



ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΑΛΑΜΑΤΑΣ
= ΠΑΡΑΡΤΗΜΑ ΣΠΑΡΤΗΣ =
ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Μάριος Σαμαράκης

**Σύλλογή, αποθήκευση, επεξεργασία και απεικόνιση
μετρήσεων ηλιοφάνειας σε πραγματικό χρόνο.**



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων καθηγητής: *Γιάννης Λιαπέρδος*, Καθηγητής Εφαρμογών.

ΣΠΑΡΤΗ 2012

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

| | |
|---|----|
| ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ..... | 3 |
| ΠΡΟΛΟΓΟΣ..... | 5 |
| 1. ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ..... | 7 |
| 1.1 Εισαγωγή..... | 9 |
| 1.1.1 Φωτοκότταρο..... | 10 |
| 1.1.2 Arduino Duemilanove..... | 11 |
| 1.1.2.1 Το υλικό (hardware) του Arduino..... | 14 |
| 1.1.2.2 Ακροδέκτες του μικροελεγκτή Arduino..... | 14 |
| 1.1.2.3 Χαρακτηριστικά του Arduino..... | 15 |
| 1.1.3 Arduino Ethernet Shield..... | 17 |
| 1.1.4 Breadboard (ή αλλιώς Ράστερ)..... | 19 |
| 1.2.1 Ηλεκτρονικός Υπολογιστής (H/Y) ή Laptop..... | 19 |
| 1.2.2 Εξωτερικό Modem / Router..... | 20 |
| 1.2.3 Καλωδίωση και Συνδεσμολογία..... | 20 |
| 1.2.4 Internet..... | 20 |
| 2. ΠΡΑΚΤΙΚΟ ΜΕΡΟΣ..... | 21 |
| 2.1 Μεθοδολογία σχεδίασης..... | 23 |
| 2.2 Προδιαγραφές..... | 24 |
| 2.3 XAMPP (X: cross, Apache HTTP Server, MySQL, PHP, Perl)..... | 30 |
| 2.4 Ρύθμιση Modem/Router με στατική IP..... | 33 |

| | |
|--|-----|
| 2.5 MySQL Workbench..... | 35 |
| 2.5.1 Παράμετροι του MySQL Workbench..... | 38 |
| 2.5.2 Δημιουργία βάσεων δεδομένων..... | 40 |
| 2.6 Κώδικας PHP..... | 51 |
| 2.6.1 Κώδικας για το Arduino..... | 55 |
| 2.6.2 Google Maps και Open Flash Chart..... | 59 |
| 3. ΣΥΜΠΕΡΑΣΜΑΤΑ..... | 65 |
| 4. ΠΑΡΑΡΤΗΜΑΤΑ..... | 69 |
| Ολοκληρωμένος κώδικας βάσης δεδομένων <code>arduino_light</code> | 70 |
| Κώδικας του αρχείου <code>test.html</code> | 84 |
| ΟΡΟΛΟΓΙΕΣ..... | 103 |
| ΠΗΓΕΣ..... | 105 |
| ΒΙΒΛΙΟΓΡΑΦΙΑ..... | 106 |
| ΕΥΡΕΤΗΡΙΟ..... | 108 |
| ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ – ΕΙΚΟΝΩΝ..... | 111 |

ΠΡΟΛΟΓΟΣ

Σκοπός της παρούσας εργασίας υπήρξε η ανάπτυξη υλικού (hardware) με το οποίο θα λαμβάνονται δεδομένα από φωτοβολταϊκό αισθητήρα τα οποία θα αποστέλλονται σε Server (μέσω Ethernet) για αποθήκευση και περαιτέρω επεξεργασία. Τα αποτελέσματα της επεξεργασίας θα απεικονίζονται σε έναν κοινό Web Browser. Θα αναπαρασταθεί δηλαδή σε πραγματικό χρόνο όλη η διαδικασία που απαιτείται έτσι ώστε να μπορεί να καταγραφεί η ηλιοφάνεια σε διάφορες περιοχές της Ελλάδας ανά τακτά χρονικά διαστήματα.

Το κείμενο αποτελείται από δύο βασικά μέρη: Το Θεωρητικό και το Πρακτικό. Το πρώτο κρίθηκε απαραίτητο (χωρίς να επεκταθεί σε βάρος του δεύτερου) προκειμένου να δώσει στο μη εξοικειωμένο αναγνώστη τις κύριες θεωρητικές γνώσεις ώστε να γίνει κατανοητή η ανάπτυξη λογισμικού (software), η χρησιμοποίηση αυτού, η σωστή επιλογή των συσκευών του συστήματος μας, καθώς και η αναπαράσταση του τρόπου δημιουργίας του τοπικού μας δικτύου και των επιμέρους διαδικασιών ώστε να έρθει εις πέρας το επιθυμητό αποτέλεσμα.

Στο πρακτικό μέρος παρουσιάζονται εκτενώς οι προδιαγραφές του κυκλώματος και οι ανάγκες οι οποίες τις επέβαλαν, έτσι ώστε να μπορούν όλα τα εργαλεία να λειτουργούν αλληλένδετα και να είναι σωστά εναρμονισμένα, καθώς και η μεθοδολογία σχεδίασης και κατασκευής, τα προβλήματα που ανέκυψαν και οι τρόποι αντιμετώπισης τους.

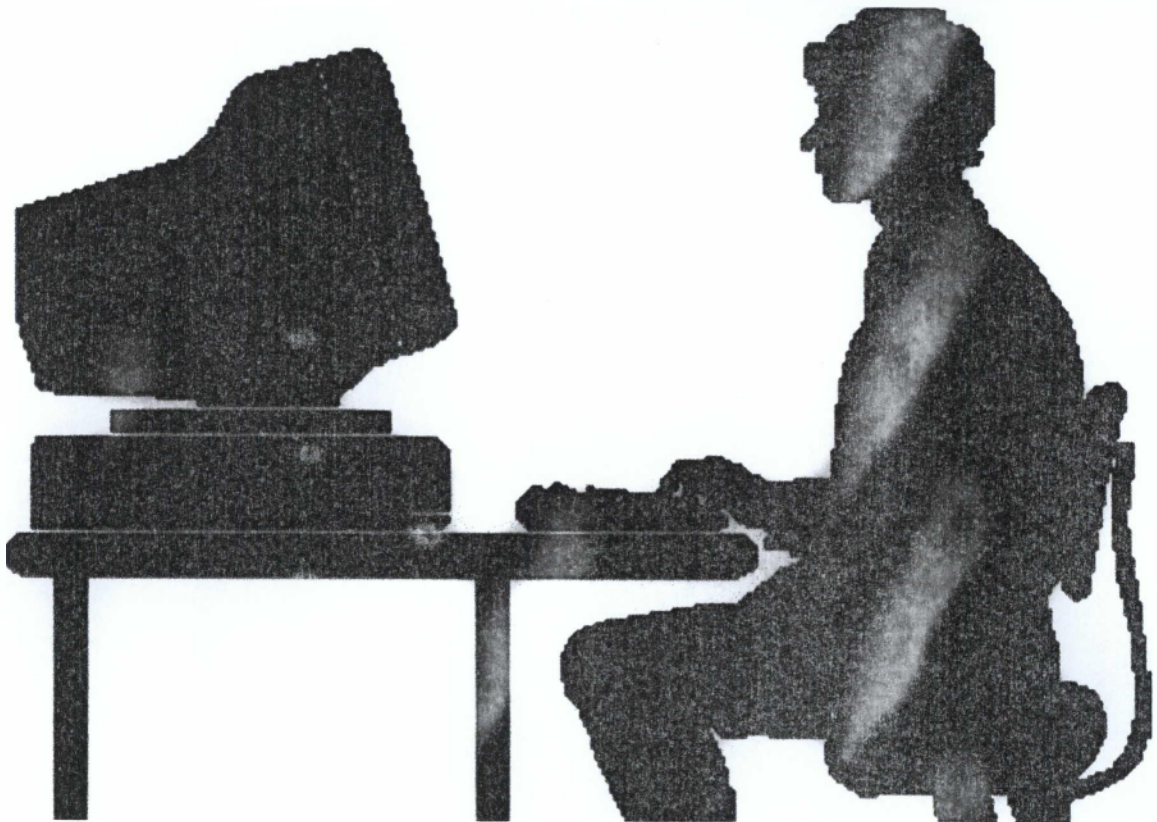
Των δύο βασικών μερών προτάχθηκε μια σύντομη Εισαγωγή, κύριος σκοπός της οποίας είναι η περιγραφή της ιδέας του θέματος της πτυχιακής ώστε να γίνει πιο εύκολη η κατανόηση του όλου κυκλώματος και συστήματος που χρησιμοποιήθηκε.

Θα ήθελα και από τη θέση αυτή να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Γιάννη Λιαπέρδο για την καθοδήγηση του και την πολύτιμη βοήθειά του κατά τη διάρκεια της εκπόνησης της εργασίας μου. Επίσης, θα ήθελα να αφιερώσω την εργασία αυτή στην Μητέρα μου, την οποία «έχασα» πριν δύο χρόνια. Της αξίζει...

Σπάρτη, Μάρτης του 2012

Μάριος Σαμαράκης

1. ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ



1.1 ΕΙΣΑΓΩΓΗ

Η ιδέα του θέματος αυτής της πτυχιακής έχει να κάνει με το γεγονός ότι, αρχικά θα πρέπει να μπορεί να μπει ένα πλάνο στο μυαλό που θα απεικονίζει τον τρόπο αναπαράστασης του όλου κυκλώματος, έτσι ώστε να καταφέρει να φτάσει στο τελικό στόχο, που είναι η απεικόνιση των αποτελεσμάτων των μετρήσεων σε μια ιστοσελίδα.

Για να μπορέσει να γίνει κάτι τέτοιο θα χρειαστεί σίγουρα να σκεφτεί κανείς πολλές παραμέτρους που πιθανότατα θα βοηθήσουν ή αντίστοιχα θα σταθούν εμπόδιο στην συνέχιση του έργου αυτού. Το γενικότερο πλάνο έχει ως εξής: 1) ένα απλό φωτοκύτταρο το οποίο έχει την ικανότητα να δέχεται τιμές ανάλογες της ηλιοφάνειας που «διαβάζει» η επιφάνεια του, μια χρονική στιγμή ή πχ. κάθε 1 λεπτό, 2) το Arduino Duemilanove και το Arduino Ethernet Shield τα οποία είναι δυο συσκευές που επικοινωνούν άμεσα μεταξύ τους και η μια «κουμπώνει» πάνω στην άλλη (το Ethernet Shield πάνω στο Duemilanove), 3) ένα breadboard στο οποίο ενώνονται με απλά καλώδια το φωτοκύτταρο και το Arduino, 4) Η/Υ (στην προκειμένη περίπτωση είναι ένα laptop για λόγους μεταφοράς) ο οποίος είναι φορτωμένος με το αντίστοιχο λογισμικό για να μπορούν να συντονιστούν όλα τα δεδομένα και τα στοιχεία μεταξύ τους ώστε να βγει το τελικό αποτέλεσμα, 5) εξωτερικό Modem / Router του οποίου είναι απαραίτητη η χρήση λόγω του ότι, οι δοκιμές θα γίνουν σε ένα απλό τοπικό δίκτυο, 6) απαραίτητη καλωδίωση όπως επίσης και τον τρόπο συνδεσμολογίας όλων των παραπάνω μεταξύ τους, και τέλος, 7) είναι απαραίτητη η πρόσβαση στο Internet ώστε, στο τελικό στάδιο να εμφανίζει τις οποιεσδήποτε τιμές καταγράφηκαν σε μια ιστοσελίδα.

Πέρα από τα προαναφερθέντα, ήταν απαραίτητη η αναζήτηση των σωστών – συμβατών προγραμμάτων καθώς και γλωσσών προγραμματισμού ώστε να μπορεί να προχωρήσει παρακάτω και να συντεθεί η εργασία αυτή. Τα προγράμματα - γλώσσες που χρησιμοποιούνται (ορισμένα περισσότερο, άλλα λιγότερο) είναι τα εξής: XAMPP (X: cross, Apache HTTP Server, MySQL, PHP, Perl), Arduino 1.0, MySQL Workbench, Mozilla Firefox & Thunderbird, Internet Explorer, Winzip, Command Prompt, Json, XML, Open flash chart, Google Maps, κ.ά.

Σαν πρώτη φάση λοιπόν, έχοντας συγκεντρώσει όλα τα στοιχεία και το πλάνο, έγινε το ξεκίνημα, συνδέθηκαν όλα σωστά και έγινε έλεγχος για το εάν επικοινωνούν όλα μεταξύ τους. Δεν ήταν απαραίτητο να είναι κάθε φορά συνδεδεμένο το φωτοκύτταρο γιατί λαμβάνονταν τυχαίες τιμές μέσω ενός κώδικα που χρησιμοποιήθηκε στο χρονικό περιθώριο που δόθηκε (δηλαδή κάθε 1 λεπτό της ώρας). Έτσι είχε την δυνατότητα να ελεγχθεί αν σε πρώτο στάδιο ήταν όλα εντάξει και «μιλούσαν» μεταξύ τους ώστε να μπορέσουμε να προχωρήσουμε βήμα – βήμα.

Παρουσιάστηκαν όπως είναι προφανές διάφορα προβλήματα κατά την πάροδο των ημερών τα οποία μετέπειτα λύθηκαν, κατόπιν ψαξίματος, βοήθειας, υπομονής και επιμονής. Αφού κατάφερα να υπερπηδήσω ορισμένα από αυτά, έφτασα στο σημείο όπου οι τυχαίες τιμές καταγράφονταν κανονικά στο Arduino και από εκεί περνούσαν στην βάση δεδομένων. Το επόμενο στάδιο ήταν εξίσου δύσκολο, έπρεπε να «σταλούν» αυτές οι τυχαίες τιμές που είχαν καταγραφεί πλέον στο site. Όλα αυτά γίνανε με τη χρήση της γλώσσας MySQL, όπως και με τις συχνές παραλλαγές και δοκιμές στον κώδικα που χρησιμοποιεί το Arduino, καθώς και με την σωστή σύνθεσή και των δυο ώστε να βγουν τα αποτελέσματα που χρειαζόντουσαν κάθε φορά.

Τέλος, «στήθηκαν» τα επιμέρους προγράμματα ώστε να μπορέσει να εμφανιστεί ένας χάρτης που θα φαινόταν ξεκάθαρα (όπως φαίνονται στο Google Earth) διάφορες περιοχές, στις οποίες βρέθηκα και κατέγραψα σε διάφορες χρονικές στιγμές της ημέρας την ηλιοφάνεια και παρατήρησα τον τρόπο μεταβολής της, μέσα από ένα γράφημα.

1.1.1 Φωτοκύτταρο

Η ονομασία φωτοκύτταρο σχετίζεται ετυμολογικά με την λέξη κύτταρο. Είναι σύνθετη λέξη και προέρχεται από τις δυο λέξεις φως και κύτταρο. Είναι λοιπόν μια συσκευή της οποίας η λειτουργία στηρίζεται στο φωτοηλεκτρικό φαινόμενο και χρησιμοποιείται για φωτομετρήσεις, στον ομιλούντα κινηματογράφο, στην τηλεόραση, στην τηλεφωτογραφία, στους φωτοπολλαπλασιαστές και σε διάφορες διατάξεις αυτοματισμού. Τα φωτοκύτταρα διακρίνονται σε φωτοκύτταρα κενού και σε φωτοκύτταρα αερίου.

Τα φωτοκύτταρα κενού αποτελούνται από αερόκενο υάλινο σωλήνα. Όταν στην κάθοδο (φωτοκάθοδος) επιπέσει φως, τότε, αν μεταξύ των ηλεκτροδίων εφαρμοστεί μια ελάχιστη διαφορά δυναμικού, το κύκλωμα διαρρέεται από ρεύμα (φωτοηλεκτρικό φαινόμενο). Αν η φωτεινή ροή που προσπίπτει στο φωτοκύτταρο διατηρείται σταθερή, ενώ η τάση V μεταξύ ανόδου - καθόδου αυξάνει συνεχώς, τότε η ένταση του ρεύματος που διαρρέει το κύκλωμα, όπως δείχνει το αμπερόμετρο, διαρκώς αυξάνει.

Όσον αφορά τα φωτοκύτταρα αερίου, υπάρχει μέσα στο σωλήνα αέριο αργό (Ar) με πίεση 2 Torr. Αν η τάση αυξάνει απεριόριστα, είναι δυνατό να γίνει αυτοτελής εκκένωση, η οποία καταστρέφει το φωτοκύτταρο για το λόγο αυτόν τοποθετείται σε σειρά στο κύκλωμα μια αντίσταση ασφαλείας.

Στην εργασία χρησιμοποιήθηκε ένα μικρό, απλό, στρογγυλό φωτοκύτταρο με 2 καλώδια τα οποία έρχονται και συνδέονται στο breadboard.

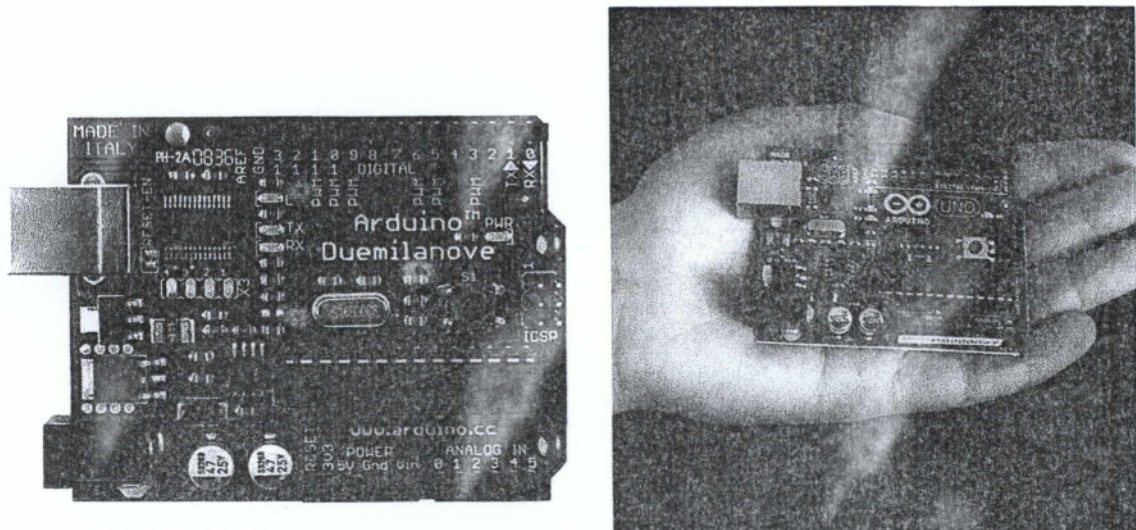


Σχήμα 1.1.1: Φωτοκύτταρο.

1.1.2 Arduino Duemilanove

Το Arduino Duemilanove είναι ένα εργαλείο για να κατασκευαστεί ένα υπολογιστικό σύστημα με την έννοια ότι αυτό θα ελέγχει συσκευές του φυσικού κόσμου, σε αντίθεση με τον κοινό 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 προγραμματιστεί μέσα από ένα απλό περιβάλλον ανάπτυξης. Ένα Arduino μπορεί να χρησιμοποιηθεί για να αναπτύξει διαδραστικά

αντικείμενα, να δεχτεί εισόδους από πληθώρα αισθητηρίων οργάνων και διακόπτες, αλλά και να ελέγχει 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 συναρμολογημένες. Το περιβάλλον ανάπτυξης του λογισμικού βασίζεται στην γλώσσα προγραμματισμού Processing και την γλώσσα προγραμματισμού Wiring, οι οποίες είναι ανοιχτού κώδικα (open source) και μπορούν να «κατεβούν» δωρεάν από το διαδίκτυο. Η γλώσσα προγραμματισμού του Arduino αποτελεί μια εφαρμογή σε software επίπεδο της καλωδίωσης. Εξομοιώνει θα λέγαμε απόλυτα το φυσικό περιβάλλον του μικροελεγκτή.



Σχήμα 1.1.2: Μικροελεγκτής Arduino Duemilanove

Υπάρχει πληθώρα άλλων μικροελεγκτών και αναπτυξιακών πλακετών 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 προαναφέρθηκαν 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 εκδοχή ενός Arduino. Ωστόσο ακόμα και αν προμηθευτεί την έτοιμη (μονταρισμένη πλακέτα) αυτή θα κοστίζει το μέγιστο 50 Euro.

Τρέχει σε διάφορα Λειτουργικά Συστήματα. Οι μηχανικοί λογισμικού, ανέπτυξαν το περιβάλλον προγραμματισμού του Arduino για Windows, Macintosh 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.

1.1.2.1 Το υλικό (hardware) του Arduino

Ο μικροελεγκτής Arduino είναι στην ουσία μια αναπτυξιακή πλακέτα που ενσωματώνει έναν ATMEL ATMEGA168. Υπάρχουν πολλές εκδόσεις του μικροελεγκτή. 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 9 Volt, είτε απευθείας από την USB θύρα του Υπολογιστή.

1.1.2.2 Ακροδέκτες του μικροελεγκτή Arduino

Ο Arduino έχει 14 ψηφιακούς ακροδέκτες Εισόδου/Εξόδου. Αυτοί μπορούν να τεθούν ως είσοδοι ή ως εξοδοι 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 20-50 KΩ. 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, και 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) και 5 (SCL). Υποστηρίζει το πρωτόκολλο I2C (TWI) χρησιμοποιώντας βιβλιοθήκες τις Γλώσσας προγραμματισμού Wiring.

Υπάρχουν και κάποιοι άλλοι ακροδέκτες:

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

Reset. Αν τεθεί σε κατάσταση LOW τότε επαννεκινεί τον Μικροελεγκτή. Σε αυτή τη γραμμή τοποθετείται ένας διακόπτης.

Επιπλέον υπάρχουν και άλλοι ακροδέκτες με ειδικές λειτουργίες:

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

1.1.2.3 Χαρακτηριστικά του Arduino

| | |
|-----------------------------|--|
| Μικροελεγκτής | ATmega168 |
| Τάση Λειτουργίας | 5V |
| Τάση Εισόδου | 7 - 12V |
| Όρια Τάσης | 6 - 20V |
| Ψηφιακοί Ακροδέκτες I/O | 14 (από τα οποία 6 παρέχουν PWM έξοδο) |
| 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 διαβαστούν 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 9 Volt από 220 V. 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 5 Volt 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 ενδεχομένως 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

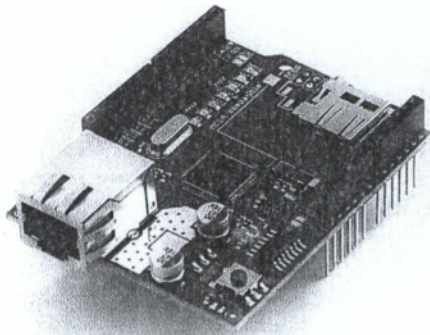
Ο Arduino Duemilανone έχει την δυνατότητα να επικοινωνεί με τον Ηλεκτρονικό Υπολογιστή, έναν άλλον Arduino ή άλλους μικροελεγκτές. Το ολοκληρωμένο ATMega 168 παρέχει σειριακή επικοινωνία TTL 5Volt UART, η οποία είναι διαθέσιμη από τους ακροδέκτες (λήψη RX) 0 και (εκπομπή TX) 1 του ολοκληρωμένου. Επιπλέον 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.

1.1.3 Arduino Ethernet Shield

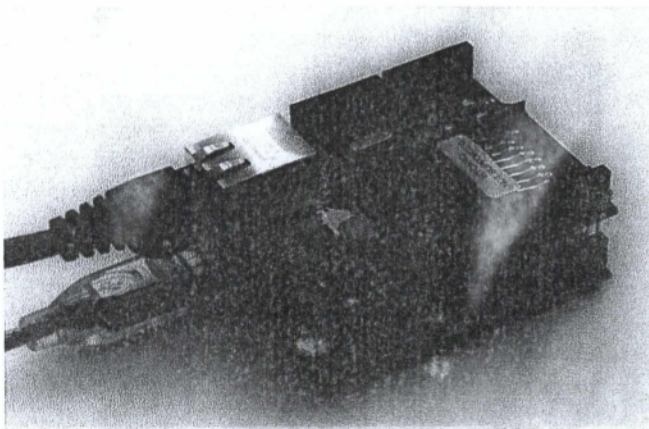
Το Arduino Ethernet Shield επιτρέπει στην πλακέτα Arduino να συνδεθεί στο internet χρησιμοποιώντας την Ethernet βιβλιοθήκη για να διαβάσει και να γράψει σε κάρτα SD μέσω της SD βιβλιοθήκης.

Για να χρησιμοποιηθεί η ασπίδα, τοποθετείται πάνω από μία πλακέτα Arduino (π.χ. το Duemilανone). Για να ανεβούν σχέδια στην πλακέτα, συνδέεται στον υπολογιστή με ένα καλώδιο USB κανονικά. Μόλις το σχέδιο έχει αποσταλεί, μπορεί να αποσυνδεθεί ο πίνακας από τον υπολογιστή και να του δοθεί ρεύμα με ένα εξωτερικό τροφοδοτικό.

Για την σύνδεση του Arduino Ethernet Shield στον υπολογιστή ή σε ένα διανομέα δικτύου ή δρομολογητή χρησιμοποιείται ένα τυπικό καλώδιο Ethernet (CAT5 ή CAT6 με RJ45 υποδοχές). Για την σύνδεση σε υπολογιστή μπορεί να απαιτείται η χρήση ενός cross - over καλωδίου (αν και πολλοί υπολογιστές, συμπεριλαμβανομένων όλων των προσφάτων υπολογιστών Mac μπορούν να κάνουν το cross - over στο εσωτερικό).



Σχήμα 1.1.3: Arduino Ethernet Shield.



Ρυθμίσεις Δικτύου

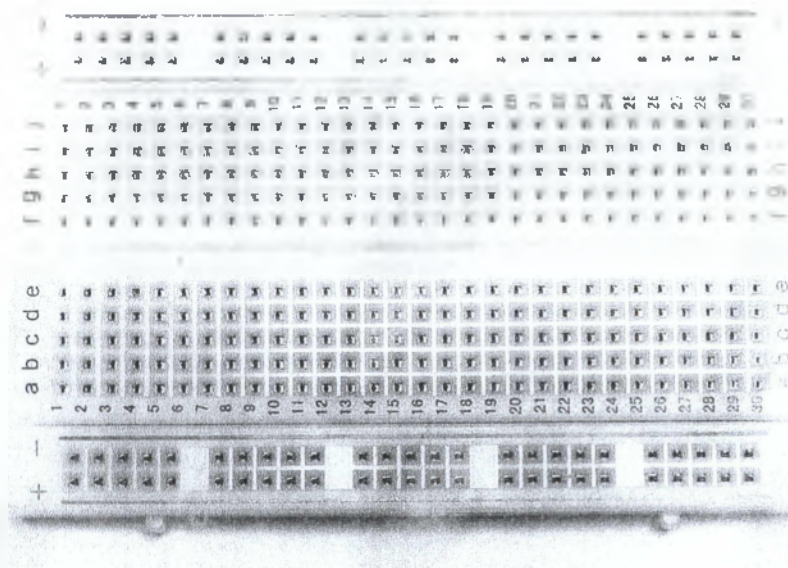
Στο Arduino Ethernet Shield δίνεται μια διεύθυνση MAC και μια σταθερή διεύθυνση IP χρησιμοποιώντας το `Ethernet.begin()` function. Η MAC είναι μια καθολικά μοναδική αναγνωριστική διεύθυνση για μια συγκεκριμένη συσκευή. Η συγκεκριμένη Ethernet Shield έρχεται μαζί με ένα αυτοκόλλητο που δείχνει τη διεύθυνση MAC που θα πρέπει να χρησιμοποιηθεί. Για τα παλαιότερα Ethernet Shield χωρίς μια ειδική MAC διεύθυνση, εφευρίσκοντας μια τυχαία θα πρέπει να δουλέψει, αλλά χωρίς να χρησιμοποιηθεί η ίδια για πολλαπλά σχέδια. Έγκυρες διευθύνσεις IP εξαρτώνται από τη διαμόρφωση του δικτύου. Είναι δυνατόν να χρησιμοποιηθεί το DHCP για τη δυναμική εκχώρηση IP στην Ethernet Shield. Προαιρετικά, μπορεί επίσης να καθοριστεί μια πύλη δικτύου και το δευτερεύον δίκτυο.

SD Card

Η τελευταία διασκευή του Ethernet Shield περιλαμβάνει μια micro - SD υποδοχή κάρτας, η οποία μπορεί να διασυνδεθεί με τη χρήση της βιβλιοθήκης SD.

1.1.4 Breadboard (ή αλλιώς Ράστερ)

Η λέξη breadboard είναι η ειδική πλακέτα άμεσης συναρμολόγησης και αποσυναρμολόγησης χωρίς κολλήσεις για πρόχειρη κατασκευή και έλεγχο ενός κυκλώματος. Πώς ξεκίνησε όμως; Τη δεκαετία του '30 στο εξωτερικό, όταν τα υλικά βρίσκονταν δύσκολα, πολλοί πειραματιστές συναρμολογούσαν το κύκλωμά τους με τις λυχνίες πάνω σε μια πρόχειρη ξύλινη βάση, συνήθως το ξύλο κοπής του ψωμιού, αντί για μεταλλικό σασί. Έτσι προέκυψε το περίφημο breadboard των ηλεκτρονικών.



Σχήμα 1.1.4: Breadboard, ενώθηκαν με απλά καλώδια το φωτοκύτταρο και το Arduino.

1.2.1 Ηλεκτρονικός Υπολογιστής (H/Y) ή Laptop

Ο H/Y (στην εργασία χρησιμοποιήθηκε ένα laptop, για λόγους μεταφοράς) είναι φορτωμένος με το αντίστοιχο λογισμικό για να μπορούν να συντονιστούν όλα τα δεδομένα και τα στοιχεία μεταξύ τους ώστε να παρθεί το τελικό αποτέλεσμα.

1.2.2 Εξωτερικό Modem / Router

Έγινε χρήση εξωτερικού Mode / Router λόγω του ότι οι δοκιμές και οι συνδέσεις έγιναν σε ένα απλό τοπικό δίκτυο.

1.2.3 Καλωδίωση και Συνδεσμολογία

Απαραίτητη είναι η καλωδίωση όπως επίσης και ο σωστός τρόπος συνδεσμολογίας όλων των συσκευών μεταξύ τους. Κρίθηκε σημαντικό το γεγονός ότι, όλα γίνανε σε ασφαλή τοποθεσία ώστε να μην υπάρξουν τυχόν ατυχήματα ή καταστροφές του υλικού.

1.2.4 Internet

Απαραίτητη είναι η πρόσβαση στο Internet ώστε να μπορούν στο τελικό στάδιο να εμφανιστούν οι οποιεσδήποτε τιμές καταγράφηκαν σε μια ιστοσελίδα.

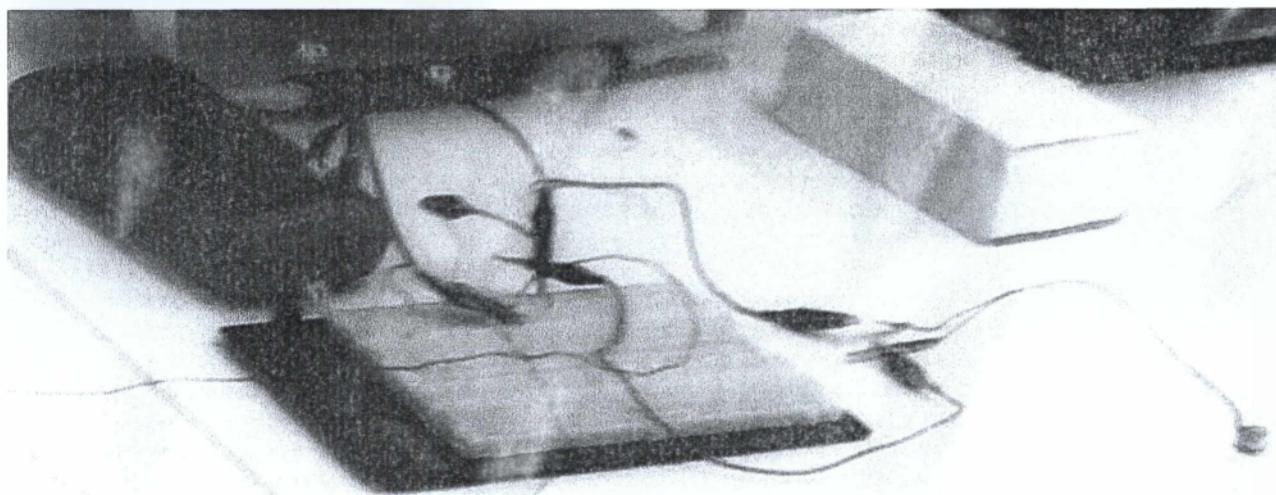
2. ΠΡΑΚΤΙΚΟ ΜΕΡΟΣ



Συλλογή, αποθήκευση, επεξεργασία και απεικόνιση μετρήσεων ηλιοφάνειας σε πραγματικό χρόνο.

2.1 Μεθοδολογία σχεδίασης

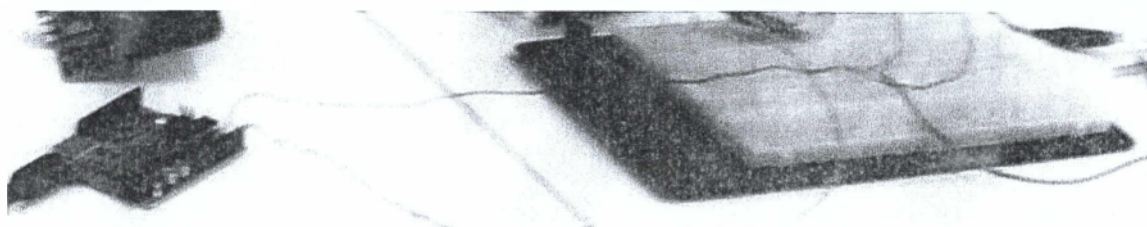
Αφού συγκεντρώθηκαν όλα τα απαραίτητα υλικά που θα συνδέονταν, ακολούθησε η δημιουργία ενός πλάνου μέσω του οποίου θα αναπαριστανόταν όλο το σύστημα. Αρχικά ελέγχθηκε το φωτοκύτταρο, αν ήταν σε καλή κατάσταση, αν τα καλώδια του ήταν φθαρμένα και γενικότερα αν λειτουργούσε. Χρησιμοποιήθηκε ένα πολύμετρο που συνδέθηκε μαζί με το φωτοκύτταρο ώστε να ελεγχθεί αν τα αποτελέσματα που εμφανίζονταν στην οθόνη του πολύμετρου ήταν έγκυρα και ανάλογα κάθε περίπτωσης. Έτσι όταν το φωτοκύτταρο βρισκόταν σε μεγάλη έκθεση φωτός τότε οι τιμές που αναδεικνύονταν στο πολύμετρο ήταν υψηλές. Στην αντίθετη περίπτωση όταν υπήρχε λιγότερο – έως καθόλου φως στον χώρο «που έβλεπε» το φωτοκύτταρο τότε οι τιμές πλησίαζαν τις μηδενικές. Αφού λοιπόν εγκρίθηκε η λειτουργία του φωτοκύτταρου όπως και το γεγονός ότι θα ήταν συμβατό και θα πληρούσε τις προϋποθέσεις για τον στόχο της εργασίας, μπήκε στο κύκλωμα. Έπειτα συνδέθηκε στο breadboard στις σωστές οπές.



Σχήμα 2.1: Φωτοκύτταρο – Πολυμετρητής – Breadboard.

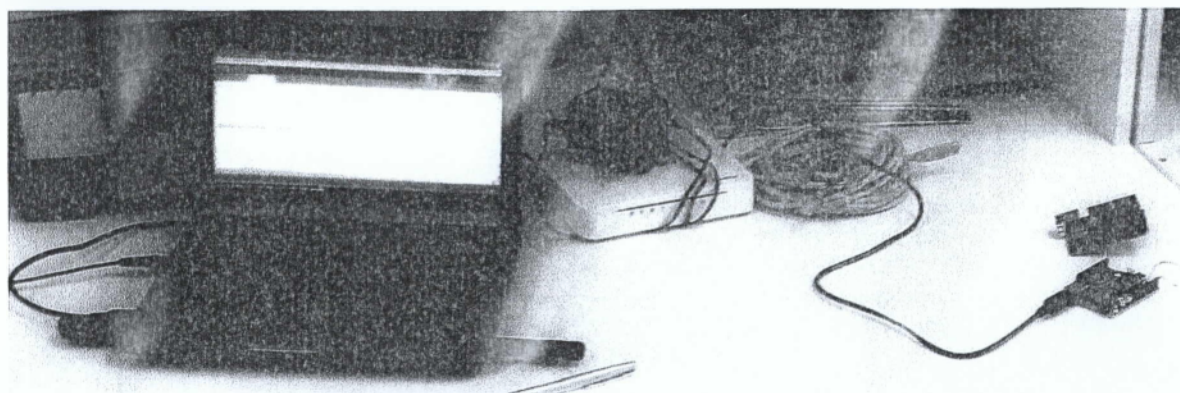
2.2 Προδιαγραφές

Κρίθηκε απαραίτητο το να μπορέσει στο επόμενο στάδιο να συνδεθεί το Arduino με το breadboard καθώς και με τον Η/Υ. Το 1^ο έγινε με δυο απλά καλώδια που ξεκίνησαν από το breadboard και κατέληξαν πάνω στις υποδοχές του Arduino



Σχήμα 2.2.1: Breadboard – Arduino.

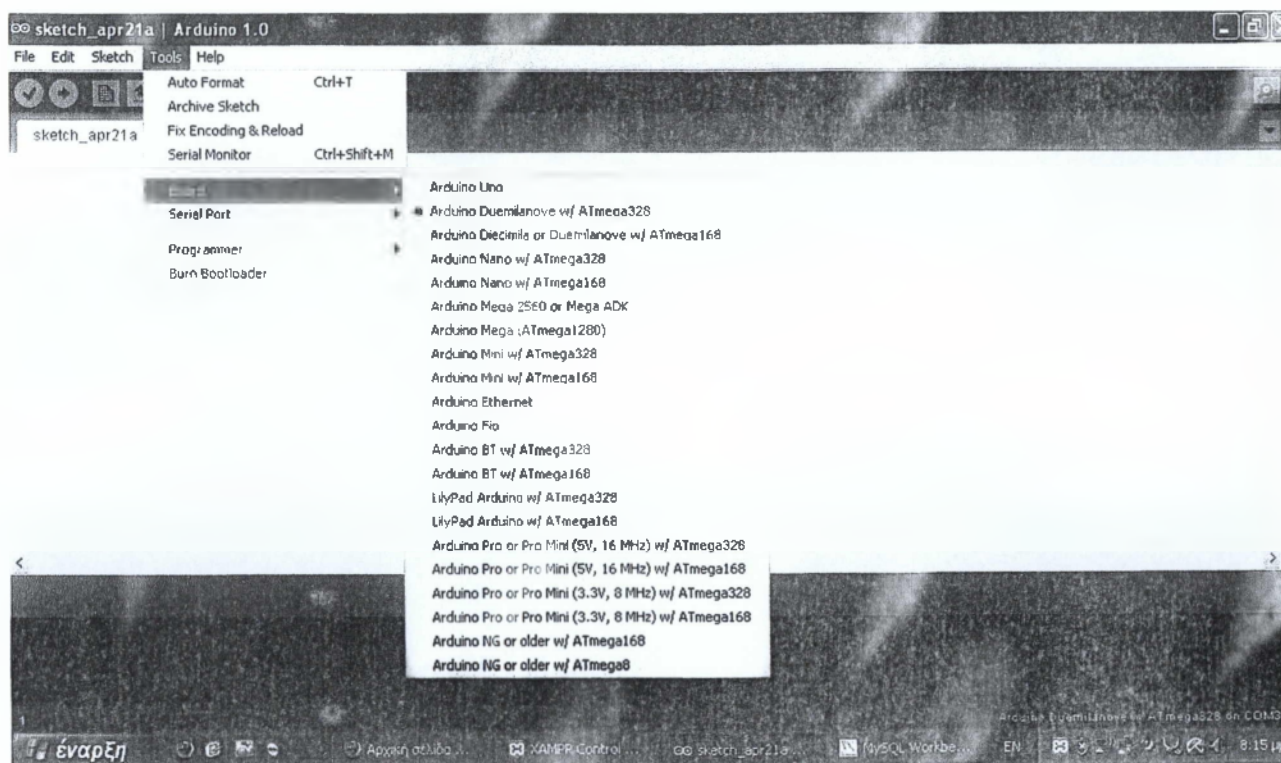
και το 2^ο, μέσω ενός απλού καλωδίου USB το οποίο ξεκίνησε από την θύρα του Arduino και κατέληξε στην θύρα του Η/Υ. Σε αυτό το σημείο αναγνωρίστηκε από τον Η/Υ το Arduino ως μια νέα εξωτερική συσκευή που δεν είχε εγκατασταθεί το λογισμικό της (drivers) και έτσι χρειάστηκε εγκατάσταση.



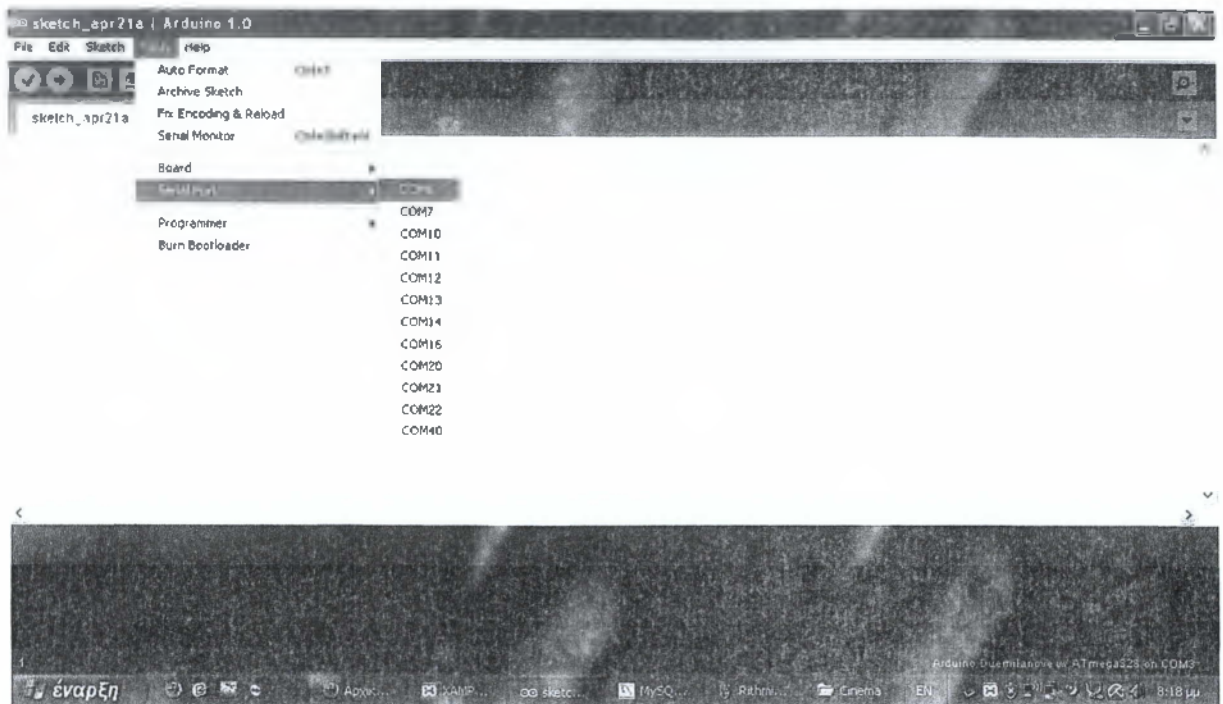
Σχήμα 2.2.2: Arduino – Η/Υ.

Η εγκατάσταση της συσκευής είναι απλή και για αυτό το λόγο παραλείφθηκε.

Μετά την ολοκλήρωση της εγκατάστασης της συσκευής Arduino ήταν απαραίτητη η εγκατάσταση του προγράμματος Arduino 1.0 το οποίο χρησιμοποιήθηκε ώστε, αρχικά να γίνουν οι απαραίτητες δοκιμές για να φανεί αν το φωτοκύτταρο επικοινωνεί με το Arduino καθώς και για το εάν περνάνε οι τιμές που καταγράφονται μέσα στο πρόγραμμα του Arduino. Ρυθμίστηκαν οι απαραίτητες παράμετροι του προγράμματος, όπως το ποιο μοντέλο του Arduino χρησιμοποιείται (από το περιβάλλον του Arduino 1.0 επιλέχτηκε: Tools – Board – Arduino Duemilanove w/ ATmega328), καθώς και η θύρα USB που ενώθηκε (Tools – Serial Port – COM3).

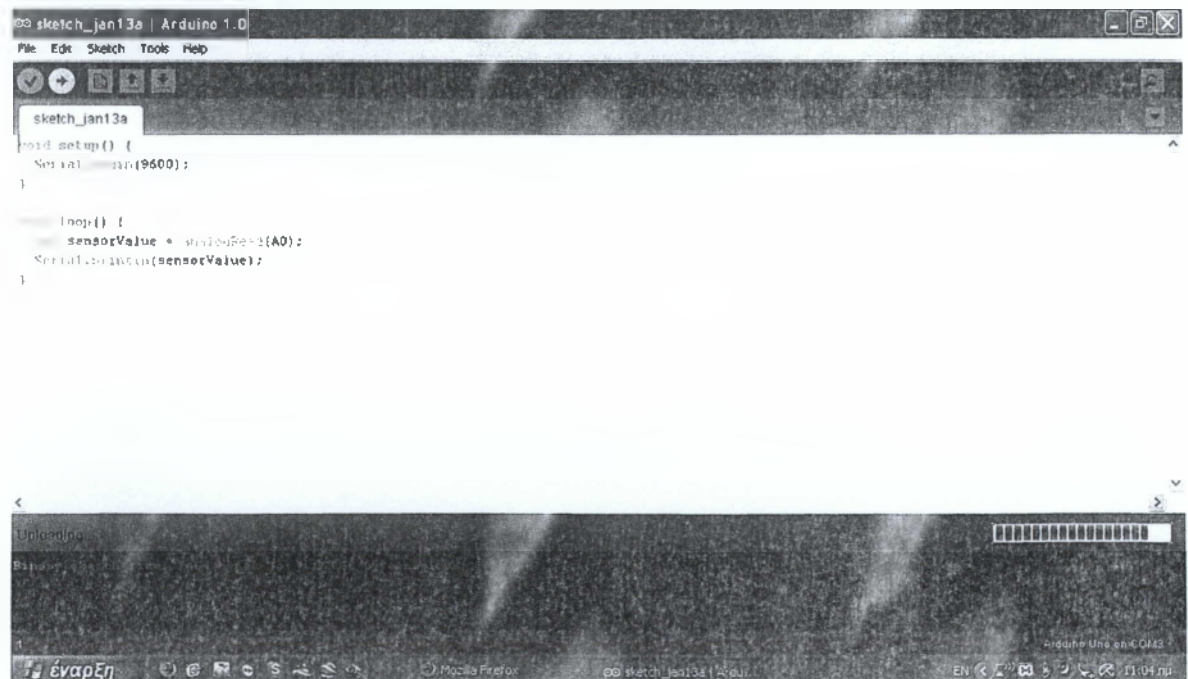


Σχήμα 2.2.3: Επιλογή συσκευής που χρησιμοποιείται.



Σχήμα 2.2.4: Επιλογή θύρας USB.

Έπειτα γράφτηκε ο απαραίτητος κώδικας που θα έκανε την δουλειά που χρειάστηκε στην προκειμένη περίπτωση, δηλαδή να στέλνει – «ανεβάζει» το φωτοκύτταρο τις τιμές που καταγράφει κάθε 1 λεπτό της ώρας.



Σχήμα 2.2.5: Κώδικας που στέλνονται τα δεδομένα από το φωτοκύτταρο.

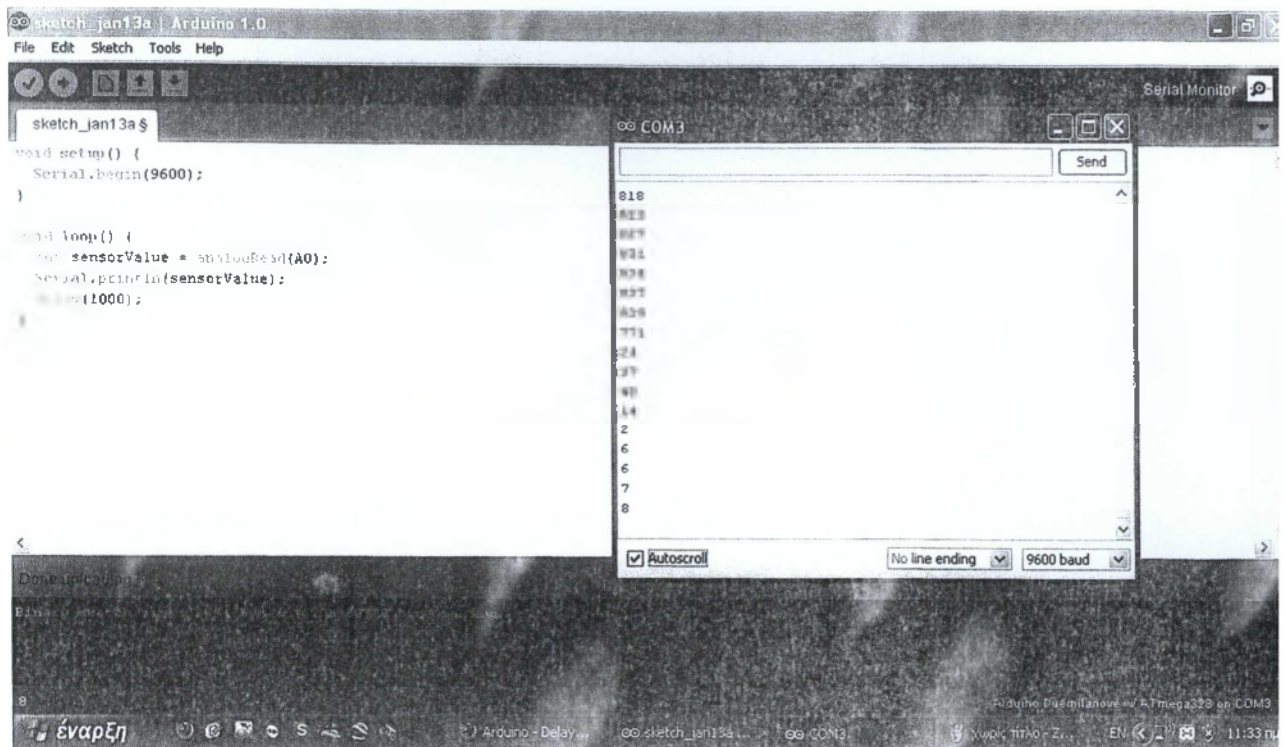
Για να γίνει αυτό χρησιμοποιήθηκε μια εντολή που κάνει αυτή ακριβώς την λειτουργία, η `delay(X)`. Να σημειωθεί ότι, στην συγκεκριμένη εντολή όπου `X` μπορεί να μπει οποιαδήποτε ακέραια τιμή.

Ο κώδικας ξεχωριστά:

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop {  
    int sensorValue = analogRead (A0);  
    Serial.println(sensorValue);  
    delay(1000);  
}
```

Μετά την ολοκλήρωση συγγραφής του κώδικα έγινε επαλήθευση (Verify), αποθήκευση (Save), και «ανέβασμα» (Upload). Τα παραπάνω γίνανε με τα κουμπιά που βρίσκονται πάνω ακριβώς από τον κώδικα στο περιβάλλον του Arduino 1.0, είναι το τσεκ (1^ο κουμπί), το βελάκι προς τα κάτω (5^ο κουμπί) και τέλος, το βελάκι προς τα δεξιά (2^ο κουμπί).

Τέλος επιλέχτηκε από πάνω δεξιά στο περιβάλλον του Arduino 1.0 το κουμπί με τον μεγεθυντικό φακό που είναι η σειριακή οθόνη (Serial Monitor).



Σχήμα 2.2.6: Εμφάνιση καταγραφών από φωτοκύτταρο στην οθόνη κάθε 1 λεπτό.

Μετά το τέλος των παραπάνω βημάτων, δείχνει το φωτοκύτταρο τις ενδείξεις που κατέγραφε στην σειριακή οθόνη του Arduino 1.0. Δηλαδή ο πίνακας που φαίνεται στα δεξιά του παραπάνω σχήματος. Επιλέχτηκε στο κάτω μέρος του παραθύρου της οθόνης το Autoscroll, που η λειτουργία του είναι να πηγαίνει πιο κάτω τη λίστα με τις τιμές που καταγράφονται όταν γεμίσει όλη με αριθμούς, ώστε να μην χρειαστεί να γίνεται συνέχεια Scroll Down χειροκίνητα. Παρατηρείται ότι οι τιμές αρχικά είναι υψηλές γιατί στην δοκιμή που έγινε υπήρχε άπλετο φως στον χώρο και μετά από λίγα λεπτά πέφτουνε απότομα γιατί μειώθηκε σημαντικά το φως που υπήρχε, ως αποτέλεσμα αυτού, οι τιμές μειώθηκαν κατά πολύ πλησιάζοντας το 0.

Παρακάτω φαίνεται το μέχρι τώρα σύστημα καθώς και η συνδεσμολογία του:



Σχήμα 2.2.7 : Πλάνο – 1^η προσπάθεια διασύνδεσης του όλου συστήματος.

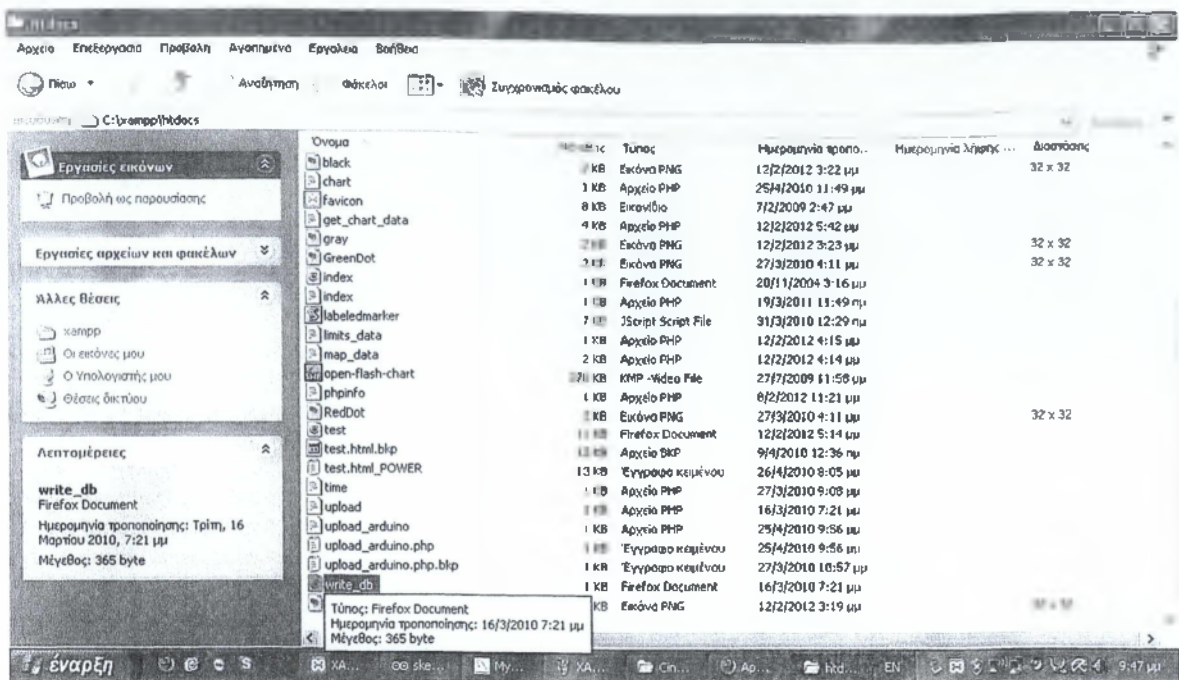
2.3 XAMPP (X: cross, Apache HTTP Server, MySQL, PHP, Perl)

Χρησιμοποιήθηκε το XAMPP που είναι το ιδανικό εργαλείο για να στηθεί ένα site στον Η/Υ και να γίνουμε διάφορες δοκιμές. Επιλέχτηκε σαν πρόγραμμα γιατί μέσω αυτού μπορεί να επικοινωνεί η βάση δεδομένων που θα χρησιμοποιηθεί αργότερα με τον Apache Server συμπεριλαμβανομένης και της PHP. Αφού εγκαταστάθηκε το πρόγραμμα λοιπόν (ο αναλυτικός τρόπος εγκατάστασης παραλείφθηκε) δεν χρειάστηκαν ρυθμίσεις ή γενικότερα παράμετροι ώστε να τρέχει σωστά.



Σχήμα 2.3.1: Το περιβάλλον του προγράμματος XAMPP.

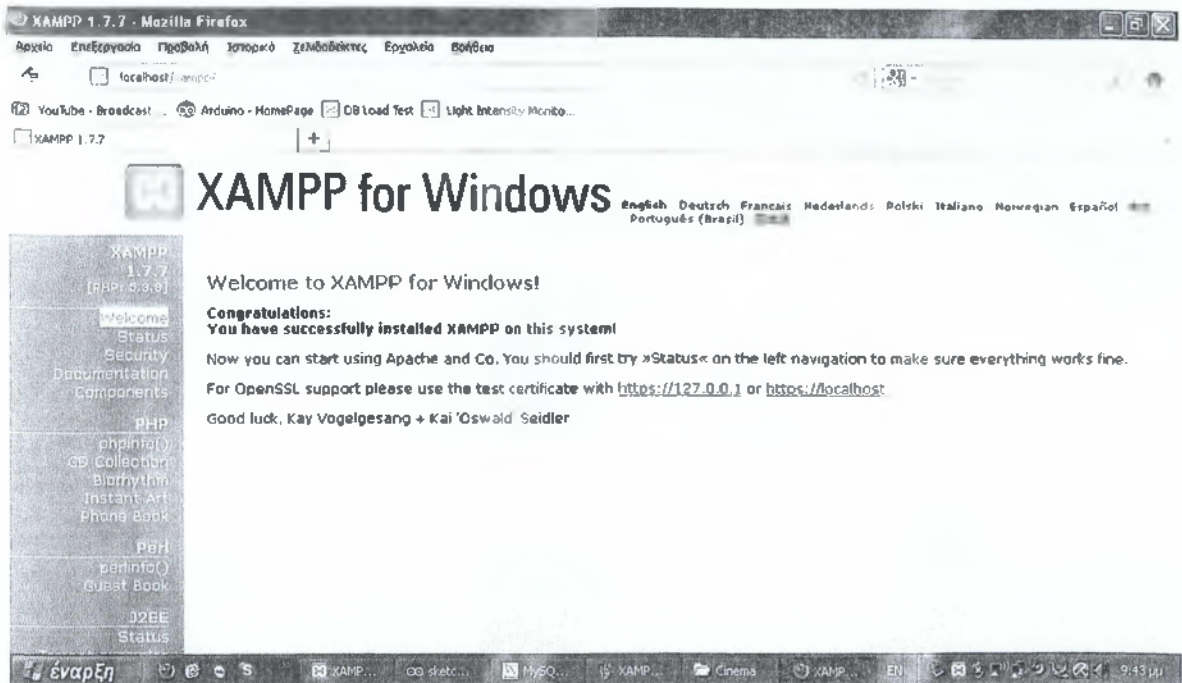
Το μόνο που κρίθηκε απαραίτητο ήταν να μπουνε τα σωστά αρχεία μέσα στους σωστούς υποφακέλους (htdocs) ώστε να εμφανίζονται σωστά στον Web Server και ως αποτέλεσμα να υπάρχει επικοινωνία μεταξύ του Apache server και της MySQL.



Σχήμα 2.3.2: Φάκελος που περιέχει τα απαραίτητα αρχεία μέσα στο htdocs.

Το μόνο που χρειάστηκε για αυτό το πρόγραμμα ήταν, να ανοίγει πρώτο πριν από το Arduino 1.0 και το MySQL Workbench ώστε να μην υπάρξει πρόβλημα συγχρονισμού του.

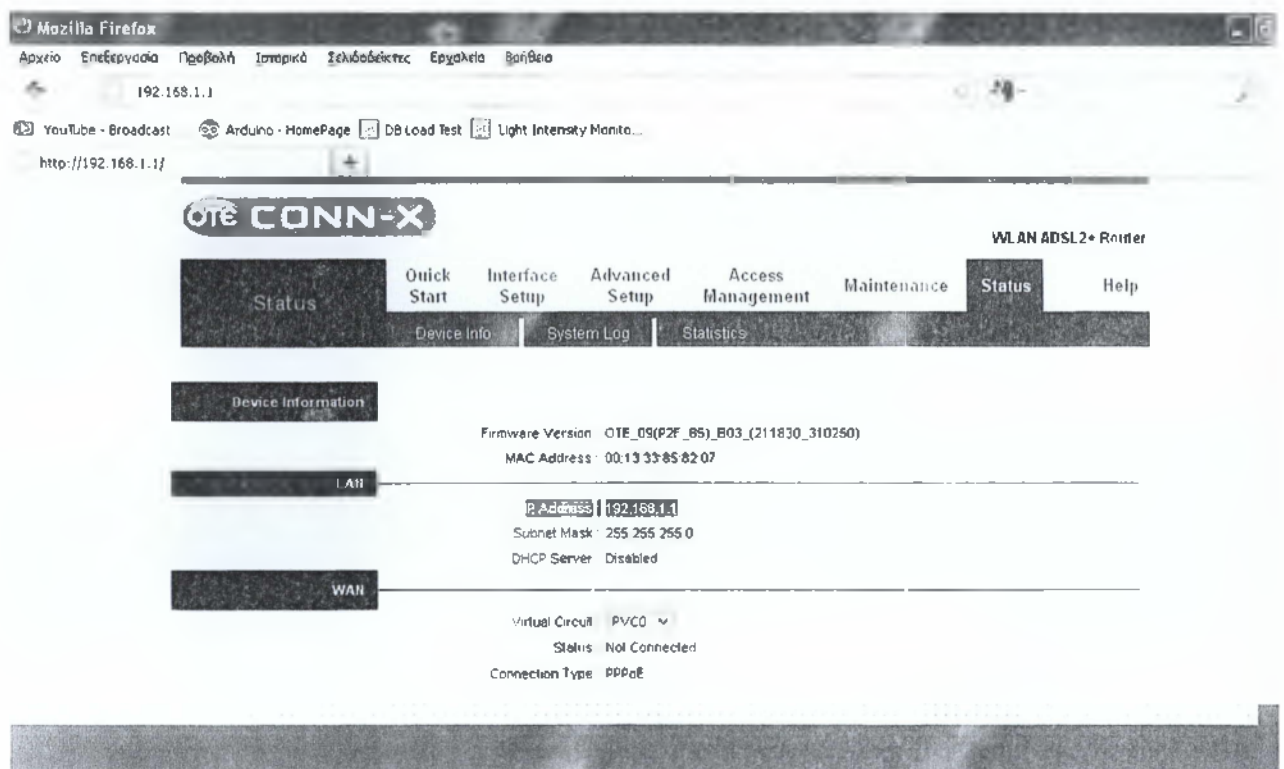
Ανοίγοντας έναν Web Browser (στην εργασία χρησιμοποιήθηκε Mozilla Firefox) και πληκτρολογώντας την εξής διεύθυνση: <http://localhost/xampp/> θα έβγαине το παρακάτω αποτέλεσμα:



Σχήμα 2.3.4: Ρυθμίσεις και παράμετροι του XAMPP μέσα από τον φυλλομετρητή.

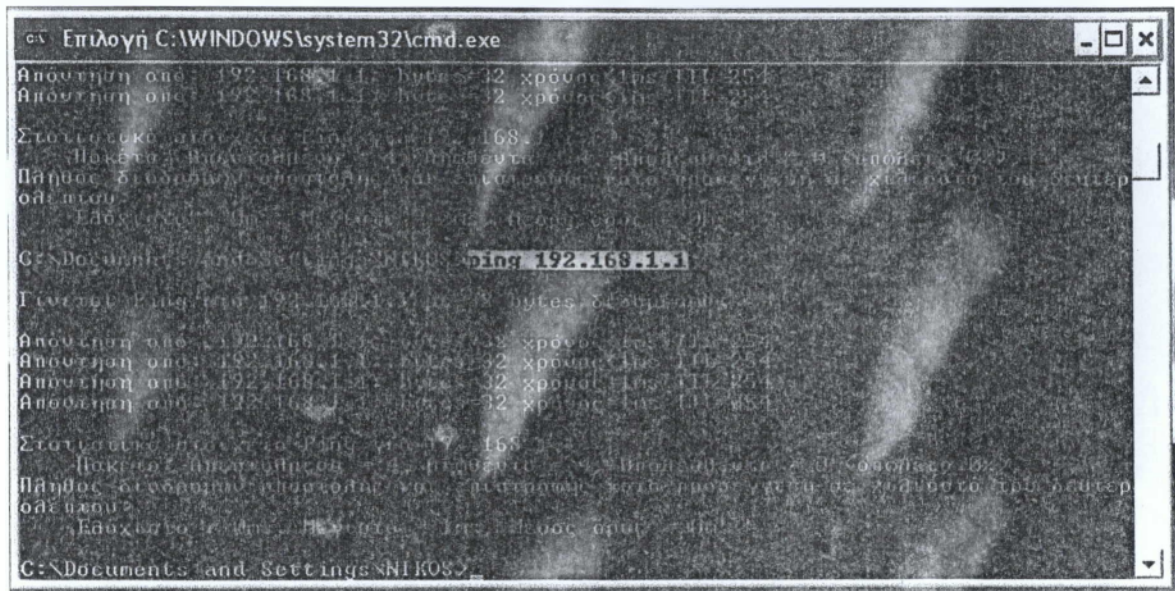
2.4 Ρύθμιση Modem/Router με στατική IP

Το ρούτερ που χρησιμοποιήθηκε είναι ένα BaudTec ISDN / TW263R4 - B2. Είναι ένα απλό ρούτερ που είναι κυρίως για οικιακή χρήση και συνδέθηκε μέσω ενός Ethernet καλωδίου RJ – 45 στον Η/Υ. Έχοντας κάνει Reset ώστε να έρθει στις αρχικές – προκαθορισμένες ρυθμίσεις του, του δόθηκε στατική IP. Αυτό έγινε μέσα από το περιβάλλον του ρούτερ δηλαδή δίνοντας για διεύθυνση στον φυλλομετρητή την: 192.168.1.1 και κάνοντας τις περαιτέρω ρυθμίσεις.



Σχήμα 2.4.1: Στατική IP που δόθηκε στο ρούτερ.

Έπειτα, για να επαληθευτεί ότι το ρούτερ ήταν σωστά ρυθμισμένο και ότι επικοινωνούσε επιτυχώς με τον Η/Υ μέσω της IP που του δόθηκε έγινε Ping μέσω του Command Prompt (έναρξη – αναζήτηση – cmd) και γράφτηκε η εντολή: ping 192.168.1.1 και πατήθηκε το Enter. Βλέποντας ότι τα πακέτα στέλνονται κανονικά και ότι δεν υπάρχουνε απολεσθέντα βγήκε το συμπέρασμα ότι το ρούτερ πλέον ήταν σωστά ρυθμισμένο και λειτουργούσε κανονικά.



Σχήμα 2.4.2: Ping μέσω cmd για αποτελέσματα πακέτων επικοινωνίας του ρούτερ.

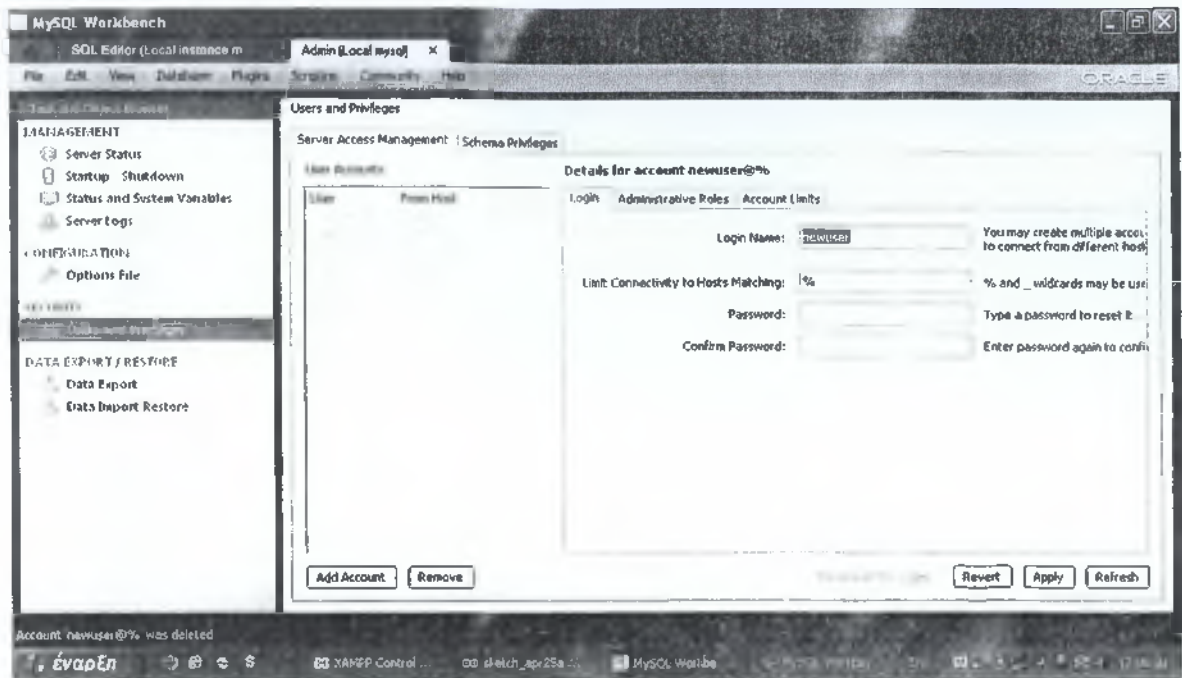
2.5 MySQL Workbench

Στην εργασία χρησιμοποιήθηκε το πρόγραμμα MySQL Workbench 5.2.37.



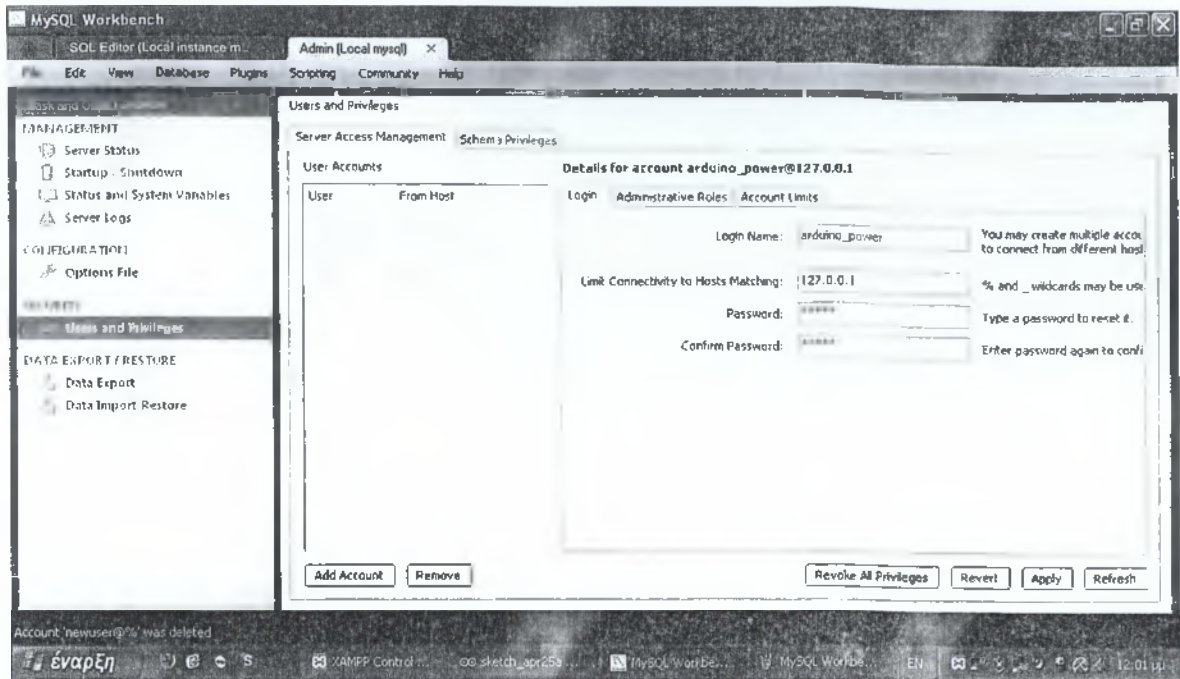
Σχήμα 2.5.1: MySQL Workbench αρχική οθόνη (home).

Αφού εγκαταστάθηκε επιτυχώς, το επόμενο βήμα ήταν να δημιουργηθεί ένας χρήστης. Αυτό έγινε από το κυρίως μενού (home) στην επιλογή Manage Security, βγάζει ένα νέο παράθυρο με τους χρήστες και τα δικαιώματα αυτών (Users and Privileges).



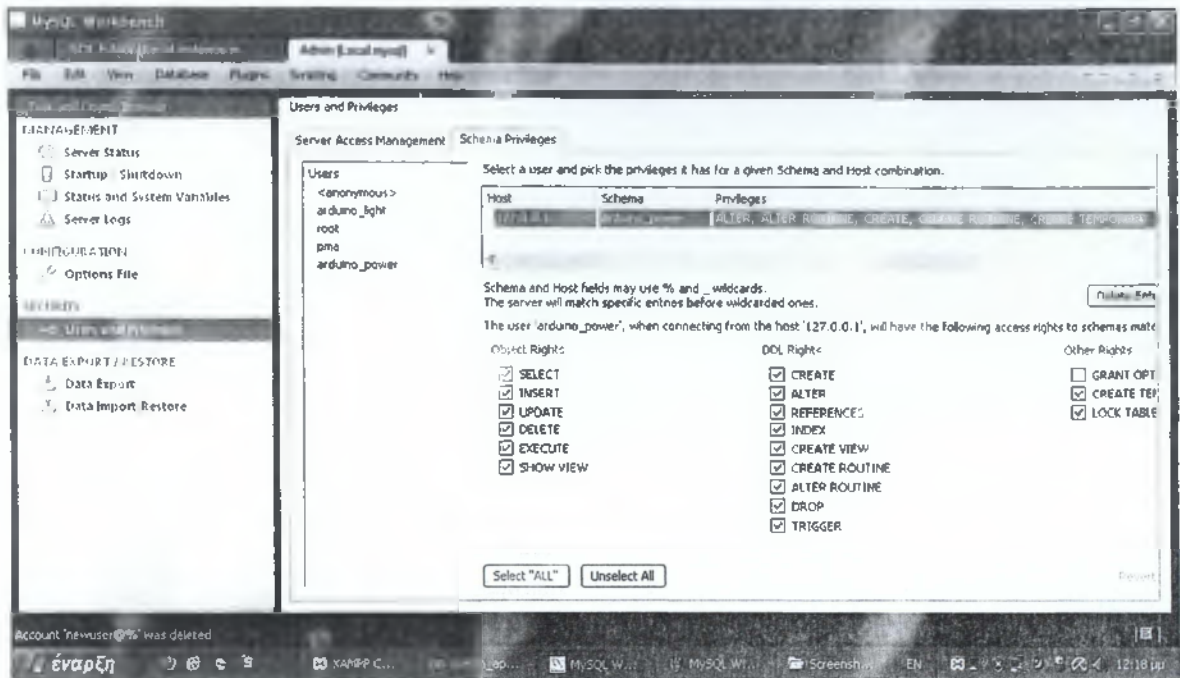
Σχήμα 2.5.2: Δημιουργία – Προσθήκη λογαριασμού στο MySQL Workbench.

Από εκεί επιλέχθηκε το: Προσθήκη Λογαριασμού (Add Account) και συμπληρώθηκαν τα κενά με τα απαραίτητα στοιχεία. Login Name: arduino_power, Limit Connectivity to Hosts Matching: 127.0.0.1, Password: power, Confirm Password: power και τέλος πατήθηκε το Apply.



Σχήμα 2.5.3: Συμπλήρωση στοιχείων του χρήστη που δημιουργήθηκε.

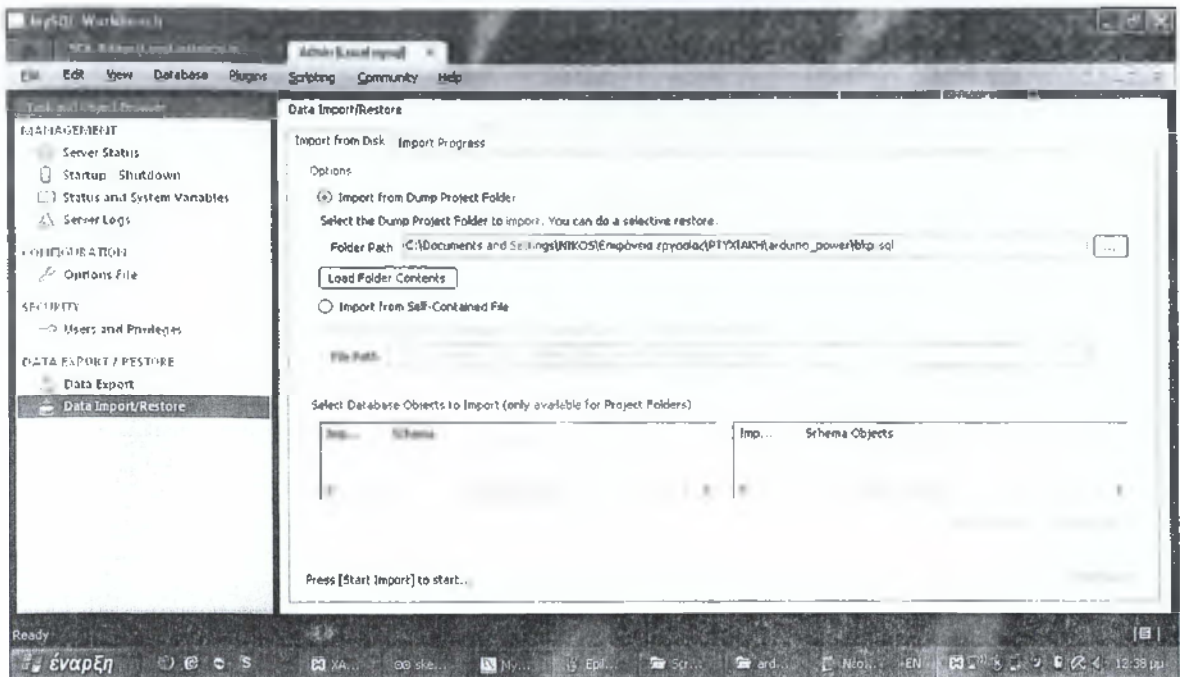
Έπειτα για να δοθούν δικαιώματα στον χρήστη που μόλις δημιουργήθηκε, επιλέχτηκε το Schema Privileges και από τους χρήστες ο `arduino_power`. Δίπλα ακριβώς εμφανίζει: Host: `127.0.0.1`, Schema: `arduino_power`, Privileges: κενό. Εδώ πρέπει να οριστεί τι είδους δικαιώματα θα έχει ο συγκεκριμένος χρήστης, αυτό γίνεται επιλέγοντας ένα-ένα από κάτω με τσεκ τα κουτάκια (είτε κάνοντας επιλογή όλων ή αντίστοιχα αποεπιλογή όλων). Στην προκειμένη περίπτωση έγινε επιλογή όλων πέραν του Grant Option. Τέλος, πατήθηκε το Apply ώστε να καταχωρηθούν τα δικαιώματα αυτά στον χρήστη.



Σχήμα 2.5.4: Επιλογή και καταχώρηση δικαιωμάτων στον χρήστη `arduino_power`.

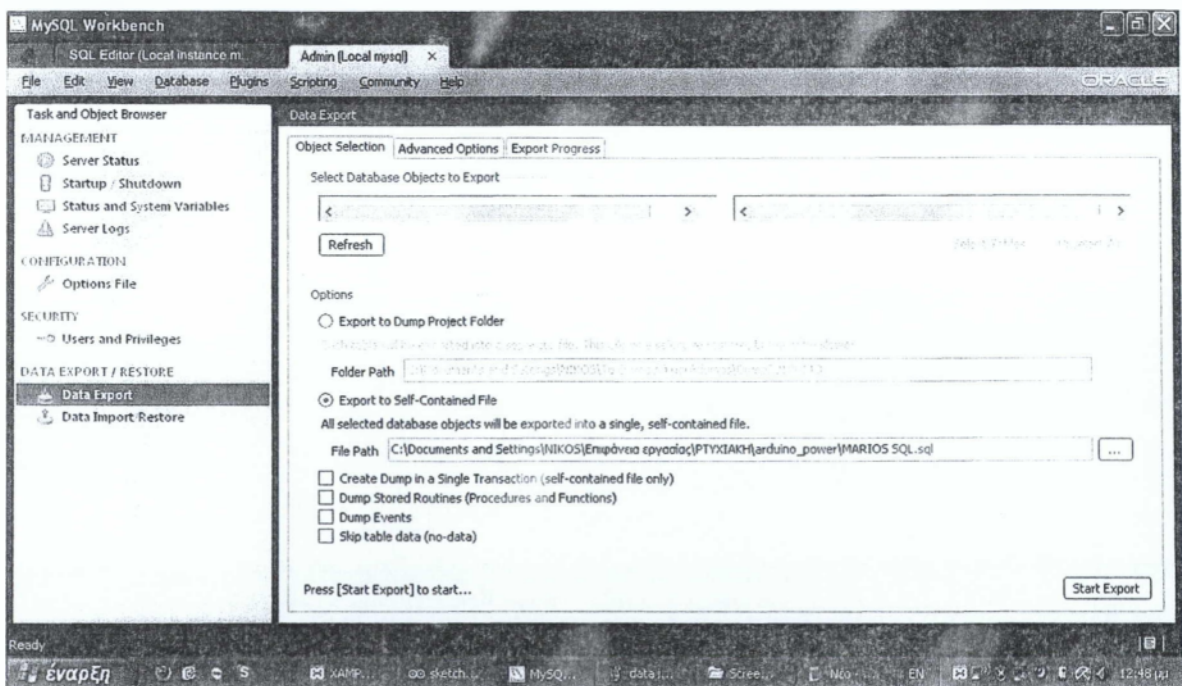
2.5.1 Παράμετροι του MySQL Workbench

Από το κεντρικό μενού επιλέγοντας το `Manage Import / Export` εμφανίζει ένα νέο παράθυρο και από εκεί γίνεται η εξαγωγή δεδομένων ή η εισαγωγή. Επιλέγοντας την εισαγωγή (`Data Import/Restore`) βγάζει κάποιες επιλογές, πατώντας στο `Import from Dump Project Folder` συμπληρώθηκε το πεδίο με: `C:\Documents and Settings\NIKOS\Επιφάνεια εργασίας\PTYXIAKH\arduino_power\bkp.sql`.



Σχήμα 2.5.1.1: Εισαγωγή δεδομένων (Data Import).

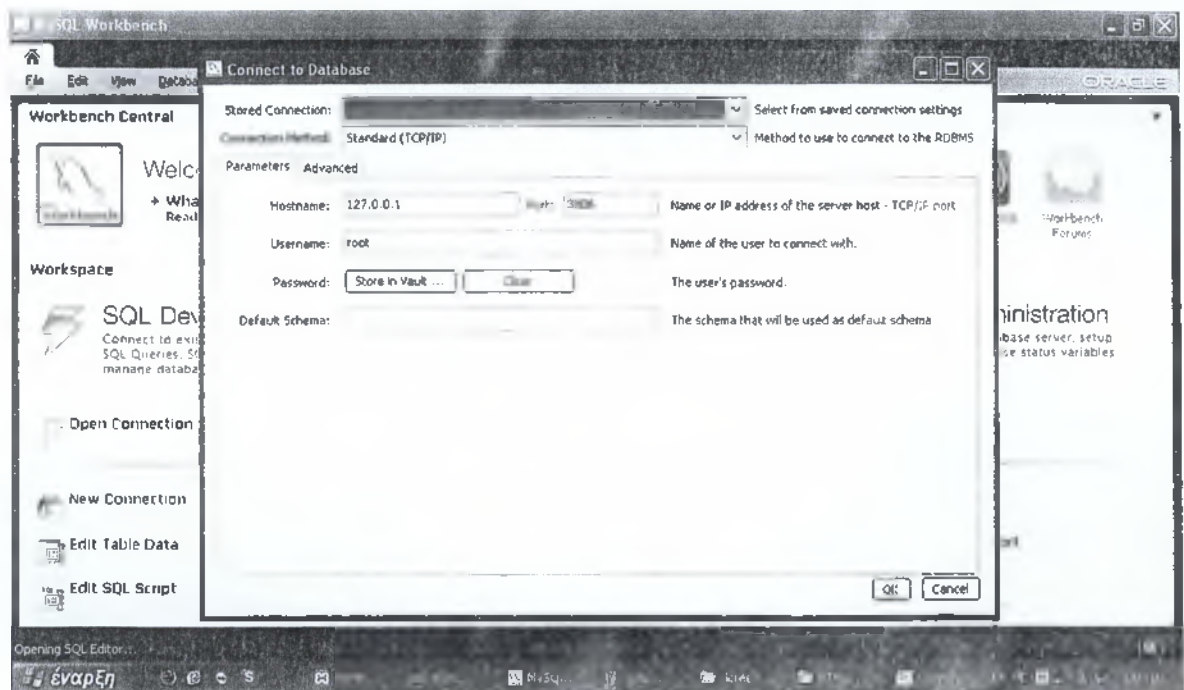
Έπειτα επιλέχτηκε πάνω από το Data Import/Restore το Data Export. Σε αυτό επιλέγοντας το Export to Self-Contained File συμπληρώθηκε το πεδίο με: C:\Documents and Setting\NIKOS\Επιφάνεια εργασίας\PTYXIAKH\arduino_power\MARIOS SQL.sql.



Σχήμα 2.5.1.2: Εξαγωγή δεδομένων (Data Export).

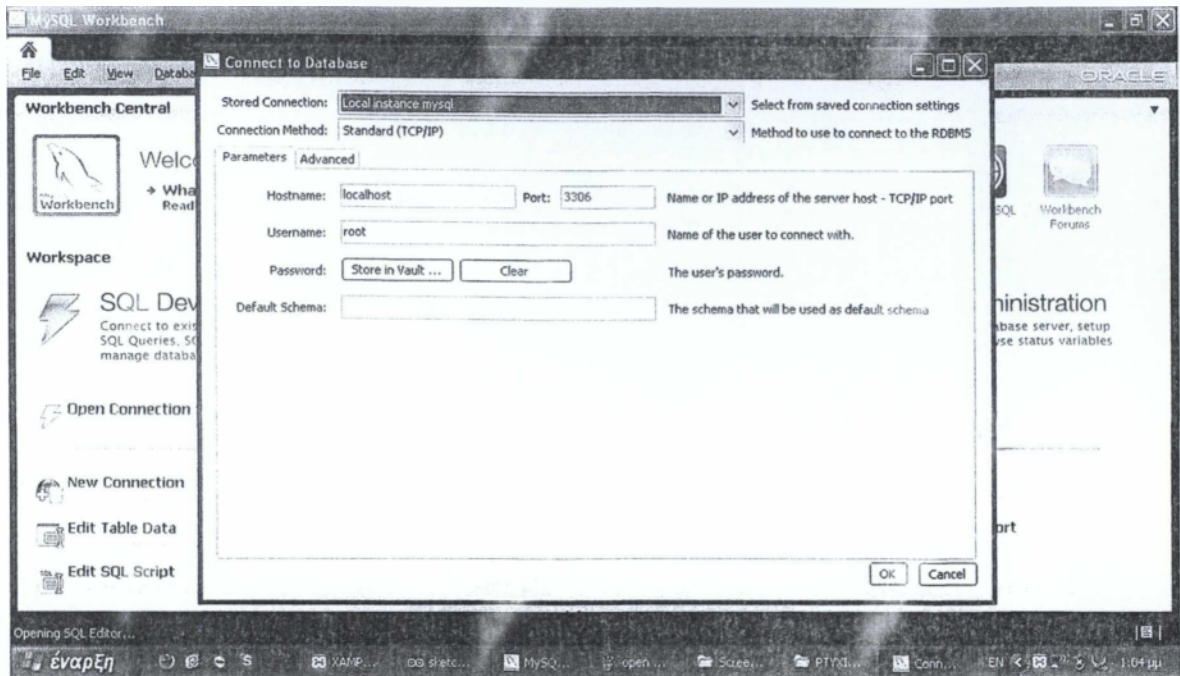
2.5.2 Δημιουργία βάσεων δεδομένων

Σε αυτό το στάδιο, κρίθηκε απαραίτητη η συγγραφή κώδικα ώστε να δημιουργηθεί η βάση δεδομένων που χρειαζότανε, για την καταγραφή και αποθήκευση των δεδομένων που στέλνονταν από το Arduino στον Η/Υ. Από το κεντρικό μενού του MySQL Workbench επιλέχτηκε το Open Connection to Start Querying.



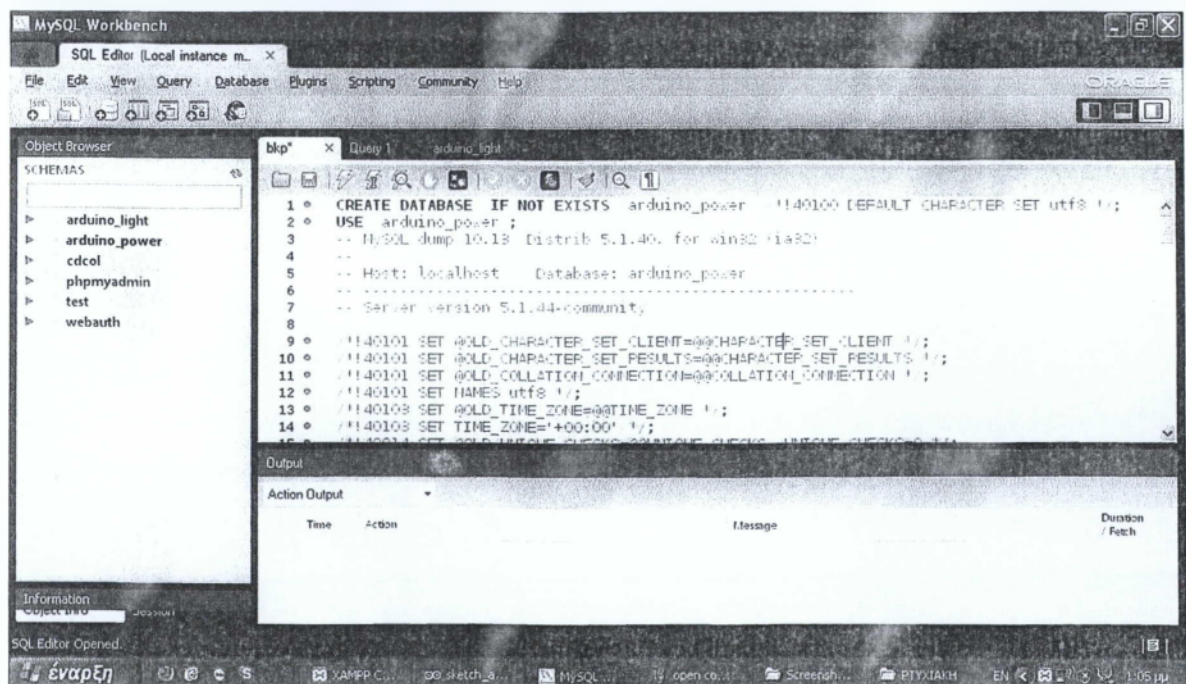
Σχήμα 2.5.2.1: Σύνδεση στην βάση δεδομένων.

Μετά στο Stored Connection επιλέχτηκε το Local instance mysql, τα υπόλοιπα πεδία μείνανε ως είχαν, και τέλος πατήθηκε το OK.



Σχήμα 2.5.2.2: Επιλογή τοπικής σύνδεσης.

Μετά από αυτό το βήμα, εμφανίζεται ένα νέο παράθυρο που φαίνεται ο κώδικας από το αποθηκευμένο αρχείο που έχει οριστεί ως βάση δεδομένων (στην προκειμένη περίπτωση είναι το bkr.sql).



Σχήμα 2.5.2.3: Εμφάνιση και επεξεργασία βάσης δεδομένων.

Ολοκληρωμένος κώδικας της βάσης δεδομένων bkp.sql:

```
CREATE DATABASE IF NOT EXISTS `arduino_power` /*!40100 DEFAULT
CHARACTER SET utf8 */;

USE `arduino_power`;

-- MySQL dump 10.13 Distrib 5.1.40, for Win32 (ia32)

--

-- Host: localhost Database: arduino_power

-----

-- Server version 5.1.44-community

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8 */;

/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;

/*!40103 SET TIME_ZONE='+00:00' */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;

/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
```

```
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
```

```
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

```
--  
-- Table structure for table `events`  
--
```

```
DROP TABLE IF EXISTS `events`;
```

```
/*!40101 SET @saved_cs_client = @@character_set_client */;
```

```
/*!40101 SET character_set_client = utf8 */;
```

```
CREATE TABLE `events` (  
  `event_id` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `station_id` varchar(4) CHARACTER SET utf8 DEFAULT NULL,  
  `timestamp` timestamp NULL DEFAULT NULL,  
  `light_intensity` decimal(10,3) DEFAULT NULL,  
  PRIMARY KEY (`event_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=130 DEFAULT CHARSET=latin1;
```

```
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--  
-- Dumping data for table `events`  
--
```

```

LOCK TABLES `events` WRITE;

/*!40000 ALTER TABLE `events` DISABLE KEYS */;

INSERT INTO `events` VALUES (3, 4,'2011-01-25 17:44:43', 12.345); /*h mporo na ta
balo ola NULL, arkei nanai 4 synolika*/

/*!40000 ALTER TABLE `events` ENABLE KEYS */;

UNLOCK TABLES;

--
-- Table structure for table `limits`
--

DROP TABLE IF EXISTS `limits`;

/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `limits` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `parameter` varchar(45) DEFAULT NULL,
  `min` decimal(10,3) DEFAULT NULL,
  `max` decimal(10,3) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--
-- Dumping data for table `limits`
--

LOCK TABLES `limits` WRITE;

/*!40000 ALTER TABLE `limits` DISABLE KEYS */;

INSERT INTO `limits` VALUES
(1,'frequency','49.000','51.000'),(2,'vrms','215.000','225.000'),(3,'power','0.000','1000.000'),(
4,'power_factor','0.000','1000.000'),(5,'thd','0.000','10.000');

/*!40000 ALTER TABLE `limits` ENABLE KEYS */;

UNLOCK TABLES;

--
-- Table structure for table `stations`
--

DROP TABLE IF EXISTS `stations`;

/*!40101 SET @saved_cs_client = @@character_set_client */;

/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `stations` (
  `station_id` int(11) NOT NULL,
  `station_code` varchar(4) CHARACTER SET utf8 DEFAULT NULL,

```

```

`station_name` varchar(45) CHARACTER SET utf8 DEFAULT NULL,
`station_latitude` decimal(8,5) DEFAULT NULL,
`station_longitude` decimal(8,5) DEFAULT NULL,
PRIMARY KEY (`station_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `stations`
--

LOCK TABLES `stations` WRITE;

/*!40000 ALTER TABLE `stations` DISABLE KEYS */;

INSERT INTO `stations` VALUES
(1,'SPRT','SPARTA','37.07627','22.42996'),(2,'KAST','KASTOREION','37.16798','22.305
16'),(3,'PELN','PELLANA','37.20545','22.32267'),(4,'KLAD','KLADAS','37.11790','22.431
16'),(5,'PERI','PERIVOLIA','37.18138','22.30345');

/*!40000 ALTER TABLE `stations` ENABLE KEYS */;

UNLOCK TABLES;

/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;

/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;

```

```

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS
*/;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION
*/;

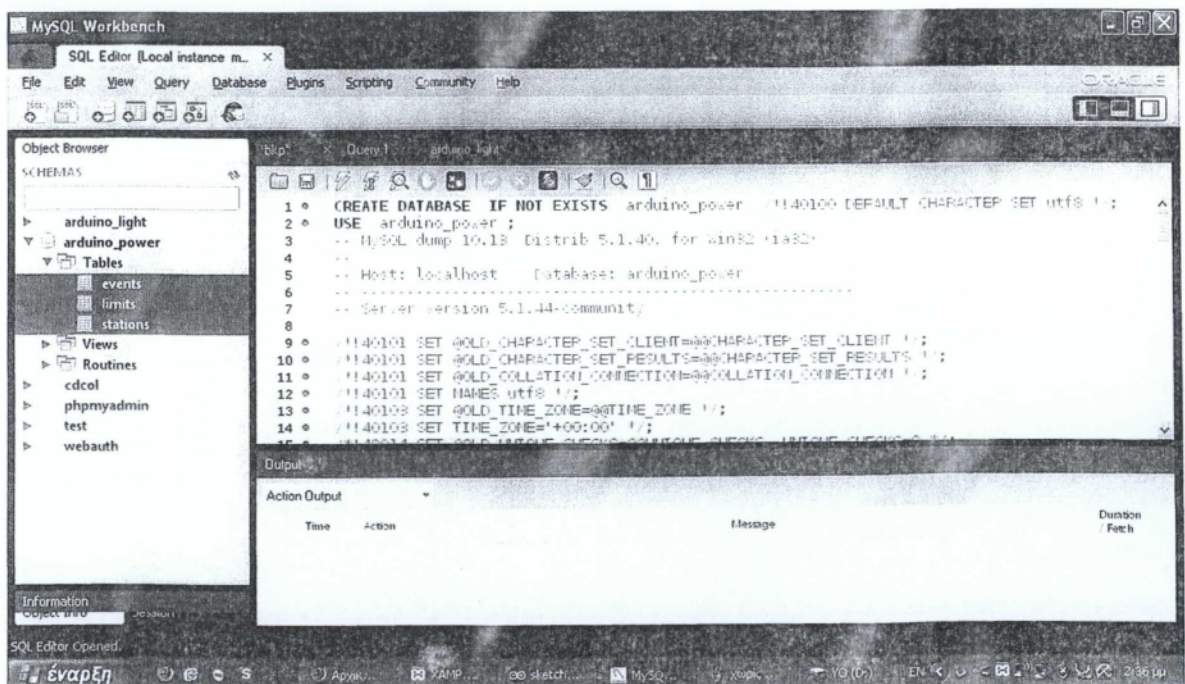
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2010-04-25 23:58:46

```

Οι πίνακες που περιέχει η βάση αυτή είναι οι εξής:

- Events
- Limits
- Stations

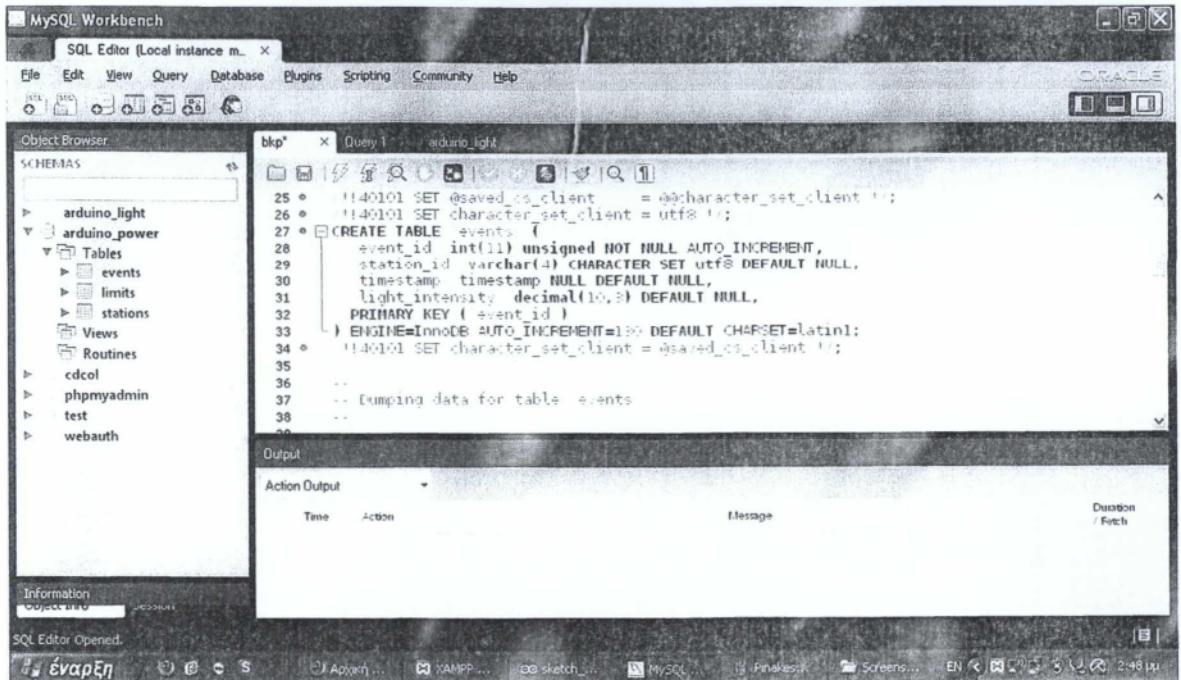


Σχήμα 2.5.2.4: Πίνακες βάσης bkr.

Συλλογή, αποθήκευση, επεξεργασία και απεικόνιση μετρήσεων ηλιοφάνειας σε πραγματικό χρόνο.

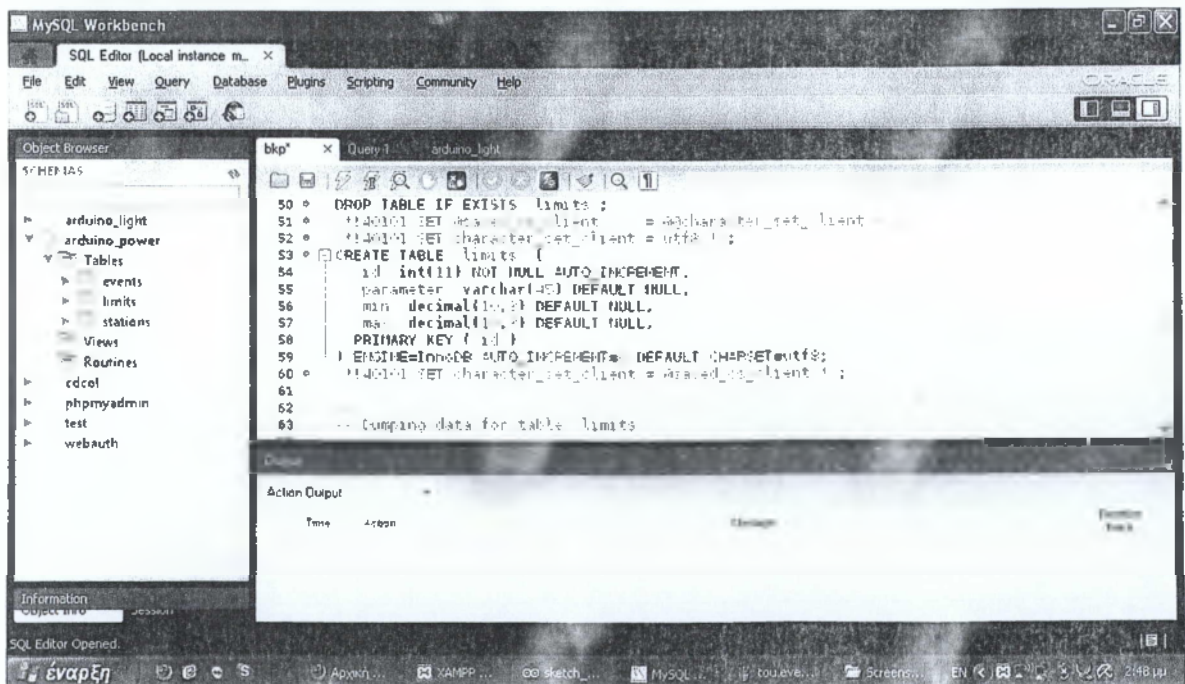
Σε κάθε πίνακα αντιστοιχούν κάποια πεδία, αυτά είναι τα εξής:

1. Στον πίνακα Events αντιστοιχεί το: `event_id`, `station_id`, `timestamp`, `light_intensity`.



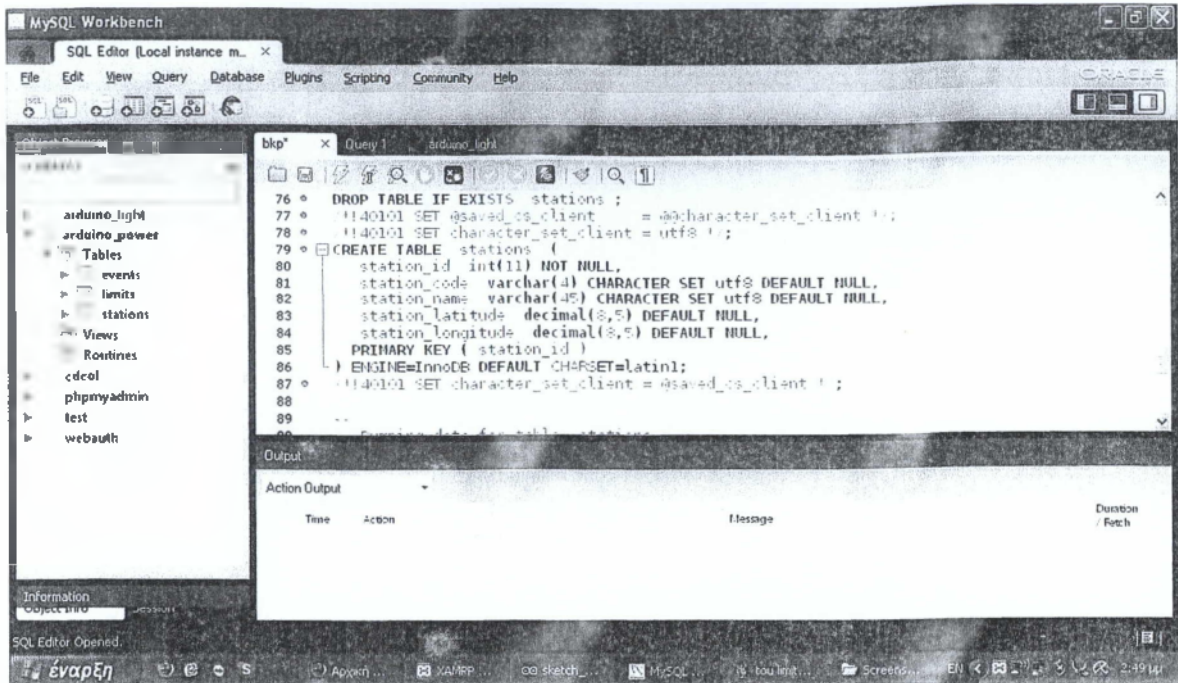
Σχήμα 2.5.2.5: Πίνακας events.

2. Στον πίνακα Limits αντιστοιχεί το: `id`, `parameter`, `min`, `max`.



Σχήμα 2.5.2.6: Πίνακας limits.

3. Στον πίνακα Stations αντιστοιχεί το: station_id, station_code, station_name, station_latitude, station_longitude.

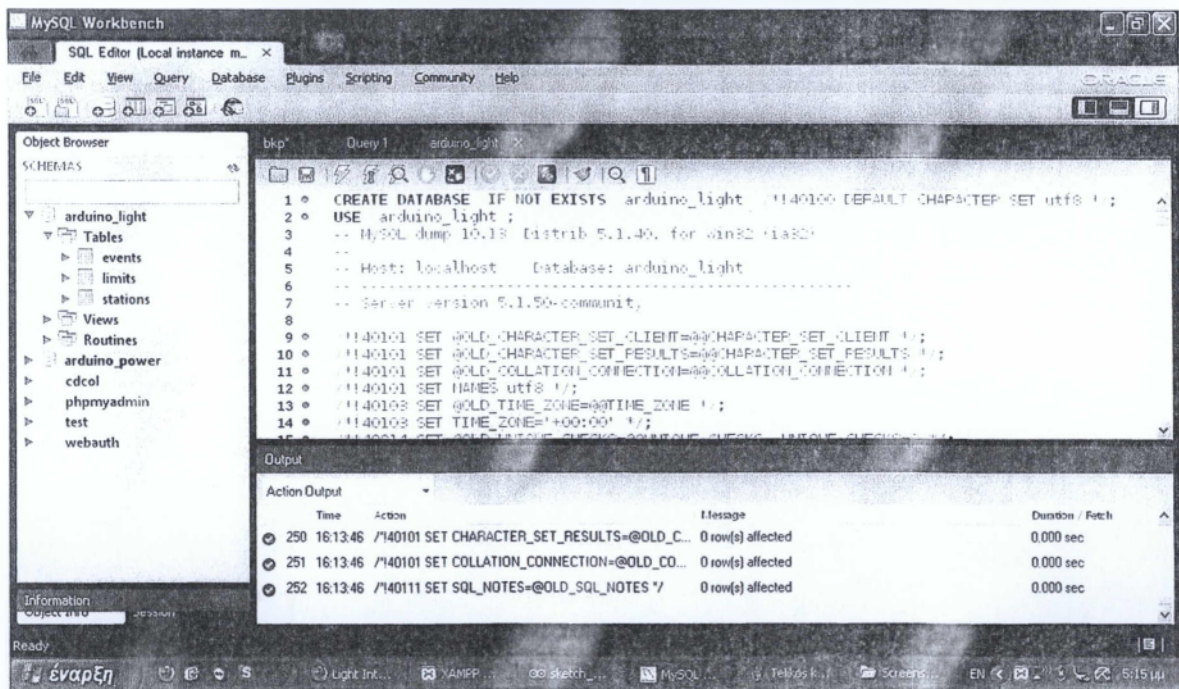


Σχήμα 2.5.2.7: Πίνακας stations.

Η χρησιμότητα του κάθε πίνακα αναφέρεται κατά σειρά παρακάτω:

- Ο πίνακας Events χρησιμοποιείται για τα γεγονότα – στοιχεία που καταγράφονται.
- Ο πίνακας Limits χρησιμοποιείται για τα όρια που τίθενται προκειμένου να καθορίζονται τα διαφορετικά επίπεδα ηλιοφάνειας που θα απεικονίζονται στον χάρτη, όπως θα δούμε στη συνέχεια.
- Ο πίνακας Stations χρησιμοποιείται για τους σταθμούς στους οποίους τοποθετήθηκε το κύκλωμα μας και γίνανε οι καταγραφές.

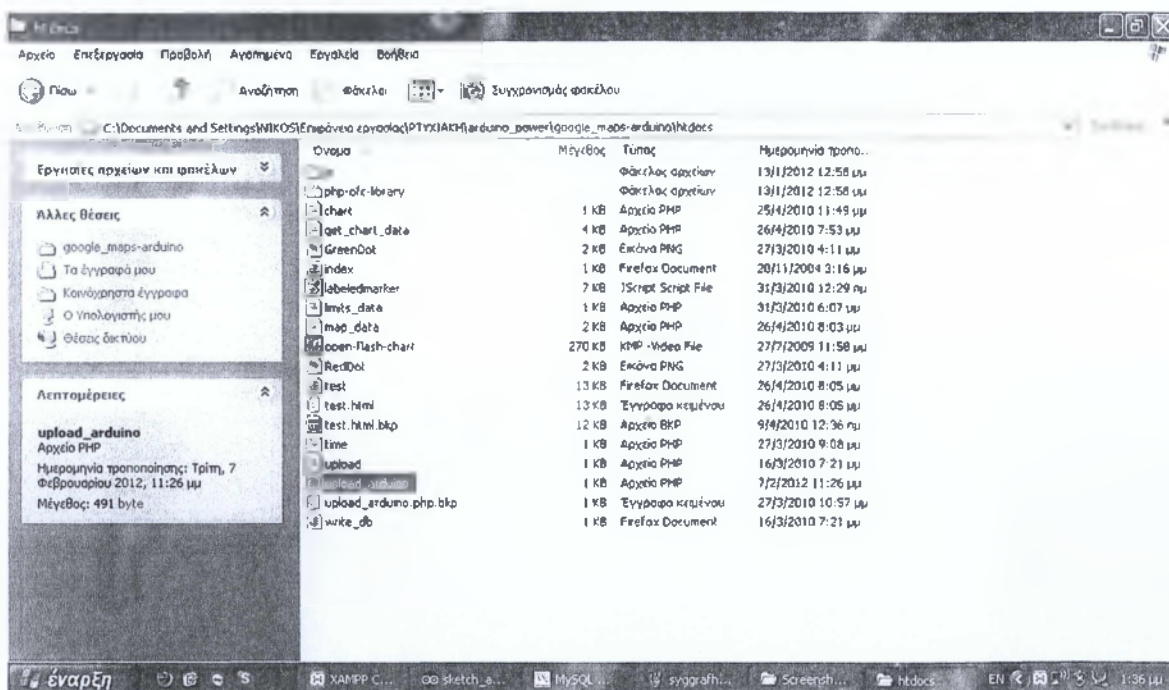
Δημιουργήθηκε άλλη μια βάση δεδομένων με όνομα `arduino_light` (αυτό γιατί θα πρέπει παρακάτω να λειτουργούν τα Google Maps και το Open Flash Chart). Δημιουργήθηκε ένας χρήστης `arduino_light` και του δόθηκαν δικαιώματα, όπως ακριβώς έγινε παραπάνω για τον χρήστη `arduino_power` (βλέπε σελίδα 35 - 38).



Σχήμα 2.5.2.8: Βάση δεδομένων `arduino_light`.

2.6 Κώδικας PHP

Γράφτηκε κώδικας σε γλώσσα PHP ο οποίος είχε σκοπό να συμβαδίζει με τη βάση δεδομένων ως αποτέλεσμα του να επιτευχθεί η σωστή σύνδεση και επικοινωνία ανάμεσα στο Arduino και στη βάση δεδομένων. Για να συμβεί αυτό έπρεπε να τροποποιηθεί κατάλληλα το αρχείο `upload_arduino.php`.



Σχήμα 2.6.1: Τοποθεσία αρχείου `upload_arduino.php`.

Ολοκληρωμένος κώδικας upload_arduino.php.

```
<?php

$con = mysql_connect('127.0.0.1','arduino_power','power');

if (!$con)

{

    die('Could not connect: ' . mysql_error());

}

mysql_select_db("arduino_power", $con);

$unixtime_val=time();

$sql="INSERT INTO events (station_id, timestamp, light_intensity)

VALUES

('$_GET[station_id]', from_unixtime('$unixtime_val'), $_GET[light_intensity])";

if (!mysql_query($sql,$con))

{

    die('Error: ' . mysql_error());

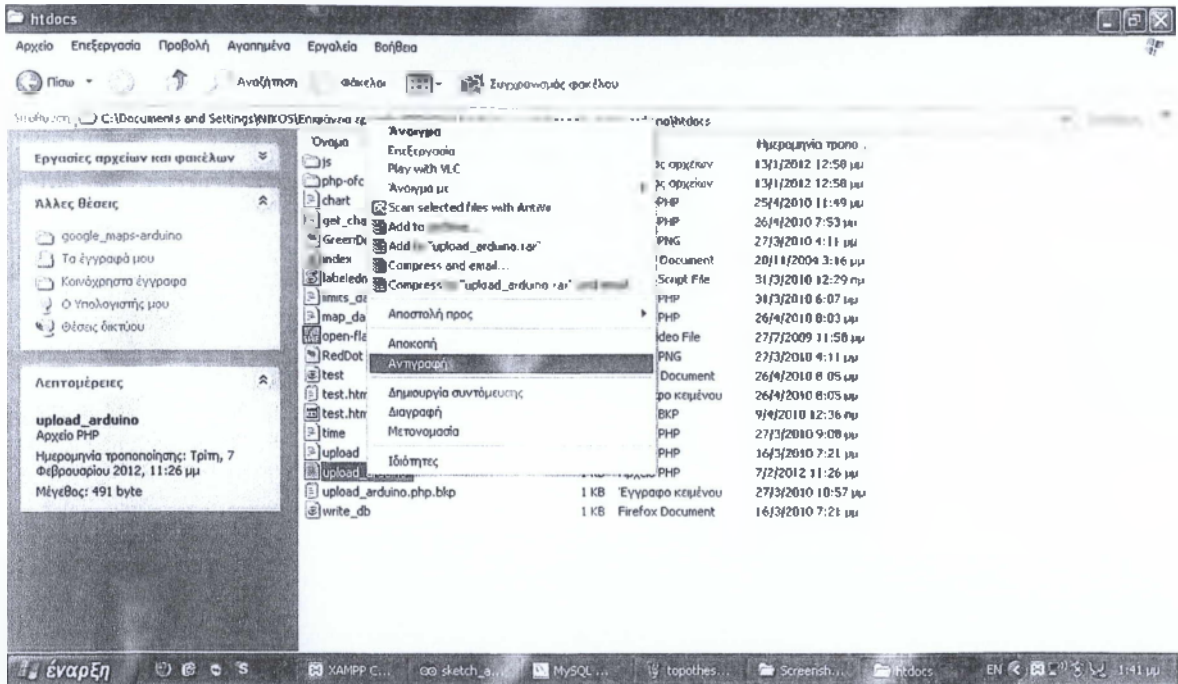
}

echo "1 record added";
```

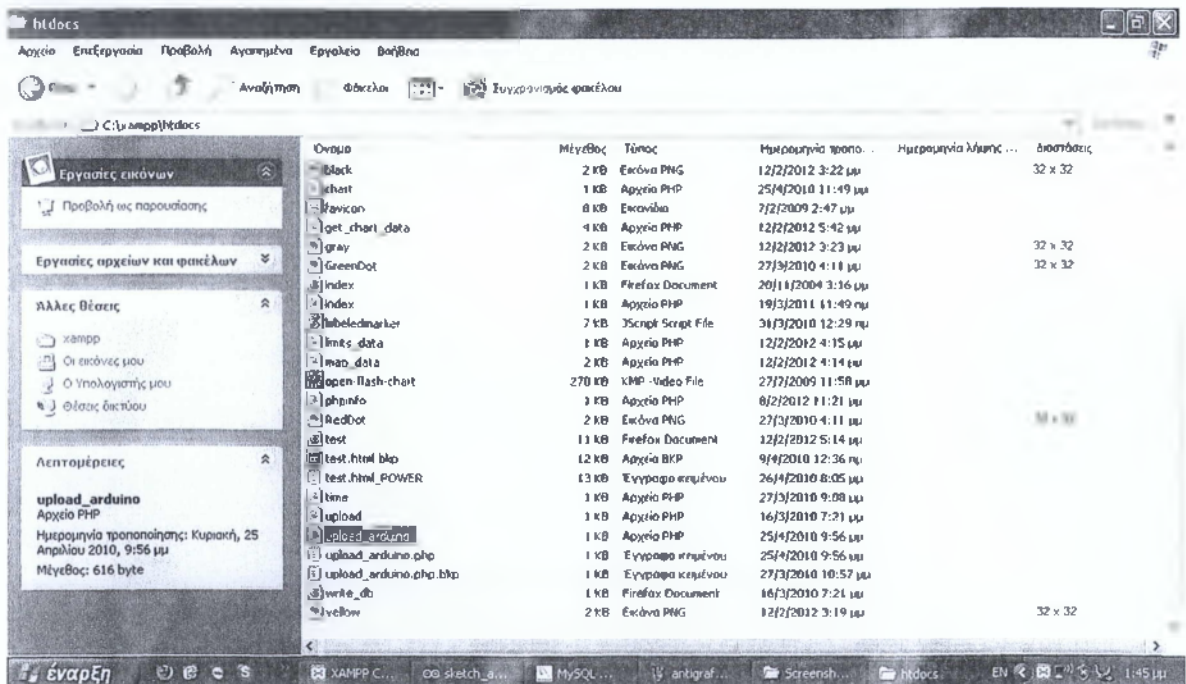
```
mysql_close($con)
```

```
?>
```

Στη συνέχεια χρειάστηκε το αρχείο αυτό να αντιγραφεί και να επικολληθεί - αντικατασταθεί μέσα στο φάκελο του XAMPP/htdocs ώστε να συμβαδίζει και με το XAMPP.



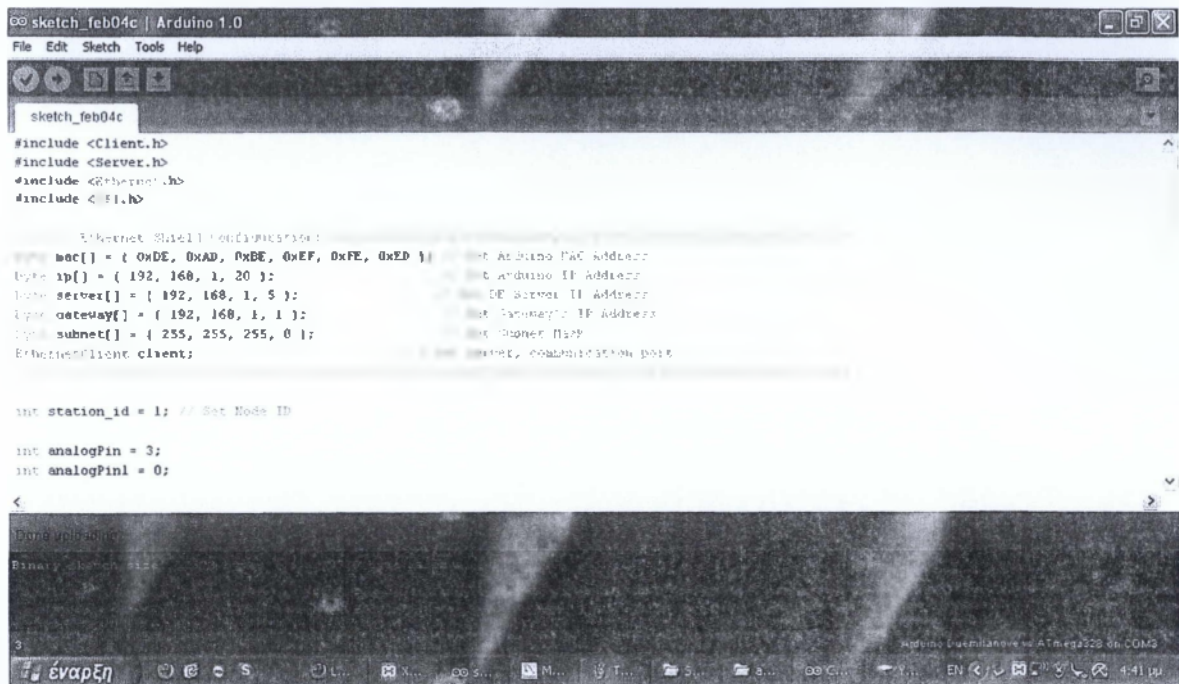
Σχήμα 2.6.2: Αντιγραφή αρχείου upload_arduino.php.



Σχήμα 2.6.3: Επικόλληση αρχείου upload_arduino.php στο XAMPP\htdocs.

2.6.1 Κώδικας για το Arduino

Γράφτηκε κώδικας στο περιβάλλον του Arduino ο οποίος κατέγραφε τα δεδομένα από το φωτοκύτταρο και τα έστειλε στην βάση δεδομένων.



```
sketch_feb04c
#include <Client.h>
#include <Server.h>
#include <Ethernet.h>
#include <SPI.h>

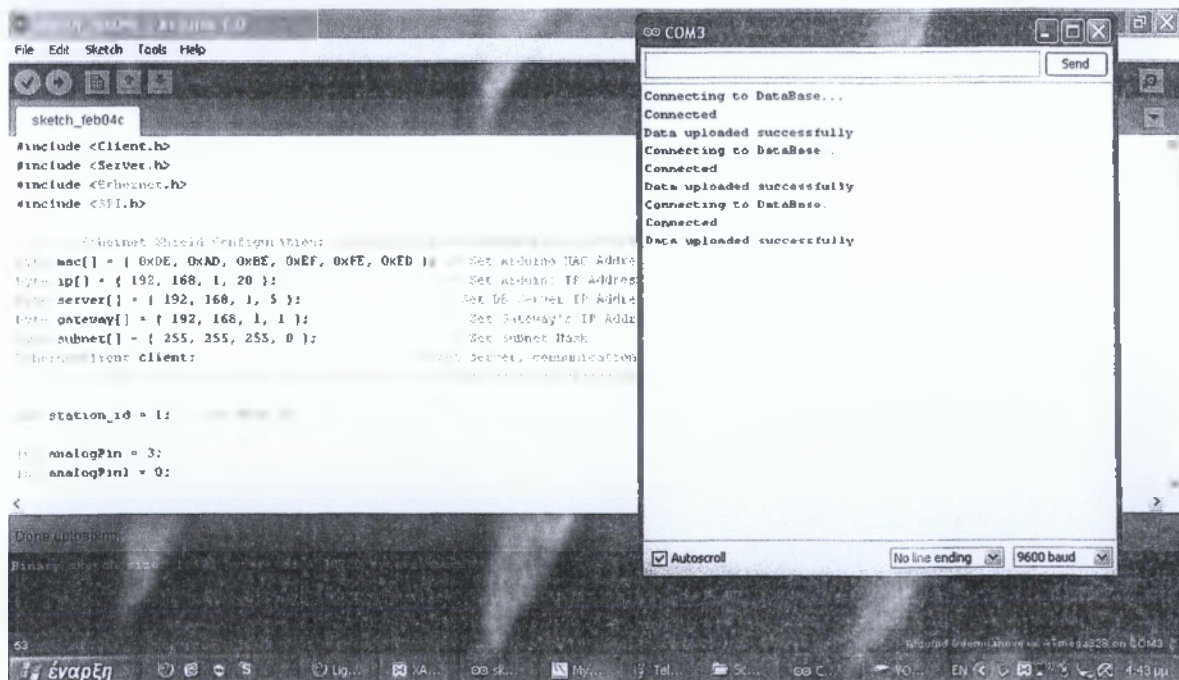
// Ethernet Shield Configuration
const byte mac[] = { 0x0E, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E }; // Set Arduino MAC Address
byte ip[] = { 192, 168, 1, 20 }; // Set Arduino IP Address
byte server[] = { 192, 168, 1, 5 }; // Set Server IP Address
byte gateway[] = { 192, 168, 1, 1 }; // Set Gateway IP Address
byte subnet[] = { 255, 255, 255, 0 }; // Set Subnet Mask

EthernetClient client;

int station_id = 1; // Set Node ID

int analogPin = 3;
int analogPinI = 0;
```

Σχήμα 2.6.1.1: Κώδικας καταγραφής δεδομένων (Arduino) – μεταφορά στη βάση.



```
sketch_feb04c
#include <Client.h>
#include <Server.h>
#include <Ethernet.h>
#include <SPI.h>

// Ethernet Shield Configuration
const byte mac[] = { 0x0E, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E }; // Set Arduino MAC Address
byte ip[] = { 192, 168, 1, 20 }; // Set Arduino IP Address
byte server[] = { 192, 168, 1, 5 }; // Set Server IP Address
byte gateway[] = { 192, 168, 1, 1 }; // Set Gateway IP Address
byte subnet[] = { 255, 255, 255, 0 }; // Set Subnet Mask

EthernetClient client;

int station_id = 1;

int analogPin = 3;
int analogPinI = 0;
```

```
Serial Monitor
Connecting to DataBase...
Connected
Data uploaded successfully
Connecting to DataBase...
Connected
Data uploaded successfully
Connecting to DataBase...
Connected
Data uploaded successfully
Autoscroll No line ending 9600 baud
```

Σχήμα 2.6.1.2: Σειριακή οθόνη (Serial Monitor) Arduino, επιτυχής αποστολή δεδομένων.

Ολοκληρωμένος κώδικας (sketch_jan13c):

```
#include <Client.h>

#include <Server.h>

#include <Ethernet.h>

#include <SPI.h>

//////// Ethernet Shield Configuration: //////////////////////////////////////

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // Set Arduino MAC Address

byte ip[] = { 192, 168, 1, 20 }; // Set Arduino IP Address

byte server[] = { 192, 168, 1, 5 }; // Set DB Server IP Address

byte gateway[] = { 192, 168, 1, 1 }; // Set Gateway's IP Address

byte subnet[] = { 255, 255, 255, 0 }; // Set Subnet Mask

EthernetClient client; // Set server, communication port

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

int station_id = 1; // Set Node ID

int analogPin = 3;

int analogPin1 = 0;

// Data upload parameters:

unsigned long time; // Stores time elapsed since power-up (in milliseconds)
```



```
unsigned long upload_interval=60000; // time interval between sending data to the DB
server (in milliseconds)
```

```
unsigned long last_ul=0; // time of last upload event (in milliseconds)
```

```
float random_light_intensity;
```

```
void setup()
```

```
{
```

```
  Ethernet.begin(mac, ip, gateway, subnet);
```

```
  Serial.begin(9600);
```

```
  randomSeed(analogRead(analogPin1));
```

```
}
```

```
void loop()
```

```
{
```

```
  for (int i=0; i < 500; i++) //Loop for calculations (one per second)
```

```
  {
```

```
    unsigned long start_time=micros();
```

```
    //Write data to DB server:
```

```
    time=millis(); //get current time (milliseconds since power-up)
```

```
    if (last_ul>time) last_ul=time; //avoid time overflow
```

```
    if ((time-last_ul)>upload_interval) //ensures uploading at specified intervals
```

```

{
  Serial.println("Connecting to DataBase...");

  client.connect(server, 80);

  if (client.connected() == true) {

    Serial.println("Connected");

    client.print("GET /upload_arduino.php?station_id=");

    client.print(station_id);

    random_light_intensity = (float (random (0,200)-100))/100;

    client.print("&light_intensity=");

    client.print(random_light_intensity);

    client.println(" HTTP/1.0");

    client.println();

    client.flush();

    client.stop();

    Serial.println("Data uploaded successfully");

    last_ul=millis(); //remember time of current upload

  } else {

    Serial.println("Connection to DataBase Failed");

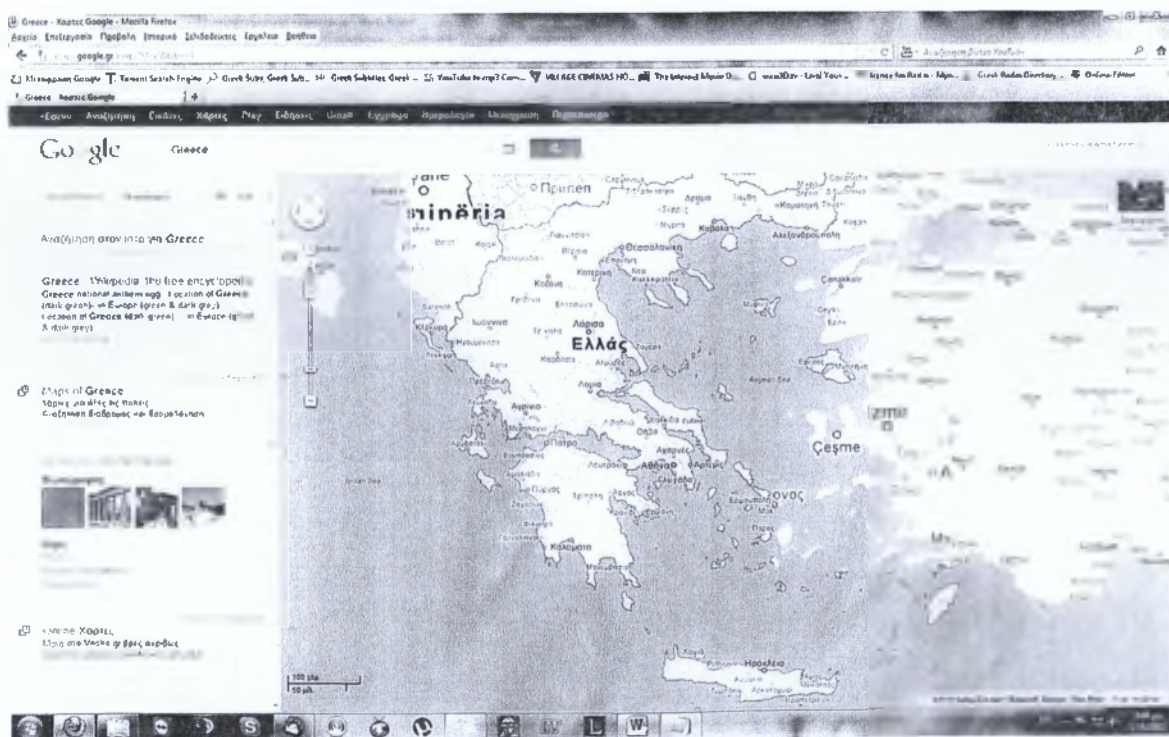
  }

}
}
}

```

2.6.2 Google Maps και Open Flash Chart

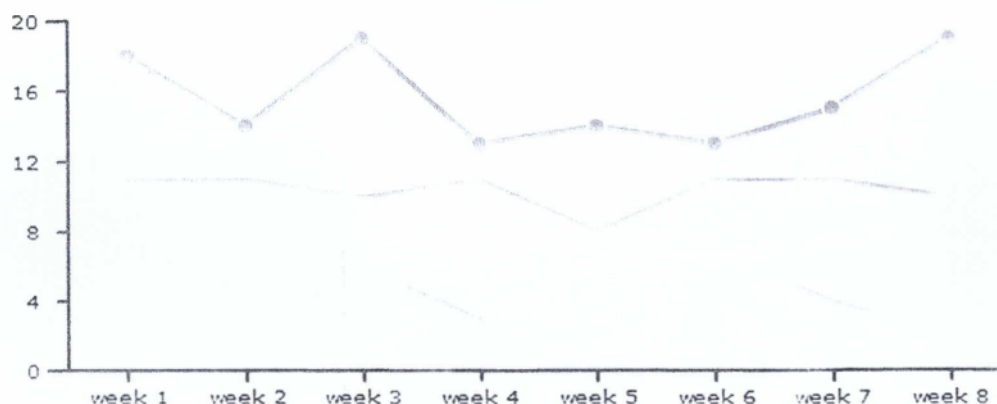
Τα Google Maps είναι μια υπηρεσία της Google που προσφέρει ισχυρή, εύχρηστη τεχνολογία χαρτογράφησης και πληροφορίες για τοπικές επιχειρήσεις, όπως η ακριβής τοποθεσία, τα στοιχεία επικοινωνίας και καθοδήγηση για τον προορισμό. Με τους Χάρτες Google, οι χρήστες μπορούν να βρίσκουν τις τοποθεσίες επιχειρήσεων και τα στοιχεία επικοινωνίας σε μία θέση και ενσωματωμένα στο χάρτη. Για παράδειγμα, εάν ένας χρήστης πραγματοποιήσει αναζήτηση για αεροδρόμια, στο χάρτη θα εμφανιστούν οι συναφείς καταχωρίσεις και οι τοποθεσίες αυτών. Ο χρήστης μπορεί να προβάλλει επίσης πρόσθετες πληροφορίες, όπως τις ώρες των πτήσεων, τους αποδεκτούς τύπους πληρωμής και τις κριτικές για τις διάφορες εταιρείες των αερογραμμών.



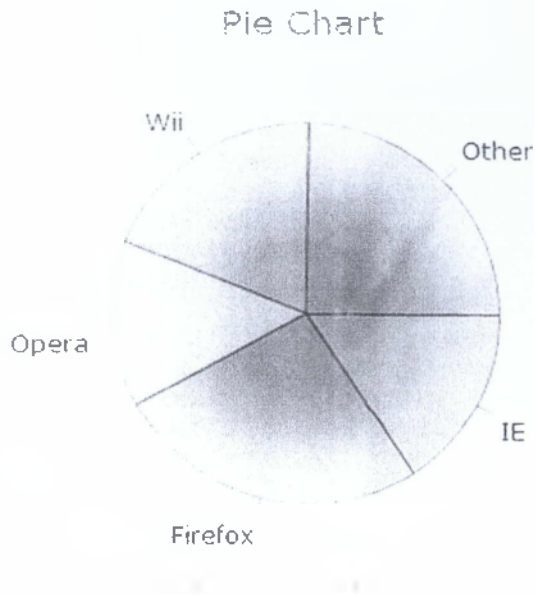
Σχήμα 2.6.2.1: Google Maps, εμφάνιση της Ελλάδας.

Το Open Flash Chart χρησιμοποιείται για να αναπαριστώνται κάποια δεδομένα (που βρίσκονται κάπου αποθηκευμένα) σε ένα γράφημα οποιουδήποτε τύπου. Όταν ένας χρήστης μεταβεί σε μια ιστοσελίδα τότε ο φυλλομετρητής «κατεβάζει» την ιστοσελίδα που περιέχει το Open Flash Chart και τέλος, το Open Flash Chart «κατεβάζει» το αρχείο δεδομένων και εμφανίζει το γράφημα. Πιο αναλυτικά: Όταν γίνει προσθήκη του Open Flash Chart στην ιστοσελίδα τότε ταυτόχρονα ορίζεται και η διαδρομή από την οποία θα παρθεί το αρχείο με τα δεδομένα. Όταν λοιπόν ο χρήστης κατεβάζει την ιστοσελίδα, το Open Flash Chart ζητάει τα διαγράμματα δεδομένων από τον διακομιστή. Ο διακομιστής ξέρει ποιος είναι ο χρήστης και έτσι μπορεί να δημιουργήσει ένα διάγραμμα για το συγκεκριμένο χρήστη που προβάλλει την συγκεκριμένη χρονική στιγμή κάποια δεδομένα.

Open Flash Chart



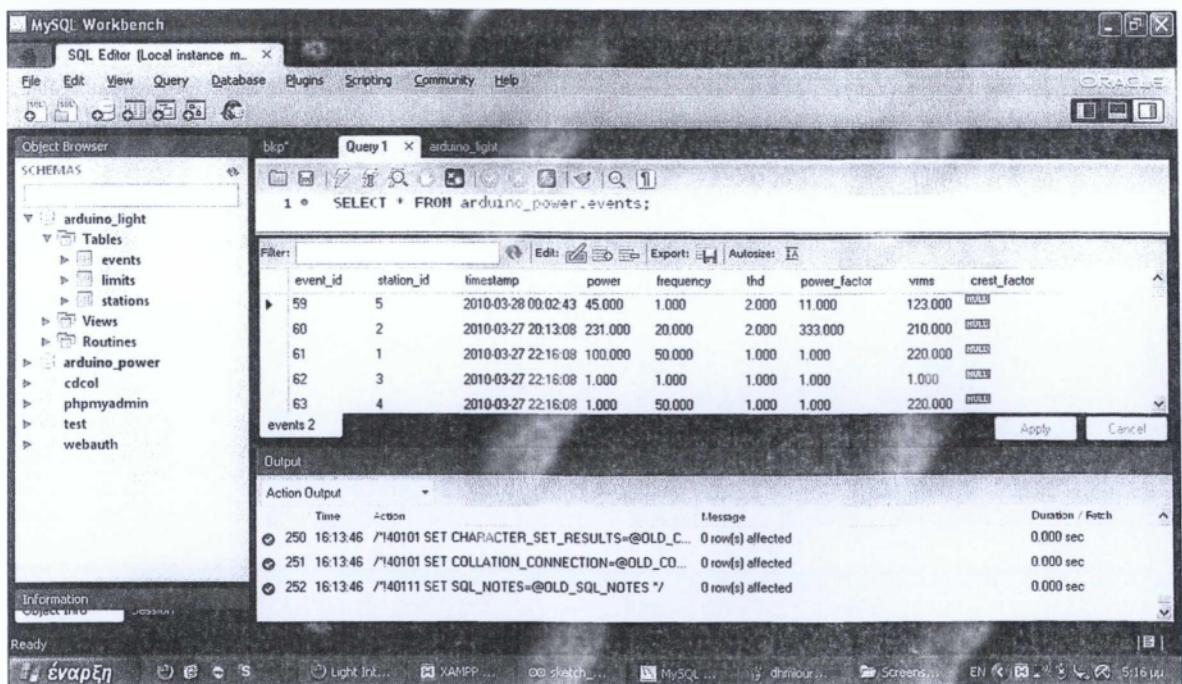
Σχήμα 2.6.2.2: 1^ο παράδειγμα διαγράμματος που χρησιμοποιεί το Open Flash Chart.



Σχήμα 2.6.2.3: 2^ο παράδειγμα διαγράμματος, γνωστό και ως «πίτα».

Έπειτα τοποθετήθηκαν κάποια αρχεία μέσα στον φάκελο `htdocs` του XAMPP (για τη σωστή αναπαράσταση των δεδομένων μας που θα δούμε παρακάτω).

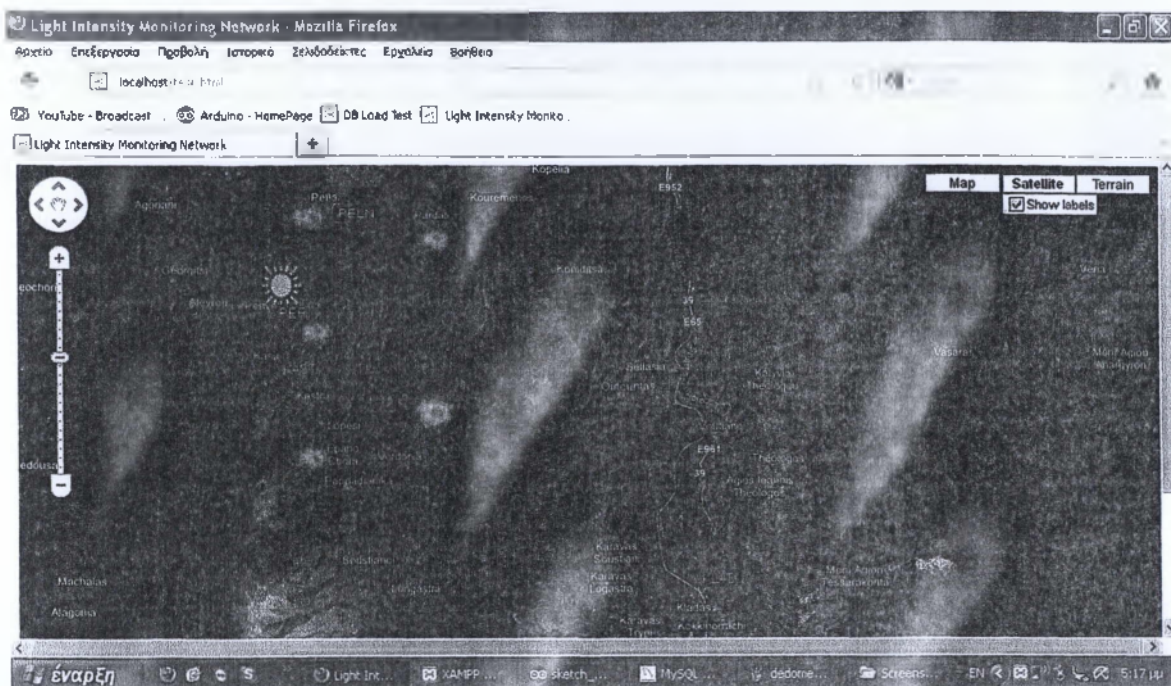
Εκτελέστηκε (πατώντας στον κίτρινο κεραυνό) η βάση `arduino_light` και εμφανιστήκαν στο Query 1 τα αποτελέσματα των καταγραφών από το φωτοκύτταρο – Arduino.



Σχήμα 2.6.2.4: Δεδομένα που καταγράφονται στη βάση.

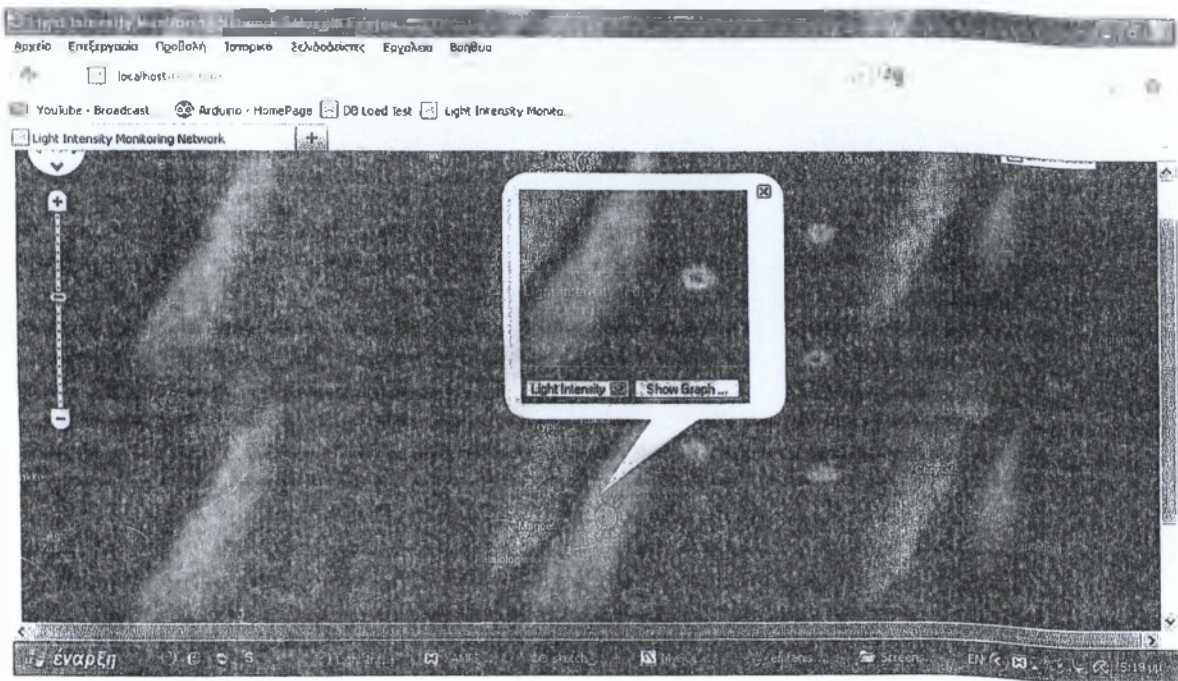
Συλλογή, αποθήκευση, επεξεργασία και απεικόνιση μετρήσεων ηλιοφάνειας σε πραγματικό χρόνο.

Ανοίγοντας τον φυλλομετρητή και πληκτρολογώντας <http://localhost/test.html> εμφανίζονται σε ένα χάρτη τα αποτελέσματα των καταγραφών της βάσης δεδομένων.

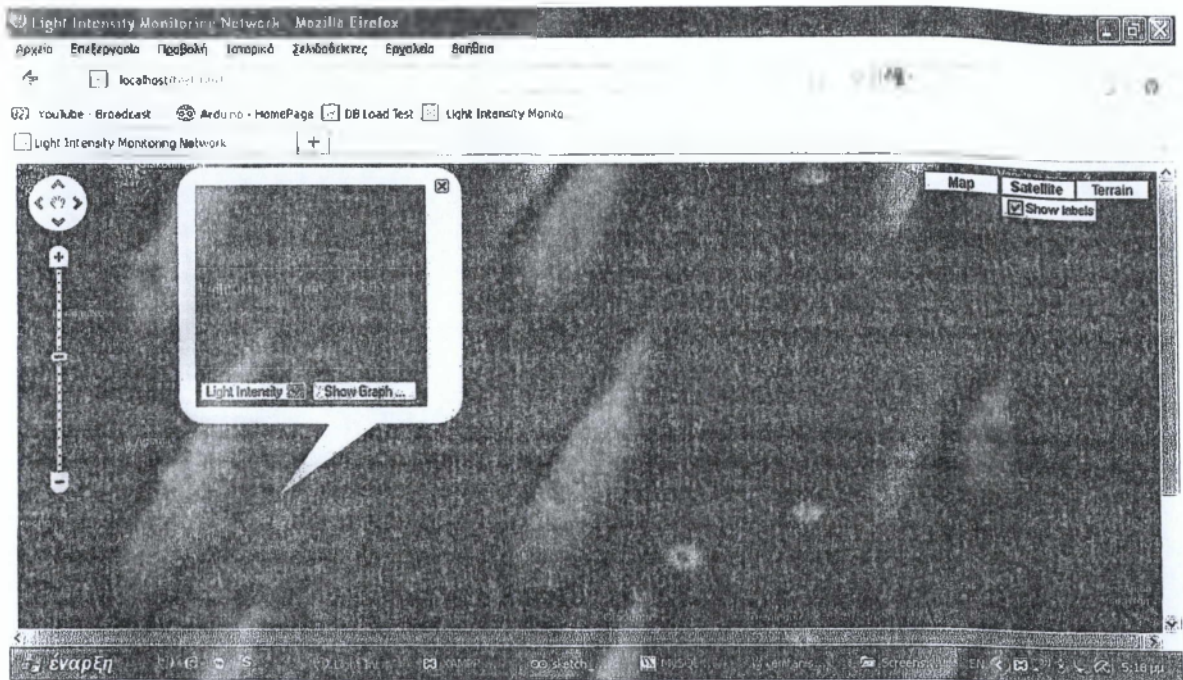


Σχήμα 2.6.2.5: Χάρτης απεικόνισης περιοχών στις οποίες γίνανε καταμετρήσεις της ηλιοφάνειας.

Εδώ φαίνεται ένα Google Map με markers που αντιστοιχούν σε κάποιες περιοχές που τοποθετήθηκε το Arduino ώστε να γίνει η δειγματοληψία. Κλικάροντας στο όνομα του κάθε σταθμού ανοίγει ένα παράθυρο που δείχνει διάφορες πληροφορίες για τον συγκεκριμένο σταθμό. Οι πληροφορίες αυτές είναι οι εξής: κωδικός σταθμού, όνομα σταθμού, τελευταία μέτρηση: ηλιοφάνεια, ημερομηνία και ώρα που έγινε η μέτρηση, ιστορικό, παράθυρο επιλογής μονάδας μέτρησης που προβάλλεται και τέλος, εμφάνιση γραφήματος.

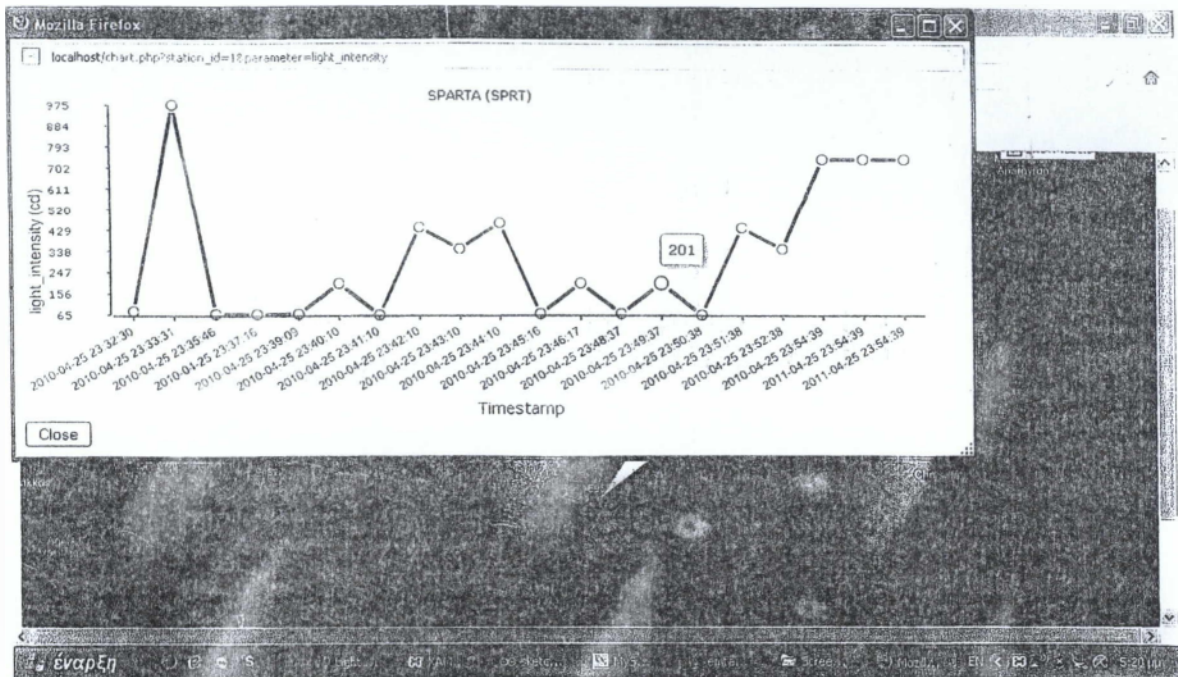


Σχήμα 2.6.2.6: Εμφάνιση πληροφοριών για την περιοχή της Σπάρτης.

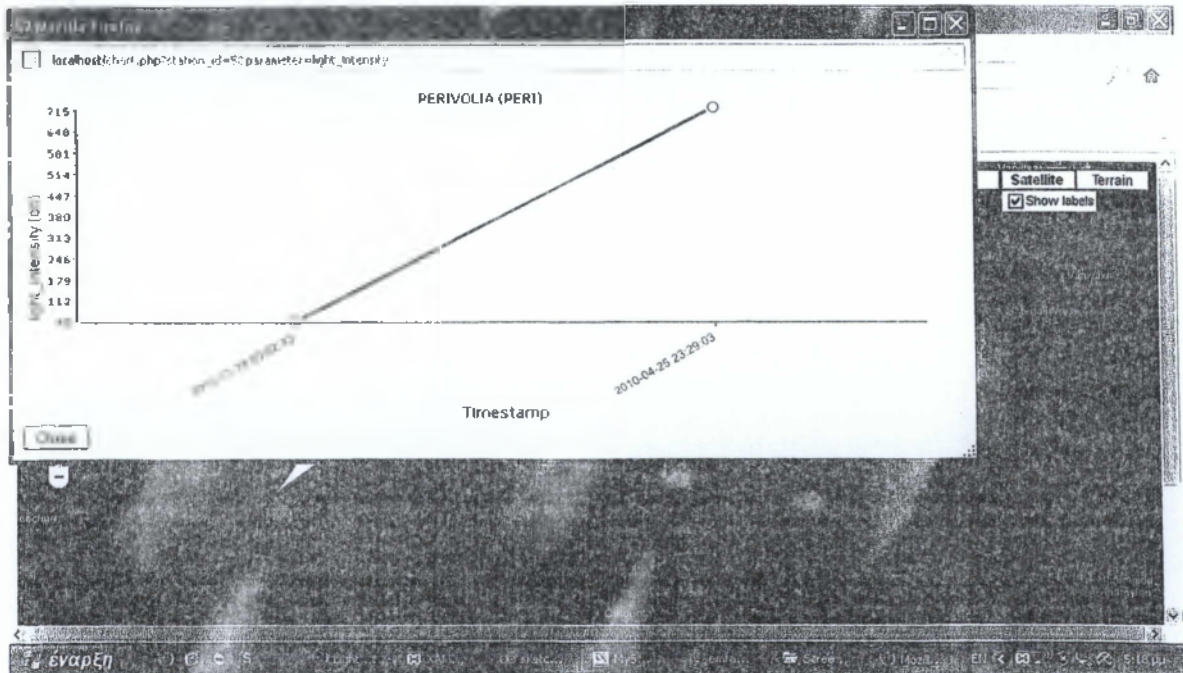


Σχήμα 2.6.2.7: Εμφάνιση πληροφοριών για την περιοχή των Περιβολιών.

Κλικάροντας στο «εμφάνιση γραφήματος» (Show Graph ...) εμφανίζεται ένα διάγραμμα με τις μετρήσεις που είναι αποθηκευμένες στη βάση.



Σχήμα 2.6.2.8: Εμφάνιση διαγράμματος για την περιοχή της Σπάρτης.



Σχήμα 2.6.2.9: Εμφάνιση διαγράμματος για την περιοχή των Περιβολιών

3. ΣΥΜΠΕΡΑΣΜΑΤΑ



3. ΣΥΜΠΕΡΑΣΜΑΤΑ

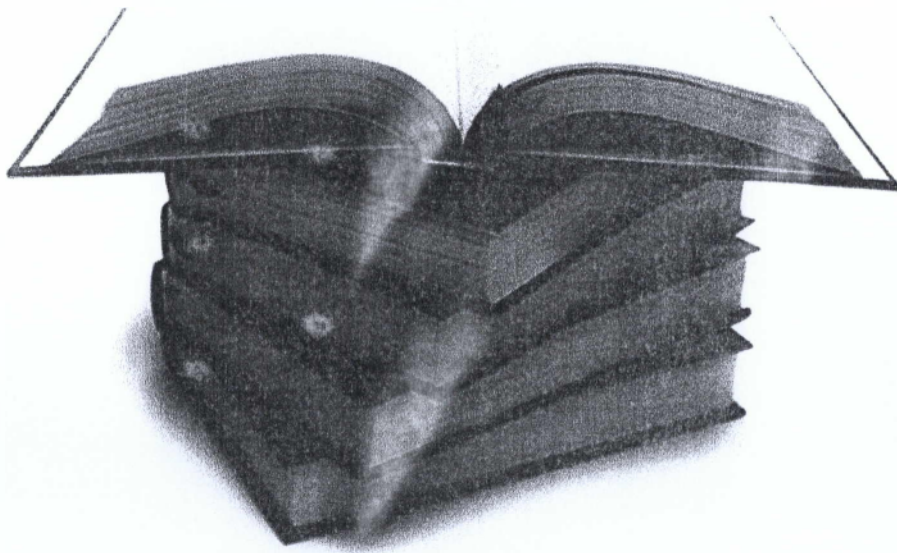
Από πλευράς software το σύστημα χρησιμοποιεί MySQL, PHP, json, xml, open flash chart, google maps. Κατόπιν συνεργασίας των ανωτέρω ολοκληρώθηκε και έφτασε εις πέρας αυτή η εργασία. Συνδυάζοντας software – hardware βγήκε το σωστό αποτέλεσμα.

Ξεκινώντας από το Arduino με τη σωστή συνδεσμολογία του με τα άλλα επιμέρους κομμάτια (φωτοκύτταρο - breadboard - modem - H/Y) καταφέραμε να συμπεράνουμε τον τρόπο με τον οποίο λειτουργεί το όλο σύστημα αυτό. Αυτό επιτυγχάνεται με το φωτοκύτταρο να καταγράφει την ηλιοφάνεια και να στέλνει πληροφορίες στο Arduino, το Arduino από εκεί με τη σειρά του να στέλνει (έχοντας γράψει τον σωστό κώδικα στο περιβάλλον του) τα δεδομένα αυτά στην βάση δεδομένων και από εκεί κατόπιν συνεννόησης με τον κώδικα (php) που είναι γραμμένος, να αναπαρίσταται σωστά το site με ένα χάρτη με τη μορφή των Google maps και του Open flash chart για τα αντίστοιχα διαγράμματα. Στον χάρτη εμφανίζονται οι περιοχές (markers) όπου συναρμολογήθηκε το σύστημα αυτό και γίνανε οι καταγραφές, πατώντας πάνω σε οποιαδήποτε από τις περιοχές αυτές εμφανίζονται τα δεδομένα και οι πληροφορίες του τόπου αυτού. Στα διαγράμματα παρουσιάζονται ορισμένες ημερομηνίες - ώρες που έχει γίνει η καταγραφή των δεδομένων συναρτήσει της ηλιοφάνειας για μια συγκεκριμένη χρονική στιγμή.

Τέλος, το χρώμα κάθε marker αλλάζει αναλόγως των τιμών που είναι καταγεγραμμένες στην βάση, δηλαδή αν έχει ηλιοφάνεια είναι λαμπερός στο χάρτη ο ήλιος, αν δεν έχει ηλιοφάνεια ο ήλιος είναι πιο σκούρος.

Πιθανές εφαρμογές του συστήματος αυτού μπορούν να γίνουν στους εξής τομείς: στη γεωργία, στη μελέτη εγκατάστασης φωτοβολταϊκών, σε οργανισμούς που εξετάζουν την ημερήσια πρόγνωση του καιρού κ.ά.

4. ΠΑΡΑΡΤΗΜΑΤΑ



Ολοκληρωμένος κώδικας βάσης δεδομένων `arduino_light`:

```
CREATE DATABASE IF NOT EXISTS `arduino_light` /*!40100 DEFAULT
CHARACTER SET utf8 */;

USE `arduino_light`;

-- MySQL dump 10.13 Distrib 5.1.40, for Win32 (ia32)

--
-- Host: localhost Database: arduino_light
--
-----
-- Server version 5.1.50-community

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8 */;

/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;

/*!40103 SET TIME_ZONE='+00:00' */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;

/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
```

```
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
```

```
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

```
--  
-- Table structure for table `events`  
--
```

```
DROP TABLE IF EXISTS `events`;
```

```
/*!40101 SET @saved_cs_client = @@character_set_client */;
```

```
/*!40101 SET character_set_client = utf8 */;
```

```
CREATE TABLE `events` (  
  `event_id` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `station_id` varchar(4) CHARACTER SET utf8 DEFAULT NULL,  
  `timestamp` timestamp NULL DEFAULT NULL,  
  `light_intensity` decimal(10,3) DEFAULT NULL,  
  PRIMARY KEY (`event_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=132 DEFAULT CHARSET=latin1;
```

```
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--  
-- Dumping data for table `events`  
--
```

```
LOCK TABLES `events` WRITE;
```

```
/*!40000 ALTER TABLE `events` DISABLE KEYS */;
```

```
INSERT INTO `events` VALUES (59,'5','2010-03-27 22:02:43','45.000'),(60,'2','2010-03-27 18:13:08','231.000'),(61,'1','2010-03-27 20:16:08','1.000'),(62,'3','2010-03-27 20:16:08','1.000'),(63,'4','2010-03-27 21:02:43','100.000'),(64,'4','2010-04-06 21:02:53','60.000'),(65,'4','2010-04-06 21:03:03','70.000'),(66,'4','2010-04-06 21:03:13','100.000'),(67,'4','2010-04-06 21:03:13','100.000'),(68,'4','2010-04-07 21:03:23','50.000'),(69,'4','2010-04-07 18:58:13','46.000'),(70,'1','2010-04-25 18:59:13','199.000'),(71,'1','2010-04-25 19:00:13','493.000'),(72,'1','2010-04-25 19:01:14','633.000'),(73,'1','2010-04-25 19:02:14','262.000'),(74,'1','2010-04-25 19:03:14','152.000'),(75,'1','2010-04-25 19:04:14','818.000'),(76,'1','2010-04-25 19:05:14','146.000'),(77,'1','2010-04-25 19:06:15','915.000'),(78,'1','2010-04-25 19:07:15','352.000'),(79,'1','2010-04-25 19:09:05','853.000'),(80,'1','2010-04-25 19:10:31','81.000'),(81,'1','2010-04-25 19:11:31','976.000'),(82,'1','2010-04-25 19:12:31','202.000'),(83,'1','2010-04-25 19:13:31','706.000'),(84,'1','2010-04-25 19:15:37','695.000'),(85,'1','2010-04-25 19:16:38','475.000'),(86,'1','2010-04-25 19:17:56','888.000'),(87,'1','2010-04-25 19:18:56','402.000'),(88,'1','2010-04-25 19:19:57','132.000'),(89,'1','2010-04-25 19:21:42','274.000'),(90,'1','2010-04-25 19:22:42','550.000'),(91,'1','2010-04-25 19:23:43','919.000'),(92,'1','2010-04-25 19:25:08','660.000'),(93,'1','2010-04-25 19:26:08','51.000'),(94,'1','2010-04-25 19:27:08','706.000'),(95,'1','2010-04-25 20:29:03','721.000'),(96,'1','2010-04-25 20:30:03','814.000'),(97,'1','2010-04-25 20:31:04','385.000'),(98,'1','2010-04-25 20:32:30','81.000'),(99,'1','2010-04-25 20:33:31','976.000'),(100,'3','2010-04-25 20:34:31','202.000'),(101,'1','2010-04-25 20:35:46','70.000'),(102,'1','2010-04-25 20:37:16','70.000'),(103,'1','2010-04-25 20:39:09','70.000'),(104,'1','2010-04-25 20:40:10','201.000'),(105,'1','2010-04-25 20:41:10','65.000'),(106,'1','2010-04-25 20:42:10','440.000'),(107,'1','2010-04-25 20:43:10','348.000'),(108,'1','2010-04-25 20:44:10','459.000'),(109,'1','2010-04-25 20:45:16','70.000'),(110,'1','2010-04-25 20:46:17','201.000'),(111,'2','2010-04-25 20:47:17','65.000'),(112,'1','2010-04-25 20:48:37','70.000'),(113,'1','2010-04-25 20:49:37','201.000'),(114,'1','2010-04-25 20:50:38','65.000'),(115,'1','2010-04-25 20:50:38','65.000'),(116,'1','2010-04-25 20:50:38','65.000'),(117,'1','2010-04-25 20:50:38','65.000'),(118,'1','2010-04-25 20:50:38','65.000'),(119,'1','2010-04-25 20:50:38','65.000'),(120,'1','2010-04-25 20:50:38','65.000'),(121,'1','2010-04-25 20:50:38','65.000'),(122,'1','2010-04-25 20:50:38','65.000'),(123,'1','2010-04-25 20:50:38','65.000'),(124,'1','2010-04-25 20:50:38','65.000'),(125,'1','2010-04-25 20:50:38','65.000'),(126,'1','2010-04-25 20:50:38','65.000');
```



```

20:51:38','440.000'),(127,'1','2010-04-25      20:52:38','348.000'),(128,'4','2010-04-25
20:53:38','459.000'),(129,'1','2010-04-25      20:54:39','744.000'),(130,'1','2011-04-25
20:54:39','744.000'),(131,'1','2011-04-25 20:54:39','744.000');

```

```

/*!40000 ALTER TABLE `event:` ENABLE KEYS */;

```

```

UNLOCK TABLES;

```

```

--

```

```

-- Table structure for table `limits`

```

```

--

```

```

DROP TABLE IF EXISTS `limits`;

```

```

/*!40101 SET @saved_cs_client = @@character_set_client */;

```

```

/*!40101 SET character_set_client = utf8 */;

```

```

CREATE TABLE `limits` (

```

```

  `id` int(11) NOT NULL AUTO_INCREMENT,

```

```

  `parameter` varchar(45) DEFAULT NULL,

```

```

  `limit1` decimal(10,3) DEFAULT NULL,

```

```

  `limit2` decimal(10,3) DEFAULT NULL.

```

```

  PRIMARY KEY (`id`)

```

```

) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

```

```

/*!40101 SET character_set_client = @saved_cs_client */;

```

```

--

```

```
-- Dumping data for table `limits`
```

```
--
```

```
LOCK TABLES `limits` WRITE;
```

```
/*!40000 ALTER TABLE `limits` DISABLE KEYS */;
```

```
INSERT INTO `limits` VALUES (1,'light_intensity','200.000','500.000');
```

```
/*!40000 ALTER TABLE `limits` ENABLE KEYS */;
```

```
UNLOCK TABLES;
```

```
--
```

```
-- Table structure for table `stations`
```

```
--
```

```
DROP TABLE IF EXISTS `stations`;
```

```
/*!40101 SET @saved_cs_client = @@character_set_client */;
```

```
/*!40101 SET character_set_client = utf8 */;
```

```
CREATE TABLE `stations` (
```

```
  `station_id` int(11) NOT NULL,
```

```
  `station_code` varchar(4) CHARACTER SET utf8 DEFAULT NULL,
```

```
  `station_name` varchar(45) CHARACTER SET utf8 DEFAULT NULL,
```

```
  `station_latitude` decimal(8,5) DEFAULT NULL,
```

```
  `station_longitude` decimal(8,5) DEFAULT NULL,
```

```
  PRIMARY KEY (`station_id`)
```

```

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `stations`
--

LOCK TABLES `stations` WRITE;

/*!40000 ALTER TABLE `stations` DISABLE KEYS */;

INSERT INTO `stations` VALUES
(1,'SPRT','SPARTA','37.07627','22.42996'),(2,'KAST','KASTOREION','37.16798','22.305
16'),(3,'PELN','PELLANA','37.20545','22.32267'),(4,'KLAD','KLADAS','37.11790','22.431
16'),(5,'PERI','PERIVOLIA','37.18138','22.30345');

/*!40000 ALTER TABLE `stations` ENABLE KEYS */;

UNLOCK TABLES;

/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;

/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS
*/;

```

```

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION
*/;

/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2012-02-12 16:44:31

CREATE DATABASE IF NOT EXISTS `arduino_power` /*!40100 DEFAULT
CHARACTER SET utf8 */;

USE `arduino_power`;

-- MySQL dump 10.13 Distrib 5.1.40, for Win32 (ia32)

--
-- Host: localhost Database: arduino_power
-----
-- Server version 5.1.50-community

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
SET

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
SET

/*!40101 SET NAMES utf8 */;

/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
TIME_ZONE

/*!40103 SET TIME_ZONE='+00:00' */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;

```

```
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,  
FOREIGN_KEY_CHECKS=0 */;
```

```
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
```

```
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

```
--
```

```
-- Table structure for table `events`
```

```
--
```

```
DROP TABLE IF EXISTS `events`;
```

```
/*!40101 SET @saved_cs_client = @@character_set_client */;
```

```
/*!40101 SET character_set_client = utf8 */;
```

```
CREATE TABLE `events` (  
  `event_id` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `station_id` varchar(4) CHARACTER SET utf8 DEFAULT NULL,  
  `timestamp` timestamp NULL DEFAULT NULL,  
  `power` decimal(10,3) DEFAULT NULL,  
  `frequency` decimal(10,3) DEFAULT NULL,  
  `thd` decimal(10,3) DEFAULT NULL,  
  `power_factor` decimal(10,3) DEFAULT NULL,  
  `vrms` decimal(10,3) DEFAULT NULL,  
  `crest_factor` decimal(10,3) DEFAULT NULL,  
  PRIMARY KEY (`event_id`),  
  KEY `station_id` (`station_id`),  
  KEY `timestamp` (`timestamp`),  
  KEY `power` (`power`),  
  KEY `frequency` (`frequency`),  
  KEY `thd` (`thd`),  
  KEY `power_factor` (`power_factor`),  
  KEY `vrms` (`vrms`),  
  KEY `crest_factor` (`crest_factor`)
```

```

PRIMARY KEY (`event_id`)

) ENGINE=InnoDB AUTO_INCREMENT=132 DEFAULT CHARSET=latin1;

/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `events`
--

LOCK TABLES `events` WRITE;

/*!40000 ALTER TABLE `events` DISABLE KEYS */;

INSERT INTO `events` VALUES (59,'5','2010-03-27
22:02:43','45.000','1.000','2.000','11.000','123.000',NULL),(60,'2','2010-03-27
18:13:08','231.000','20.000','2.000','333.000','210.000',NULL),(61,'1','2010-03-27
20:16:08','100.000','50.000','1.000','1.000','220.000',NULL),(62,'3','2010-03-27
20:16:08','1.000','1.000','1.000','1.000','1.000',NULL),(63,'4','2010-03-27
20:16:08','1.000','50.000','1.000','1.000','220.000',NULL),(64,'4','2010-04-06
21:02:43','100.000','50.000','2.000','200.000','211.000',NULL),(65,'4','2010-04-06
21:02:53','60.000','55.000','3.000','1.000','216.000',NULL),(66,'4','2010-04-06
21:03:03','70.000','49.000','1.500','0.000','221.000',NULL),(67,'4','2010-04-06
21:03:13','200.000','52.000','0.000','0.000','210.000',NULL),(68,'4','2010-04-07
21:03:13','100.000','48.000','1.000','-1.000','205.000',NULL),(69,'4','2010-04-07
21:03:23','50.000','53.000','0.900','-0.500','213.000',NULL),(78,'1','2010-04-25
18:45:38',NULL,'49.000','50.050',NULL,NULL,NULL),(79,'1','2010-04-25
18:46:38',NULL,'49.000','50.050',NULL,NULL,NULL),(80,'1','2010-04-25
18:58:13','46.000','49.000','50.050','0.000','19.690','1.880'),(81,'1','2010-04-25
18:59:13','199.000','49.000','50.050','0.000','19.690','1.880'),(82,'1','2010-04-25
19:00:13','493.000','49.000','50.050','-1.000','19.690','1.880'),(83,'1','2010-04-25
19:01:14','633.000','49.000','50.050','-1.000','19.690','1.880'),(84,'1','2010-04-25
19:02:14','262.000','49.000','50.050','0.000','19.690','1.880'),(85,'1','2010-04-25

```

19:03:14', '152.000', '49.000', '50.050', '-1.000', '19.690', '1.880'), (86, '1', '2010-04-25
19:04:14', '818.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (87, '1', '2010-04-25
19:05:14', '146.000', '49.000', '50.050', '-1.000', '19.690', '1.880'), (88, '1', '2010-04-25
19:06:15', '915.000', '49.000', '50.050', '-1.000', '19.690', '1.880'), (89, '1', '2010-04-25
19:07:15', '352.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (90, '1', '2010-04-25
19:09:05', '853.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (91, '1', '2010-04-25
19:10:31', '81.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (92, '1', '2010-04-25
19:11:31', '976.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (93, '1', '2010-04-25
19:12:31', '202.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (94, '1', '2010-04-25
19:13:31', '706.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (95, '1', '2010-04-25
19:15:37', '695.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (96, '1', '2010-04-25
19:16:38', '475.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (97, '1', '2010-04-25
19:17:56', '888.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (98, '1', '2010-04-25
19:18:56', '402.000', '49.000', '50.050', '-1.000', '19.690', '1.880'), (99, '1', '2010-04-25
19:19:57', '132.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (100, '1', '2010-04-25
19:21:42', '274.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (101, '1', '2010-04-25
19:22:42', '550.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (102, '1', '2010-04-25
19:23:43', '919.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (103, '1', '2010-04-25
19:25:08', '660.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (104, '1', '2010-04-25
19:26:08', '51.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (105, '1', '2010-04-25
19:27:08', '706.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (106, '1', '2010-04-25
20:29:03', '721.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (107, '1', '2010-04-25
20:30:03', '814.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (108, '1', '2010-04-25
20:31:04', '385.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (109, '1', '2010-04-25
20:32:30', '81.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (110, '1', '2010-04-25
20:33:31', '976.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (111, '1', '2010-04-25
20:34:31', '202.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (112, '1', '2010-04-25
20:35:46', '70.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (113, '1', '2010-04-25
20:37:16', '70.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (114, '1', '2010-04-25
20:39:09', '70.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (115, '1', '2010-04-25
20:40:10', '201.000', '49.000', '50.050', '1.000', '19.690', '1.880'), (116, '1', '2010-04-25
20:41:10', '65.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (117, '1', '2010-04-25
20:42:10', '440.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (118, '1', '2010-04-25
20:43:10', '348.000', '49.000', '50.050', '0.000', '19.690', '1.880'), (119, '1', '2010-04-25

```

20:44:10','459.000','49.000','50.050','0.000','19.690','1.880'),(120,'1','2010-04-25
20:45:16','70.000','49.000','50.050','0.000','19.690','1.880'),(121,'1','2010-04-25
20:46:17','201.000','49.000','50.050','1.000','19.690','1.880'),(122,'1','2010-04-25
20:47:17','65.000','49.000','50.050','0.000','19.690','1.880'),(123,'1','2010-04-25
20:48:37','70.000','49.000','50.050','-0.570','19.690','1.880'),(124,'1','2010-04-25
20:49:37','201.000','49.000','50.050','0.920','19.690','1.880'),(125,'1','2010-04-25
20:50:38','65.000','49.000','50.050','-0.740','19.690','1.880'),(126,'1','2010-04-25
20:51:38','440.000','49.000','50.050','-0.020','19.690','1.880'),(127,'1','2010-04-25
20:52:38','348.000','49.000','50.050','-0.330','19.690','1.880'),(128,'1','2010-04-25
20:53:38','459.000','49.000','50.050','-0.850','19.690','1.880'),(129,'1','2010-04-25
20:54:39','744.000','49.000','50.050','-0.800','19.690','1.880'),(130,'1','2011-04-25
20:54:39','744.000','999.000','999.000','-0.800','19.690','1.880'),(131,'1','2011-04-25
20:54:39','744.000','999.990','999.000','-0.800','19.690','1.880');

```

```

/*!40000 ALTER TABLE `events` ENABLE KEYS */;

```

```

UNLOCK TABLES;

```

```

-- Table structure for table `limits`

```

```

DROP TABLE IF EXISTS `limits`;

```

```

/*!40101 SET @saved_cs_client = @@character_set_client */;

```

```

/*!40101 SET character_set_client = utf8 */;

```

```

CREATE TABLE `limits` (

```

```

  `id` int(11) NOT NULL AUTO_INCREMENT,

```

```

  `parameter` varchar(45) DEFAULT NULL,

```

```

  `min` decimal(10,3) DEFAULT NULL,

```



```

`max` decimal(10,3) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `limits`
--

LOCK TABLES `limits` WRITE;

/*!40000 ALTER TABLE `limits` DISABLE KEYS */;

INSERT INTO `limits` VALUES
(1,'frequency','49.000','51.000'),(2,'vrms','215.000','225.000'),(3,'power','0.000','1000.000'),(
4,'power_factor','0.000','1000.000'),(5,'thd','0.000','10.000');

/*!40000 ALTER TABLE `limits` ENABLE KEYS */;

UNLOCK TABLES;

--
-- Table structure for table `stations`
--

DROP TABLE IF EXISTS `stations`;

/*!40101 SET @saved_cs_client = @@character_set_client */;

```

```

/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `stations` (
  `station_id` int(11) NOT NULL,
  `station_code` varchar(4) CHARACTER SET utf8 DEFAULT NULL,
  `station_name` varchar(45) CHARACTER SET utf8 DEFAULT NULL,
  `station_latitude` decimal(8,5) DEFAULT NULL,
  `station_longitude` decimal(8,5) DEFAULT NULL,
  PRIMARY KEY (`station_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `stations`
--

LOCK TABLES `stations` WRITE;

/*!40000 ALTER TABLE `stations` DISABLE KEYS */;

INSERT INTO `stations` VALUES
(1,'SPRT','SPARTA','37.07627','22.42996'),(2,'KAST','KASTOREION','37.16798','22.305
16'),(3,'PELN','PELLANA','37.20545','22.32267'),(4,'KLAD','KLADAS','37.11790','22.431
16'),(5,'PERI','PERIVOLIA','37.18138','22.30345');

/*!40000 ALTER TABLE `stations` ENABLE KEYS */;

UNLOCK TABLES;

/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

```

```
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;

/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS
*/;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION
*/;

/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2012-02-12 16:44:32
```

κώδικας του αρχείου test.html για να μπορεί να εμφανίζεται σωστά το site φαίνεται παρακάτω:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-
com:vml">
  <head>
    <style type="text/css">
div.LabeledMarker_markerLabel {
  background-color:blue;
  color: white;
  font-family: Arial;
  font-size: 12px;
  width: 40px;
  text-align: center;
}
</style>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Power-Quality Monitoring Network</title>
```

```
<script type="text/javascript">

function submitmyform(f) {

f.target = 'foo';

window.open(",f.target,'menubar=no,scrollbars=no,resizeable=no,          toolbar=no,
location=no, directories=no, status=no, width=840, height=340');

f.submit();

return false;

}

</script>
```

```
<script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAAzr2EBOX
UKnm_jVnk0OJl7xSosDVG8KKPEl-m51RBrvYughuyMxQ-
i1QfUnH94QxWIa6N4U6MouMmBA&hl=en"

type="text/javascript"></script>
```

```
<script src="labeledmarker.js"></script>
```

```
<script type="text/javascript">
```

```
var map;  
  
var markerArray=[];  
  
var redIcon;  
  
var greenIcon;  
  
var stationsArray= [];  
  
var codesArray=[];  
  
var namesArray=[];  
  
var frequencyArray=[];  
  
var vrmsArray=[];  
  
var thdArray=[];  
  
var powerArray=[];  
  
var power_factorArray=[];  
  
var crest_factorArray=[];  
  
var timestampArray=[];  
  
var minlimitsArray=[];  
  
var maxlimitsArray=[];  
  
var htmlArray=[];  
  
var timeout_timer;  
  
  
var ajaxRequest_time;  
  
var ajaxRequest_limits;  
  
var ajaxRequest_station_data;
```

```

function refresh_time()
{
    if(ajaxRequest_time.readyState == 4){
        //document.controlForm.time.value
        ajaxRequest_time.responseText;

        document.getElementById("time_txt").innerHTML=ajaxRequest_time.responseText;

        //alert(ajaxRequest_time.responseText);
    }
}

```

```

function check_limits(value, min, max)
{
    out_of_limits=false;

    if ((parseFloat(value)>parseFloat(max)) || (parseFloat(value)<parseFloat(min))){
        out_of_limits=true;
    };

    return out_of_limits;
}

```

```

function refresh_limits()
{
    if(ajaxRequest_limits.readyState == 4){
        for (var i = 0; i <= 4; i++) {

            minlimitsArray[i]=ajaxRequest_limits.responseXML.getElementsByTagName("min")[i].childNodes[0].nodeValue;

            maxlimitsArray[i]=ajaxRequest_limits.responseXML.getElementsByTagName("max")[i].childNodes[0].nodeValue;

        };
    };
}

function refresh_data()
{
    // Exit function if infowindow is open
    if (map.getInfoWindow().isHidden()===false) {return;};

    var color_marker_red;

```



```

        if ((ajaxRequest_station_data.readyState == 4) &&
ajaxRequest_limits.readyState == 4){

        // Add markers to the map

        for (var i = 0; i <=
ajaxRequest_station_data.responseXML.getElementsByTagName("id").length-1; i++) {

            //Remove existing markers

            if (markerArray[i]!=null)
{map.removeOverlay(markerArray[i]);}

            color_marker_red=0;

            stationsArray[i] = new
GLatLng(ajaxRequest_station_data.responseXML.getElementsByTagName("lat")[i].child
Nodes[0].nodeValue
,
ajaxRequest_station_data.responseXML.getElementsByTagName("long")[i].childNodes[0
].nodeValue);

```

```
codesArray[i]=ajaxRequest_station_data.responseXML.getElementsByTagName("code")[i].childNodes[0].nodeValue;
```

```
namesArray[i]=ajaxRequest_station_data.responseXML.getElementsByTagName("name")[i].childNodes[0].nodeValue;
```

```
frequencyArray[i]=ajaxRequest_station_data.responseXML.getElementsByTagName("frequency")[i].childNodes[0].nodeValue;
```

```
vrmsArray[i]=ajaxRequest_station_data.responseXML.getElementsByTagName("vrms")[i].childNodes[0].nodeValue;
```

```
powerArray[i]=ajaxRequest_station_data.responseXML.getElementsByTagName("power")[i].childNodes[0].nodeValue;
```

```
power_factorArray[i]=ajaxRequest_station_data.responseXML.getElementsByTagName("power_factor")[i].childNodes[0].nodeValue;
```

```
crest_factorArray[i]=ajaxRequest_station_data.responseXML.getElementsByTagName("crest_factor")[i].childNodes[0].nodeValue;
```

```
thdArray[i]=ajaxRequest_station_data.responseXML.getElementsByTagName("thd")[i].childNodes[0].nodeValue;
```

```
timestampArray[i]=ajaxRequest_station_data.responseXML.getElementsByTagName("timestamp")[i].childNodes[0].nodeValue;
```

```

        htmlArray[i] ='<div style="width:200px; background-
color:#FFDEAD"> <table border="0"> <tr><td style="font-family:Arial; font-
size:11px;"> Station Code: ' + codesArray[i] + '</td></tr>'

        +   '<tr><td style="font-family:Arial; font-
size:11px;"> Station Name: ' + namesArray[i] + '</td></tr>';

        htmlArray[i]=htmlArray[i]+   '<tr><td style="font-
family:Arial; font-size:11px; font-style:italic;"> <br> Latest Measurements: <br>
</td></tr>';

        if(check_limits(frequencyArray[i],minlimitsArray[0],maxlimitsArray[0])==true){

                htmlArray[i]=htmlArray[i]+'<tr><td
style="font-family:Arial; font-size:11px; color:red; "> Frequency (Hz): ' +
frequencyArray[i] + '</td></tr>';

                color__marker_red=1;

        }

        else

        {

                htmlArray[i]=htmlArray[i]+'<tr><td
style="font-family:Arial; font-size:11px; "> Frequency (Hz): ' + frequencyArray[i] +
'</td></tr>';

        };

```

```

        if(check_limits(vrmsArray[i],minlimitsArray[1],maxlimitsArray[1])==true){

```

```

        htmlArray[i]=htmlArray[i]+'<tr><td
style="font-family:Arial; font-size:11px; color:red; "> Vrms (V): ' + vrmsArray[i] +
'</td></tr>';

```

```

        color_marker_red=1;

```

```

    }

```

```

    else

```

```

    {

```

```

        htmlArray[i]=htmlArray[i]+'<tr><td
style="font-family:Arial; font-size:11px; "> Vrms (V): ' + vrmsArray[i] + '</td></tr>';

```

```

    };

```

```

    if(check_limits(powerArray[i],minlimitsArray[2],maxlimitsArray[2])==true){

```

```

        htmlArray[i]=htmlArray[i]+'<tr><td
style="font-family:Arial; font-size:11px; color:red; "> Power (W): ' + powerArray[i] +
'</td></tr>';

```

```

        color_marker_red=1;

```

```

    }

```

```

    else

```

```

    {

```

```

        htmlArray[i]=htmlArray[i]+'<tr><td
style="font-family:Arial; font-size:11px; "> Power (W): ' + powerArray[i] + '</td></tr>';

```

```

    };

```

```

        if(check_limits(power_factorArray[i],minlimitsArray[3],maxlimitsArray[3])==true
    ){

                htmlArray[i]=htmlArray[i]+'<tr><td
style="font-family:Arial; font-size:11px; color:red; "> Power Factor: ' +
power_factorArray[i] + '</td></tr>';

                color_marker_red=1;

        }

        else

        {

                htmlArray[i]=htmlArray[i]+'<tr><td
style="font-family:Arial; font-size:11px; "> Power Factor: ' + power_factorArray[i] +
'</td></tr>';

        };

```

```

        if(check_limits(crest_factorArray[i],minlimitsArray[3],maxlimitsArray[3])==true){

                htmlArray[i]=htmlArray[i]+'<tr><td
style="font-family:Arial; font-size:11px; color:red; "> Crest Factor: ' +
crest_factorArray[i] + '</td></tr>';

                color_marker_red=1;

        }

        else

        {

```

```
htmlArray[i]=htmlArray[i]+'<tr><td  
style="font-family:Arial; font-size:11px; "> Crest Factor: ' + crest_factorArray[i] +  
'</td></tr>';
```

```
};
```

```
if(check_limits(thdArray[i],minlimitsArray[4],maxlimitsArray[4])==true){  
  
htmlArray[i]=htmlArray[i]+'<tr><td  
style="font-family:Arial; font-size:11px; color:red; "> THD: ' + thdArray[i] + '</td></tr>';
```

```
color_marker_red=1;
```

```
}
```

```
else
```

```
{
```

```
htmlArray[i]=htmlArray[i]+'<tr><td  
style="font-family:Arial; font-size:11px; "> THD: ' + thdArray[i] + '</td></tr>';
```

```
};
```

```
htmlArray[i]=htmlArray[i]+ ' <tr><td style="font-  
family:Arial; font-size:11px;"> Timestamp: ' + timestampArray[i] + '</td></tr>';
```

```
htmlArray[i]=htmlArray[i]+ '<tr><td> </td></tr>';
```

```
htmlArray[i]=htmlArray[i]+ '<tr><td style="font-  
family:Arial; font-size:11px; font-style:italic;"> <br> History: <hr> </td></tr> ' ;
```

```
htmlArray[i]=htmlArray[i]+ '<tr><td style="font-  
family:Arial; font-size:11px;"> <form action="chart.php" target="_blank"  
onsubmit="return submitmyform(this);" method="GET"> <input type="hidden"  
id="station_id" name="station_id" value="" + (i+1) + "" /> <select name="parameter"  
style="font-size:11px;"> <option value="frequency" >Frequency</option> <option  
value="vrms">Vrms</option> <option value="power">Power</option> <option  
value="power_factor">Power Factor </option> <option value="crest_factor">Crest Factor  
</option> <option value="thd">THD </option> </select> <input type="submit"  
value="Show Graph ..." style="font-size:11px;"/> </form> </td></tr> ' ;
```

```
htmlArray[i]=htmlArray[i]+ '</div>';
```

```
var
```

```
latlng=stationsArray[i];
```

```
markerOptions = {
```

```
    "icon": greenIcon,
```

```
    "clickable": true,
```

```
    "title": namesArray[i],
```

```

        "labelText": codesArray[i],
        "labelOffset": new GSize(0, 0)
    };

    if (color_marker_red==1) {
        markerOptions = {
            "icon": redIcon,
            "clickable": true,
            "title": namesArray[i],
            "labelText": codesArray[i],
            "labelOffset": new GSize(0, 0)
        };
    };

    markerArray[i] = createMarker(latlng, markerOptions,
htmlArray[i]);

    map.addOverlay(markerArray[i]);

}

```



```

    }

}

function createMarker(point,marker_options,html) {

    var marker = new LabeledMarker(point, marker_options);

    GEvent.addListener(marker, "click", function() {

        marker.openInfoWindowHtml(html);

    });

    return marker;

}

```

```

function lookForNewMarkers()

{

// code goes here to call server for new data

// add new data to the overlay

    ajaxRequest_time.open("GET", "time.php", true);

    ajaxRequest_limits.open("GET", "limits_data.php", true);

    ajaxRequest_station_data.open("GET", "map_data.php", true);

```

```
ajaxRequest_time.send(null);
```

```
ajaxRequest_limits.send(null);
```

```
ajaxRequest_station_data.send(null);
```

```
timeout_timer=setTimeout('lookForNewMarkers()',10000);
```

```
}
```

```
function initialize() {
```

```
if (GBrowserIsCompatible()) {
```

```
map = new GMap2(document.getElementById("map_canvas"));
```

```
//map.setMapType(G_HYBRID_MAP);
```

```
map.setCenter(new GLatLng(37.14, 22.4), 12);
```

```
map.setUIToDefault();
```

```

// Create red marker icon

redIcon = new GIcon(G_DEFAULT_ICON);

redIcon.image = "RedDot.png";

    redIcon.iconSize = new GSize(35, 40);

    redIcon.shadowSize = new GSize(45, 40);

// Create green marker icon

greenIcon = new GIcon(G_DEFAULT_ICON);

greenIcon.image = "GreenDot.png";

    greenIcon.iconSize = new GSize(35, 40);

    greenIcon.shadowSize = new GSize(45, 40);

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

try{

    // Opera 8.0+, Firefox, Safari

    ajaxRequest_time = new XMLHttpRequest();

    ajaxRequest_limits = new XMLHttpRequest();

```

```

ajaxRequest_station_data = new XMLHttpRequest();

} catch (e){

    // Internet Explorer Browsers

    try{

        ajaxRequest_time = new ActiveXObject("Msxml2.XMLHTTP");

        ajaxRequest_limits = new ActiveXObject("Msxml2.XMLHTTP");

        ajaxRequest_station_data = new
ActiveXObject("Msxml2.XMLHTTP");

    } catch (e) {

        try{

            ajaxRequest_time = new
ActiveXObject("Microsoft.XMLHTTP");

            ajaxRequest_limits = new
ActiveXObject("Microsoft.XMLHTTP");

            ajaxRequest_station_data = new
ActiveXObject("Msxml2.XMLHTTP");

        } catch (e){

            // Something went wrong

            alert("Your browser is incompatible.");

            return false;

        }

    }
}

```

```

<body onload="initialize()" onunload="GUnload()" >
<div id="map_canvas" style="width: 1000px; height: 550px"></div>
<form name='controlForm' style="font-family:Arial; font-size: 12px;font-weight:bold;" >

```

</head>

</script>

}

}

lookForNewMarkers());



ajaxRequest_station_data.onreadystatechange = refresh_data;

ajaxRequest_limits.onreadystatechange = refresh_limits;

ajaxRequest_time.onreadystatechange = refresh_time;

}

```
Time: <span id="time_txt"></span>
</form>
</body>
</html>
```

ΟΡΟΛΟΓΙΕΣ

- ❖ **Hardware:** Στην επιστήμη υπολογιστών ως υλικό ορίζεται το σύνολο των φυσικών εξαρτημάτων ενός υπολογιστή, όπως π.χ. ηλεκτρικά και ηλεκτρονικά στοιχεία, μικροτσίπ κλπ. Το υλικό καθοδηγείται κατά τη λειτουργία του από το λογισμικό.
- ❖ **Ethernet:** Το Ethernet είναι το συνηθέστερα χρησιμοποιούμενο πρωτόκολλο ενσύρματης τοπικής δικτύωσης υπολογιστών. Αναπτύχθηκε από την εταιρεία Xerox κατά τη δεκαετία του '70 και έγινε δημοφιλές αφότου η Digital Equipment Corporation και η Intel, από κοινού με τη Xerox, προχώρησαν στην προτυποποίησή του το 1980. Το 1985 το Ethernet έγινε αποδεκτό επίσημα από τον οργανισμό IEEE ως το πρότυπο 802.3 για ενσύρματα τοπικά δίκτυα (LAN).
- ❖ **Web browser:** Ένας Web browser (φυλλομετρητής ιστοσελίδων, πλοηγός Web, πρόγραμμα περιήγησης Web ή περιηγητής Ιστού) είναι ένα λογισμικό που επιτρέπει στον χρήστη του να προβάλλει, και να αλληλεπιδρά με κείμενα, εικόνες, βίντεο, μουσική, παιχνίδια και άλλες πληροφορίες συνήθως αναρτημένες σε μια ιστοσελίδα ενός ιστότοπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο. Το κείμενο και οι εικόνες σε μια ιστοσελίδα μπορεί να περιέχουν υπερσυνδέσμους προς άλλες ιστοσελίδες του ίδιου ή διαφορετικού ιστότοπου. Ο Web browser επιτρέπει στον χρήστη την γρήγορη και εύκολη πρόσβαση σε πληροφορίες που βρίσκονται σε διάφορες ιστοσελίδες και ιστότοπους εναλλάσσοντας τις ιστοσελίδες μέσω των υπερσυνδέσμων. Οι φυλλομετρητές χρησιμοποιούν τη γλώσσα μορφοποίησης HTML για την προβολή των ιστοσελίδων, για αυτό η εμφάνιση μιας ιστοσελίδας μπορεί να διαφέρει ανάλογα με τον browser.
- ❖ **Software:** Με τον όρο λογισμικό υπολογιστών, ή λογισμικό (software), ορίζεται η συλλογή από προγράμματα υπολογιστών, διαδικασίες και οδηγίες χρήσης που εκτελούν ορισμένες εργασίες σε ένα υπολογιστικό σύστημα.

- ❖ Φωτοκύτταρο: Η ονομασία φωτοκύτταρο σχετίζεται ετυμολογικά με την λέξη κύτταρο. Είναι σύνθετη λέξη και προέρχεται από τις δυο λέξεις φως και κύτταρο. Είναι λοιπόν μια συσκευή της οποίας η λειτουργία στηρίζεται στο φωτοηλεκτρικό φαινόμενο και χρησιμοποιείται για φωτομετρήσεις, στον ομιλούντα κινηματογράφο, στην τηλεόραση, στην τηλεφωτογραφία, στους φωτοπολλαπλασιαστές και σε διάφορες διατάξεις αυτοματισμού.

- ❖ Breadboard: Η λέξη breadboard είναι η ειδική πλακέτα άμεσης συναρμολόγησης και αποσυναρμολόγησης χωρίς κολλήσεις για πρόχειρη κατασκευή και έλεγχο ενός κυκλώματος.

- ❖ Η/Υ (Ηλεκτρονικός Υπολογιστής): Ο ηλεκτρονικός υπολογιστής είναι μια μηχανή κατασκευασμένη κυρίως από ψηφιακά ηλεκτρονικά κυκλώματα και δευτερευόντως από ηλεκτρικά και μηχανικά συστήματα, και έχει ως σκοπό να επεξεργάζεται πληροφορίες. Ο ηλεκτρονικός υπολογιστής είναι ένα αυτοματοποιημένο, ηλεκτρονικό, ψηφιακό επαναπρογραμματιζόμενο σύστημα γενικής χρήσης το οποίο μπορεί να επεξεργάζεται δεδομένα βάσει ενός συνόλου προκαθορισμένων οδηγιών, των εντολών που συνολικά ονομάζονται πρόγραμμα.

- ❖ Internet: Το Διαδίκτυο, γνωστό συνήθως και με την αγγλική άκλιτη ονομασία Internet, είναι ένα μέσο μαζικής επικοινωνίας (ΜΜΕ), όπως είναι για παράδειγμα και η τηλεόραση αν και η τελευταία έχει πολύ περιορισμένο αμφίδρομο χαρακτήρα. Ως μέσο έχει διπλή υπόσταση: η υλική (που αποτελείται από τον συνδυασμό δικτύων βασισμένων σε λογισμικό και υλικό), και η άυλη (αυτό, δηλαδή, που "κάνει" / προσφέρει στην κοινωνία το Διαδίκτυο ως μέσο).

ΠΗΓΕΣ

1. <http://el.science.wikia.com/wiki/%CE%A6%CF%89%CF%84%CE%BF%CE%BA%CF%8D%CF%84%CF%84%CE%B1%CF%81%CE%BF> (φωτοκύτταρο)
2. <http://www.arduino.cc/> (arduino)
3. <http://www.microplanet.gr/tutorials/microcontrollers/arduino> (μικροελεγκτής arduino)
4. <http://en.wikipedia.org/wiki/Breadboard> (breadboard)
5. <http://maps.google.gr/maps?hl=el&tab=wl> (google maps)
6. <http://teethgrinder.co.uk/open-flash-chart/> (open flash chart)
7. <https://developers.google.com/maps/?hl=el-GR> (google maps API)
8. <https://developers.google.com/maps/documentation/javascript/tutorial?hl=el-GR> (google maps javascript)

ΒΙΒΛΙΟΓΡΑΦΙΑ

- 1) Χαριτάντης Γιάννης, Ηλεκτρονικά 1.
- 2) Malvino Albert Paul, Ηλεκτρονική.
- 3) Βασίλειος Τ. Ταμπακάς, Εισαγωγή στις Βάσεις Δεδομένων.
- 4) Stephens Ryan, Plew Ron, Μάθετε την SQL σε 24 ώρες.
- 5) ANDREW S. TANENBAUM, Δίκτυα Υπολογιστών.
- 6) Dubois Paul, Hinz Stefan, Pedersen Carsten, Ο επίσημος οδηγός MySQL 5.
- 7) Ramakrishnan Raghu, Gehrke Johannes, Συστήματα διαχείρισης βάσεων δεδομένων, Τόμος Β.
- 8) Melonie Julie C, Μάθετε PHP, MySQL και Apache, Όλα σε Ένα.

EYPETHPIO

hardware, 5, 12, 14, 67, 103

ethernet, 5, 9, 33, 103

web browser, 5, 32, 103

software, 5, 12, 13, 17, 67, 103

φωτοκύτταρο, 9, 10, 11, 19, 23, 25, 61, 104

arduino duemilanove, 9, 11, 12, 14, 17, 25

arduino ethernet shield, 9, 17

breadboard, 9, 19

H/Y, 9, 19, 24, 32

modem / router, 9, 19, 32

internet, 9, 19

XAMPP, 9, 29, 31

mysql Workbench, 9, 30, 34, 35, 37, 39

mozilla firefox, 9

mozilla thunderbird, 9

internet explorer, 9

winzip, 9

command prompt, 9, 32

json, 9, 79

xml, 9, 79

open flash chart, 9, 72, 73, 79

google maps, 9, 72, 79

google earth, 9

USB, 16, 23, 25

πολύμετρο, 23

IP, 33

data import, 39

data export, 39

php, 65, 66, 67, 79

serial monitor, 27, 55

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ - ΕΙΚΟΝΩΝ

| | |
|---|----|
| Σχήμα 1.1.1: Φωτοκύτταρο..... | 11 |
| Σχήμα 1.1.2: Μικροελεγκτής Arduino Duemilanove..... | 12 |
| Σχήμα 1.1.3: Arduino Ethernet Shield..... | 18 |
| Σχήμα 1.1.4: Breadboard, ενώθηκαν με απλά καλώδια το φωτοκύτταρο και το Arduino.. | 19 |
| Σχήμα 2.1: Φωτοκύτταρο – Πολυμετρητής – Breadboard..... | 23 |
| Σχήμα 2.2.1: Breadboard – Arduino..... | 24 |
| Σχήμα 2.2.2: Arduino – H/Y..... | 24 |
| Σχήμα 2.2.3: Επιλογή συσκευής που χρησιμοποιείται..... | 25 |
| Σχήμα 2.2.4: Επιλογή θύρας USB..... | 26 |
| Σχήμα 2.2.5: Κώδικας που στέλνονται τα δεδομένα από το φωτοκύτταρο..... | 26 |
| Σχήμα 2.2.6: Εμφάνιση καταγραφών από φωτοκύτταρο στην οθόνη κάθε 1 λεπτό..... | 28 |
| Σχήμα 2.2.7 : Πλάνο – 1η προσπάθεια ένωσης όλου του συστήματος..... | 29 |
| Σχήμα 2.3.1: Το περιβάλλον του προγράμματος XAMPP..... | 30 |
| Σχήμα 2.3.2: Φάκελος που περιέχει τα απαραίτητα αρχεία μέσα στο htdocs..... | 31 |
| Σχήμα 2.3.4: Ρυθμίσεις και παράμετροι του XAMPP μέσα από τον φυλλομετρητή..... | 32 |
| Σχήμα 2.4.1: Στατική IP που δόθηκε στο ρούτερ..... | 33 |
| Σχήμα 2.4.2: Ping μέσω cmd για αποτελέσματα πακέτων επικοινωνίας του ρούτερ..... | 34 |
| Σχήμα 2.5.1: MySQL Workbench αρχική οθόνη (home)..... | 35 |
| Σχήμα 2.5.2: Δημιουργία – Προσθήκη λογαριασμού στο MySQL Workbench..... | 36 |
| Σχήμα 2.5.3: Συμπλήρωση στοιχείων του χρήστη που δημιουργήθηκε..... | 37 |

| | |
|--|----|
| Σχήμα 2.5.4: Επιλογή και καταχώρηση δικαιωμάτων στον χρήστη <code>arduino_power</code> | 38 |
| Σχήμα 2.5.1.1: Εισαγωγή δεδομένων (Data Import)..... | 39 |
| Σχήμα 2.5.1.2: Εξαγωγή δεδομένων (Data Export)..... | 39 |
| Σχήμα 2.5.2.1: Σύνδεση στην βάση δεδομένων..... | 40 |
| Σχήμα 2.5.2.2: Επιλογή τοπικής σύνδεσης..... | 41 |
| Σχήμα 2.5.2.3: Εμφάνιση και επεξεργασία βάσης δεδομένων..... | 41 |
| Σχήμα 2.5.2.4: Πίνακες βάσης <code>bkr</code> | 47 |
| Σχήμα 2.5.2.5: Πίνακας <code>events</code> | 48 |
| Σχήμα 2.5.2.6: Πίνακας <code>limits</code> | 48 |
| Σχήμα 2.5.2.7: Πίνακας <code>stations</code> | 49 |
| Σχήμα 2.5.2.8: Βάση δεδομένων <code>arduino_light</code> | 50 |
| Σχήμα 2.6.1: Τοποθεσία αρχείου <code>upload_arduino.php</code> | 51 |
| Σχήμα 2.6.2: Αντιγραφή αρχείου <code>upload_arduino.php</code> | 54 |
| Σχήμα 2.6.3: Επικόλληση αρχείου <code>upload_arduino.php</code> στο <code>XAMP\htdocs</code> | 54 |
| Σχήμα 2.6.1.1: Κώδικας καταγραφής δεδομένων (Arduino) – μεταφορά στη βάση..... | 55 |
| Σχήμα 2.6.1.2: Σειριακή οθόνη (Serial Monitor) Arduino, επιτυχής αποστολή δεδομένων..... | 55 |
| Σχήμα 2.6.2.1: Google Maps, εμφάνιση της Ελλάδας..... | 59 |
| Σχήμα 2.6.2.2: 1ο παράδειγμα διαγράμματος που χρησιμοποιεί το Open Flash Chart..... | 60 |
| Σχήμα 2.6.2.3: 2ο παράδειγμα διαγράμματος, γνωστό και ως «πίτα»..... | 61 |
| Σχήμα 2.6.2.4: Δεδομένα που καταγράφοντε στη βάση..... | 61 |
| Σχήμα 2.6.2.5: Χάρτης απεικόνισης περιοχών στις οποίες γίνανε καταμετρήσεις της ηλιοφάνειας..... | 62 |

| | |
|--|----|
| Σχήμα 2.6.2.6: Εμφάνιση πληροφοριών για την περιοχή της Σπάρτης..... | 63 |
| Σχήμα 2.6.2.7: Εμφάνιση πληροφοριών για την περιοχή των Περιβολιών..... | 63 |
| Σχήμα 2.6.2.8: Εμφάνιση διαγράμματος για την περιοχή της Σπάρτης..... | 64 |
| Σχήμα 2.6.2.9: Εμφάνιση διαγράμματος για την περιοχή των Περιβολιών..... | 64 |