε αυτήν την περίπτωση, το μοναδικό που έχουμε να κάνουμε είναι να γράψουμε στην γραμμή διευθύνσεων του browser της επιλογής μας την διεύθυνση <http://localhost/phpmyadmin> και θα έχουμε πρόσβαση στην διαχείριση της βάσης μας.

Παρουσίαση Βάσης Δεδομένων του project

Στις παρακάτω εικόνες παρουσιάζουμε το περιβάλλον και τις δυνατότητες διαχείρισης της βάσης δεδομένων που χρησιμοποιήσαμε στην εργασία.



Εικόνα 4-16: Στην εικόνα βλέπουμε την αρχική σελίδα της πλατφόρμας διαχείρισης και επιλέγουμε την βάση δεδομένων μας με τον τίτλο **arduino_db**

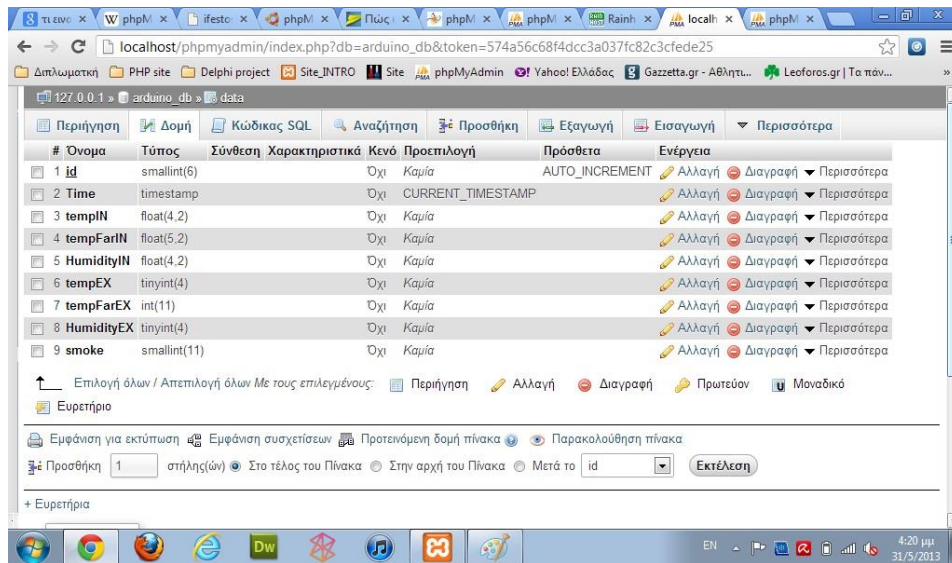


Εικόνα 4-17: Παρατηρούμε ότι η database αποτελείται από 1 table με την ονομασία **data** και επιλέγουμε **περιήγηση** για να δούμε τα επιμέρους στοιχεία του.

| id | Time | tempIN | tempFarIN | HumidityIN | tempEX | tempFarEX | HumidityEX | smoke |
|------|---------------------|--------|-----------|------------|--------|-----------|------------|-------|
| 5236 | 2013-04-20 17:04:04 | 23.81 | 74.86 | 29.00 | 27 | 80 | 27 | 206 |
| 5235 | 2013-04-20 17:03:59 | 23.81 | 74.86 | 29.00 | 27 | 80 | 27 | 206 |
| 5234 | 2013-04-20 17:03:54 | 23.81 | 74.86 | 29.00 | 27 | 80 | 27 | 206 |
| 5233 | 2013-04-20 17:03:49 | 23.81 | 74.86 | 29.00 | 27 | 80 | 28 | 206 |
| 5232 | 2013-04-20 17:03:44 | 23.81 | 74.86 | 29.00 | 27 | 80 | 27 | 206 |
| 5231 | 2013-04-20 17:03:39 | 23.81 | 74.86 | 29.00 | 27 | 80 | 28 | 206 |
| 5230 | 2013-04-20 17:03:34 | 23.81 | 74.86 | 29.00 | 27 | 80 | 28 | 206 |
| 5229 | 2013-04-20 17:03:29 | 23.81 | 74.86 | 29.00 | 27 | 80 | 28 | 206 |
| 5228 | 2013-04-20 17:03:24 | 23.81 | 74.86 | 29.00 | 27 | 80 | 28 | 206 |
| 5227 | 2013-04-20 17:03:19 | 23.81 | 74.86 | 29.00 | 27 | 80 | 28 | 206 |
| 5226 | 2013-04-20 17:03:14 | 23.81 | 74.86 | 29.00 | 27 | 80 | 27 | 806 |

Εικόνα 4-18: Στην εικόνα αυτή βλέπουμε ένα screenshot από την βάση δεδομένων, το table της οποίας αποτελείται από τα εξής χαρακτηριστικά τα οποία ανανεώνονται αυτόματα όταν ο αισθητήρας αποστέλλει data

- id: αποτελεί το πρωτεύον κλειδί της βάσης μας και στην ουσία είναι η ταυτότητα της κάθε εγγραφής.
- Time: αναφέρει την ημερομηνία και την ώρα της κάθε εγγραφής
- tempIN: είναι η στήλη που εμφανίζει την θερμοκρασία (σε βαθμούς °C) η οποία καταγράφεται από τον αισθητήρα ενσύρματης μεταφοράς, δηλαδή τον αισθητήρα που είναι απευθείας συνδεδεμένος στο pc.
- tempFarIN: παρουσιάζει τις θερμοκρασίες (σε βαθμούς Fahrenheit) που καταγράφει ο αισθητήρας ενσύρματης μεταφοράς δεδομένων.
- HumidityIN: παρουσιάζει το ποσοστό υγρασίας (%) που επίσης καταγράφει ο ενσύρματος αισθητήρας
- tempEX: μας δείχνει (σε βαθμούς °C) την θερμοκρασία που καταγράφει και αποστέλλει ο αισθητήρας Ασύρματης μεταφοράς δεδομένων.
- tempFarEX: μας παρουσιάζει την θερμοκρασία (σε βαθμούς Fahrenheit) που καταγράφει ο ασύρματος αισθητήρας
- HumidityEX: Αποτυπώνει το ποσοστό της υγρασίας (%) που καταγράφει ο ασύρματος αισθητήρας
- smoke: Στην στήλη αυτή καταγράφεται ο βαθμός που αποτυπώνει την ύπαρξη ή μη καπνού, και γενικότερα αερίου κοντά στον αισθητήρα ενσύρματης μεταφοράς



Εικόνα 4-19: Στην εικόνα βλέπουμε την δομή του table της βάσης μας με στοιχεία επιπλέον αυτών που αναφέραμε παραπάνω και αφορούν τα χαρακτηριστικά του table.

Κώδικας υπεύθυνος για την μεταφορά δεδομένων στην βάση

Το παρακάτω κομμάτι κώδικα βρίσκεται μέσα στον κεντρικό κώδικα της εργασίας (tempsite.ino) και είναι υπεύθυνο για τη αποστολή των δεδομένων που έχουμε λάβει από τους αισθητήρες, στην βάση δεδομένων μας

```
client.print("GET /arduinoodbcon.php?");
Serial.print("GET http://localhost/arduinoodbcon.php?");
```

Μέσω αυτού του κώδικα ουσιαστικά στέλνουμε τα δεδομένα μας στο αρχείο arduinoodbcon.php που με την σειρά του δημιουργεί το link με την βάση δεδομένων και ταξινομεί τα data στα αντίστοιχα κελιά στην database. Αναλυτικά, την διαδικασία θα την παρουσιάσουμε στο κεφάλαιο που αφορά την αρχιτεκτονική του συστήματος.

Κεφάλαιο 5

Παρουσίαση Συστήματος (User's manual)

5.1 Έναρξη του προγράμματος (step-by-step διαδικασία)

Αφού έχουν γίνει όλες οι απαραίτητες ρυθμίσεις στην εγκατάσταση της πλατφόρμας arduino, έχει γίνει η σύνδεση με τα Windows και έχουν γίνει οι ρυθμίσεις των drivers, μπορούμε να ξεκινήσουμε να βάζουμε σε λειτουργία το πρόγραμμα. Αρχικά, κάνοντας διπλό κλικ στο αρχείο tempsite.ino (*) ανοίγει το πρόγραμμα με τον κώδικα της εφαρμογής μας μέσω του οποίου οι αισθητήρες διαβάζουν, αποθηκεύουν και στέλνουν τα δεδομένα των θερμοκρασιών, υγρασίας και καπνού στη βάση μας.

(*) (.ino είναι η κατάληξη όλων των αρχείων που τρέχουν στην πλατφόρμα arduino)

Ταυτόχρονα, με το άνοιγμα του κώδικα της εφαρμογής, θα κάνουμε διπλό κλικ και στο αρχείο με τίτλο ledonoff.pde το οποίο χρησιμεύει στο να μπορέσουμε να κάνουμε on-off στην εφαρμογή (δηλαδή στην λήψη και την αποστολή δεδομένων από τους αισθητήρες) μέσα από το site μας.

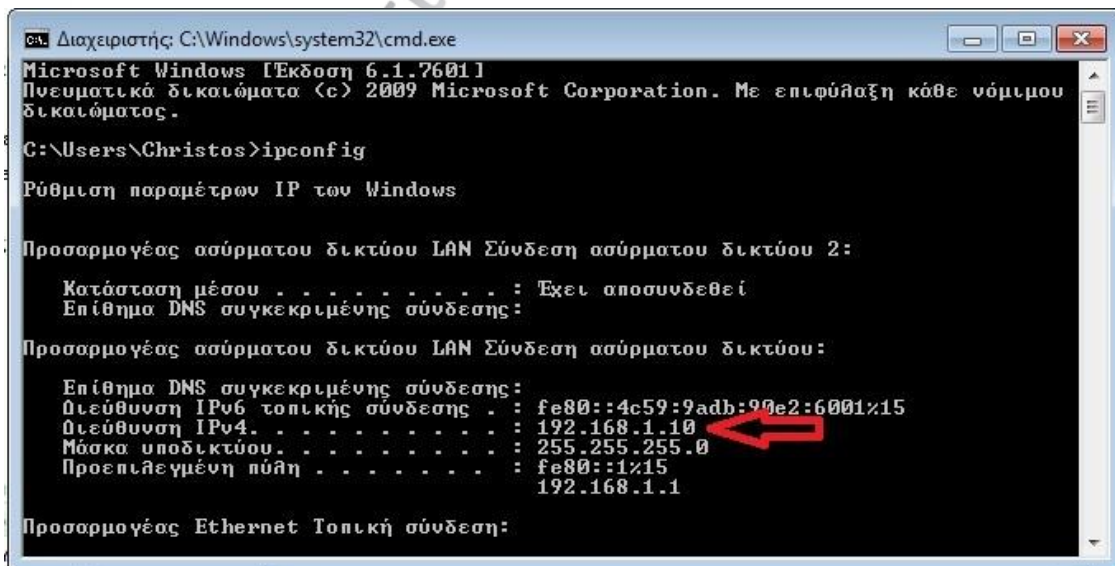
Επομένως, με δεδομένη την λειτουργία αυτή μέσω του site, θα πρέπει να ανοίξουμε και το αρχείο αυτό που βρίσκεται στον φάκελο onoff_led.

Το πρόγραμμα ανοίγει και λειτουργεί μέσα από την πλατφόρμα Processing 1.5.1 η οποία μπορεί και λειτουργεί ταυτόχρονα με την πλατφόρμα Arduino.

Έτσι, από την στιγμή που έχουμε ανοιχτά και έτοιμα να λειτουργήσουν και τα δύο προγράμματα με τους κώδικες έτοιμους για debug, ακολουθούμε τα παρακάτω βήματα:

1^ο βήμα

Ανοίγουμε το command prompt και πληκτρολογούμε στην γραμμή εντολών την εντολή ipconfig. Με αυτόν τον τρόπο βλέπουμε ποια είναι η διεύθυνση ip του υπολογιστή μας (Ipn4)



```

C:\Windows\system32\cmd.exe
Microsoft Windows [Έκδοση 6.1.7601]
Πνευματικά δικαιώματα (c) 2009 Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου
δικαιώματος.
C:\Users\Christos>ipconfig

Ρύθμιση παραμέτρων IP των Windows

Προσαρμογές ασύρματου δικτύου LAN Σύνδεση ασύρματου δικτύου 2:
    Κατάσταση μέσου . . . . . : Έχει αποσυνδεθεί
    Επίθημα DNS συγκεκριμένης σύνδεσης:

Προσαρμογές ασύρματου δικτύου LAN Σύνδεση ασύρματου δικτύου:
    Επίθημα DNS συγκεκριμένης σύνδεσης:
    Διεύθυνση IPv6 τοπικής σύνδεσης . . : fe80::4c59:9adb:90e2:6001%15
    Διεύθυνση IPv4. . . . . : 192.168.1.10
    Μάσκα υποδικτύου. . . . . : 255.255.255.0
    Προεπιλεγμένη πύλη . . . . . : fe80::1%15
    192.168.1.1

Προσαρμογές Ethernet Τοπική σύνδεση:
  
```

Εικόνα 5-1 Διεύθυνση που μας ενδιαφέρει μέσα από το Command Prompt


```

tempste
#define ONE_WIRE_BUS 2 //the onewire bus is connected to pin 7 on arduino
#define TEMPERATURE_PRECISION 10 //resolution of the sensors is set to 10bit(somehow won't work)

// Ethernet settings
uint8_t hwaddr[6] = {
  0x90, 0xA2, 0xDA, 0x0D, 0x1A, 0x88}; // mac-adress of arduino
uint8_t ipaddr[4] = {
  192, 168, 1, 2}; // IP-adress of arduino spon DhcpAddressPrinter ← ip από 2ο βήμα
uint8_t gwaddr[4] = {
  192, 168, 1, 1}; // IP-adress of gateway ( for later DNS implementation)
uint8_t subnet[4] = {
  255, 255, 255, 0}; // subnetmask ( for later DNS implementation)
uint8_t serverip[4] = {
  192, 168, 1, 10}; // IP-adress of server arduino sends data to ← ip από 1ο βήμα
uint8_t serverport = 80; // the port the arduino talks to

EthernetClient client; // make a new instance from type "Client" named "client", giving it
// serverip and serverport

```


Εικόνα 5-3 Συμπλήρωση ip επικοινωνίας με Internet και έναρξη προγράμματος

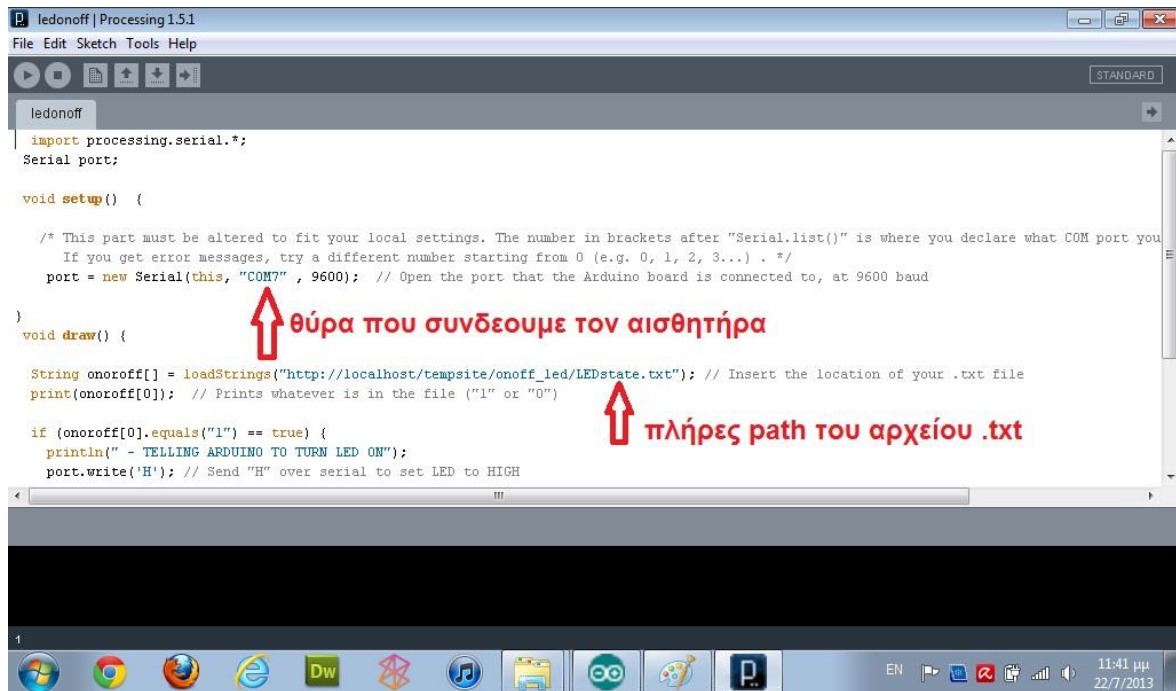
4^ο βήμα

Αμέσως μετά την επιτυχή φόρτωση του προγράμματος από την πλατφόρμα arduino, πηγαίνω στο φάκελο onoff_led και ανοίγω το αρχείο ledonoff το οποίο όπως αναφέραμε και προηγουμένως ανοίγει αυτόματα με την πλατφόρμα Processing 1.5.1. Το αρχείο αυτό χρησιμεύει στην λειτουργία της ενεργοποίησης και της απενεργοποίησης του αισθητήρα, μέσα από site της εφαρμογής.

Οι ρυθμίσεις που απαιτούνται πριν «τρέξουμε» και αυτό το πρόγραμμα είναι οι εξής:

- Στην ενότητα void setup() στο τμήμα του κώδικα που γράφει port = new Serial (this, "COM7",9600); αντικαθιστούμε το COM7 με την θύρα του H/Y στην οποία έχουμε συνδέσει το σύστημα αισθητήρων.
- Στην ενότητα void draw() στο τμήμα του κώδικα που αναγράφει string onoff[] = load strings (" "); θα βάλουμε το πλήρες path στο οποίο βρίσκεται το αρχείο text LEDstate.txt και το οποίο μας δείχνει την παρούσα κατάσταση του LED δίνοντας μας την δυνατότητα να παρακολουθήσουμε αν το σύστημα αισθητήρων είναι ανοιχτό ή κλειστό (0=κλειστό 1=ανοιχτό)

Αφού ρυθμίσουμε και αυτές τις παραμέτρους, πατάμε στο κουμπί  έτσι ώστε να τρέξουμε και το πρόγραμμα αυτό.



Εικόνα 5-4 Έναρξη προγράμματος Processing για προγραμματισμό του LED

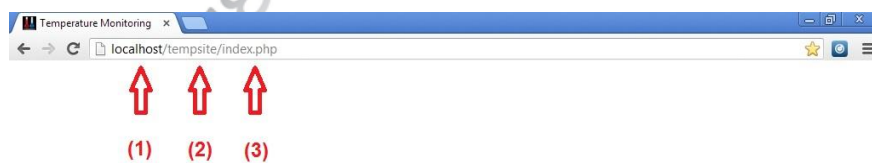
5.2 Περιγραφή και ανάλυση του interface του website

Στην ενότητα αυτή θα περιγράψουμε και θα περιηγηθούμε στο website που απεικονίζει και παρουσιάζει τις θερμοκρασίες και τις μετρήσεις που λαμβάνουμε από τους αισθητήρες.

Αρχικά πρέπει να ανοίξουμε:

- το πρόγραμμα XAMPP το οποίο περιέχει τον host για ιστοσελίδες Http Apache,
- την βάση δεδομένων My SQL
- και έναν διεργαστή για κώδικα γραμμένο σε PHP

και στη συνέχεια ανοίγουμε τον browser και πληκτρολογούμε την παρακάτω διεύθυνση:



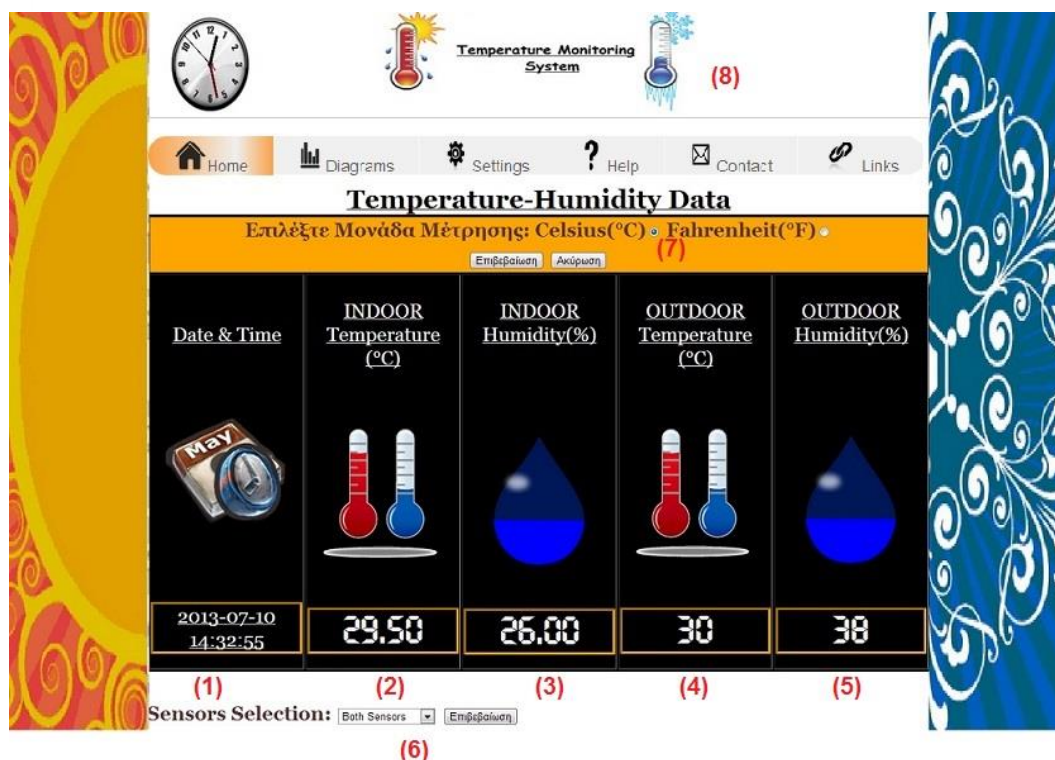
Εικόνα 5-5 Η διεύθυνση του Website της εφαρμογής

- (1) localhost: ο εξηρητητής του site, τα αρχεία του οποίου βρίσκονται στον φάκελο C:/xampp/htdocs
- (2) tempsite : ο φάκελος μέσα στον οποίο υπήρχαν τα αρχεία του site
- (3) index.php: αρχείο σε γλώσσα PHP που περιέχει την html μορφή του και αποτελεί την αρχική σελίδα του site.

Ξεκινάμε την αρχική σελίδα (HOME) η οποία αποτελεί την βασική σελίδα απεικόνισης των δεδομένων που λαμβάνουμε μέσω των αισθητήρων.

Στην σελίδα αυτή απεικονίζεται η ημερομηνία και ώρα που έγινε η καταγραφή μας και στην συνέχεια βλέπουμε την θερμοκρασία και το ποσοστό (%) υγρασίας που καταγράφει ο ενσύρματος αισθητήρας και την θερμοκρασία και υγρασί του ασύρματου. Επιπλέον υπάρχει η δυνατότητα να αλλάξουμε την μονάδα μέτρησης των δειγμάτων μας απο βαθμούς Κελσίου (ο C) σε βαθμούς Fahrenheit (ο F) και αντίστροφα.

Επίσης μπορούμε να απομονώσουμε και να βλέπουμε τα data μόνο απο τον ενσύρματο αισθητήρα, μόνο απο τον ασύρματο και επίσης έχουμε την δυνατότητα παρακολούθησης και των δυο αισθητήρων ταυτόχρονα.



Εικόνα 5-6 Κεντρική σελίδα (HOME) του site

Υπόμνημα

- (1): Απεικόνιση ημερομηνίας και ώρας δειγματοληψίας
- (2): Θερμοκρασία σε βαθμούς Κελσίου (° C) απο καταγραφή ενσύρματου Αισθητήρα.
- (3): Ποσοστό υγρασίας (%) απο καταγραφή ενσύρματου αισθητήρα.
- (4): Θερμοκρασία σε βαθμούς (° C) από καταγραφή ασύρματου – απομακρυσμένου αισθητήρα
- (5): Ποσοστό υγρασίας (%) απο καταγραφή ασύρματου απομακρυσμένου Αισθητήρα.
- (6): Επιλογή για απεικόνιση μόνο ασύρματου, μόνο ενσύρματου ή και των δυο αισθητήρων.

(7): Επιλογή για μονάδα μέτρησης θερμοκρασίας ανάμεσα σε βαθμούς ° C ή ° F

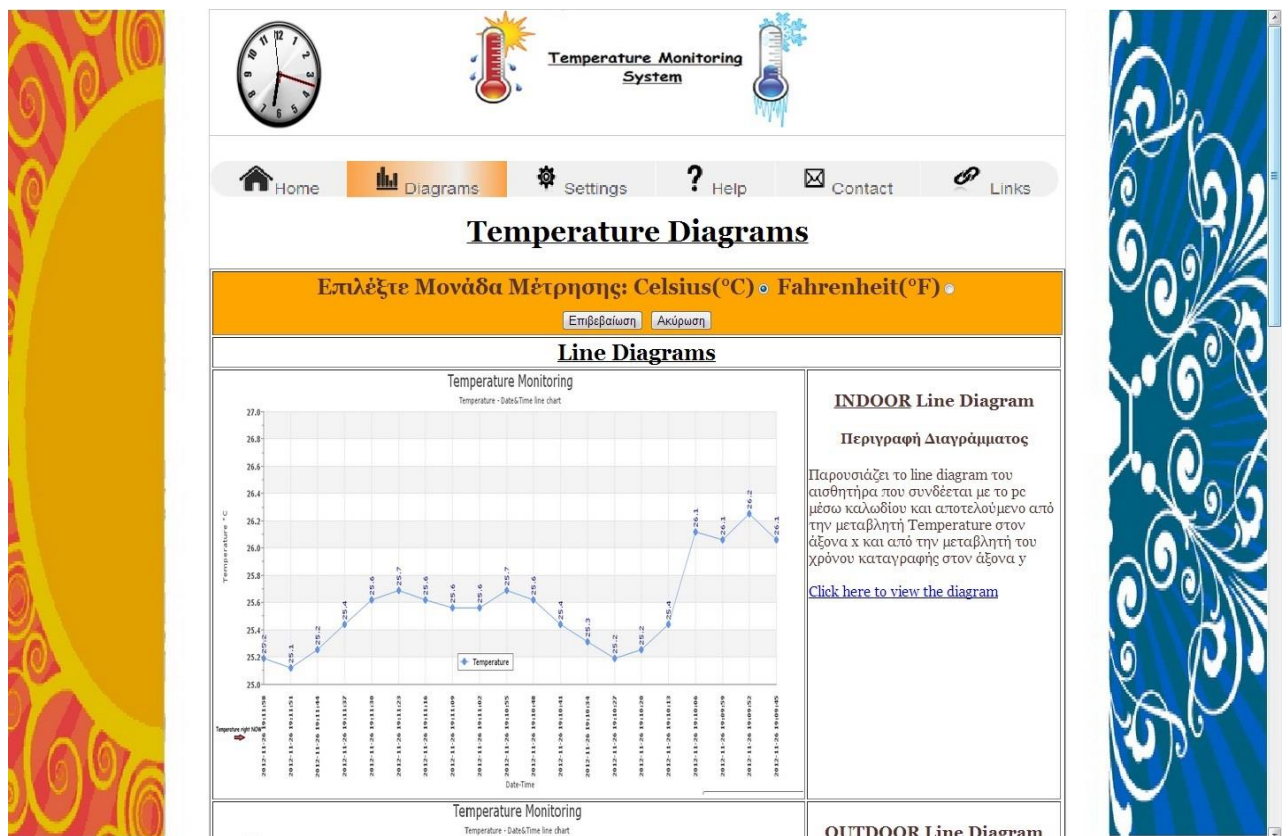
(8): Επιστροφή στην βασική σελίδα.

*Αναφέρουμε οτι το site ανανεώνεται δυναμικά κάθε 10 seconds

Στη συνέχεια, η δεύτερη σελίδα – ενότητα του site περιλαμβάνει την διαγραμματική (Diagrams) απεικόνιση των αλλαγών της θερμοκρασίας σε συνάρτηση με το χρόνο σε δυο διαφορετικά είδη διαγραμμάτων:

Γραμμικό διάγραμμα και ραβδόγραμμα. Συγκεκριμένα, υπάρχει αναλυτική διαγραμματική αναπαράσταση τόσο για τον ενσύρματο όσο και για τον ασύρματο αισθητήρα και φυσικά με δυνατότητα επιλογής παρουσίασης σε βαθμούς ο C και σε βαθμούς ο F.

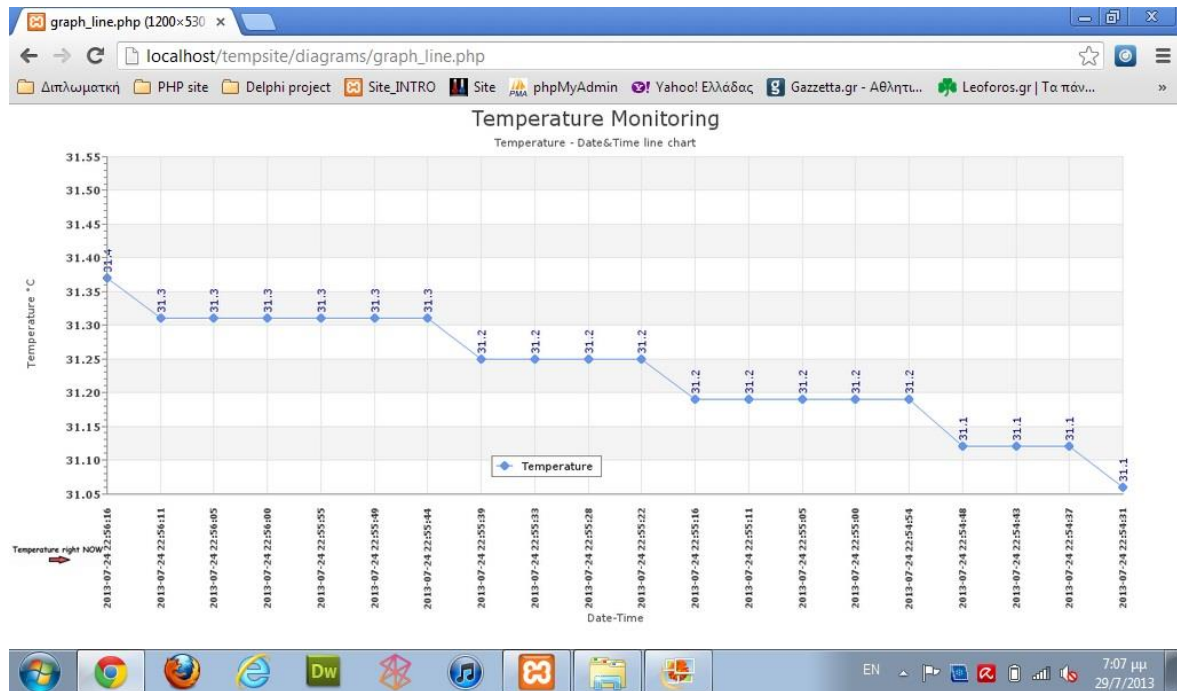
Αναλυτικότερα:



Εικόνα 5-7 Γραμμικό Διάγραμμα

Σε αυτό το screenshot βλέπουμε ένα γραμμικό διάγραμμα που αναπαριστά την μεταβολή της θερμοκρασίας που καταγράφει ο ενσύρματος αισθητήρας (ο C) σε συνάρτηση με τον χρόνο δειγματοληψίας.

Αν κάνουμε κλικ πάνω στο διάγραμμα ή στο link που υπάρχει μας εμφανίζει το διάγραμμα σε μεγαλύτερο και αναλυτικότερο μέγεθος.

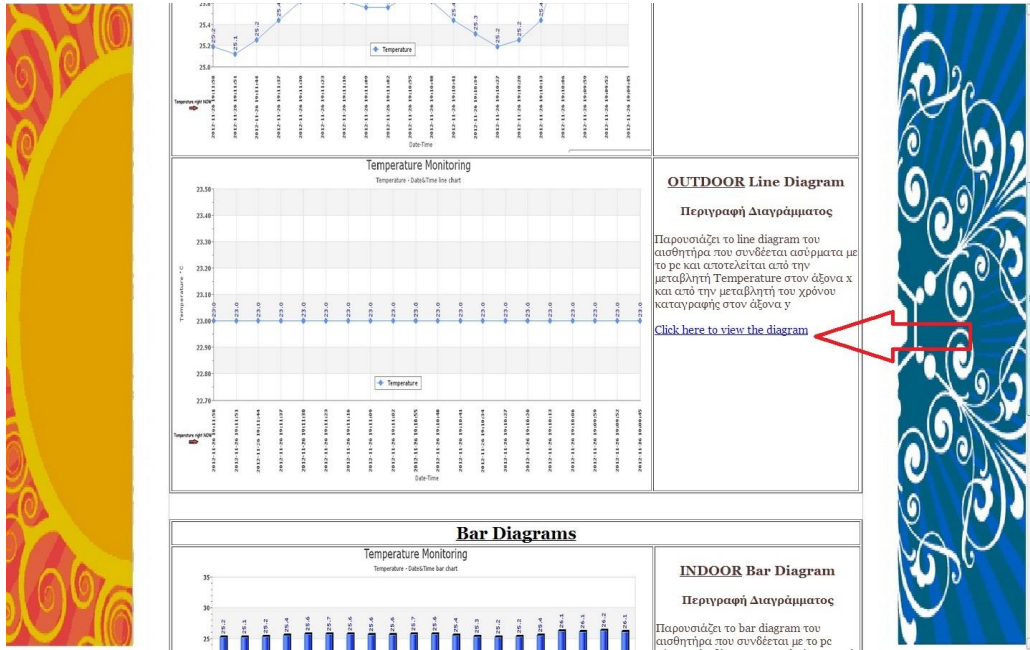


Εικόνα 5-8 Γραμμικό Διάγραμμα αναλυτικό

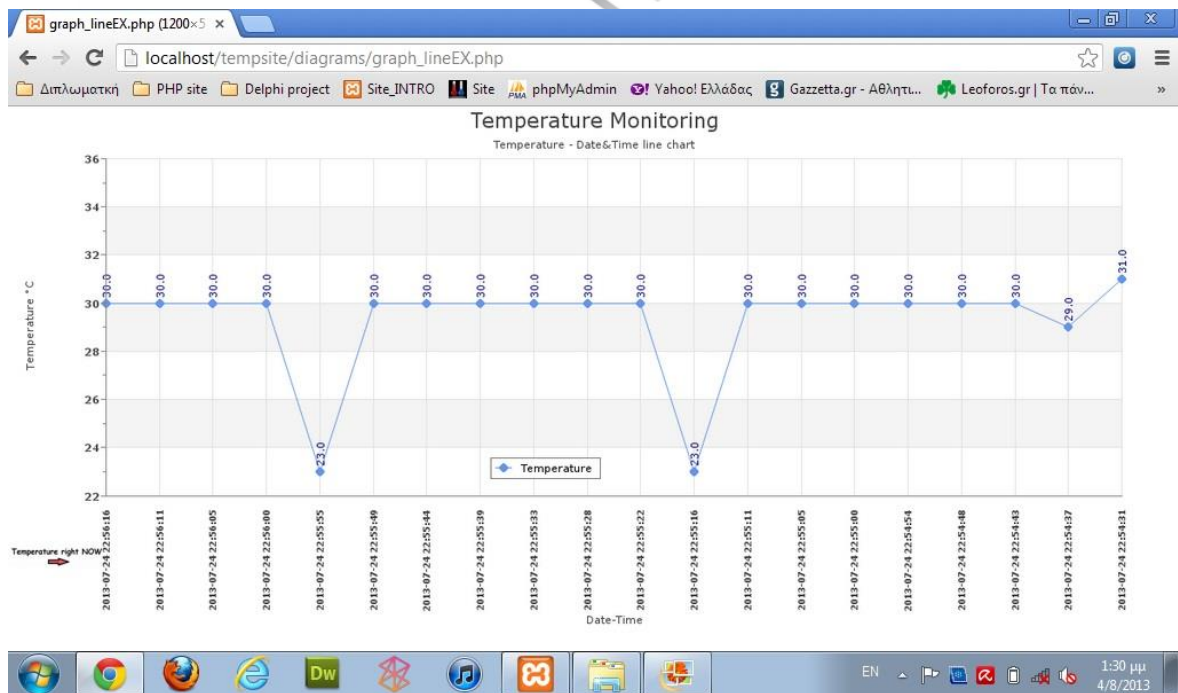
Όπως παρατηρούμε υπάρχει πάντα η δυνατότητα επιλογής ανάμεσα σε βαθμούς C και σε βαθμούς F όπου εμφανίζονται ακριβώς τα ίδια διαγράμματα με την επιλεγόμενη μονάδα μέτρησης.

* Σημειώνουμε ότι τα διαγράμματα, όπως και η ιστοσελίδα ανανεώνονται δυναμικά κάθε 20 sec ώστε να λαμβάνει και να απεικονίζει τα καινούρια δεδομένα.

Ακριβώς κάτω από το γραμμικό διάγραμμα παρουσίασης των μεταβολών του ενσύρματου αισθητήρα, υπάρχει το αντίστοιχο γραμμικό διάγραμμα του ασύρματου με τις ίδιες ακριβώς δυνατότητες που ναλύουμε και παραπάνω. Πρόκειται για διαγραμματική αναπαράσταση συνάρτησης θερμοκρασίας – χρόνου καταγραφής των δεδομένων που στέλνει ο απομακρυσμένος αισθητήρας.



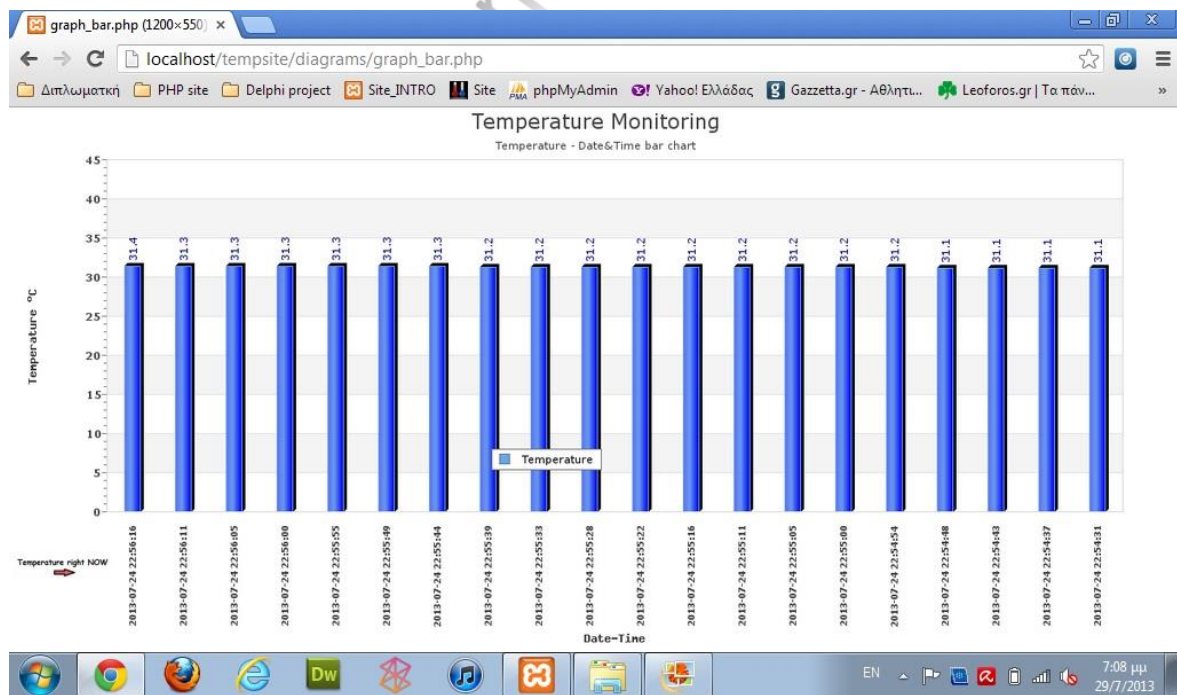
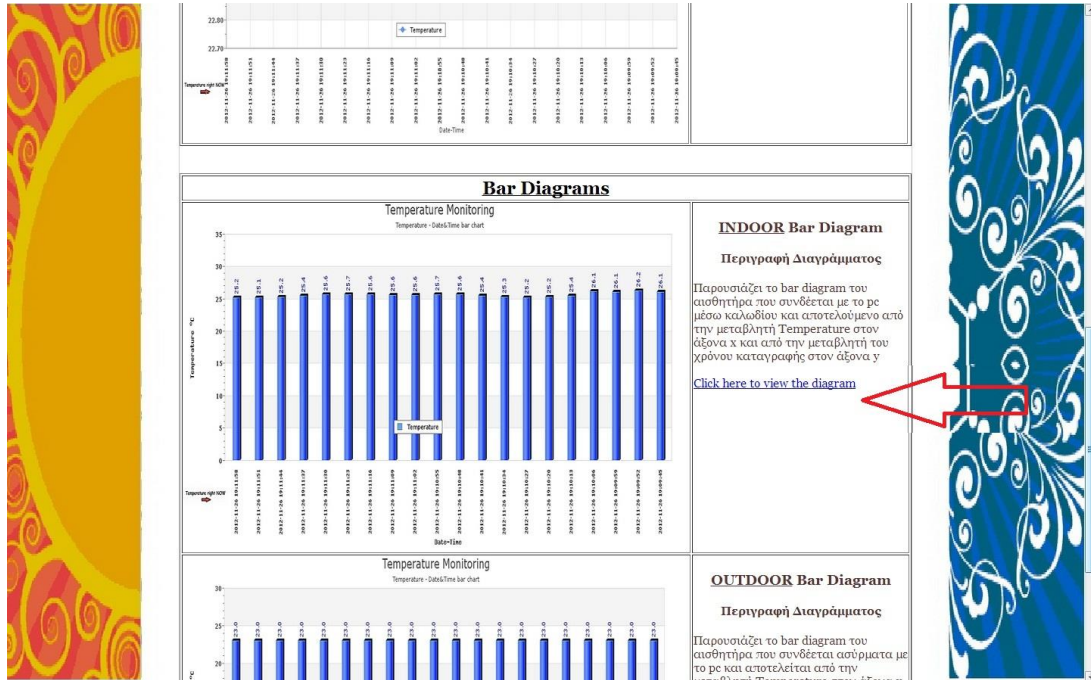
Ομοίως όπως προαναφέρθηκε, αν κάνουμε κλικ στο διάγραμμα αυτό, τότε στο καινούριο παράθυρο που θα εμφανιστεί, παρουσιάζεται αναλυτικά το γραμμικό διάγραμμα του απομακρυσμένου αισθητήρα.



Εικόνα 5-9 / 5-10 Γραμμικό Διάγραμμα από data ασύρματου αισθητήρα

Εν συνεχεία, υπάρχουν τα ραβδογράμματα που ομοίως, αναλύουν και παρουσιάζουν τις μεταβολές της θερμοκρασίας σε συνάρτηση με τον χρόνο. Δίνεται επίσης η δυνατότητα επιλογής μεταξύ των μονάδων μέτρησης ο C και ο F.

Αρχικά το πρώτο ραβδόγραμμα δείχνει την μεταβολή των θερμοκρασιών που λαμβάνουμε απο τον ενσύρματο αισθητήρα, σε συνάρτηση με τον χρόνο και όπως και προηγουμένως, αν κάνουμε κλικ στο διάγραμμα τότε στην καινούρια σελίδα βλέπουμε αναλυτικά το ραβδόγραμμα που ανανεώνεται και αυτό αυτόματα κάθε 20sec.



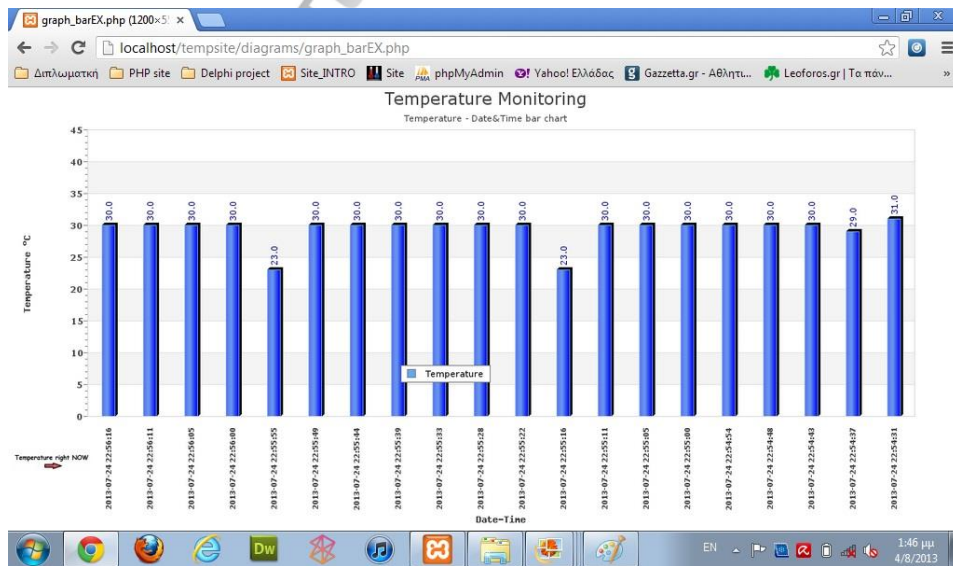
Εικόνα 5-11 / 5-12 Ραβδόγραμμα από data ενσύρματου αισθητήρα

Κατασκευή συστήματος καταγραφής, μελέτης και παρουσίασης θερμοκρασιών και πυρανίχνευσης με χρήση αισθητήρων

Τέλος, υπάρχει και ραβδόγραμμα που παρουσιάζει τις μεταβολές των θερμοκρασιών μέσω των απομακρυσμένων –ασύρματων αισθητήρων. Είναι το τελευταίο διάγραμμα στην λίστα της σελίδας Diagrams του site και αποτελείται από την μεταβλητή temperature στον άξονα X και από την μεταβλητή του χρόνου καταγραφής, στον άξονα Y.



Μετά κλικ στο διάγραμμα:



Εικόνα 5-13 / 5-14 Ραβδόγραμμα από data Ασύρματου αισθητήρα

Συνεχίζουμε την ανάλυση του interface του site με την σελίδα Settings, η οποία είναι βασική καθώς μέσω της σελίδας αυτής μπορούμε να ενεργοποιήσουμε και να απενεργοποιήσουμε τον αισθητήρα και έτσι να ξεκινήσουμε ή να τερματίσουμε την λειτουργία του project.

Παράλληλα, στην σελίδα αυτή παρουσιάζεται ένα ιστορικό των ήδη καταγεγραμμένων δεδομένων που έχουν ληφθεί με δυνατότητα επιλογής ανάμεσα στις 25, 50 ή 100 τελευταίες εγγραφές δεδομένων.

Αναλυτικότερα:



Εικόνα 5-15 Οθόνη με Settings του site

Στο screenshot παρατηρούμε ότι η σελίδα έχει 2 online buttons μέσω των οποίων ενεργοποιούμε και απενεργοποιούμε ολόκληρο το σύστημα καταγραφής. Η λειτουργία αυτή βασίζεται στη χρήση συναρτήσεων που λειτουργούν στο παρασκήνιο και μέσω της πλατφόρμας processing συνδέουν το site με το hardware δηλαδή τους αισθητήρες.

Εκτός όμως των online buttons και της λειτουργίας αυτής, υπάρχει και ιστορικό καταγεγραμμένων δεδομένων όπου ο χρήστης με την χρήση Radio Button επιλέγει πόσες από τις τελευταίες εγγραφές θέλει να δει. Στην σελίδα που αναδύεται (π.χ. στα 25 τελευταία δεδομένα)

Data History

| Date & Time | Temperature Indoor(C) | Temperature Indoor(Far) | Humidity Indoor(%) | Temperature Outdoor(C) | Temperature Outdoor(Far) | Humidity Outdoor(%) |
|---------------------|-----------------------|-------------------------|--------------------|------------------------|--------------------------|---------------------|
| 2013-07-24 22:56:16 | 31.37 | 88.47 | 15.00 | 30 | 86 | 24 |
| 2013-07-24 22:56:11 | 31.31 | 88.36 | 15.00 | 30 | 86 | 23 |
| 2013-07-24 22:56:05 | 31.31 | 88.36 | 15.00 | 30 | 86 | 24 |
| 2013-07-24 22:56:00 | 31.31 | 88.36 | 15.00 | 30 | 86 | 24 |
| 2013-07-24 22:55:55 | 31.31 | 88.36 | 15.00 | 23 | 73 | 23 |
| 2013-07-24 22:55:49 | 31.31 | 88.36 | 15.00 | 30 | 86 | 23 |
| 2013-07-24 22:55:44 | 31.31 | 88.36 | 15.00 | 30 | 86 | 24 |
| 2013-07-24 22:55:39 | 31.25 | 88.25 | 15.00 | 30 | 86 | 24 |
| 2013-07-24 22:55:33 | 31.25 | 88.25 | 15.00 | 30 | 86 | 24 |
| 2013-07-24 22:55:28 | 31.25 | 88.25 | 15.00 | 30 | 86 | 24 |
| 2013-07-24 22:55:22 | 31.25 | 88.25 | 14.00 | 30 | 86 | 23 |
| 2013-07-24 22:55:16 | 31.19 | 88.14 | 15.00 | 23 | 73 | 23 |
| 2013-07-24 22:55:11 | 31.19 | 88.14 | 14.00 | 30 | 86 | 24 |
| 2013-07-24 22:55:05 | 31.19 | 88.14 | 14.00 | 30 | 86 | 24 |
| 2013-07-24 22:55:00 | 31.19 | 88.14 | 14.00 | 30 | 86 | 23 |
| 2013-07-24 22:54:54 | 31.19 | 88.14 | 14.00 | 30 | 86 | 23 |
| 2013-07-24 22:54:48 | 31.12 | 88.02 | 14.00 | 30 | 86 | 24 |
| 2013-07-24 22:54:43 | 31.12 | 88.02 | 14.00 | 30 | 86 | 23 |
| 2013-07-24 22:54:37 | 31.12 | 88.02 | 14.00 | 29 | 84 | 29 |
| 2013-07-24 22:54:31 | 31.06 | 87.91 | 14.00 | 31 | 88 | 29 |
| 2013-07-24 22:54:26 | 31.06 | 87.91 | 14.00 | 31 | 88 | 29 |
| 2013-07-24 22:54:20 | 31.06 | 87.91 | 14.00 | 31 | 88 | 29 |
| 2013-07-24 22:54:14 | 31.06 | 87.91 | 14.00 | 31 | 88 | 29 |
| 2013-07-24 22:54:09 | 31.00 | 87.80 | 14.00 | 31 | 88 | 29 |
| 2013-07-24 22:54:03 | 31.00 | 87.80 | 15.00 | 31 | 88 | 29 |

INDOOR Highest/Lowest Data **OUTDOOR Highest/Lowest Data**
 Highest Recorded Indoor Temperature (°C) is: 61.25 °C recorded at 2012-11-26 18:28:31 Highest Recorded Outdoor Temperature (°C) is: 56 °C recorded at 2013-04-20 16:14:58
 Lowest Recorded Indoor Temperature (°C) is: 17.19 °C recorded at 2013-01-10 17:16:12 Lowest Recorded Outdoor Temperature (°C) is: 0 °C recorded at 2013-02-22 14:34:53
 Highest Recorded Indoor Humidity (%) is: 38.00 % recorded at 2013-02-22 14:32:31 Highest Recorded Outdoor Humidity (%) is: 127 % recorded at 2012-12-27 15:33:38
 Lowest Recorded Indoor Humidity (%) is: 12.00 % recorded at 2013-07-23 22:07:51 Lowest Recorded Outdoor Humidity (%) is: -128 % recorded at 2012-12-27 17:54:24

Επιπλέψτε στην προηγούμενη σελίδα

(2) (3)

Εικόνα 5-16 Παρουσίαση των 25 τελευταίων εγγραφών

Υπόμνημα

- (1) Πίνακας με αναλυτική παρουσίαση όλων των δεδομένων, σε βαθμούς °C και °F, που λαμβάνονται τόσο από τον ενσύρματο όσο και από τον ασύρματο αισθητήρα.
- (2) Μέγιστη και ελάχιστη καταγεγραμμένη θερμοκρασία καθώς και το ποσοστό % υγρασίας που έχει αποσταλεί ως data από τον ενσύρματο αισθητήρα στην βάση δεδομένων.
- (3) Μέγιστη και ελάχιστη καταγεγραμμένη θερμοκρασία καθώς και το ποσοστό % υγρασίας που έχει αποσταλεί ως data από τον ασύρματο αισθητήρα στην βάση δεδομένων.

Παρόμοια, με την επιλογή παρουσίασης των τελευταίων 25 καταγραφών, υπάρχει και η επιλογή για 50 ή 100 τελευταίες, με τις ίδιες αναλύσεις, και μέγιστες και ελάχιστες καταγραφές όπως παραπάνω.

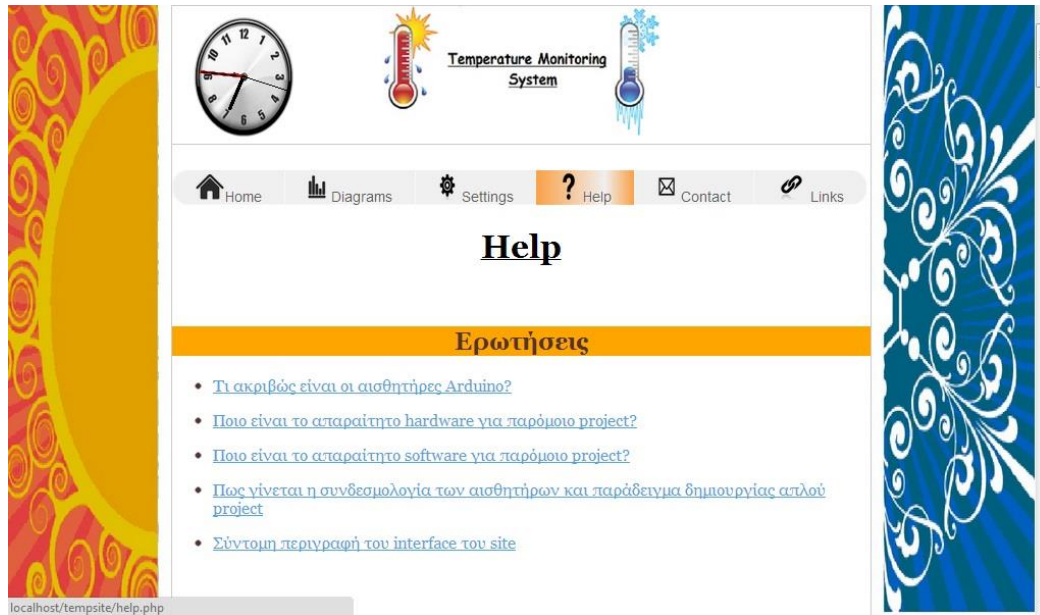
Ενδεικτικά για τα τελευταία 50 δεδομένα το screenshot είναι το εξής:

| Data History | | | | | | | |
|---------------------|-----------------------|-------------------------|--------------------|------------------------|--------------------------|---------------------|--|
| Date & Time | Temperature Indoor(C) | Temperature Indoor(Far) | Humidity Indoor(%) | Temperature Outdoor(C) | Temperature Outdoor(Far) | Humidity Outdoor(%) | |
| 2013-07-24 22:56:16 | 31.37 | 88.47 | 15.00 | 30 | 86 | 24 | |
| 2013-07-24 22:56:11 | 31.31 | 88.56 | 15.00 | 30 | 86 | 23 | |
| 2013-07-24 22:56:06 | 31.31 | 88.56 | 15.00 | 30 | 86 | 24 | |
| 2013-07-24 22:56:00 | 31.31 | 88.56 | 15.00 | 30 | 86 | 24 | |
| 2013-07-24 22:55:54 | 31.31 | 88.56 | 15.00 | 30 | 86 | 23 | |
| 2013-07-24 22:55:48 | 31.31 | 88.56 | 15.00 | 30 | 86 | 23 | |
| 2013-07-24 22:55:44 | 31.31 | 88.56 | 15.00 | 30 | 86 | 24 | |
| 2013-07-24 22:55:39 | 31.25 | 88.25 | 15.00 | 30 | 86 | 24 | |
| 2013-07-24 22:55:33 | 31.25 | 88.25 | 15.00 | 30 | 86 | 24 | |
| 2013-07-24 22:55:28 | 31.25 | 88.25 | 15.00 | 30 | 86 | 24 | |
| 2013-07-24 22:55:23 | 31.25 | 88.25 | 14.00 | 30 | 86 | 23 | |
| 2013-07-24 22:55:18 | 31.19 | 88.14 | 15.00 | 23 | 73 | 23 | |
| 2013-07-24 22:55:13 | 31.19 | 88.14 | 14.00 | 30 | 86 | 24 | |
| 2013-07-24 22:55:08 | 31.19 | 88.14 | 14.00 | 30 | 86 | 24 | |
| 2013-07-24 22:55:00 | 31.19 | 88.14 | 14.00 | 30 | 86 | 23 | |
| 2013-07-24 22:54:54 | 31.19 | 88.14 | 14.00 | 30 | 86 | 23 | |
| 2013-07-24 22:54:48 | 31.12 | 88.02 | 14.00 | 30 | 86 | 24 | |
| 2013-07-24 22:54:43 | 31.12 | 88.02 | 14.00 | 30 | 86 | 23 | |
| 2013-07-24 22:54:37 | 31.12 | 88.02 | 14.00 | 29 | 84 | 29 | |
| 2013-07-24 22:54:31 | 31.06 | 87.91 | 14.00 | 31 | 88 | 29 | |
| 2013-07-24 22:54:26 | 31.06 | 87.91 | 14.00 | 31 | 88 | 29 | |
| 2013-07-24 22:54:20 | 31.06 | 87.91 | 14.00 | 31 | 88 | 29 | |
| 2013-07-24 22:54:14 | 31.06 | 87.91 | 14.00 | 31 | 88 | 29 | |
| 2013-07-24 22:54:08 | 31.00 | 87.80 | 14.00 | 31 | 88 | 29 | |
| 2013-07-24 22:54:03 | 31.00 | 87.80 | 15.00 | 31 | 88 | 28 | |
| 2013-07-24 22:53:58 | 31.37 | 88.47 | 16.00 | 31 | 88 | 29 | |
| 2013-07-24 22:53:53 | 31.44 | 88.59 | 15.00 | 31 | 88 | 29 | |
| 2013-07-25 02:08:09 | 30.00 | 86.00 | 12.00 | 30 | 86 | 20 | |
| 2013-07-25 02:08:03 | 29.94 | 85.89 | 12.00 | 30 | 86 | 127 | |
| 2013-07-25 02:07:57 | 29.94 | 85.89 | 19.00 | 30 | 86 | 20 | |
| 2013-07-25 02:07:51 | 29.87 | 85.77 | 12.00 | 30 | 86 | 20 | |
| 2013-07-25 02:07:46 | 29.87 | 85.77 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:08:40 | 29.94 | 85.89 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:08:34 | 29.87 | 85.77 | 18.00 | 30 | 86 | 127 | |
| 2013-07-25 02:08:28 | 29.87 | 85.77 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:08:24 | 29.87 | 85.77 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:08:18 | 29.87 | 85.77 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:08:14 | 29.87 | 85.77 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:08:08 | 29.87 | 85.77 | 18.00 | 30 | 86 | 127 | |
| 2013-07-25 02:08:03 | 29.87 | 85.77 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:07:58 | 29.87 | 85.77 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:07:52 | 29.87 | 85.77 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:07:46 | 29.87 | 85.77 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:07:40 | 29.94 | 85.89 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:07:34 | 29.94 | 85.89 | 18.00 | 30 | 86 | 127 | |
| 2013-07-25 02:07:28 | 29.94 | 85.89 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:07:24 | 29.94 | 85.89 | 18.00 | 20 | 88 | 20 | |
| 2013-07-25 02:07:18 | 29.94 | 85.89 | 18.00 | 20 | 88 | 0 | |
| 2013-07-25 02:07:12 | 29.87 | 85.77 | 18.00 | 30 | 86 | 20 | |
| 2013-07-25 02:07:04 | 29.87 | 85.77 | 18.00 | 30 | 86 | 20 | |

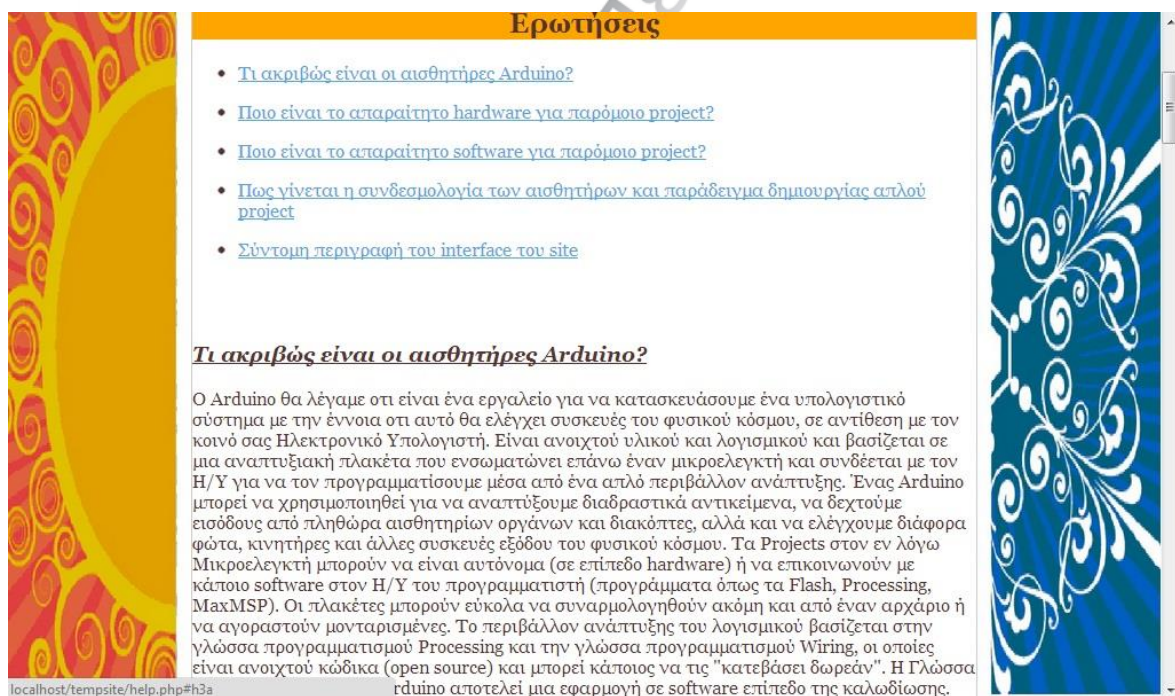
INDOOR Highest/Lowest Data **OUTDOOR Highest/Lowest Data**
 Highest Recorded Indoor Temperature (°C) is: 61.25 °C recorded at 2012-11-26 18:28:31 Highest Recorded Outdoor Temperature (°C) is: 56 °C recorded at 2013-04-20 16:14:58
 Lowest Recorded Indoor Temperature (°C) is: 17.19 °C recorded at 2013-01-10 17:16:11 Lowest Recorded Outdoor Temperature (°C) is: 0 °C recorded at 2013-02-21 14:54:53
 Highest Recorded Indoor Humidity (%) is: 58.00 % recorded at 2013-02-22 14:32:31 Highest Recorded Outdoor Humidity (%) is: 127 % recorded at 2012-12-27 15:55:38

Εικόνα 5-17 Παρουσίαση των τελευταίων 50 εγγραφών

Επόμενο κομμάτι της σελίδας είναι η ενότητα Help όπου πολύ συνοπτικά υπάρχουν κάποιες βασικές ερωτήσεις - απαντήσεις και πληροφορίες σχετικά με τους αισθητήρες Arduino που χρησιμοποιήθηκαν για την υλοποίηση του project, οδηγίες και απαραίτητα εργαλεία για την κατασκευή παρόμοιων εφαρμογών και ένας ενδεικτικός οδηγός του website. Ουσιαστικά πρόκειται για ένα συνοπτικό online manual με πληροφορίες σχετικές με το project και το website.



Οι ερωτήσεις με το μπλε χρώμα γραμματοσειράς είναι link τα οποία παραπέμπουν το χρήστη στο αντίστοιχο κομμάτι κειμένου που παρέχει την απάντηση,



Εικόνα 5-18 / 5-19 Οθόνη με μενού βοήθειας του site

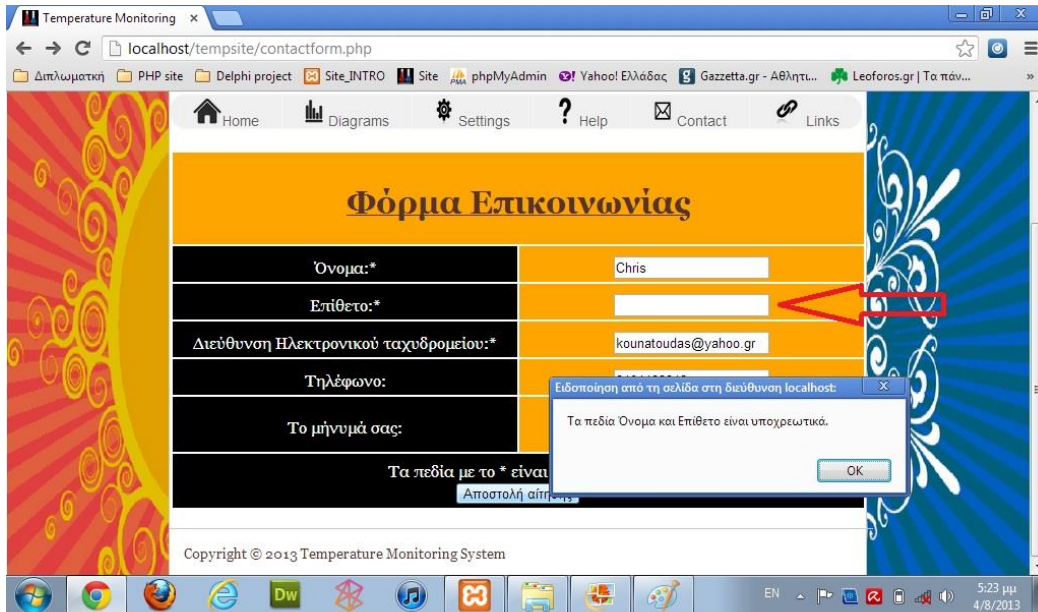
Στην συνέχεια, και στην σελίδα Contact, ο χρήστης έχει μπροστά του μία φόρμα επικοινωνίας στην οποία συμπληρώνει τα στοιχεία του και συντάσσει το μήνυμα που θέλει να αποστείλει στον διαχειριστή της ιστοσελίδας.

The screenshot shows the 'Temperature Monitoring System' website. At the top, there is a navigation menu with icons for Home, Diagrams, Settings, Help, Contact, and Links. The 'Contact' menu item is highlighted. Below the navigation is a large orange banner with the text 'Φόρμα Επικοινωνίας'. The form consists of several input fields: 'Όνομα:*', 'Επίθετο:*', 'Διεύθυνση Ηλεκτρονικού ταχυδρομείου:*', 'Τηλέφωνο:', and 'Το μήνυμά σας:'. Below the message field is a button labeled 'Αποστολή αίτησης'. At the bottom of the form, there is a note: 'Τα πεδία με το * είναι υποχρεωτικά'. The footer of the page reads 'Copyright © 2013 Temperature Monitoring System'.

Ο χρήστης συμπληρώνει υποχρεωτικά το όνομα του, το επώνυμό του και την διεύθυνση ηλεκτρονικού ταχυδρομείου και στην συνέχεια το μήνυμά του. Σε περίπτωση που όλα τα στοιχεία έχουν την σωστή μορφή, τότε το μήνυμα αποστέλλεται επιτυχώς.

The screenshot shows the same website after a successful submission. The main content area now displays a large orange banner with the text 'Το μήνυμά σας εστάλη επιτυχώς'. Below this banner is a button labeled 'Επιστρέψτε στην προηγούμενη σελίδα'. The footer of the page remains 'Copyright © 2013 Temperature Monitoring System'.

Σε διαφορετική περίπτωση, μας εμφανίζει μήνυμα λάθους με την κατάλληλη επισήμανση για το πού έγινε αυτό



Εικόνα 5-20 / 5-21 / 5-22 Εικόνες από συμπλήρωση φόρμας επικοινωνίας στο site

Τελευταία ενότητα - σελίδα του website του project είναι η σελίδα Links η οποία περιέχει κάποια χρήσιμα links σχετικά με το αντικείμενο της εργασίας που είναι η καταγραφή, μελέτη και αναλυτική παρουσίαση δεδομένων θερμοκρασίας και υγρασίας σε εσωτερικό χώρο. Ενδεικτικά links όπως αυτό της EMY ή επίσης ιστοσελίδες καιρικών προβλέψεων αλλά ακόμη και σελίδες σχετικές με το hardware και το software που χρησιμοποιήθηκε για την υλοποίηση του project.

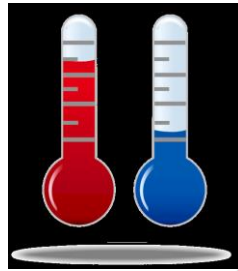


Εικόνα 5-23 Οθόνη με χρήσιμα Links μέσω της ιστοσελίδας

5.3 Περιγραφή και ανάλυση του interface της Delphi εφαρμογής

Πέρα από το website που αποτελεί τον βασικό τρόπο παρουσίασης και ανάλυσης των δεδομένων που κατέγραψαν οι αισθητήρες, έχει κατασκευαστεί και μία desktop εφαρμογή σε Delphi και η οποία παρουσιάζει επίσης τις θερμοκρασίες των δύο αισθητήρων αλλά επιπλέον και την μέτρηση που καταγράφει ο αισθητήρας καπνού Gas Sensor MQ-2 που είναι ενσωματωμένος πάνω στον ενσύρματο αισθητήρα. Η εφαρμογή ανανεώνεται αυτόματα κάθε 10 seconds έτσι ώστε να συμβαδίζει με τα τελευταία data που αποθηκεύονται στην βάση δεδομένων μας.

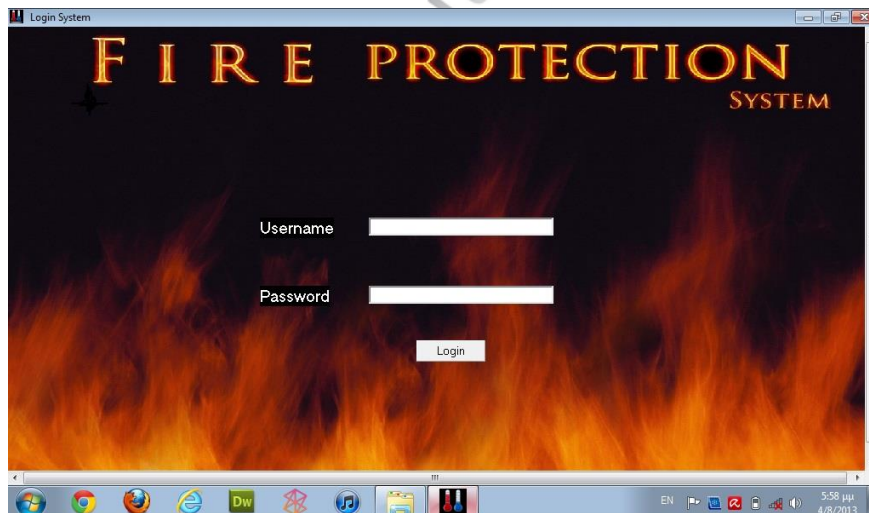
Η εφαρμογή ονομάζεται Fire Protection System και στην επιφάνεια εργασίας έχει το εξής εικονίδιο:



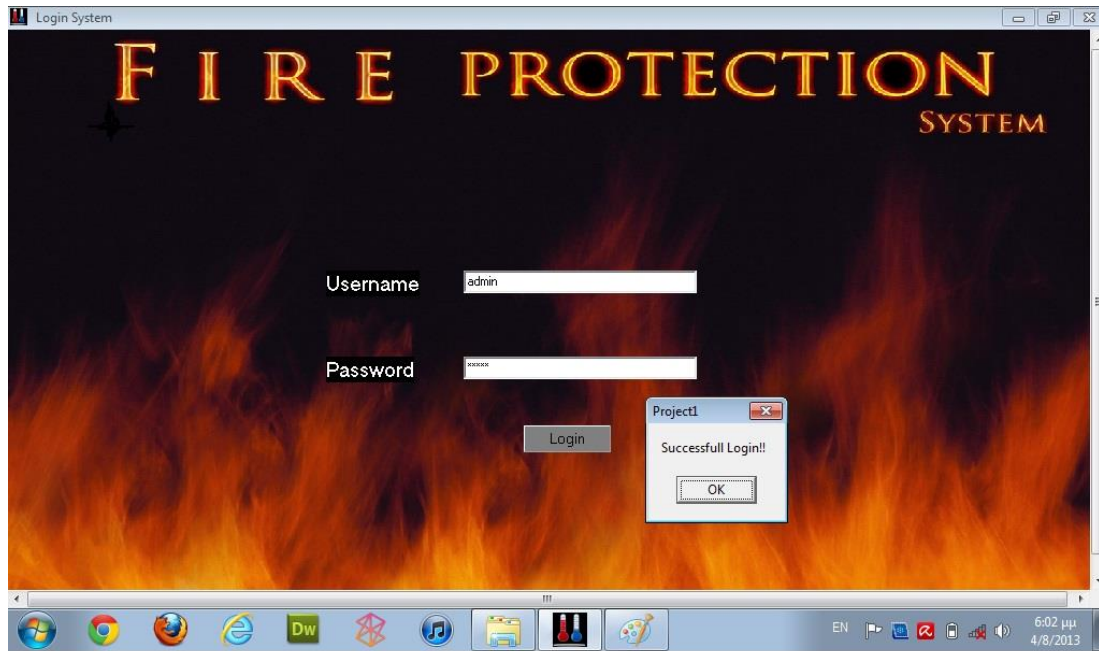
Εικόνα 5-24 Εικονίδιο Delphi εφαρμογής

Κάνοντας διπλό κλικ πάνω του, ανοίγει στον χρήστη η αρχική σελίδα της εφαρμογής που περιέχει το login system. Σε αυτήν, ο χρήστης πρέπει να συμπληρώσει το username και το password και να πατήσει το login. (στην προκειμένη περίπτωση το username είναι admin και το password είναι 12345).

Όπως είναι προφανές μας εμφανίζει η εφαρμογή το κατάλληλο μήνυμα, τόσο σε περίπτωση επιτυχημένου όσο και σε αυτή του αποτυχημένου login.



Εικόνα 5-25 Αρχική σελίδα της Delphi εφαρμογής μας

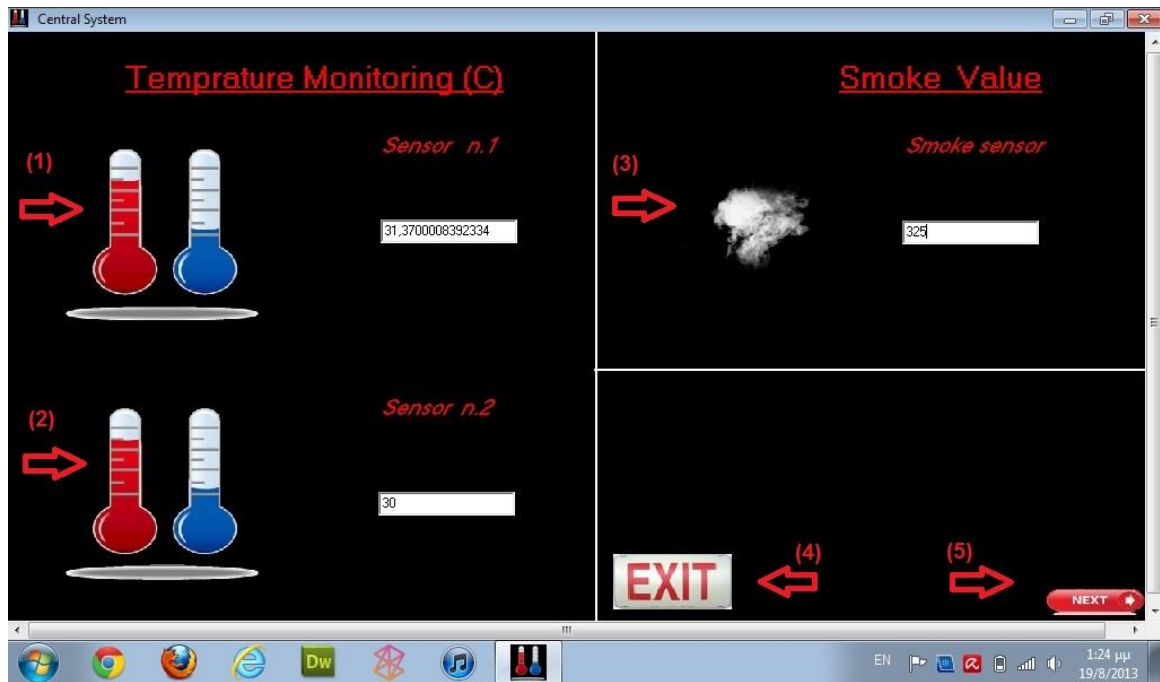


Εικόνα 5-26 Μήνυμα επιτυχημένης εισόδου



Εικόνα 5-27 Μήνυμα αποτυχημένης εισόδου

Μετά την επιτυχημένη καταχώρηση του username και password μπαίνουμε στην σελίδα του κεντρικού συστήματος όπου απεικονίζονται οι θερμοκρασίες που λαμβάνουμε από τους 2 αισθητήρες θερμοκρασίας καθώς επίσης και το νούμερο που απεικονίζει την μέτρηση καπνού από το σύστημα.



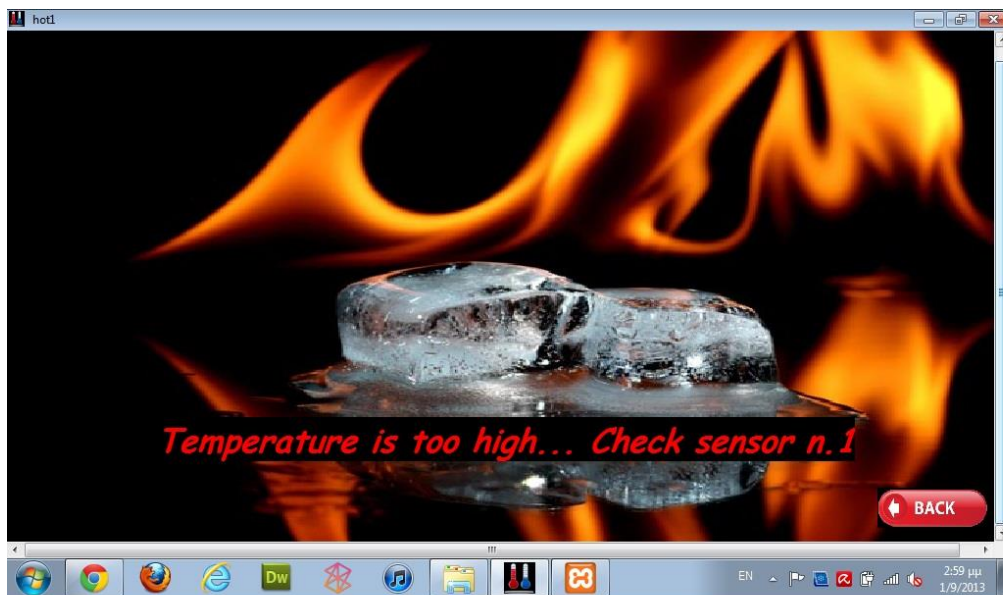
Εικόνα 5-28 Οθόνη Κεντρικού Συστήματος

Αναλυτικότερα, ακολουθεί το σχετικό υπόμνημα:

- (1) Θερμοκρασία (σε βαθμούς °C) που καταγράφει ο ενσύρματος, συνδεδεμένος απευθείας στο PC αισθητήρας.
- (2) Θερμοκρασία (σε βαθμούς °C) που καταγράφει ο ασύρματος - απομακρυσμένος αισθητήρας.
- (3) Αριθμοί που απεικονίζουν την ποσότητα καπνού που καταγράφεται.
- (4) Έξοδος από την εφαρμογή (Τερματισμός)
- (5) Είσοδος στην φόρμα επικοινωνίας.

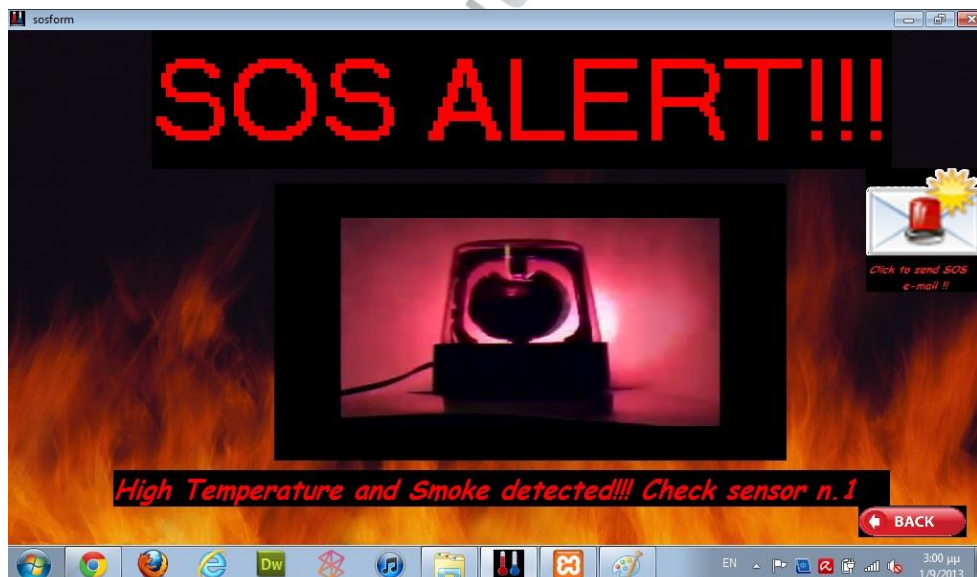
Προειδοποιητικά μηνύματα σε περιπτώσεις ακραίων καταγραφών από το σύστημα

Χαρακτηριστικό της εφαρμογής είναι η εμφάνιση μηνυμάτων στον χρήστη μέσω των οποίων τον προειδοποιεί για καταγραφεί θερμοκρασίας πάνω ή κάτω από τα επιτρεπτά όρια καθώς επίσης και για την ύπαρξη καπνού. Ενδεικτικά κάποια screenshots από τα παραπάνω μηνύματα:



Εικόνα 5-29 Προειδοποιητικό μήνυμα ανόδου θερμοκρασίας

Στην εικόνα βλέπουμε το σχετικό προειδοποιητικό μήνυμα που εμφανίζεται στην οθόνη του PC και στην συγκεκριμένη περίπτωση τονίζει την αυξημένη θερμοκρασία που κατέγραψε ο αισθητήρας n.1 (σαν αισθητήρα n.1 έχουμε ορίσει εξ' αρχής τον ενσύρματο)



Εικόνα 5-30 Προειδοποιητικό μήνυμα για άνοδο θερμοκρασίας και ύπαρξης καπνού

Στην εικόνα αυτή τονίζεται η σοβαρότητα του κινδύνου και η άμεση επέμβαση που χρειάζεται καθώς το σύστημα ειδοποιεί τον χρήστη της εφαρμογής ότι κατέγραψε αυξημένη θερμοκρασία σε συνδυασμό με την ύπαρξη καπνού. Στην εικόνα βλέπουμε ότι υπάρχει σχετικό banner που παραπέμπει τον χρήστη στην

φόρμα επικοινωνίας προκειμένου να αποστείλει άμεσα το κατάλληλο μήνυμα στις αρμόδιες αρχές ή στον admin.

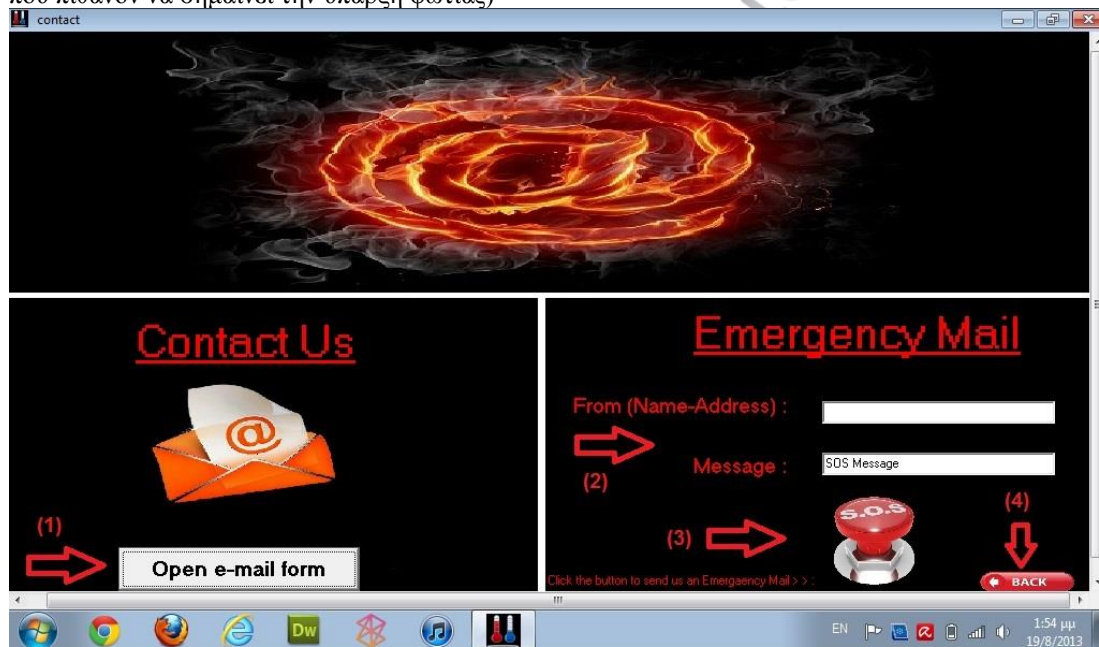
Φόρμα Επικοινωνίας

Όπως αναφέραμε παραπάνω, κάνοντας κλικ στο κουμπί Next βρισκόμαστε στην οθόνη του κεντρικού συστήματος, μπαίνουμε στην σελίδα επικοινωνίας της εφαρμογής. Σε αυτήν, μπορούμε να επικοινωνήσουμε με δύο τρόπους. Στο κομμάτι Contact Us, πατώντας στο κουμπί Open e-mail form η εφαρμογή συνδέεται με το Microsoft Outlook (ή όποιο άλλο παρόμοιο πρόγραμμα έχουμε προεπιλεγμένο) και μας ανοίγει την φόρμα σύνταξης και αποστολής e-mail.

Πέραν αυτού του τρόπου όμως, υπάρχει και η δυνατότητα αποστολής έκτακτου - επείγοντος e-mail, ένα είδος SOS e-mail, το οποίο αποστέλλεται απευθείας στον διαχειριστή της εφαρμογής.

Συγκεκριμένα, στην ενότητα Emergency Mail, ο χρήστης συμπληρώνει το ονοματεπώνυμό του και την διεύθυνσή του, καθώς και το μήνυμα που θέλει να στείλει στον διαχειριστή.

Εν συνεχεία, πατάμε το κουμπί κάτω από την φόρμα συμπλήρωσης (SOS button) και το mail αποστέλλεται στον λογαριασμό του server. Ο συγκεκριμένος τρόπος επικοινωνίας είναι χρήσιμος σε περίπτωση έκτακτης ανάγκης (π.χ. ανίχνευση καπνού σε συνδυασμό με απότομη αύξηση θερμοκρασίας που πιθανόν να σημαίνει την ύπαρξη φωτιάς)



Εικόνα 5-31 Φόρμα Επικοινωνίας της Delphi εφαρμογής

Υπόμνημα

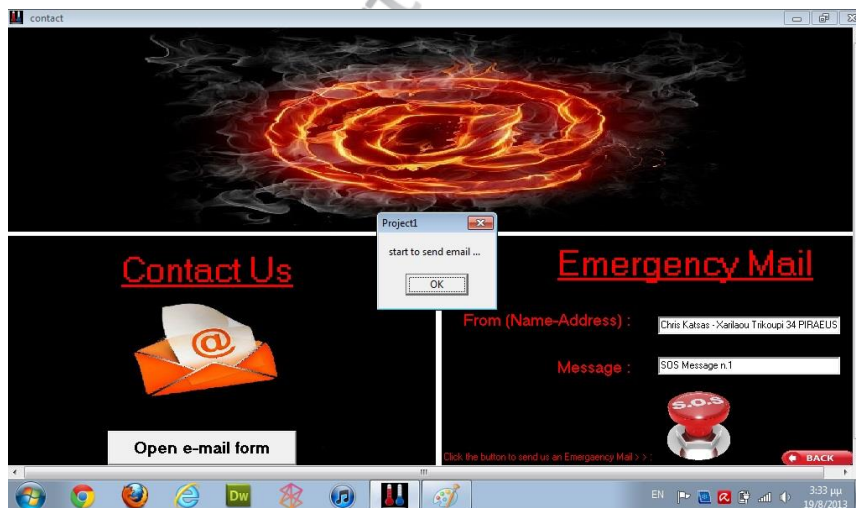
- (1) Κουμπί ενεργοποίησης φόρμας επικοινωνίας μέσω Microsoft Outlook (default πρόγραμμα)
- (2) Φόρμα επικοινωνίας για αποστολή SOS e-mail
- (3) Κουμπί αποστολής έκτακτου - emergency mail
- (4) Κουμπί επιστροφής σε προηγούμενη σελίδα (οθόνη κεντρικού συστήματος)

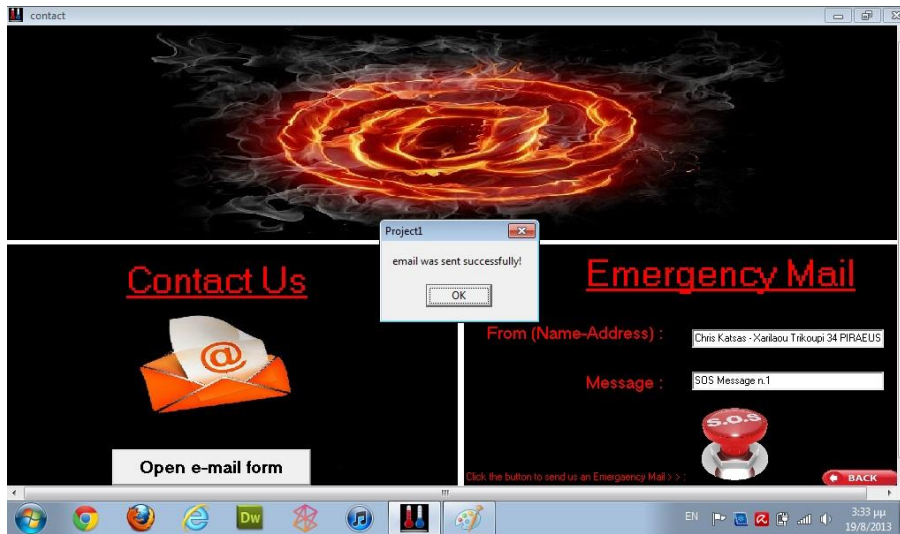
Ενδεικτική παρουσίαση αποστολής Emergency MailΒήμα 1ο

Συμπληρώνουμε την φόρμα επικοινωνίας με τα στοιχεία και το μήνυμα που θέλουμε να αποστείλουμε και στην συνέχεια κάνω κλικ στο κουμπί αποστολής mail.

**Εικόνα 5-32 Συμπλήρωση στοιχείων για SOS mail**Βήμα 2ο

Πατώντας το κουμπί, εμφανίζεται το μήνυμα ότι έχει ξεκινήσει η διαδικασία αποστολής, και πατώντας το OK το μήνυμα ότι το mail εστάλη με επιτυχία

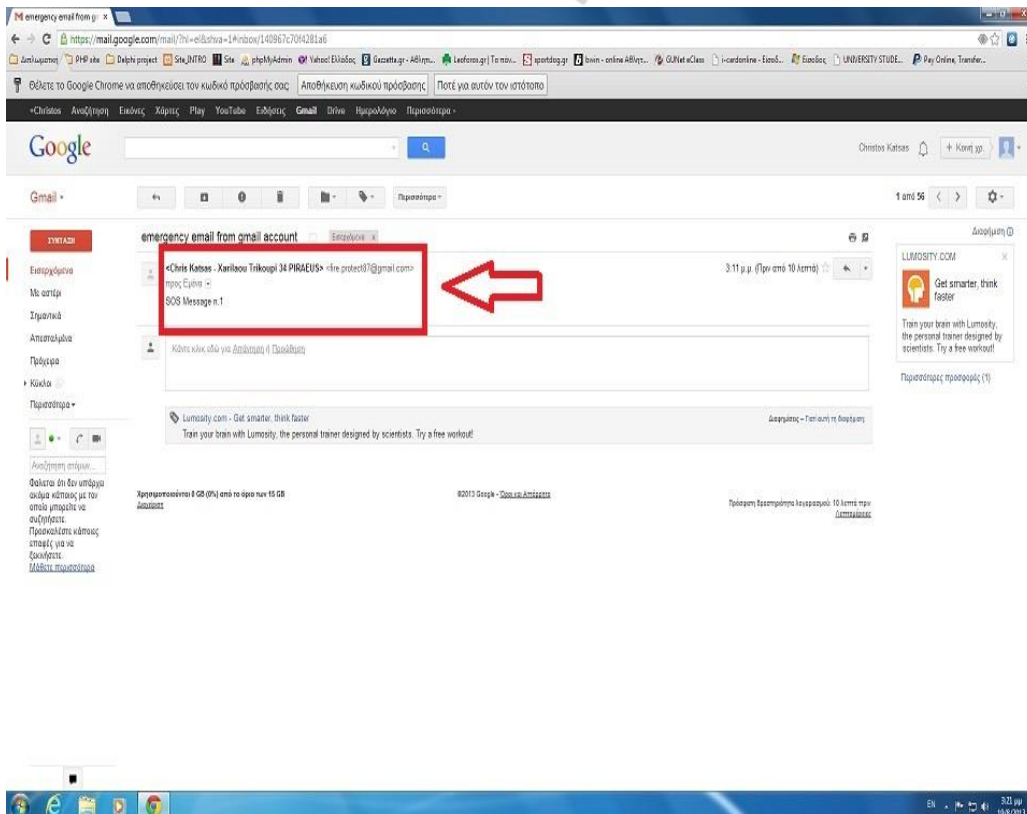
**Εικόνα 5-33 Διαδικασία αναμονής για αποστολή mail**



Εικόνα 5-34 Επιτυχημένη αποστολή SOS mail

Βήμα 3ο

Προς επιβεβαίωση, επισκεπτόμαστε το λογαριασμό mail που έχουμε δηλώσει ότι ανήκει στον διαχειριστή και βλέπουμε ότι το mail που στείλαμε μέσω της εφαρμογής υπάρχει στα εισερχόμενα.



Εικόνα 5-35 Επιβεβαίωση λήψης του Emergency mail

Κατασκευή συστήματος καταγραφής, μελέτης και παρουσίασης θερμοκρασιών και πυραίνχενυσης με χρήση αισθητήρων

ΚΕΦΑΛΑΙΟ 6

Αρχιτεκτονική Συστήματος

6.1 Αρχιτεκτονική προγράμματος για χρήση στην πλατφόρμα Arduino

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες, και προέρχεται από το IDE για την γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει τον προγραμματισμό σε χρήστες που μπορεί να μην είναι εξοικειωμένοι με την ανάπτυξη λογισμικού.

Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκυλών και επίσης είναι σε θέση να μεταγλωττίζει με μόνο ένα κλικ. Το πρόγραμμα που δημιουργήσαμε και γράφτηκε για arduino ονομάζεται σκίτσο (sketch).

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring μια παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc.

Λόγω της καταγωγής της από τη C, στη γλώσσα του Arduino, μπορούν να χρησιμοποιηθούν ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στη C.

Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για τη διαχείριση του ειδικού hardware του Arduino. Τα προγράμματα του Arduino διαιρούνται σε τρία μέρη: δομή (structure), τιμές (values) και συναρτήσεις (functions).

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

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

Εικόνα 6-1 Πίνακας εντολών γλώσσας Arduino

Δομή προγράμματος

Ένα τυπικό πρόγραμμα του Arduino έχει την εξής δομή:

```
// δηλώσεις μεταβλητών

void setup() {
    // αρχικοποιήσεις
}

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

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

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

Αναλύοντας το κομμάτι που αφορά την εφαρμογή, όπως έχει αναφερθεί σε προηγούμενο κεφάλαιο, για την υλοποίηση του project, χρησιμοποιήθηκαν 2 προγράμματα Arduino (σε γλώσσα προγραμματισμού Processing) τα οποία λειτουργούν ταυτόχρονα στο παρασκήνιο. Το πρώτο, είναι και το βασικό ,κύριο, πρόγραμμα που ονομάζεται tempsite.ino, "τρέχει" μέσω της πλατφόρμας arduino και είναι υπεύθυνο για την καταγραφή και αποθήκευση όλων των δεδομένων, όπως επίσης και για τον προγραμματισμό και λειτουργία όλου του συστήματος αισθητήρων δηλαδή του project. Το επόμενο, είναι το πρόγραμμα ledonoff.pde που "τρέχει" στην πλατφόρμα Processing και είναι υπεύθυνο για τον προγραμματισμό του LED ένδειξης λειτουργίας του συστήματος (LED που φανερώνει αν το σύστημα αισθητήρων είναι ενεργοποιημένο ή όχι). Ένα επιπλέον, βοηθητικό κυρίως, πρόγραμμα είναι το site_DhcpAddressPrinter που σκοπό έχει να εκτυπώσει στην σειριακή οθόνη της πλατφόρμας arduino την ip που έχει η πλακέτα Arduino Ethernet ώστε να στέλνουμε τα data στο internet.

Εξετάζοντας αναλυτικότερα τα προγράμματα:

A. tempsite.ino

```

#include <dht11.h>
#include <SPI.h>
#include <Wire.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <HTTPClient.h> // libraries that we are going to use
#include <Dhcp.h>
#include <Dns.h>
#include <Ethernet.h>
#include <EthernetClient.h>
#include <EthernetServer.h>
#include <EthernetUdp.h>
#include <util.h>

#define ONE_WIRE_BUS 2 //the onewire bus is connected to pin 2 on arduino
#define TEMPERATURE_PRECISION 10 //resolution of the sensors is set to 10bit

// Ethernet settings
uint8_t hwaddr[6] = {
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00}; // mac-adress of arduino
uint8_t ipaddr[4] = {
  192, 168, 1, x}; //IP-adress of arduino from DhcpAddressPrinter.ino program
uint8_t gwaddr[4] = {
  192, 168, 1, 1}; // IP-adress of gateway ( for later DNS implementation)
uint8_t subnet[4] = {
  255, 255, 255, 0}; // subnetmask ( for later DNS implementation)
uint8_t serverip[4] = {
  192, 168, 1, y}; // IP-adress of server arduino sends data to
uint8_t serverport = 80; // the port the arduino talks to

EthernetClient client; // make a new instance from type "Client" named "client", giving it
// serverip and serverport

dht11 DHT11;

OneWire oneWire(ONE_WIRE_BUS); // setup a oneWire instance to communicate with
//any OneWire devices

```

```

DallasTemperature sensors(&oneWire); // Pass our oneWire reference to Dallas Temperature
// variable to store the number of sensors
bool connected = false; // yes-no variable (boolean) to store if the arduino is
// connected to the server
int i,j,k = 0; // variable to count the sendings to the server

int incomingByte;

char buffer[50];
int var[10];
int pos=0,y=0,t=0;

////////////////// END OF DECLARATION AND INITIATION PART//////////////////
//////////////////////////////////////
////////////////// MAIN PART//////////////////

void setup() {

  DHT11.attach(5); // in which pin is DHT11 sensor connected to

  // initialize both serial ports:

  Serial.begin(9600); // Arduino Mega Serial
  Serial3.begin(9600); // Arduino Uno Serial

  pinMode(3, OUTPUT); // in which pin is our LED connected to
  Ethernet.begin(hwaddr, ipaddr); // start up ethernet
  sensors.begin();
}

//-----//

void loop() {

  if(Serial.available(>0){

    while(Serial.available(>0)
      incomingByte = Serial.read();

    if (incomingByte == 'H') {

```

```

digitalWrite(3, HIGH);} // LED is ON

if (incomingByte == 'L'){
digitalWrite(3,LOW); // LED is OFF
delay(20000);
}
}

// send data only when you receive data:
if (Serial3.available(>0)
{

////////////////////////////////// GET DATA FROM WIRELESS SENSOR //////////////////////////////////

// read the incoming byte:
char inByte = Serial3.read();
//Serial.write(inByte);

if(inByte==',' || inByte=='\n') // ends Byte receive if reaches comma or space
{
buffer[pos] = 0;

var[y] = atoi(buffer); // atoi = turns char into int
y++;
pos=0; // gets to the start all over

if(inByte=='\n')
{

////////////////////////////////// GET DATA FROM WIRED- CONNECTED TO PC SENSOR //////////////////////////////////

if (client.connect(serverip,serverport))
{
// if connected, set variable connected to "true"

connected = true;

sensors.requestTemperatures(); // send the request for temperature to sensors (all sensors)

int chk = DHT11.read(); // request to get data from DHT11 sensor

//delay(100);

```

```

float temp = sensors.getTempCByIndex(i); // take temperature reading from sensor "i" and
                                          store it to the variable "temp"
float tempF = (temp * 1.8 + 32); // converts C into F degree
float tempFar = tempF;
float humid = (DHT11.humidity); // stores humidity data from DHT11 sensor
float Humidity = humid;
int smoke = analogRead(A0); // stores into an int our smoke sensor's data

```

```

//////////////////// PRINTING DATA FROM WIRELESS SENSOR //////////////////////

```

```

y=0;

```

```

int humidx = (var[0]);
int tempx = (var[1]);
int tempFx = (tempx * 1.8 + 32);

```

```

////////////////////// COMMANDS FOR SAFETY REASONS. IF THERE IS A MALFUNCTION AT THE
TRASMISSION OF DATA BETWEEN SENSORS AND PC, THE SYSTEM USES THESE
COMMANDS//////////////////////////////////////

```

```

if (tempx >=49){

    //var[pos-1]=atoi(buffer);
    //var[pos-1]=var[1];
    tempx =29;
}

if (tempFx!= tempx * 1.8 + 32){
    tempFx= tempx * 1.81 + 32 ;
}

if (humidx > 90){
    var[pos-1] = atoi(buffer);
    var[0]=var[pos-1];
    humidx = (var[0]);
}

```

```

////////////////////////////////// END OF SAFETY COMMANDS ////////////////////////////////////

/// STORE ALL DATA INTO MY DATABASE//////////////////////////////////
/// SERIAL.PRINT= PRINT DATA TO SERIAL MONITOR///  

/// CLIENT.PRINT= PRINT DATA TO DATABASE //////  


client.print("GET /arduinodbcon.php?");
Serial.print("GET http://localhost/arduinodbcon.php?");

//////////////////////////////////

/// PRINT DATA FROM WIRED-CONNECTED TO PC SENSOR ////////////////////////////////////

client.print("t");
Serial.print("t");
client.print("=");
Serial.print("=");
client.print(temp); // temperature data
Serial.print(temp);

client.print("&&");
Serial.print("&&");

client.print("tf");
Serial.print("tf");
client.print("=");
Serial.print("=");
client.print(tempFar); // temperature data (fahrenheit)
Serial.print(tempFar);

client.print("&&");
Serial.print("&&");

client.print("h");
Serial.print("h");
client.print("=");
Serial.print("=");
client.print(Humidity); // humidity data
Serial.print(Humidity);

client.print("&&");
Serial.print("&&");

```

```
client.print("sx");  
Serial.print("sx");  
client.print("=");  
Serial.print("=");  
client.print(smoke); // smoke data  
Serial.println(smoke);
```

```
client.print("&&");  
Serial.print("&&");
```

```
////////////////// PRINT DATA FROM WIRELESS SENSOR ////////////////////
```

```
client.print("tx");  
Serial.print("tx");  
client.print("=");  
Serial.print("=");  
client.print(tempx); // temperature data  
Serial.print(tempx);
```

```
client.print("&&");  
Serial.print("&&");
```

```
client.print("tfx");  
Serial.print("tfx");  
client.print("=");  
Serial.print("=");  
client.print(tempFx); // temperature data (fahrenheit)  
Serial.print(tempFx);
```

```
client.print("&&");  
Serial.print("&&");
```

```
client.print("hx");  
Serial.print("hx");  
client.print("=");  
Serial.print("=");  
client.print(humidx); // humidity data  
Serial.println(humidx);
```

//////////////////////////////// INTERNET – HOST – PROTOCOL SETTINGS //////////////////////////////////

```

    client.println(" HTTP/1.1");
    Serial.println(" HTTP/1.1");
    client.println("Host: localhost");
    Serial.println("Host: localhost");
    client.println("User-Agent: Arduino");
    Serial.println("User-Agent: Arduino");
    client.println("Accept: text/html");
    Serial.println("Accept: text/html");
    client.println("Connection: close");
    Serial.println("Connection: close");
    client.println();
    Serial.println();

}
else
{
    Serial.println("Cannot connect to Server"); // else block if the server connection fails
(debugging)
}

Serial.println();
client.stop(); // stop the connection and set
connected = false; // "connected" to false
delay(3000); // EVERY 3 SECONDS WE GET AND SEND DATA!
}
}
else
{
    buffer[pos] = inByte;
    pos++;
}
}
}

```


B. ledonoff.pde + πρόγραμμα onoff_led.ino

Τα δύο αυτά προγράμματα είναι υπεύθυνα για τον προγραμματισμό και τον έλεγχο της ενεργοποίησης της συσκευής μέσω του site. Η διαδικασία έχει ως εξής:

Αρχικά γίνεται compile το πρόγραμμα ledonoff.pde που "τρέχει" στην πλατφόρμα processing.

```
import processing.serial.*;
Serial port;

void setup() {

  /* This part must be altered to fit your local settings. The number in brackets after "Serial.list()" is
  where you declare what COM port your Arduino is connected to.
  */
  port = new Serial(this, "COM7", 9600);
  // Open the port that the Arduino board is connected to, at 9600 baud

}
void draw() {

  String onoroff[] = loadStrings("http://localhost/tempsite/onoff_led/LEDstate.txt"); // Insert the location
of your .txt file in which the integer 1 or 0 is transferred to
  print(onoroff[0]); // Prints whatever is in the file ("1" or "0")

  if (onoroff[0].equals("1") == true) {
    println(" - TELLING ARDUINO TO TURN LED ON");
    port.write('H'); // Send "H" over serial to set LED to HIGH

  } else {

    println(" - TELLING ARDUINO TO TURN LED OFF");
    port.write('L'); // Send "L" over serial to set LED to LOW
  }
  delay(2000); // Set your desired interval here, in milliseconds
}
```

Όπως παρατηρούμε από τον κώδικα, ουσιαστικά στο buffer αποθηκεύονται ο αριθμός 1 ή 0 που συνδέονται μέσω led.php με το site και την ενότητα Settings. Όταν στο buffer υπάρχει ο αριθμός 1 τότε το πρόγραμμα στέλνει στο port του το byte 'H' και το LED ανάβει. Σε αντίθετη περίπτωση, (buffer = 0) το πρόγραμμα στέλνει στο port του το byte 'L' και το LED σβήνει. Η αποκωδικοποίηση του byte 'H' ή 'L' ώστε να μεταφράζεται στην ενεργοποίηση ή απενεργοποίηση του LED γίνεται μέσω του προγράμματος onoff_led.ino που λαμβάνει από το ledonoff.pde το byte 'H' ή 'L'.

Αναλυτικά ο κώδικας του προγράμματος

C. onoff_led.ino:

```
const int ledPin = 3; // the pin that the LED is attached to - change this if you have a separate LED
connected to another pin
int incomingByte; // a variable to read incoming serial data into

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // see if there's incoming serial data:
  if (Serial.available() > 0) {
    // read the oldest byte in the serial buffer:
    incomingByte = Serial.read();
    // if it's a capital H (ASCII 72), turn on the LED:
    if (incomingByte == 'H') {
      digitalWrite(ledPin, HIGH);
    }
    // if it's an L (ASCII 76) turn off the LED:
    if (incomingByte == 'L') {
      digitalWrite(ledPin, LOW);
    }
  }
}
```

D. site DhcpAddressPrinter

```
#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address for your controller below.
byte mac[] = {
  0x90, 0xA2, 0xDA, 0x0D, 0x1A, 0x88 };

byte ip[]={192,168,1,10}; //διευθυνση IPv4 από cmd
byte server[]={192,168,1,1}; // διεύθυνση server

// Initialize the Ethernet client library
// with the IP address and port of the server
// that you want to connect to (port 80 is default for HTTP):
EthernetClient client;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
```

Κατασκευή συστήματος καταγραφής, μελέτης και παρουσίασης θερμοκρασιών και πυρανίχνευσης με χρήση αισθητήρων

```

while (!Serial) {
  ; // wait for serial port to connect.
}

// start the Ethernet connection:
if (Ethernet.begin(mac) == 0) {
  Serial.println("Failed to configure Ethernet using DHCP");
  // no point in carrying on, so do nothing forevermore:
  for(;;)
    ;
}
// print your local IP address:
Serial.print("My IP address: ");
for (byte thisByte = 0; thisByte < 4; thisByte++) {
  // print the value of each byte of the IP address:
  Serial.print(Ethernet.localIP()[thisByte], DEC);
  Serial.print(".");
}
Serial.println();
}

void loop() {
}

```

Ο κώδικας που παρουσιάσαμε αφορά τα προγράμματα που "τρέχουν" για τον προγραμματισμό των πλακετών που είναι συνδεδεμένες στον ενσύρματο αισθητήρα.

Πέραν αυτού, υπάρχει και ο ασύρματος αισθητήρας όπου η επικοινωνία και η μεταφορά δεδομένων με το κεντρικό ενσύρματο σύστημα πλακετών γίνεται μέσω συσκευών που ονομάζονται Xbee. Το Xbee πρόκειται για module ασύρματης επικοινωνίας που αναπτύχθηκε από την εταιρία Digi και διαθέτει το δικό του πρωτόκολλο επικοινωνίας και λειτουργίας.

Είδη δικτύων:

Τα Xbee μπορούν να υποστηρίξουν είτε Zigbee είτε Digimesh δίκτυα:

- Zigbee

Τα Zigbee είναι ιεραρχικά δίκτυα και προϋποθέτουν τη ρύθμιση της κάθε μονάδας Xbee ως προς το ρόλο της στο δίκτυο. Σε ένα τέτοιο δίκτυο οι διακρίσεις μεταξύ κόμβων είναι τρεις: Coordinator, Router και End Device. Ο Coordinator μπορεί να είναι ένας μόνο, και έχει τις πιο μεγάλες δυνατότητες στο δίκτυο.

- Digimesh

Δίκτυο που ανέπτυξε και τα δικαιώματα του οποίου διαθέτει η Digi. Αυτή η περίπτωση δικτύου είναι οριζόντια, επιτρέπει δηλαδή την επικοινωνία μεταξύ όλων των κόμβων οι οποίοι είναι ομότιμοι.

- Μοντέλα και εκδόσεις

Τα Xbee έχουν κυκλοφορήσει σε εκδόσεις Series 1, Series 2, ZB, 900MHz, και Digimesh 2.4GHz. Αν και η εκπομπή στα 900MHz διαθέτει μεγαλύτερη ικανότητα στο να διαπερνά φυσικά εμπόδια, η χρήση της συχνότητας αυτής απαγορεύεται στην Ευρώπη. Η κάθε έκδοση διαχωρίζεται σε κανονικό και PRO μοντέλο, τα οποία έχουν κυρίως διαφορές εμβέλειας.

Τα Xbee υποστηρίζουν επίσης διάφορες επιλογές κεραίας.

- Xbee και Arduino

Το σύστημα Xbee μπορεί να συνδυαστεί με το Arduino για ασύρματη επικοινωνία. Χρειάζονται 2 τουλάχιστον μονάδες Xbee, αντίπατερ USB και Xbee Shield ή Explorer regulated για σύνδεση με την πλακέτα του Arduino ή με breadboard. Αρχικά οι μονάδες Xbee πρέπει να συνδεθούν μέσω USB με υπολογιστή για να ρυθμιστούν διάφορες επιλογές καθώς και το κανάλι επικοινωνίας τους. Αυτό γίνεται με το λογισμικό της Digi XCT-U.

- Σύνδεση στο Arduino

Το Xbee για να συνδεθεί με το Arduino απαιτεί 2 ψηφιακά πιν και τροφοδοσία 3.3V. Τα πιν, Receive και Transmit στο Xbee, συνδέονται με τα ψηφιακά πιν TX(transmit) και RX(receive) αντίστοιχα στο Arduino.

Ακολουθώς, η σειριακή επικοινωνία (Serial communication) παραλαμβάνεται από το Xbee. Οι πληροφορίες που αφορούν το Serial Port μεταβιβάζονται στο κανάλι που είναι ρυθμισμένα τα Xbee, ώστε να επικοινωνούν μεταξύ τους

Στη συνέχεια θα παρουσιάσουμε το πρόγραμμα που θα χρησιμοποιήσουμε για να συνδέσουμε αρχικά την ασύρματη πλακέτα με τον υπολογιστή, να ορίσουμε σε ποιο pin βρίσκεται το αισθητήριο θερμοκρασίας – υγρασίας και στη συνέχεια να δώσουμε εντολή να εκτυπώσει στην σειριακή του οθόνη τα δεδομένα που λαμβάνει από το αισθητήριο. Αυτά στην συνέχεια καλούνται ως byte ένα προς ένα από το κεντρικό μας πρόγραμμα tempsite.ino, μετατρέποντας με την κατάλληλη εντολή από σύνολο χαρακτήρων, ως ακέραιους αριθμούς και αποθηκεύονται στην βάση:

***Σημείωση:** Η εντολή που δίνουμε για να εμφανιστεί το οτιδήποτε, στη σειριακή οθόνη της κεντρικής ενσύρματης πλακέτας, είναι Serial.print() ενώ η εντολή με την οποία καλούμε τα data που εμφανίζονται αρχικά στη σειριακή οθόνη της απομακρυσμένης πλακέτας, και τα θέλουμε να εμφανιστούν στη σειριακή οθόνη του προγράμματος της κεντρικής ενσύρματης πλακέτας είναι Serial3.print()

Ο κώδικας με τον οποίο αρχικά ρυθμίζουμε τις παραμέτρους της ασύρματης πλακέτας είναι ο εξής:

```
#include <dht11.h> // library that we'll use

dht11 DHT11;
void setup()
{
  DHT11.attach(13); // pin on wireless shield where our sensor is attached
  Serial.begin(9600);
}

void loop()
{
  int chk = DHT11.read(); // reads and stores data from sensor
  int sensorValue;

  Serial.print(DHT11.humidity);
  Serial.print(",");
  Serial.print(DHT11.temperature);
  Serial.print(",");

  sensorValue = analogRead(A0); // read analog input pin 0

  Serial.print(sensorValue, DEC); // prints the value read

  Serial.print("\n");
  delay(1000);
}
```

6.2 Μεταφορά δεδομένων και αποθήκευση στη βάση

Σε προηγούμενη ενότητα μελετήσαμε το interface της πλατφόρμας διαχείρισης της βάσης δεδομένων και αναφέραμε το κομμάτι εντολών στο πρόγραμμά μας το οποίο είναι αρμόδιο για να αποστέλλει τα data που λαμβάνουμε από τους αισθητήρες.

Ο συγκεκριμένος κώδικας στο πρόγραμμα tempsite.ino είναι ο εξής:

```
client.print ("GET /arduinoDbcon.php?");
Serial.print("GET http://localhost/arduinoDbcon.php?");
```

Ουσιαστικά, σε αυτό το τμήμα του κώδικα, το πρόγραμμα καλεί το αρχείο arduinoDbcon.php και στέλνει σε αυτό όλα τα data που έχει λάβει. Στην συνέχεια, το παραπάνω αρχείο είναι υπεύθυνο για την σύνδεσή μας με την βάση δεδομένων σε πρώτο στάδιο, και την ταξινόμηση των δεδομένων στα αντίστοιχα κελιά που έχουμε δημιουργήσει στην βάση. Παράλληλα εμφανίζει κατάλληλα μηνύματα λάθους σε περίπτωση λάθους βάσης, ή μη επίτευξης σύνδεσης με την βάση δεδομένων που έχουμε ορίσει.

Αναλυτικά ο κώδικας του αρχείου arduinoDbcon.php

```
<?php
#header ('Refresh : 5');
```

Κατασκευή συστήματος καταγραφής, μελέτης και παρουσίασης θερμοκρασιών και πυρανόχενωσης με χρήση αισθητήρων

```
header('Content-type: text/plain');
```

```
echo date("d.m.Y-H:i:s") . " Temperatur= " . $_GET['t'];
```

```
$link = mysql_connect("localhost", "root", "") or die("Can't connect to database: " . mysql_error());  
//connect to the MySQL-Server
```

```
mysql_select_db("arduino_db") or die("Wrong database name"); //select the database
```

```
$query = "INSERT INTO data(tempIN,tempFarIN,HumidityIN,tempEX,tempFarEX,HumidityEX,smoke)  
VALUES (" . $_GET['t'] . "," . $_GET['f'] . "," . $_GET['h'] . "," . $_GET['tx'] . "," . $_GET['fx'] .  
"," . $_GET['hx'] . "," . $_GET['sx'] . ")";  
//insert the value from GET request of arduino
```

```
$result = mysql_query($query) or die("ERROR: " . mysql_error());  
//give the browser feedback (if it worked: send the SQL-Query, if it didn't send "ERROR" plus the SQL  
error
```

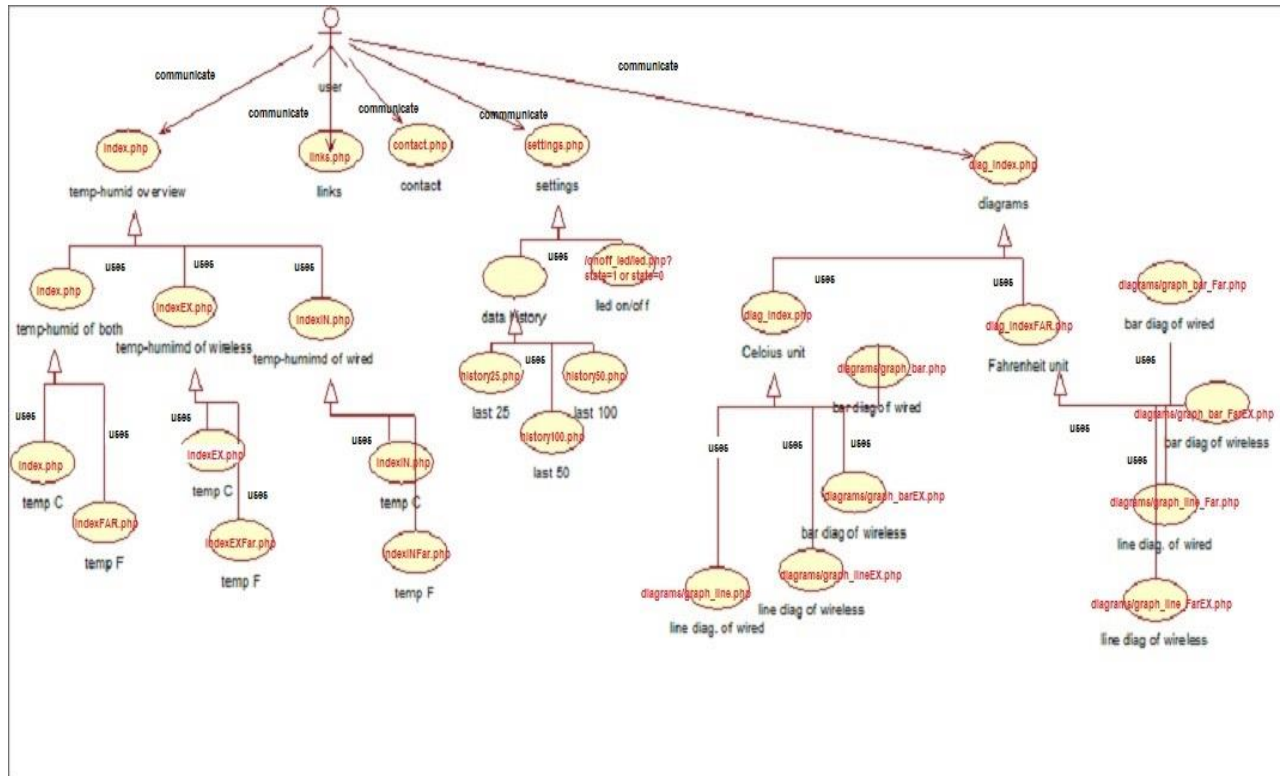
```
mysql_close($link); //close link to MySQL-Server  
?>
```

6.3 Διαδικασία, χαρακτηριστικά και τρόπους κατασκευής του website

Το website απεικόνισης, ανάλυσης και παρουσίασης των θερμοκρασιών, της υγρασίας και γενικότερα των δεδομένων μας, είναι γραμμένο σε γλώσσα PHP σε συνδυασμό με scripts εντός του κώδικα τα οποία είναι σε javascript. Χρησιμοποιήσαμε και κώδικα CSS για την δημιουργία και μορφοποίηση του στυλ των ενοτήτων της ιστοσελίδας.

Η αρχική σελίδα είναι η index.php. Στη συνέχεια περιηγούμαστε στο site και τις λειτουργίες του (παρουσίαση του interface έγινε σε προηγούμενο κεφάλαιο).

Στο ακόλουθο διάγραμμα Use Case σε UML παρουσιάζεται ποια διαφορετική σελίδα από αυτές που έχουμε δημιουργήσει και βρίσκονται στον φάκελο κατασκευής του website ανοίγει σε κάθε μία διαφορετική περίπτωση χρήσης.



Εικόνα 6-2 Διάγραμμα Use Case σε UML για την λειτουργία του website

Για παράδειγμα, όταν βλέπουμε ως χρήστες στην οθόνη του pc την θερμοκρασία σε βαθμούς Fahrenheit, από data που αποστέλλει μόνο ο ασύρματος αισθητήρας, τότε η σελίδα που τρέχει στο παρασκήνιο είναι η `indexEXFar.php`.

Χρήση αρχείων και Libraries από Internet

- Πρέπει να σημειώσουμε ότι για την κατασκευή των διαγραμμάτων που απεικονίζουν τα data που λαμβάνουμε και βρίσκονται στην ενότητα `diagrams` του website, χρησιμοποιήθηκε η βιβλιοθήκη `JpGraphPHP library Pro Version 3.5.x`.

Η βιβλιοθήκη αυτή περιλαμβάνει ουσιαστικά τα διάφορα μοντέλα διαγραμμάτων και δίνει στον δημιουργό της ιστοσελίδας, την δυνατότητα να συνδέσει το διάγραμμα με την βάση δεδομένων και να απεικονίζει με ποικιλία διαγραμμάτων, τις διάφορες μεταβολές.

Μπορούμε επίσης να ρυθμίσουμε και επιπλέον παραμέτρους, όπως ορισμένα `Legend` για επεξήγηση του ποια γραμμή ή μπάρα στο εκάστοτε διάγραμμα αντιστοιχεί σε ποιο data και τέλος δίνεται η δυνατότητα προσθήκης εικόνων ή διαφορετικών χρωμάτων σε γραμμές ή μπάρες γραφημάτων.

- Στην ενότητα Settings, η δυνατότητα on/off που μας δίνει το website, συνδέεται με τον κώδικα του προγράμματος ledonoff.rde που αναλύσαμε στην προηγούμενη ενότητα
- Για την απεικόνιση των θερμοκρασιών, υγρασίας και γενικότερα των αριθμών στην αρχική σελίδα (digital style) χρησιμοποιήθηκε αρχείο που περιείχε έτοιμο το digital style και αυτό που χρειάστηκε, ήταν να προσαρμόσουμε τις ρυθμίσεις αυτού του αρχείου στο αρχείο CSS του πίνακα που εμφανίζεται στην αρχική (Home) σελίδα.
- Τέλος, η λειτουργία του αναλογικού ρολογιού στην πάνω αριστερά πλευρά του site βασίζεται στην εξής λειτουργία: Περιστρέφουμε τις τρεις εικόνες από τις οποίες αποτελείται το ρολόι ανάλογα με την πραγματική ώρα tt/mm/ss.
Χωρίζουμε την ώρα αυτή σε ώρες, λεπτά και δευτερόλεπτα. Στην συνέχεια, με τις κατάλληλες μαθηματικές πράξεις, μετατρέπουμε κάθε μία από αυτές τις αριθμητικές μεταβλητές σε γωνίες μεταξύ 0 έως 360 μοίρες όπου κάθε μία από τις 3 μεταβλητές – γωνίες θα αντιστοιχεί στην εικόνα του δείκτη της ώρας, των λεπτών και των δευτερολέπτων. Μετά, γίνεται χρήση ενός timer function σε JQuery για να επαναλαμβάνεται αυτή η διαδικασία κάθε sec. Αυτό θα κάνει τους pointers να φαίνονται ότι κινούνται όπως τα πραγματικά ρολόγια

ΚΕΦΑΛΑΙΟ 7

Κατασκευή Delphi εφαρμογής

Όπως αναφέρθηκε και στο κεφάλαιο user's manual, πέραν του website, υπάρχει και η Delphi εφαρμογή, μία desktop εφαρμογή που σκοπό έχει, πέραν της παρουσίασης θερμοκρασίας θερμοκρασιών και ποσότητας καπνού που ανιχνεύει ο αισθητήρας, να ενημερώνει με τα κατάλληλα μηνύματα τον χρήστη για το που υπάρχει το πρόβλημα (δηλαδή σε ποιον αισθητήρα). Συγκεκριμένα ενημερώνει, εάν καταγράφηκε πολύ υψηλή ή χαμηλή θερμοκρασία, αν υπάρχει μεγάλη ποσότητα καπνού και φυσικά παρέχει την δυνατότητα επικοινωνίας με τον admin της εφαρμογής.

Η εφαρμογή είναι εξολοκλήρου κατασκευασμένη σε Delphi. Αναλυτικότερα παρουσιάζουμε τα units της εφαρμογής:

Unit1

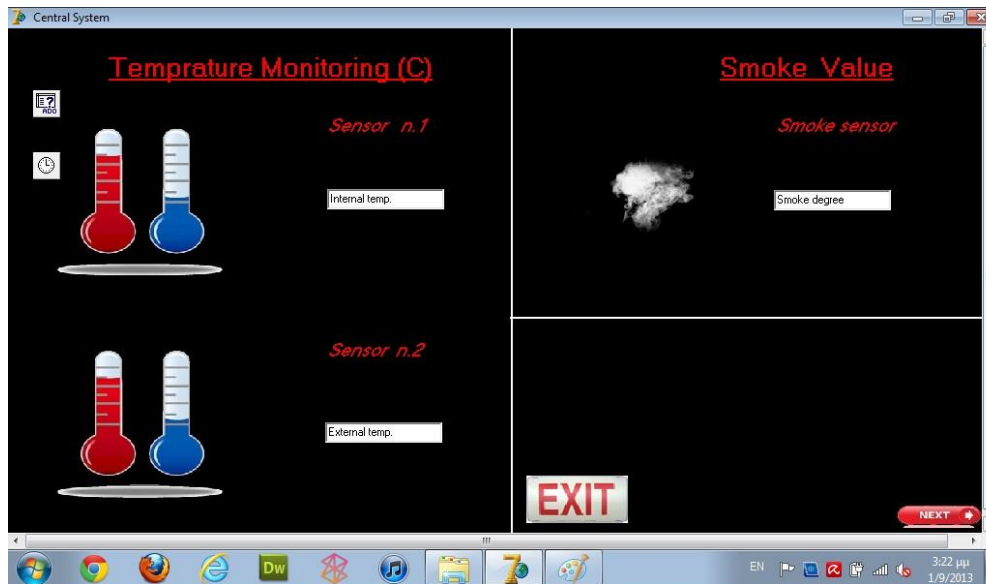


Εικόνα 7-1 Το unit1 κατά την κατασκευή της Delphi εφαρμογής

Κατασκευή συστήματος καταγραφής, μελέτης και παρουσίασης θερμοκρασιών και πυρανίχνευσης με χρήση αισθητήρων

Το unit1 είναι η πρώτη σελίδα της εφαρμογής, αυτή κατά την οποία ο χρήστης καλείται να συμπληρώσει το username και το password για την είσοδο. Στον κώδικα του unit1, πέραν της εικόνας για φόντο και των labels και panels για την συμπλήρωση των στοιχείων, έχει προστεθεί ένας timer με interval 1 second προκειμένου να ανάβει και να σβήνει το κουμπί Login κάτω από την φόρμα. Επίσης, υπάρχει η εντολή showMessage(' '); για την εμφάνιση του κατάλληλου μηνύματος σε περίπτωση επιτυχημένης ή αποτυχημένης εισόδου στο σύστημα.

Unit2



Εικόνα 7-2 Το unit2 κατά την κατασκευή της Delphi εφαρμογής

Στο unit2 μπαίνουμε στην κεντρική σελίδα της εφαρμογής όπου παρουσιάζονται τα data που λαμβάνουμε από τους αισθητήρες. Το τμήμα αυτό της Delphi εφαρμογής είναι και το πλέον σημαντικό διότι συνδέει την βάση δεδομένων μας με την εφαρμογή. Το unit αυτό περιλαμβάνει την εικόνα που χρησιμοποιούμε ως background τα Labels και Edit Fields στα οποία εμφανίζονται τα data, ένα button που ονομάζεται Next και στέλνει τον χρήστη στην επόμενη σελίδα της εφαρμογής, και το κουμπί Exit που τερματίζει την εφαρμογή με την εντολή application.terminate

Το πιο σημαντικό μέρος αυτού του unit είναι η σύνδεσή μας με την database όπως αυτή παρουσιάζεται βήμα προς βήμα παρακάτω:

- (α) Στην Form που έχουμε, προσθέτουμε από την καρτέλα ADO το compose ADOQuery
- (β) Στην καρτέλα Object Inspector του ADOQuery που βάλαμε, εντοπίζουμε την ιδιότητα ConnectionString και επιλέγουμε το μικρό κουτί με τις 3 τελείες μέσα
- (γ) Μέσω του Radio Button επιλέγουμε Use Connection String και πατάμε το κουμπί Build
- (δ) Από την λίστα που εμφανίζεται, επιλέγουμε Microsoft OLE DB Provider for ODBC Drivers και κάνουμε κλικ στο κουμπί Επόμενο

(ε) Μέσω ξανά του Radio Button επιλέγουμε Χρήση Συμβολοσειράς σύνδεσης και κάνουμε κλικ στο κουμπί Δόμηση βρίσκοντας και επιλέγοντας στην συνέχεια, την βάση δεδομένων στο σημείο που την έχουμε αποθηκεύσει στον υπολογιστή. Το username και όποιο πιθανό password έχουμε ορίσει για την είσοδο στην database, το συμπληρώνουμε στο πλαίσιο που αναγράφει Όνομα Χρήστη και Κωδικός Πρόσβασης. Κάνοντας κλικ στο κουμπί 'Δοκιμή Σύνδεσης' εμφανίζεται το κατάλληλο μήνυμα σε περίπτωση επιτυχημένης σύνδεσης ανάμεσα στην βάση μας και στην Delphi εφαρμογή.

Όσον αφορά το Connection String του project, αν ακολουθήσουμε τα παραπάνω βήματα, στην τελική του μορφή είναι το εξής:

```
Provider=MSDASQL.1;Persist Security Info=False;User ID=root;Extended  
Properties="Driver=MySQL ODBC 5.1 Driver;  
SERVER=127.0.0.1;UID=root;DATABASE=arduino_db;PORT=3306"
```

Παρατηρούμε ότι το Connection String περιλαμβάνει μεταξύ άλλων, τις επιλογές που κάναμε στα προηγούμενα βήματα και συγκεκριμένα:

```
UserID = root;  
Extended Properties = "Driver = MySQL ODBC 5.1 Driver;  
SERVER = 127.0.0.1;  
UID = root;  
DATABASE = arduino_db; (Είναι η database μας)  
PORT = 3306; (Η default port)
```

Τέλος, σε αυτό το unit έχει προστεθεί και timer με interval 10 seconds που ανανεώνει το unit ώστε να λαμβάνει τα νέα data από την βάση δεδομένων. Κάθε 10 sec. ανανεώνονται τα δεδομένα που έλαβε ώστε να συμβαδίζει με τα τελευταία δεδομένα που μπήκαν στην βάση μας. Επίσης στον κώδικα του unit υπάρχουν όλες οι εντολές της μορφής if...then που ορίζουν τα όρια πάνω και κάτω από τα οποία η εφαρμογή μας εμφανίζει το προειδοποιητικό μήνυμα ότι κάτι δεν λειτουργεί σωστά.

Unit3



Εικόνα 7-3 Το unit3 κατά την κατασκευή της Delphi εφαρμογής

Το unit3 έχει να κάνει με την φόρμα επικοινωνίας μεταξύ του χρήστη και του administrator είτε μέσω της φόρμας επικοινωνίας και του Outlook είτε μέσω του Emergency Mail απευθείας στο Inbox που έχει οριστεί από τον server. Όσον αφορά την εφαρμογή, το unit αυτό, πέραν της εικόνας που ορίζεται ως background περιέχει και Edit Boxes και Labels για την συμπλήρωση των στοιχείων και του μηνύματος σε περίπτωση του Emergency Mail. Επίσης, περιέχει το κουμπί Button1 υπεύθυνο για την αποστολή Email μέσω του Outlook , το κουμπί Back που επιστρέφει τον χρ'στη στην προηγούμενη οθόνη (κεντρικό σύστημα) καθώς επίσης και την εικόνα Image1 πατώντας πάνω στην οποία αποστέλλονται απευθείας σε inbox του λογαριασμού του server τα στοιχεία που αναγράψαμε στα Edit Boxes.

Unit4 ,Unit5 ,Unit6 ,Unit7 ,Unit8



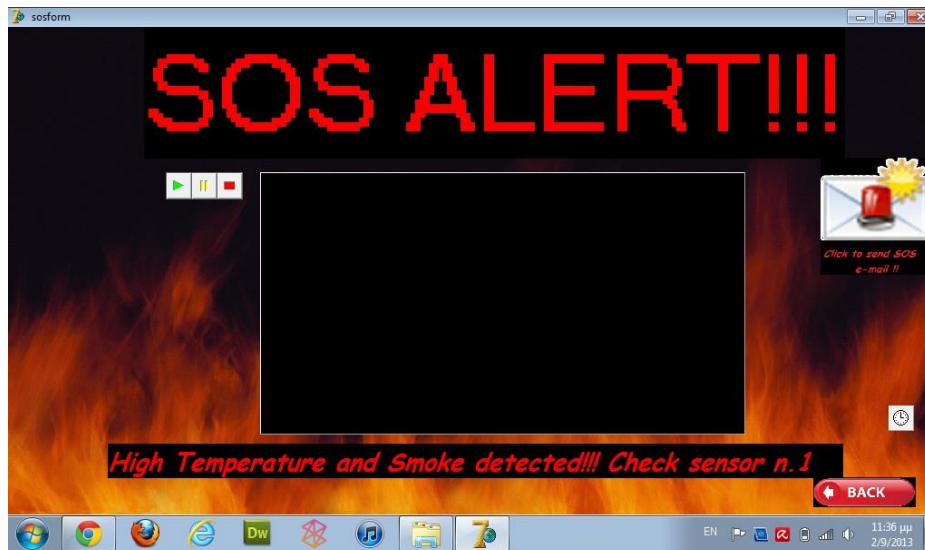
Εικόνα 7-4 Τα units που αντιστοιχούν στα προειδοποιητικά μηνύματα κατά την διαδικασία κατασκευής της Delphi εφαρμογής

Τα παραπάνω Units είναι τα μηνύματα που εμφανίζονται στην οθόνη του user σε περίπτωση που οι αισθητήρες καταγράφουν θερμοκρασίες ανώτερες ή κατώτερες από τα πάνω και κάτω όρια θερμοκρασίας και καπνού που έχουν οριστεί στο Unit2. Έχουν όλα παρόμοια διάταξη: Περιέχουν image σχετική με το περιεχόμενο του μηνύματος, ως φόντο, Label που αναγράφει το απαραίτητο προειδοποιητικό μήνυμα, έναν timer με interval 1 second που σκοπό έχει να συνδέεται με το Label, ώστε το τελευταίο να δίνει την εικόνα ότι αναβοσβήνει, και τέλος όλα έχουν το κουμπί Back το οποίο μας επιστρέφει στην οθόνη του κεντρικού συστήματος.

Unit9 (unit9.jpg)

Το unit αυτό αφορά το μήνυμα που εμφανίζεται στην οθόνη του χρήστη στην περίπτωση που το σύστημα αισθητήρων ανιχνεύσει την ύπαρξη καπνού σε συνδυασμό με την αύξηση θερμοκρασίας.

Στο unit έχουμε χρησιμοποιήσει 2 labels, το heatlabel που εμφανίζει το μήνυμα για την αύξηση της θερμοκρασίας και ύπαρξη καπνού και το sos label που σε συνδυασμό με τον timer που υπάρχει, εμφανίζει ένα μήνυμα SOS να αναβοσβήνει στην οθόνη. Επιπλέον, υπάρχει image με τον τίτλο mail που χρησιμοποιείται ως σύνδεσμος για την καρτέλα επικοινωνίας της εφαρμογής (unit3), το κουμπί back που μας επαναφέρει στην οθόνη του κεντρικού συστήματος (unit2) και τέλος το panel και το componse του Media Player που παρουσιάζει στον χρήστη το βίντεο που τονίζει την σοβαρότητα και τον πιθανό κίνδυνο



Εικόνα 7-5 Το unit9 κατά την κατασκευή της Delphi εφαρμογής

ΚΕΦΑΛΑΙΟ 8

Συμπεράσματα

Το παρόν σύστημα ελέγχου θερμοκρασίας και πυρανίχνευσης είναι ένα σύστημα ελέγχου για οικιακή και όχι μόνο χρήση καθώς μπορεί να επεκταθεί και να χρησιμοποιηθεί σε ευρύτερη κλίμακα (θερμοκήπια, άλση, κήπους κλπ).

Συνδυάζει την παρουσίαση και ανάλυση θερμοκρασιών τόσο αριθμητικά, όσο και διαγραμματικά, πράγμα χρήσιμο για μακροπρόθεσμη μελέτη μεταβολών των δεδομένων για στατιστικές μελέτες και αρχείο.

Επιπλέον η πρόσβαση στο website είναι πού εύκολη και άνετη από διαφορετικού τύπου συσκευές (pc, tablets, κινητά) ώστε να μπορεί να υπάρχει άμεση και live εικόνα του χώρου όπου έχει τοποθετηθεί το σύστημα αισθητήρων μας απ' όπου και αν βρίσκεται ο χρήστης.

Τέλος, με την desktop εφαρμογή, ο χρήστης έχει την δυνατότητα, πέραν της live παρακολούθησης των δεδομένων θερμοκρασίας και ποσότητας καπνού που ανιχνεύει ο αισθητήρας, να λάβει άμεσα στην οθόνη του υπολογιστή του το κατάλληλο προειδοποιητικό μήνυμα όταν καταγράψει κάτι πάνω από τα επιτρεπτά όρια. Το πλέον σημαντικό όμως, είναι η δυνατότητα άμεσης αποστολής Emergency e-mail σε περίπτωση κινδύνου.

Στην συγκεκριμένη εφαρμογή, το mail συνδέεται με το inbox του administrator, ωστόσο, με την κατάλληλη ρύθμιση και σε περίπτωση επέκτασης του project, το επείγον email μπορεί να αποστέλλεται στις αρμόδιες αρχές για την άμεση και γρήγορη επίλυση του όποιου πιθανού προβλήματος.

ΚΕΦΑΛΑΙΟ 9

Μελλοντικές Επεκτάσεις

Το project με την μορφή που υπάρχει αυτή την στιγμή μπορεί να επεκταθεί και να προσφέρει υπηρεσίες σε μεγαλύτερη κλίμακα και με πολλούς διαφορετικούς τρόπους.

Ενδεικτικά, πιθανές επεκτάσεις θα μπορούσαν να είναι:

- i) Δυνατότητα επικοινωνίας με τον admin ή γενικότερα με τον αρμόδιο αποδέκτη ενός επείγοντος μηνύματος σε περίπτωση έκτακτης ανάγκης, μέσω SMS που θα φτάνει στο κινητό του παραλήπτη. Η επέκταση αυτή της εφαρμογής μπορεί να γίνει με την προσθήκη μίας GSM Shield στο σύστημα αισθητήρων που έχει κατασκευαστεί, σε συνδυασμό με μία προπληρωμένη κάρτα κινητής τηλεφωνίας. Στην συνέχεια, με τον κατάλληλο προγραμματισμό της GSM πλακέτας μέσω της πλατφόρμας arduino, ορίζουμε έναν ή παραπάνω αποδέκτες και η διαδικασία επικοινωνίας λειτουργεί όπως στην Delphi εφαρμογή, στο κομμάτι του Emergency email.
- ii) Μπορούμε να αυξήσουμε τον αριθμό και τον τύπο των αισθητήρων ανάλογα με την χρήση και το σημείο που τοποθετούμε την συσκευή και θέλουμε να καταγράψουμε και να συλλέξουμε τα δεδομένα (π.χ. αισθητήρας ατμοσφαιρικής πίεσης ή κίνησης, ή φωτιάς). Γενικότερα, μπορούμε να τροποποιήσουμε τον τύπο των δεδομένων που λαμβάνουμε, ωστόσο η κεντρική λειτουργία του project παραμένει η ίδια.
- iii) Σενάρια σε συνδυασμό με πρακτική εφαρμογή του τομέα της τεχνητής νοημοσύνης. Για παράδειγμα, επέκταση του υπάρχοντος project θα μπορούσε να είναι η αδρανοποίηση συσκευών που ελέγχονται μέσω pc και η λειτουργία ειδικών σημάνσεων που να δείχνουν προς την έξοδο κινδύνου, σε περίπτωση έκτακτων συνθηκών και ανίχνευσης καπνού.
- iv) Χρήση 3D απεικόνισης (π.χ. σε γλώσσα VRML) για τρισδιάστατη απεικόνιση του χώρου όπου έχουμε τοποθετήσει τον αισθητήρα έτσι ώστε να δίνει στον χρήστη την δυνατότητα καλύτερου ελέγχου και βέλτιστης εικόνας του χώρου που ανιχνεύει - εξετάζει ο αισθητήρας.
- v) Εφαρμογή για κινητά - tablets για άμεση και εικόνα σε πραγματικό χρόνο, των μεταβολών και τροποποιήσεων στα δεδομένα που εξετάζονται στον χώρο που ενδιαφέρει τον χρήστη.

ΚΕΦΑΛΑΙΟ 10

Βιβλιογραφία

- [1] Arduino Workshop, A Hands-On Introduction with 65 Projects by John Boxall, May 2013
- [2] Getting Started with Arduino By Massimo Banzi ,Publisher: O'Reilly Media,Released: February 2009
- [3] Arduino Programming notebook, Brian W. Evans
- [4] Arduino in a Nutshell by Prof. Jan Borchers, Version 1.8 (Aug 5, 2013)
- [5] Exploring Arduino, Tools and Techniques for engineering wizardry by Jeremy Blum
- [6] Beginning Arduino [Paperback] by Michael McRoberts

Δικτυογραφία

- [1] Arduino – HomePage : <http://www.arduino.cc/>
- [2] Processing.org : <http://www.processing.org/>
- [3] Free tutorials on HTML, CSS and PHP - Build your own website - HTML.net : <http://html.net/>
- [4] iPhone-like button in Photoshop Tutorial | Yesterdayishere : <http://yesterdayishere.com/now/iphone-like-button-in-photoshop/>
- [5] W3Schools Online Web Tutorials : <http://www.w3schools.com/>
- [6] Arduino Tutorials – tronixstuff : <http://tronixstuff.com/tutorials/>
- [7] Arduino Tutorial - Learn electronics and microcontrollers using Arduino! : <http://www.ladyada.net/learn/arduino/index.html>
- [8] Xbee Adapter - wireless Arduino programming : <http://www.ladyada.net/make/xbee/arduino.html>
- [9] Datalogging with Arduino | code, circuits, & construction : <http://www.tigoe.com/pcomp/code/arduinowiring/1043/>
- [10] Save data of temperature and humidity on MySQL with Arduino Uno and Wifly : <http://www.instructables.com/id/Save-data-of-temperature-and-humidity-on-MySQL-wit/>
- [11] Arduino + Temperature + Humidity : <http://www.instructables.com/id/Arduino-Temperature-Humidity/?ALLSTEPS>
- [12] Temperature and Humidity Datalogger + Webserver : <http://www.instructables.com/id/Temperature-and-Humidity-Datalogger-Webserver/>
- [13] Arduino and Java | Silveira Neto : <http://silveiraneto.net/2009/03/01/arduino-and-java/>
- [14] XBEE Tutorial : <https://sites.google.com/site/xbeetutorial/>
- [15] 20 Best Useful Digital Fonts for LED Banner Designing – DJDESIGNERLAB : <http://djdesignerlab.com/2010/11/14/20-best-useful-digital-fonts-for-led-banner-designing/>
- [16] Delphi to send email using Gmail SMTP server | Blog Kuliah : <http://vkuliah.wordpress.com/2010/04/09/delphi-to-send-email-using-gmail-smtp-server/>
- [17] Send Email using Gmail in Delphi : <http://www.emailarchitect.net/easendmail/kb/delphi.aspx?cat=2>