

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1595

**ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ  
ΑΠΟΜΑΚΡΥΣΜΕΝΗΣ ΜΕΤΡΗΣΗΣ ΚΑΙ ΚΑΤΑΓΡΑΦΗΣ ΤΗΣ  
ΚΑΤΑΝΑΛΩΣΗΣ ΤΟΥ ΑΝΑΛΟΓΙΚΟΥ ΜΕΤΡΗΤΗ ΤΗΣ Δ.Ε.Η.  
ΜΕΣΩ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ ΒΑΣΙΣΜΕΝΟ ΣΤΗΝ ΠΛΑΤΦΟΡΜΑ  
ARDUINO**

ΣΠΟΥΔΑΣΤΗΣ: ΝΤΙΝΕ ΒΑΣΙΛΕΙΟΣ (6564)

ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΕΣ: ΤΟΠΑΛΗΣ ΕΥΑΓΓΕΛΟΣ  
ΧΑΔΕΛΛΗΣ ΛΟΥΚΑΣ

ΠΑΤΡΑ 2016

## Περίληψη

Η παρούσα πτυχιακή έχει ως αντικείμενο την κατασκευή ενός συστήματος εποπτείας με το οποίο, από το αναλογικό ρολόι της Δ.Ε.Η. θα μπορούμε να υπολογίζουμε την συνολική και την στιγμιαία κατανάλωση της εγκατάστασης που είναι τοποθετημένος ο μετρητής της Δ.Ε.Η..

Πιο συγκεκριμένα για την υλοποίηση αυτού του συστήματος θα χρησιμοποιηθεί η αναπτυξιακή πλατφόρμα Arduino. Πρόκειται για μια πλατφόρμα ανοικτού κώδικα που περιέχει μια πλακέτα με ενσωματωμένο τον μικροελεγκτή και εισόδους/εξόδους η οποία μπορεί να προγραμματιστεί με την γλώσσα Wiring. Σε αυτή την πλατφόρμα μπορούν να συνδεθούν ενσύρματα ή ασύρματα διάφοροι αισθητήρες ή συσκευές δράσης , ενώ παράλληλα δίνει τη δυνατότητα διασύνδεσης με το διαδίκτυο.

Στην δική μας περίπτωση ένας αισθητήρας θα παρακολουθεί τον δίσκο του μετρητή της Δ.Ε.Η. και από διάφορους υπολογισμούς θα βρίσκουμε την συνολική και την στιγμιαία κατανάλωση της εγκατάστασης (που είναι τοποθετημένος ο μετρητής της Δ.Ε.Η.) και παράλληλα θα γίνεται μια σύγκριση των τιμών αυτών με την ένδειξη του μετρητή Bizy Type F Plug της εταιρίας MEAZON .

Όλες οι μετρήσεις που γίνονται θα καταγράφονται σε μια βάση δεδομένων και θα δημιουργηθεί και ένας δυναμικός ιστοχώρος στον οποίο θα απεικονίζονται όλες οι μετρήσεις και θα βγάζουμε κάποια στατιστικά.

Ο σκοπός αυτού του συστήματος είναι η απομακρυσμένη καταγραφή της κατανάλωσης μιας εγκατάστασης χωρίς να απαιτείται η ανθρώπινη παρουσία (προσωπικό της Δ.Ε.Η. να κάνει την μέτρηση) και επίσης η απομακρυσμένη παρακολούθηση της κατανάλωσης από τον χρήστη της εγκατάστασης.

## Περιεχόμενα

Περίληψη .....	ii
ΚΕΦΑΛΑΙΟ 1: Εισαγωγή.....	6
ΚΕΦΑΛΑΙΟ 2: Τεχνολογίες, Πλατφόρμες και Εργαλεία Ανάπτυξης που χρησιμοποιήθηκαν.....	7
2.1 HTML .....	7
2.2 CSS.....	8
2.3 JavaScript.....	9
2.4 PHP .....	10
2.5 MySQL .....	10
2.6 Apache HTTP .....	11
2.7 Τεχνολογίες που χρησιμοποιήθηκαν .....	12
2.7.1 ZigBee .....	12
2.8. Μικροελεγκτές .....	13
2.8.1. Arduino .....	13
2.8.1.1 Ιστορική αναδρομή .....	13
2.8.1.2 Γενικά για το Arduino .....	14
2.8.1.3 Εκδόσεις Arduino .....	14
2.8.1.4 Shields .....	17
2.9 Εργαλεία ανάπτυξης λογισμικού .....	17
2.9.1 XAMPP .....	17
2.9.2 WinSCP .....	18
2.9.3 Putty .....	18
2.9.4 Notepad++.....	19
2.9.5 Λογισμικό Arduino IDE 1.6.4 .....	20
2.9.5.1 Γλώσσα του Arduino.....	20
2.9.5.2 Δομή προγράμματος.....	24
2.9.5.3 Μεταβλητές.....	25
2.9.5.4 Σταθερές.....	25
2.9.5.5 Πίνακες – Arrays.....	25
2.9.5.6 Αριθμητικοί τελεστές .....	25
2.9.5.7 Τελεστές σύγκρισης.....	26
2.9.5.8 Λογικοί τελεστές.....	26
2.10 AmCharts: JavaScript Charts & Maps .....	27
ΚΕΦΑΛΑΙΟ 3 : Υλικό Μέρος Συστήματος .....	28
3.1 Αναλογικός Μετρητής Ενέργειας.....	28
3.2 Arduino DUE .....	30

3.2.1 Τεχνικά χαρακτηριστικά Arduino DUE .....	30
3.2.3 Τροφοδοσία Arduino.....	31
3.2.4 ΜΝΗΜΗ .....	32
3.2.5 ΕΙΣΟΔΟΙ & ΕΞΟΔΟΙ.....	32
3.2.6 USB Προστασία από Υπερένταση .....	36
3.3 Ethernet Shield.....	36
3.4 Αισθητήρας TCRT5000: .....	38
3.5 Bizy Type F Plug .....	40
3.6 Meazon Gateway Advanced.....	41
3.7 Τοπολογία Διασυνδέσεις .....	41
ΚΕΦΑΛΑΙΟ 4: Λογισμικό μέρος του συστήματος.....	43
4.1 Προγραμματισμός Arduino .....	43
4.1.1 Εισαγωγή βιβλιοθηκών .....	44
4.1.2 Δήλωση - αρχικοποίηση μεταβλητών και ρυθμίσεις Ethernet.....	44
4.1.3 Συνάρτηση void setup().....	45
4.1.4 Συνάρτηση void loop().....	45
4.2 Προγραμματισμός BeagleBone Meazon.....	49
4.3 Βάση Δεδομένων.....	54
4.3.1 Δημιουργία Βάσης Δεδομένων .....	54
4.3.2 Σύνδεση βάσης δεδομένων με Arduino και BeagleBone .....	57
4.4 Ιστοσελίδα που υλοποιήθηκε και οι λειτουργίες της .....	58
4.4.1 Αρχική σελίδα.....	58
4.4.2 Σελίδα απεικόνισης των μετρήσεων .....	61
4.4.3 Σελίδα απεικόνισης μετρήσεων σε πίνακα.....	66
4.4.4 Σελίδες Επιλογής Διαγραμμάτων .....	68
4.4.5 Ιστοσελίδες Διαγραμμάτων .....	71
4.4.5.1 Simple diagrams: .....	73
4.4.5.1.1 Αρχείο data_all.php.....	73
4.4.5.2 Statistics Diagrams from Arduino.....	79
4.4.5.2.1 Διάγραμμα μέγιστης στιγμιαίας κατανάλωσης ανά ημέρα .....	79
4.4.5.2.2 Διάγραμμα μέσης στιγμιαίας κατανάλωσης ανά ημέρα.....	80
4.4.5.2.3 Διάγραμμα μέγιστης στιγμιαίας κατανάλωσης ανά μήνα .....	82
4.4.5.2.4 Διάγραμμα μέσης στιγμιαίας κατανάλωσης ανά μήνα.....	83
4.4.5.3 Statistics Diagrams from BeagleBone.....	84
4.4.5.4 Compare diagrams: .....	87
4.4.5.4.1 Διάγραμμα σύγκρισης στιγμιαίας κατανάλωσης από Arduino και BeagleBone.....	87

4.4.5.4.2 Διάγραμμα σύγκρισης συνολικής κατανάλωσης από Arduino και BeagleBone .....	90
4.4.5.4.3 Συμπεράσματα συγκριτικών διαγραμμάτων .....	93
Κεφάλαιο 5: Συμπεράσματα και Μελλοντικές Επεκτάσεις .....	95
Βιβλιογραφία .....	96
Παράρτημα.....	98

## ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

Ο ηλεκτρισμός πλέον αποτελεί μέρος της καθημερινότητας μας, είναι αναγκαίος στο σπίτι, στην εργασία, στην ψυχαγωγία, στην επικοινωνία ακόμα και στα παιχνίδια μας. Για αυτό το λόγο σε μία κοινωνία όπου όλα μπορούμε να τα μετρήσουμε και να τα κοστολογήσουμε έτσι και εδώ μας δίνεται αυτή η δυνατότητα όπου μπορούμε να μετρήσουμε την ενέργεια που καταναλώνουμε και να την περιορίσουμε όπου αυτό είναι εφικτό. Εκείνο που θα πρέπει να τονιστεί είναι ότι, όταν μιλάμε για εξοικονόμηση ηλεκτρικής ενέργειας, δεν σημαίνει τη στέρησή μας από την ικανοποίηση των καθημερινών μας αναγκών και ούτε τη μείωση της άνεσής μας.

Υπάρχουν πολλοί τρόποι για την μέτρηση της κατανάλωσης της ενέργειας. Μερικοί από αυτούς είναι: με ένα βατόμετρο, με έναν αναλυτή ενέργειας, με έναν μετρητή ράγας μονοφασικό ή τριφασικό και άλλοι πολλοί. Στην παρούσα πτυχιακή εργασία παρουσιάζεται ένας τρόπος με τον οποίο δεν χρειάζεται να κάνουμε καμία παρεμβολή στο κύκλωμα μας, απλά με ένα Arduino και έναν αισθητήρα ανάκλασης μπορούμε να ξέρουμε ανά πάσα στιγμή την κατανάλωση μας, και την στιγμιαία αλλά και την συνολική.

Για την καλύτερη κατανόηση της ενέργειας που καταναλώνουμε θα αποθηκεύουμε τις μετρήσεις μας σε μια βάση δεδομένων, θα δημιουργήσουμε κάποια διαγράμματα και θα εξάγουμε κάποια στατιστικά. Για την ευκολότερη περιήγηση στα παραπάνω θα δημιουργήσουμε μια ιστοσελίδα στην οποία θα έχουμε όλα τα παραπάνω.

Για να είμαστε όμως σίγουροι ότι ο μετρητής που φτιάξαμε μετράει σωστά θα συγκρίνουμε τις μετρήσεις μας με ένα εμπορικό προϊόν, συγκεκριμένα με την ένδειξη του μετρητή Bizy Type F Plug της εταιρίας MEAZON.

Στη συνέχεια αναφέρονται τα κεφάλαια τα οποία θα ασχοληθούμε και θα αναπτύξουμε με λίγες λεπτομέρειες για το τι αφορά το καθένα :

Κεφάλαιο 1: Γίνεται μία απλή εισαγωγή στο με τι έχει σχέση η παρούσα εργασία.

Κεφαλαίο 2: Γίνεται αναφορά στις Τεχνολογίες, στις Πλατφόρμες και στα Εργαλεία Ανάπτυξης που χρησιμοποιήθηκαν.

Κεφαλαίο 3: Γίνεται αναλυτική αναφορά στο υλικό μέρος του συστήματος.

Κεφάλαιο 4: Αναφέρεται στο λογισμικό μέρος του συστήματος.

και στο Κεφάλαιο 5 που είναι και το τελευταίο, γίνεται έκθεση των συμπερασμάτων και των μελλοντικών επεκτάσεων του συστήματος που υλοποιήσαμε.

## ΚΕΦΑΛΑΙΟ 2: Τεχνολογίες, Πλατφόρμες και Εργαλεία Ανάπτυξης που χρησιμοποιήθηκαν

### 2.1 HTML

Η HTML (είναι τα αρχικά από **H**yper**T**ext **M**arkup **L**anguage, ή στα ελληνικά Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.



Εικόνα 2.1: Παράδειγμα HTML

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από (>)» και «μικρότερο από (<)» για παράδειγμα: <html>. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης. Ανάμεσα στις ετικέτες, μπορεί να τοποθετηθεί κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και να τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Με τα στοιχεία της HTML κτίζονται όλοι οι ιστότοποι. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML (κυρίως ως προς την εμφάνιση).

Μερικά παραδείγματα των στοιχείων της html είναι:

- Η κεφαλίδα του εγγράφου HTML: <head>...</head>
- Οι επικεφαλίδες: <h1>Επικεφαλίδα1</h1>  
<h2>Επικεφαλίδα2</h2>  
<h3>Επικεφαλίδα3</h3>  
<h4>Επικεφαλίδα4</h4>  
<h5>Επικεφαλίδα5</h5>  
<h6>Επικεφαλίδα6</h6>
- Οι παράγραφοι: <p>Παράγραφος 1</p> <p>Παράγραφος 2</p>
- Αλλαγή γραμμής: <br>  
Σημείωση: Η διαφορά ανάμεσα στο <br> και το <p> είναι ότι το «br» αλλάζει γραμμή χωρίς να αλλάζει την σημαντική δομή της σελίδας, ενώ το «p» τεμαχίζει τη σελίδα σε παραγράφους. Το «br» είναι ένα άδειο στοιχείο, δηλαδή δεν έχει περιεχόμενο, ούτε και χρειάζεται ετικέτα τερματισμού.
- Σύνδεσμος: Για τη δημιουργία του χρησιμοποιείται η ετικέτα <a>. Η ιδιότητα href περιέχει τη σελίδα προορισμού του συνδέσμου. Ένα παράδειγμα συνδέσμου:  
<a href="http://el.wikipedia.org/">Ένας σύνδεσμος στη Βικιπαίδεια!</a>

- **Σχόλια:** <!-- Αυτό είναι ένα σχόλιο -->
- **Εικόνα:** 
- **Έντονη γραφή:** <strong>το κείμενο ή τη λέξη που θελω να φαινονται έντονα</strong>
- **Εικόνα που να σε παραπέμπει σε σύνδεσμο:**  
 <a href="κάποιος\_σύνδεσμος">  
     
</a>

Ένα συνηθισμένο πρόγραμμα υπολογιστή, το πρόγραμμα Hello world, στην HTML υλοποιείται έτσι:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού.

## 2.2 CSS

CSS είναι τα αρχικά από τις λέξεις Cascading Style Sheets δηλαδή Διαδοχικά Φύλλα Στυλ. Η CSS είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης (HTML και XHTML), δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Συγκεκριμένα να διαμορφώνει χαρακτηριστικά όπως, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

Στην HTML για να αλλάξουμε το χρώμα κάποιου κειμένου ή το χρώμα ενός πίνακα, θα πρέπει να βρούμε το χρώμα αυτό μέσα στον κώδικα και να το αλλάξουμε. Η διαδικασία αυτή μπορεί να φαντάζει εύκολη όταν έχουμε να διαχειριστούμε μια μόνο σελίδα, αλλά

```
h1 { color: white;
      background: orange;
      border: 1px solid bla
      padding: 0 0 0 0;
      font-weight: bold;
    }
/* begin: seaside-theme */

body {
  background-color: white;
  color: black;
  font-family: Arial, sans-serif;
  margin: 0 4px 0 0;
  border: 12px solid;
}
```

CSS

Εικόνα 1.2: Παράδειγμα CSS



ένα site αποτελείται από δεκάδες σελίδες τις οποίες χρειάζεται να διαχειριζόμαστε εύκολα και γρήγορα. Αν για παράδειγμα, θέλουμε να αλλάξουμε τα χρώματα στο κύριο μενού του site μας, το οποίο επαναλαμβάνεται σε όλες τις σελίδες, θα έπρεπε να ανοίξουμε κάθε σελίδα του site και να αλλάξουμε τα χρώματα του φόντου και των links του μενού. Η διαδικασία αυτή εκτός από χρονοβόρα είναι και κουραστική.

Με την χρήση της CSS μπορούμε να ορίσουμε χρώματα και μεγέθη οργανωμένα σε στυλ και έπειτα να εφαρμόσουμε τα στυλ αυτά στα στοιχεία των σελίδων του site μας. Με αυτόν τον τρόπο, κάθε φορά που αλλάζουμε το χρώμα ενός στυλ, αλλάζει το χρώμα όλων των στοιχείων που έχουν αναφορά στο στυλ αυτό.

Εκτός από την ευκολία στην διαχείριση ενός site, ένα άλλο σημαντικό πλεονέκτημα της χρήσης CSS στις σελίδες είναι ο "καθαρότερος" κώδικας, χωρίς πολλές ιδιότητες στις ετικέτες οι οποίες τον κάνουν δυσανάγνωστο. Επιπλέον κάνει γρηγορότερη την πλοήγηση καθώς το αρχείο, μέσα στο οποίο ορίζονται τα στυλ, "διαβάζεται" από τον browser μόνο μια φορά και έπειτα αποθηκεύεται στην cache memory, μειώνοντας έτσι το μέγεθος της πληροφορίας που γίνεται download από τους browsers.

## 2.3 JavaScript

Η JavaScript είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Διερμηνευμένη γλώσσα προγραμματισμού σημαίνει ότι η υλοποίηση της συνήθως αποτελείται από έναν διερμηνέα. Θεωρητικά, οποιαδήποτε γλώσσα μπορεί να είναι είτε μεταγλωττισμένη είτε διερμηνευμένη, έτσι ο διαχωρισμός αυτός εφαρμόζεται μόνο με βάση την συνήθη πρακτική υλοποίησης, και όχι κάποια συγκεκριμένη ιδιότητα της γλώσσας.

Αρχικά η JavaScript αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων βασισμένη στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξη της είναι επηρεασμένη από τη C. Η JavaScript παρόλο που αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, είναι δύο γλώσσες που δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστραφή, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Επίσης η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων, όπως: τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).

Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript.

Ο κώδικας Javascript μιας σελίδας περικλείεται από τις ετικέτες της HTML `<script type="text/javascript">` και `</script>`.

Για παράδειγμα, ο ακόλουθος κώδικας Javascript εμφανίζει ένα πλαίσιο διαλόγου με το κείμενο "Γεια σου, κόσμε!":

```
<script type="text/javascript">
alert('Γεια σου, κόσμε!');
</script>
```

## 2.4 PHP

Η PHP είναι μία πολύ διαδεδομένη γλώσσα προγραμματισμού γενικής χρήσης, που είναι κατάλληλη για προγραμματισμό διαδικτυακών εφαρμογών και μπορεί να εισαχθεί σε κώδικα HTML.



Εικόνα 2.3: Λογότυπο php

Είναι μια γλώσσα προγραμματισμού ειδικά για την κατασκευή δυναμικών ιστοσελίδων, δηλαδή σελίδων που μπορούν να τροποποιούνται από το διαχειριστή τους online ή να διαφοροποιούνται, ανάλογα με τα χαρακτηριστικά του χρήστη που τις προβάλλει, όπως για παράδειγμα, το λειτουργικό του σύστημα, η διεύθυνση IP του κ.ά.

Δεν χρησιμοποιείται για την αισθητική διαμόρφωση μιας σελίδας, αλλά για τον χειρισμό των λειτουργιών και εργασιών που θα διεκπεραιώνει. Συνεπώς, ο κώδικας που γράφεται για μια ιστοσελίδα σε γλώσσα PHP δεν γίνεται άμεσα αντιληπτός αλλά μετά από την επέμβαση του χρήστη στην ιστοσελίδα.

Για να γίνει αυτό κατανοητό: η PHP χρησιμοποιείται ευρέως για τον χειρισμό ιστοσελίδων με δυνατότητες όπως η εγγραφή χρηστών (user registration), τα fora κ.ά. Λειτουργεί με την βοήθεια της HTML και πλέον και με την XHTML (νέα αναθεωρημένη έκδοση της HTML).

Επίσης μπορεί να χρησιμοποιηθεί μαζί με την MySQL για την διαχείριση δεδομένων μέσα σε βάσεις. Για παράδειγμα, σε μια ιστοσελίδα που είναι απαραίτητη η εγγραφή των χρηστών, η PHP μπορεί να αποθηκεύει τα ονόματα και τους κωδικούς τους σε μια βάση δεδομένων.

## 2.5 MySQL

Πριν αναφερθούμε στην MySQL, θα πούμε λίγα λόγια για την Βάση Δεδομένων. Η Βάση Δεδομένων είναι μια συλλογή από σχετικά μεταξύ τους δεδομένα – αντικείμενα. Σκοπός της είναι να διευκολύνει τον χρήστη στην εισαγωγή,

διαγραφή, χειρισμό και επεξεργασία μεγάλου όγκου δεδομένων. Τα δεδομένα που χειρίζεται είναι δυναμικά, δηλαδή αλλάζουν συνεχώς.



Εικόνα 2.2: Λογότυπο MySQL

Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Πιο συγκεκριμένα είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι .

Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU (General Public License), καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.

Τα δεδομένα στη MySQL αποθηκεύονται σε πίνακες. Ένας πίνακας αποτελείται από στήλες και γραμμές και είναι μια συλλογή από σχετικές καταχωρήσεις δεδομένων. Οι βάσεις δεδομένων είναι χρήσιμες γιατί η αποθήκευση των πληροφοριών γίνεται σε κατηγορίες. Η διαχείριση των δεδομένων στις βάσεις, γίνεται μέσω των SQL εντολών, στη MySQL. Η πιο σημαντική ίσως κατηγορία εντολών που χρησιμοποιείται είναι τα queries. Ένα query είναι ένα ερώτημα ή ένα αίτημα. Με τη MySQL, επιτρέπεται η διερεύνηση σε μια βάση δεδομένων και η επιστροφή ζητηθέντων πληροφοριών.

Η MySQL χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το Flickr, το YouTube, η Wikipedia, το Google, το Facebook και το Twitter.

## 2.6 Apache HTTP

Ο Apache HTTP είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όταν ένας χρήστης επισκέπτεται ένα ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους εξυπηρετητές ιστού, αυτό γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το

Linux, το Unix και το Mac OS X. Κυκλοφόρησε υπό την άδεια λογισμικού Apache και είναι λογισμικό ανοιχτού κώδικα. Συντηρείται από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).

Ο Apache χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσης Δεδομένων π.χ. Oracle, MySQL.

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και την πλατφόρμα .NET. Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστότοπους.

## 2.7 Τεχνολογίες που χρησιμοποιήθηκαν

### 2.7.1 ZigBee

Το ZigBee είναι μία από τις πιο νέες τεχνολογίες στο χώρο των ασύρματων δικτύων προσωπικού χώρου (WPANs). Προήλθε από τη συνεργασία της εταιρείας ZigBee Alliance με την επιτροπή IEEE 802.15.4 και παρέχει τη δυνατότητα για συνδέσεις συσκευών με χαμηλό ρυθμό μετάδοσης, χαμηλό κόστος και χαμηλή κατανάλωσης ισχύος.

Το ZigBee είναι μια ασύρματη τεχνολογία που αναπτύσσεται για να καλύψει τις μοναδικές ανάγκες των χαμηλού κόστους, χαμηλής ισχύος, ασύρματων δικτύων αισθητήρων. Συγκεκριμένα το ZigBee είναι το όνομα μιας προδιαγραφής για μια ακολουθία υψηλού επιπέδου πρωτοκόλλων επικοινωνίας που χρησιμοποιούν οι μικροί, χαμηλής ισχύος ψηφιακοί δεκτές βασισμένοι στο 802.15.4 πρότυπο της IEEE για τα ασύρματα προσωπικά τοπικά δίκτυα (WPAN), όπως για παράδειγμα τα ασύρματα ακουστικά που συνδέονται με τα κινητά τηλέφωνα. Η τεχνολογία προορίζεται να είναι απλούστερη και φτηνότερη από άλλα ασύρματα προσωπικά, τοπικά δίκτυα (WPAN), όπως το Bluetooth. Το ZigBee στοχεύει στις εφαρμογές ραδιοσυχνότητας (RF) που απαιτούν ένα χαμηλό ρυθμό μεταφοράς δεδομένων, μεγάλη ζωή μπαταριών, και εξασφαλισμένη δικτύωση. Τα πρότυπα εκμεταλλεύονται πλήρως το 802.15.4 πρότυπο της IEEE και λειτουργούν στις χωρίς άδεια ζώνες παγκοσμίως στις ακόλουθες συχνότητες: 2.400-2.484 GHz, 902-928 MHz και 868.0-868.6 MHz.

#### **Σκοπός, χρησιμότητα, σχέση με την προϋπάρχουσα τεχνολογία.**

Το ZigBee είναι χαμηλού κόστους, χαμηλής ισχύος, ασύρματα πρότυπα δικτύωσης πλέγματος. Το χαμηλότερο κόστος επιτρέπει στην τεχνολογία για να επεκταθεί ευρέως στις ασύρματες εφαρμογές ελέγχου και παρακολούθησης, η χαμηλή κατανάλωση ισχύος επιτρέπει τη μακρύτερη ζωή με μικρότερες μπαταρίες, και η δικτύωση πλέγματος παρέχει υψηλή αξιοπιστία και μεγαλύτερη ακτίνα λειτουργίας.

#### **Το πλεονέκτημα ZigBee**

Χαρακτηριστικά γνωρίσματα πρωτοκόλλου ZigBee:

- Χαμηλός κύκλος καθυκόντων - παρέχει μακριά ζωή μπαταριών
- Χαμηλή λανθάνουσα κατάσταση
- Υποστηρίζει πολλές τοπολογίες δικτύων: Στατικός, δυναμικός, αστέρι και πλέγμα (Static, dynamic, star and mesh)
- Άμεσο απλωμένο φάσμα ακολουθίας (DSSS) Direct Sequence Spread Spectrum
- Μέχρι 65.000 κόμβοι σε ένα δίκτυο
- 128-bit AES encryption κρυπτογράφηση - παρέχει ασφαλείς συνδέσεις μεταξύ των συσκευών
- Αποφυγή συγκρούσεων
- Ποιοτική ένδειξη συνδέσεων
- Σαφής αξιολόγηση των καναλιών

## 2.8. Μικροελεγκτές

### 2.8.1. Arduino



Εικόνα 2.3: Λογότυπο Arduino

#### 2.8.1.1 Ιστορική αναδρομή

Το 2005 από μαθητές ξεκίνησε ένα σχέδιο, με σκοπό να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο. Έτσι οι ιδρυτές Massimo Banzi και David Cueartielles ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας. Ονόμασαν το σχέδιο Arduino από τον Arduino της Ivrea.

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

### 2.8.1.2 Γενικά για το Arduino

Το Arduino είναι μια πλατφόρμα ανοικτού κώδικα που περιέχει μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, την οποία μπορούμε να την προγραμματίσουμε με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++ ).

Το Arduino χρησιμοποιείται και για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά μπορεί να συνδεθεί και με υπολογιστή μέσω προγραμμάτων όπως για παράδειγμα τα: Processing, Max/MSP, Pure Data, SuperCollider. Το Arduino μπορεί να το αγοράσει κάποιος έτοιμο προ-συναρμολογημένο ή μπορεί να κατασκευάσει μόνος του επειδή όλα του τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν.

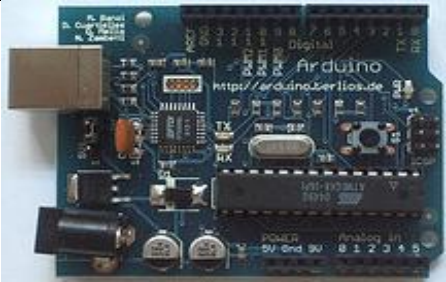
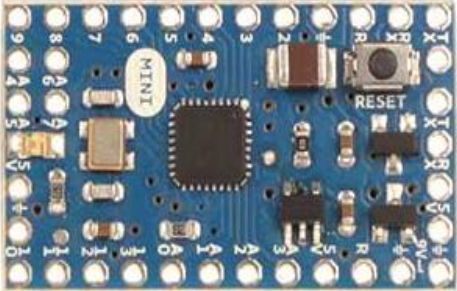
Αφού κατασκευαστεί, ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου.

Το κύριο πλεονέκτημά του είναι η τεράστια κοινότητα που το υποστηρίζει και η οποία έχει δημιουργήσει, συντηρεί και επεκτείνει μια ανάλογου μεγέθους online γνωσιακή βάση.

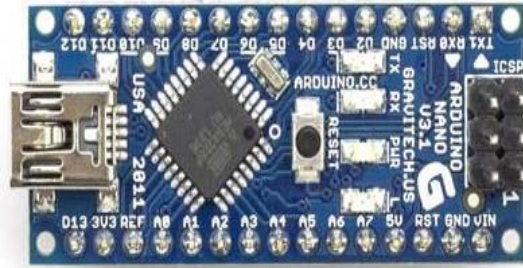
### 2.8.1.3 Εκδόσεις Arduino

Μερικές από τις εκδόσεις του Arduino Hardware που έχουν χρησιμοποιηθεί εμπορικά μέχρι τώρα φαίνονται στον παρακάτω πίνακα:

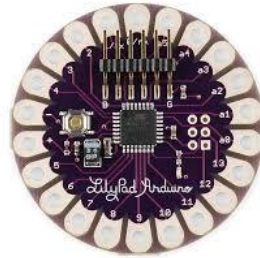
Πίνακας 2.1: Εκδόσεις Arduino

<p>Το Arduino Extreme, με ένα USB interface για προγραμματισμό χρησιμοποιώντας τεχνολογία ATmega8.</p>	 A photograph of the Arduino Extreme board, a blue PCB with a USB Type-B connector on the left, a 9V battery jack, and various electronic components. The text "Arduino" and "http://www.arduino.cc" are visible on the board.
<p>Το Arduino Mini, μία έκδοση μινιατούρας του Arduino χρησιμοποιώντας τεχνολογία surface-mounted ATmega168</p>	 A photograph of the Arduino Mini board, a small blue PCB with a USB Type-B connector on the left, a reset button, and various electronic components. The text "MINI" and "RESET" are visible on the board.

Το Arduino Nano, ένα ακόμα πιο μικρό, USB τροφοδοτούμενη εκδοχή του Arduino χρησιμοποιώντας τεχνολογία surface-mounted ATmega168 (ATmega328 για την νεότερη έκδοση).



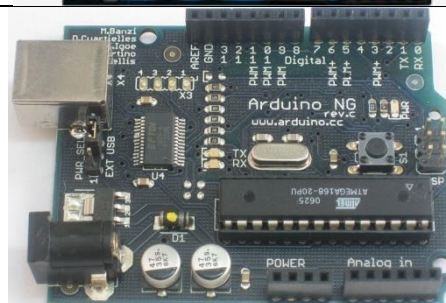
Το LilyPad Arduino, ένα μιμηταστικό σχέδιο για εφαρμογές ένδυσης και E-textiles χρησιμοποιώντας τεχνολογία surface-mounted AT-mega328



Το Arduino NG, με ένα USB interface για προγραμματισμό και χρησιμοποιώντας τεχνολογία ATmega8.



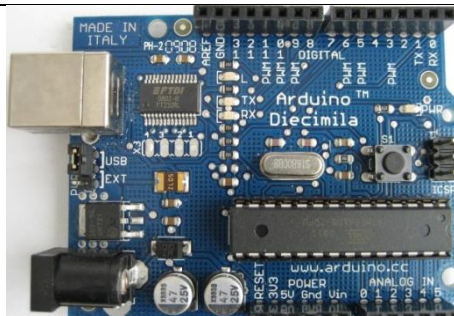
Το Arduino NG plus, με ένα USB interface για προγραμματισμό και χρησιμοποιώντας τεχνολογία ATmega168.



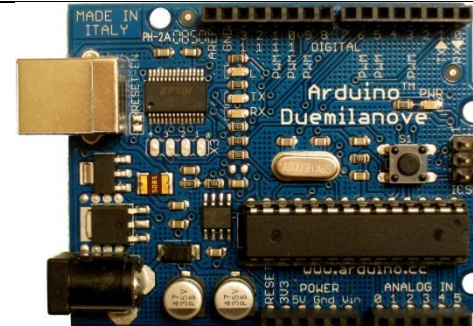
Το Arduino Bluetooth, με Bluetooth interface για προγραμματισμό χρησιμοποιώντας τεχνολογία ATmega168.



Το Arduino Diecimila, με ένα USB interface και χρησιμοποιεί τεχνολογία ATmega168 σε ένα DIP28 πακέτο.



Το Arduino Duemilavone (“2009”), χρησιμοποιεί τεχνολογία ATmega168 (ATmega328 για την καινούργια έκδοση) και τροφοδοτείται μέσω ενέργειας USB/DC, αυτόματα εναλλασσόμενης.



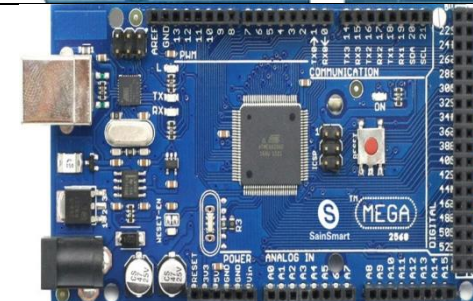
Το Arduino Mega, χρησιμοποιώντας τεχνολογία surface-mounted ATmega1280 για περαιτέρω I/O και μνήμη.



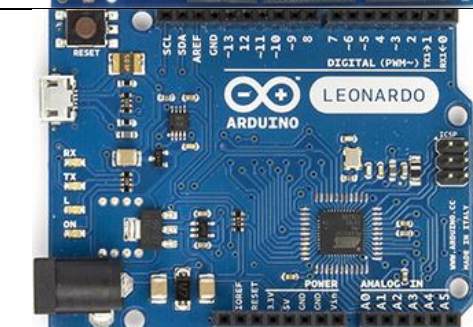
Το Arduino Uno, χρησιμοποιώντας την ίδια τεχνολογία ATmega328 όπως το τελευταίο μοντέλο Duemilavone, αλλά ενώ το Duemilavone χρησιμοποιεί ένα FTDI chipset για το USB, το Uno χρησιμοποιεί τεχνολογία ATmega8U2 προγραμματισμένο ως σειριακός μετατροπέας.



Το Arduino Mega2560, χρησιμοποιεί τεχνολογία surface-mounted ATmega2560 φέρνοντας την ολική μνήμη στα 256kB. Επίσης ενσωματώνει τη νέα τεχνολογία ATmega8U2 (ATmega16U2 σε αναθεώρηση τύπου 3) USB chipset.

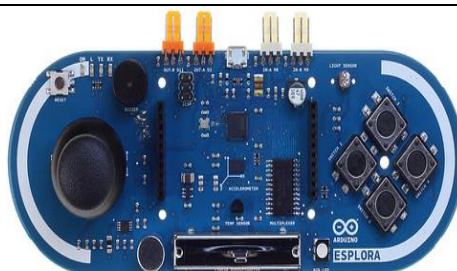


Το Arduino Leonardo, με ένα ATmega32U4 chip που εξαλείφει την ανάγκη για συνδεσιμότητα μέσω USB και μπορεί να χρησιμοποιηθεί ως ψηφιακό πληκτρολόγιο ή ποντίκι. Κυκλοφόρησε στο Maker Faire Bay Area το 2012.

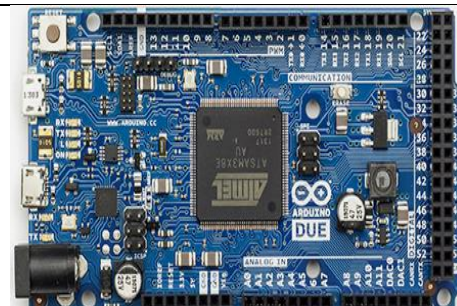




Το Arduino Esplora, με εμφάνιση που παραπέμπει σε χειριστήριο κονσόλας βιντεοπαιχνιδιών με joystick και ενσωματωμένους αισθητήρες για ήχο, φως, θερμοκρασία και επιτάχυνση



Το Arduino Due είναι ένα μικροχειριστήριο board βασισμένο στην τεχνολογία Atmel SAM3X8E ARM Cortex-M3 CPU. Είναι το πρώτο board της Arduino βασισμένη σε επεξεργαστή 32-bit ARM microcontroller.



### 2.8.1.4 Shields

Τα Arduino και τα Arduino συμβατά boards χρησιμοποιούν την τεχνολογία των shields, τυπωμένων boards επέκτασεων κυκλωμάτων που συνδέονται στα κανονικά παρεχόμενα Arduino pin-headers. Τα shields μπορούν να παρέχουν έλεγχο στα motors, GPS, Ethernet, LCD εικόνας ή breadboarding (προτυποποίησης).

## 2.9 Εργαλεία ανάπτυξης λογισμικού

### 2.9.1 XAMPP



Εικόνα 2.6: Λογότυπο XAMPP

Το XAMPP είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl.

Με βάση το πακέτο προγραμμάτων που περιέχει “βγήκε” και το όνομα του:

- X (αναφέρεται στο «cross-platform» που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας)
- Apache HTTP εξυπηρετητής
- MySQL
- PHP
- Perl

Το XAMPP συμπεριλαμβάνει επίσης τα πακέτα OpenSSL και το phpMyAdmin.

Αρχικά το XAMPP προοριζόταν για εργαλείο ανάπτυξης και δοκιμής ιστοσελίδων τοπικά στον υπολογιστή χωρίς να είναι απαραίτητη η σύνδεση στο διαδίκτυο. Για να είναι δυνατή η χρήση του, πολλές σημαντικές λειτουργίες ασφάλειας έχουν απενεργοποιηθεί. Στην πράξη το XAMPP ορισμένες φορές χρησιμοποιείται και για την φιλοξενία ιστοσελίδων. Υπάρχει ειδικό εργαλείο το οποίο περιέχεται στο XAMPP για την προστασία με κωδικό των σημαντικών μερών. Το XAMPP υποστηρίζει την δημιουργία και διαχείριση βάσεων δεδομένων τύπου MySQL και SQLite.

Όταν το XAMPP εγκατασταθεί στον τοπικό υπολογιστή διαχειρίζεται τον localhost ως ένα απομακρυσμένο κόμβο, ο οποίος συνδέεται με το πρωτόκολλο μεταφοράς αρχείων FTP. Η σύνδεση στον localhost μέσω του FTP μπορεί να γίνει με το όνομα χρήστη «newuser» και το κωδικό «wamprr». Για την βάση δεδομένων MySQL υπάρχει ο χρήστης «root» χωρίς κωδικό πρόσβασης.

## 2.9.2 WinSCP

Το WinSCP (Windows Secure Copy) είναι ένα δωρεάν και ανοιχτού κώδικα SFTP, FTP, WebDAV και SCP client για τα Microsoft Windows. Η κύρια λειτουργία του είναι η ασφαλής μεταφορά αρχείων μεταξύ ενός τοπικού και ενός απομακρυσμένου υπολογιστή. Πέρα από αυτό, το WinSCP προσφέρει βασικό διαχειριστή αρχείων και συγχρονισμό αρχείων. Για την ασφαλή μεταφορά, χρησιμοποιεί Secure Shell (SSH) και υποστηρίζει το πρωτόκολλο SCP εκτός από την SFTP.

Η ανάπτυξη του WinSCP άρχισε γύρω στον Μάρτιο του 2000 και συνεχίζεται. Αρχικά φιλοξενήθηκε από το Οικονομικό Πανεπιστήμιο στην Πράγα, όπου ο συντάκτης του εργαζόταν εκεί εκείνη την εποχή. Από 16 Ιουλίου του 2003, είναι υπό την άδεια GNU GPL και φιλοξενείται στο SourceForge.net.

Το WinSCP είναι βασισμένο στην εφαρμογή του πρωτοκόλλου SSH από το PuTTY και το FTP πρωτόκολλο από FileZilla. Είναι επίσης διαθέσιμο ως plugin για το διαχειριστή αρχείων Altap Salamander, και υπάρχει και ένα third-party plugin για το διαχειριστή αρχείων FAR.

Μπορείτε να “κατεβάσετε” την εφαρμογή «WinSCP» ελεύθερα από την επίσημη ιστοσελίδα του.

## 2.9.3 Putty

Η εφαρμογή «PUTTY» μας επιτρέπει να συνδεθούμε και να διαχειριστούμε ένα απομακρυσμένο λειτουργικό που υποστηρίζει το πρωτόκολλο SSH (Secure Shell), όπως για παράδειγμα ένα Unix λειτουργικό σύστημα. Με το SSH πρωτόκολλο, δημιουργούμε ένα ασφαλές κανάλι επικοινωνίας μεταξύ εμάς και του απομακρυσμένου λειτουργικού συστήματος έτσι ώστε κατά την διάρκεια της σύνδεσής μας, κανείς κακόβουλος ενδιάμεσος να μπορεί να «διαβάσει» τα δεδομένα που ανταλλάσσουμε με το απομακρυσμένο σύστημα.

Μπορείτε να κατεβάσετε δωρεάν την εφαρμογή “putty.exe” από την επίσημη ιστοσελίδα της.

## 2.9.4 Notepad++

Το Notepad++ είναι ένας επεξεργαστής κειμένου (text editor). Είναι ένα εργαλείο συγγραφής κώδικα, το οποίο είναι ελαφρύ πρόγραμμα με πάρα πολλές δυνατότητες.

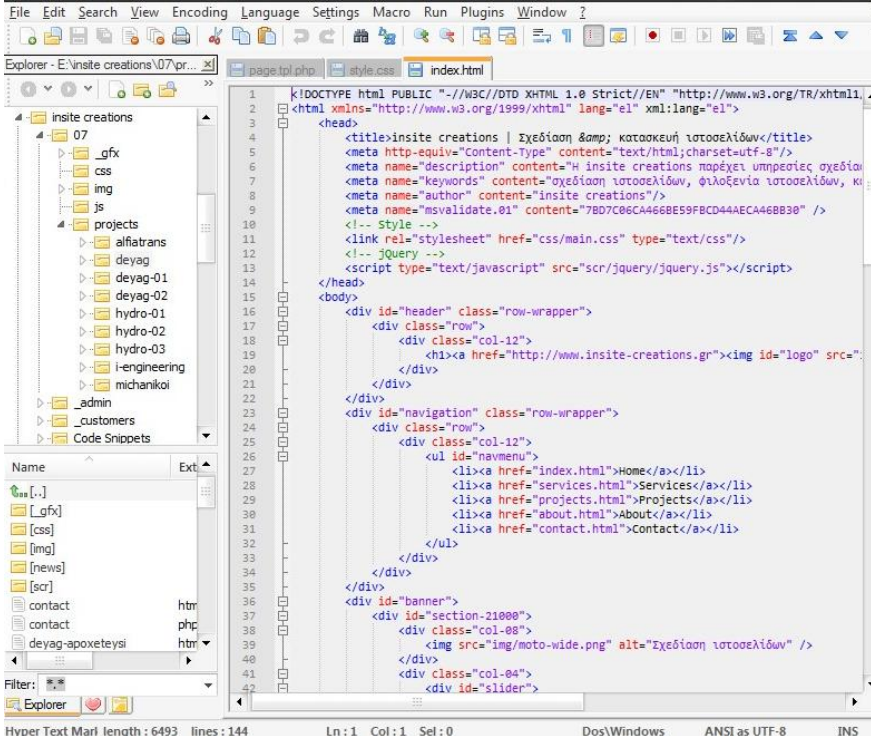
Το βασικό του χαρακτηριστικό είναι το syntax highlighting, το οποίο χρωματίζει τον κώδικα ανάλογα με την δομή του και διευκολύνει την ανάπτυξη του. Για κάθε γλώσσα προγραμματισμού υπάρχει διαφορετικό χρωματικό στυλ. Το πρόγραμμα έχει έτοιμα στυλ για πολλές γλώσσες αλλά δίνει και την δυνατότητα κάποιος να τα ρυθμίσει στις ανάγκες του ή να φτιάξει και καινούργια.

Επίσης δίνει την δυνατότητα επεξεργασίας σε πολλά έγγραφα συγχρόνως τα οποία ανοίγουν σε ξεχωριστές καρτέλες έτσι ώστε να μπορούν επεξεργαστούν τα αρχεία ενός ολόκληρου project. Επιπλέον χάρη στο multi view έχει τη δυνατότητα απεικόνισης και επεξεργασίας δύο εγγράφων συγχρόνως.

Ένα πολύ δυνατό χαρακτηριστικό του προγράμματος είναι οι εκτεταμένες δυνατότητες εύρεσης κειμένου στο αρχείο που είναι ανοικτό ή στα αρχεία ενός ολόκληρου φακέλου. Εκτός από τον κλασικό τρόπο έρευνας μπορεί να χρησιμοποιήσει και regular expressions για ακόμα πιο ακριβή αποτελέσματα.

Το πρόγραμμα υποστηρίζει την χρήση plugins, τα οποία επεκτείνουν τις δυνατότητες του. Υπάρχουν πολλά τέτοια plugins για να κατεβάσετε και να στήσετε το πρόγραμμα όπως σας βολεύει.

Επιπλέον προσφέρει την δυνατότητα καταγραφής και εκτέλεσης μακροεντολών, ώστε να αυτοματοποιήσετε τις εργασίες που εκτελείτε συχνά.



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1
2 <html xmlns="http://www.w3.org/1999/xhtml" lang="el" xml:lang="el">
3
4 <head>
5 <title>insite creations | Σχεδίαση &amp; κατασκευή ιστοσελίδων</title>
6 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
7 <meta name="description" content="Η insite creations παρέχει υπηρεσίες σχεδία
8 <meta name="keywords" content="σχεδίαση ιστοσελίδων, φιλοξενία ιστοσελίδων, κ
9 <meta name="author" content="insite creations"/>
10 <meta name="msvalidate.01" content="78D7C06CA466BE59FBCD44AEC4468B30" />
11 <!-- Style -->
12 <link rel="stylesheet" href="css/main.css" type="text/css"/>
13 <!-- jquery -->
14 <script type="text/javascript" src="scr/jquery/jquery.js"></script>
15 </head>
16 <body>
17 <div id="header" class="row-wrapper">
18 <div class="row">
19 <div class="col-12">
20 <h1><a href="http://www.insite-creations.gr">
24 <div class="row">
25 <div class="col-12">
26 <ul id="navmenu">
27 <li><a href="index.html">Home</a></li>
28 <li><a href="services.html">Services</a></li>
29 <li><a href="projects.html">Projects</a></li>
30 <li><a href="about.html">About</a></li>
31 <li><a href="contact.html">Contact</a></li>
32 </ul>
33 </div>
34 </div>
35 </div>
36 <div id="banner">
37 <div id="section-21000">
38 <div class="col-08">
39 
40 </div>
41 <div class="col-04">
42 <div id="slider">
```

Εικόνα 2.7 Περιβάλλον Notepad++

## 2.9.5 Λογισμικό Arduino IDE 1.6.4

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει τον προγραμματισμό στους καλλιτέχνες και τους νέους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστείτε αρχεία make ή να τρέξετε προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται *σκίτσο* (sketch).

### 2.9.5.1 Γλώσσα του Arduino

Τα Arduino προγράμματα είναι γραμμένα σε C ή C++. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring", από το πρωτότυπο σχέδιο Wiring, γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες. Οι χρήστες πρέπει μόνο να ορίσουν δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:

-`setup()`:μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις

-`loop()`:μία συνάρτηση που καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί

Ένα τυπικό πρώτο πρόγραμμα για έναν μικροελεγκτή αναβοσβήνει απλά ένα LED. Στο περιβάλλον του Arduino, ο χρήστης μπορεί να γράψει ένα πρόγραμμα σαν αυτό:

```
#define LED_PIN 13

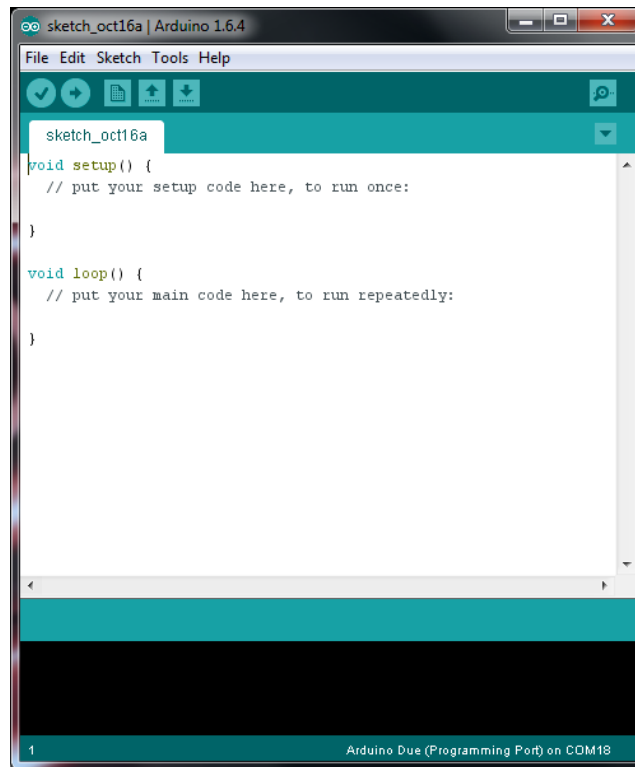
void setup () {
  pinMode (LED_PIN, OUTPUT); // δηλώνουμε το pin 13 για ψηφιακή έξοδο
}

void loop () {
  digitalWrite (LED_PIN, HIGH); // ανάβουμε το LED
  delay (1000); // καθυστέρηση 1 second (1000 milliseconds)
  digitalWrite (LED_PIN, LOW); // σβήνουμε το LED
  delay (1000); // καθυστέρηση 1 second
}
```

Ένα χαρακτηριστικό των περισσότερων πλακετών Arduino είναι ότι έχουν ένα LED και μία αντίσταση φορτίου που συνδέονται μεταξύ του pin 13 και του εδάφους, ένα βολικό χαρακτηριστικό για πολλά απλά τεστ. Ο προηγούμενος κώδικας δεν θα αναγνωριστεί από ένα κανονικό μεταγλωττιστή C++ ως έγκυρο πρόγραμμα, έτσι ώστε όταν ο χρήστης κάνει κλικ στο κουμπί "Upload to I / O board" στο IDE, ένα αντίγραφο του κώδικα θα γραφτεί σε ένα προσωρινό αρχείο με ένα παραπάνω include στην κορυφή και μία πολύ απλή συνάρτηση `main()` στο τέλος, για να φτιάξει ένα έγκυρο C++ πρόγραμμα.

Το IDE του Arduino χρησιμοποιεί το GNU toolchain και το AVR Libc για να μεταγλωττίζει προγράμματα και το avrdude για να φορτώνει προγράμματα στην πλακέτα.

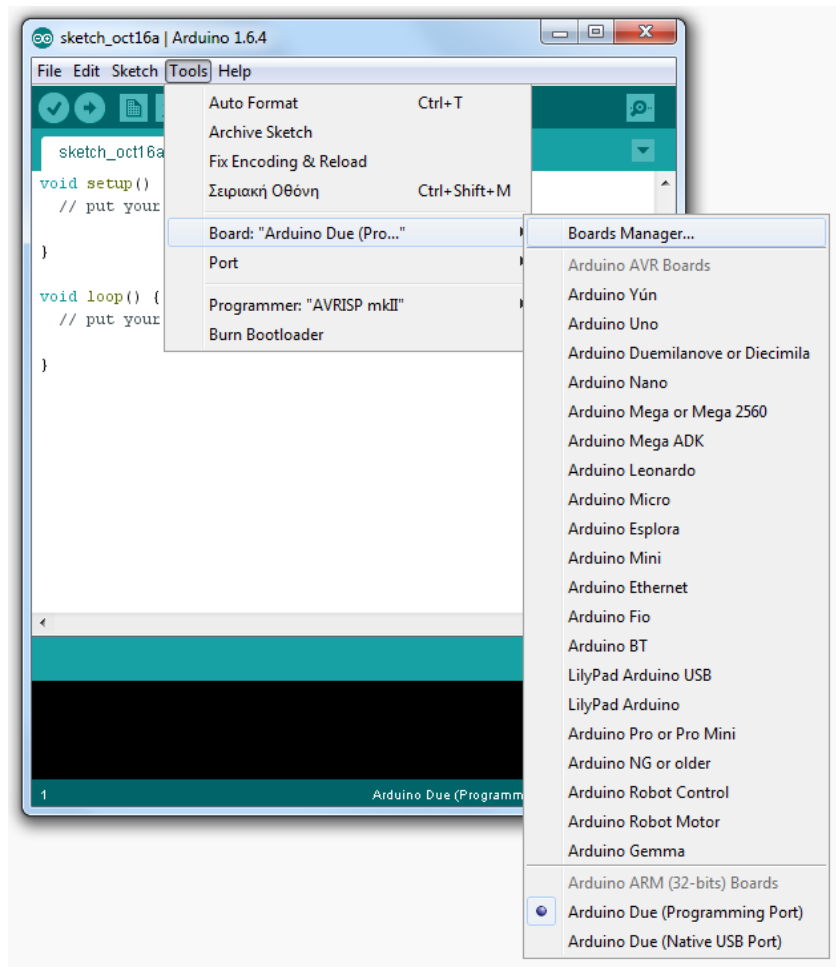
Αφού κατεβάσουμε το Arduino ide στον υπολογιστή μας το ανοίγουμε και μας εμφανίζεται η παρακάτω εικόνα:



Εικόνα 2.8: Περιβάλλον Arduino IDE 1.6.4

Όπως βλέπουμε το Arduino IDE μας έχει ήδη έτοιμη την μορφή του προγράμματος με τις δυο υποχρεωτικές συναρτήσεις.

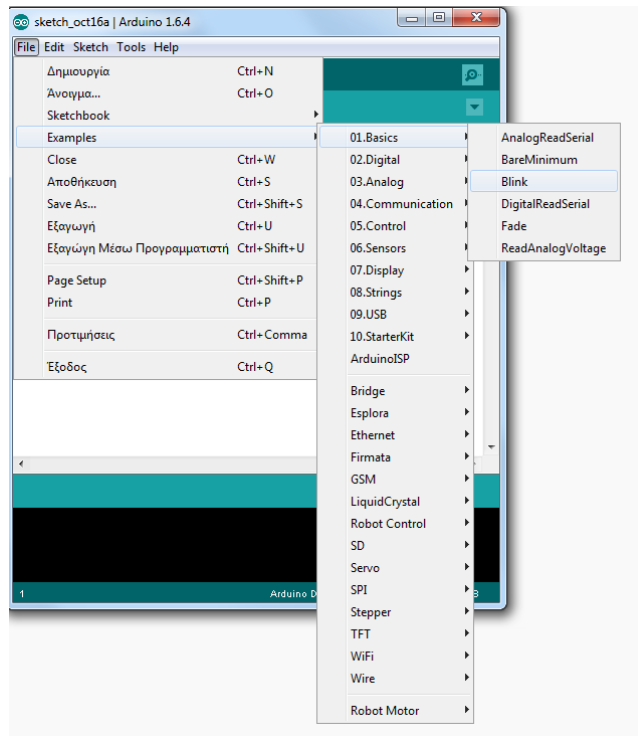
Αρχικά πρέπει να δηλώσουμε στο Arduino IDE την πλακέτα του Arduino που χρησιμοποιούμε, για να το κάνουμε αυτό πηγαίνουμε Tools -> Board > και επιλέγουμε την πλακέτα μας. Σε περίπτωση που η πλακέτα μας δεν βρίσκεται σε αυτή την λίστα (όπως στην περίπτωση μας για το Arduino DUE) τότε κάνουμε κλικ στο "Board Manager..." και ψάχνουμε ή γράφουμε στην αναζήτηση το όνομα της πλακέτας Arduino που διαθέτουμε, για να κατεβάσουμε τους απαραίτητους drivers.



Εικόνα 2.9: Επιλογή πλακέτας στο Arduino IDE 1.6.4

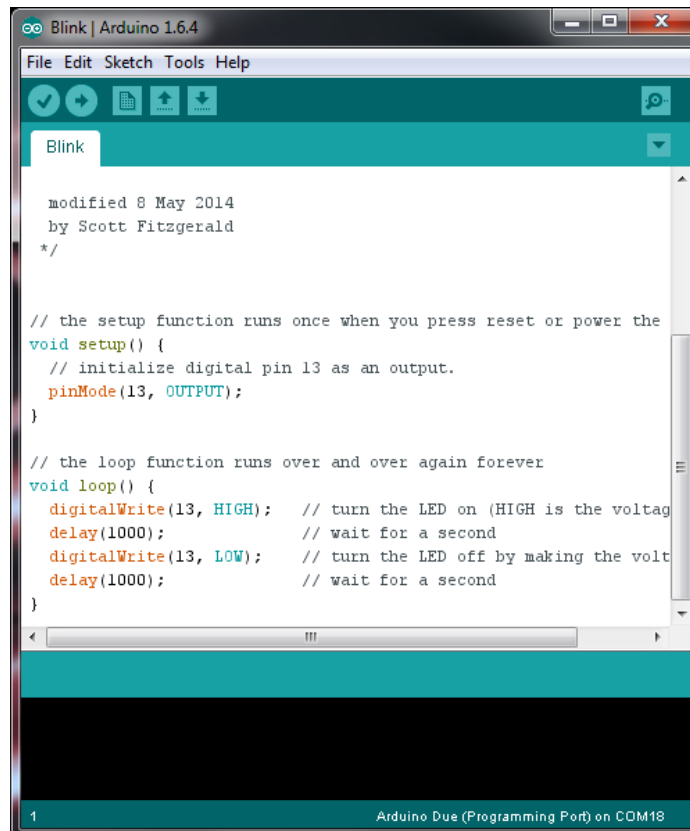
Αφού ολοκληρωθεί η εγκατάσταση των drivers μετά πηγαίνουμε και δηλώνουμε την θύρα στην οποία έχουμε συνδέσει το Arduino. Tools-> Port .

Τώρα μπορούμε να γράψουμε ένα πρόγραμμα ή να δοκιμάσουμε κάποιο από τα υπάρχον παραδείγματα. Για να το κάνουμε αυτό ακολουθούμε την διαδρομή που φαίνεται στην παρακάτω εικόνα:



Εικόνα 2.10: Άνοιγμα παραδείγματος

Και θα εμφανιστεί η παρακάτω εικόνα με το πρόγραμμα που ανάβει και σβήνει ένα LED που είδαμε παραπάνω:



Εικόνα 2.11: Παράδειγμα sketch

Τώρα αφού γράψαμε/φορτώσαμε το πρόγραμμα μας το Arduino IDE μας παρέχει τις εξής δυνατότητες:



Εικόνα 2.12: Εργαλειοθήκη

1. Verify / Compile, ελέγχει για συντακτικά λάθη στον κώδικα μας.
2. Upload, μεταγλωττίζει τον κώδικα μας και τον φορτώνει στο Arduino.
3. New, δημιουργεί ένα νέο sketch.
4. Open, μας επιτρέπει να ανοίξουμε κάποιο από τα υπάρχων sketch.
5. Save, αποθηκεύει ένα sketch.
6. Serial Monitor, ανοίγει την σειριακή οθόνη.

### 2.9.5.2 Δομή προγράμματος

Ένα τυπικό πρόγραμμα του Arduino έχει την εξής δομή:

```
sketch_oct17a §
// δηλώσεις μεταβλητών
void setup() {
  // αρχικοποιήσεις
}

void loop() {
  // το βασικό πρόγραμμα που επαναλαμβάνεται
}
```

Εικόνα 2.13; Δομή προγράμματος



Υπάρχουν δύο βασικές συναρτήσεις σε ένα τυπικό πρόγραμμα, η συνάρτηση `setup` και η συνάρτηση `loop`.

Η συνάρτηση `setup()` εκτελείται στην αρχή του προγράμματος και για μία μόνο φορά. Χρησιμοποιείται για τις αρχικοποιήσεις των μεταβλητών, τις δηλώσεις των `pin` (αν θα είναι είσοδος ή έξοδος) και τις αρχικοποιήσεις των βιβλιοθηκών.

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

### 2.9.5.3 Μεταβλητές

Μεταβλητή στις γλώσσες προγραμματισμού γενικά ονομάζουμε ένα γλωσσικό αντικείμενο που μπορεί να λάβει διάφορες τιμές, μία κάθε φορά. Οι τιμές μιας μεταβλητής περιορίζονται συνήθως σε ένα τύπο δεδομένων.

Οι βασικοί τύποι δεδομένων στο Arduino είναι:

1. `byte`: αποθηκεύει μια αριθμητική τιμή 8-bit χωρίς δεκαδικά ψηφία, παίρνουν τιμές από 0 μέχρι 255.
2. `int`: ακραίοι, παίρνουν τιμές από -32,768 μέχρι 32767.
3. `long`: μεγάλοι μεγέθους ακαριαίοι, παίρνουν τιμές από -2,147,483,648 μέχρι 2,147,483,647
4. `float`: πραγματικοί αριθμοί, παίρνουν τιμές από  $3.4 \times 10^{-38}$  μέχρι  $3.4 \times 10^{38}$

Τις μεταβλητές μπορούμε να τις δηλώσουμε στην αρχή του προγράμματός μας:

```
int myvariable;
```

Μπορούμε επίσης να δώσουμε αρχική τιμή στη μεταβλητή ταυτόχρονα με τη δήλωσή της:

```
int myvariable = 47;
```

### 2.9.5.4 Σταθερές

Σταθερές είναι αντικείμενα τα οποία παίρνουν μόνο μία τιμή, και δηλώνονται μαζί με τις μεταβλητές:

```
#define ledPin 13
```

### 2.9.5.5 Πίνακες – Arrays

Πίνακα ονομάζουμε διάταξη δεδομένων μιας ή περισσότερων διαστάσεων η οποία είναι συγκεκριμένου τύπου δεδομένων. Για παράδειγμα αν έχουμε ένα πίνακα ακεραίων 5 θέσεων τον οποίο ονομάζουμε `myarray` τον δηλώνουμε όπως βλέπουμε παρακάτω:

```
int myarray[5];
```

Για να δώσουμε τιμή στο τέταρτο στοιχείο του πίνακα `myarray` γράφουμε:

```
myarray[3]=12;
```

επίσης μπορούμε να γεμίσουμε τον πίνακα ταυτόχρονα με την δήλωσή του:

```
int myarray[]={12,45,32,61,55};
```

### 2.9.5.6 Αριθμητικοί τελεστές

Οι αριθμητικοί τελεστές καλύπτουν τις βασικές πράξεις: πρόσθεση, αφαίρεση, πολλαπλασιασμό, διαίρεση (+, -, \*, /). Για παράδειγμα μπορούμε να κάνουμε την πρόσθεση δύο ακέραιων και το αποτέλεσμα να εκχωρηθεί σε μία μεταβλητή:

```
sum = 458 + 954;
```

### 2.9.5.7 Τελεστές σύγκρισης

Με τους τελεστές σύγκρισης μπορούμε να ελέγξουμε αν μία συγκεκριμένη συνθήκη μεταξύ μεταβλητών ή σταθερών είναι “Αληθής”. Ποιοι συγκεκριμένα υπάρχουν οι παρακάτω τελεστές σύγκρισης στο Arduino:

Πίνακας 2.2: Τελεστές Σύγκρισης

<code>x == y</code>	το x είναι ίσο με το y
<code>x != y</code>	το x είναι άνισο του y
<code>x &lt; y</code>	το x είναι μικρότερο με το y
<code>x &gt; y</code>	το x είναι μεγαλύτερο με το y
<code>x &lt;= y</code>	το x είναι μικρότερο ή ίσο με το y
<code>x &gt;= y</code>	το x είναι μεγαλύτερο ή ίσο με το y

### 2.9.5.8 Λογικοί τελεστές

Με τους λογικούς τελεστές μπορούμε να συγκρίνουμε δύο ή περισσότερες εκφράσεις, δίνοντας αποτέλεσμα “Αληθής” ή “Ψευδής”. Υπάρχουν τρεις λογικοί τελεστές:

Πίνακας 2.3: Λογικοί τελεστές

Λογικό ΚΑΙ	<code>&amp;&amp;</code> - επιστρέφει “Αληθής” αν όλες οι εκφράσεις είναι “Αληθείς”
Λογικό Ή	<code>   </code> - επιστρέφει “Αληθής” αν μία από τις εκφράσεις είναι “Αληθείς”
Λογικό ΟΧΙ	<code>!</code> - επιστρέφει “Αληθής” αν η έκφραση είναι “Ψευδής”

Παράδειγμα:

```
if(x >0&& x <5){
    //κώδικας
}
```

Στο παραπάνω κομμάτι κώδικα γίνεται έλεγχος αν το x είναι μεγαλύτερο από το 0

ΚΑΙ μικρότερο από 5 τότε εκτελείται ο κώδικας που βρίσκεται μέσα στις αγκύλες. Με λίγα λόγια η πρόταση `if()` ελέγχει αν η συνθήκη μέσα στις παρενθέσεις είναι “Αληθής”.

Ένα άλλο παράδειγμα:

```
if(!x>0){  
    //κώδικας  
}
```



Εικόνα 2.14: Λογότυπο AMCHARTS

Εδώ γίνεται έλεγχος αν το  $x$  είναι μεγαλύτερο από 0, αν αυτή η συνθήκη ΔΕΝ ισχύει τότε έχουμε το αποτέλεσμα “Αληθής” και εκτελείται ο κώδικας μέσα στις αγκύλες.

## 2.10 AmCharts: JavaScript Charts & Maps

Τα AmCharts είναι μία αυτόνομη και ανεξάρτητη προηγμένη βιβλιοθήκη που παρέχει δωρεάν γραφήματα και διαγράμματα με διαφορετικά στυλ όπως στήλη, πίτα, πυραμίδα κ.α. Ένα σημαντικό χαρακτηριστικό της εφαρμογής είναι ότι μπορούν προστεθούν στα διαγράμματα της ιδιότητες όπως μεγέθυνση, χρώματα, τρόπο απεικόνισης στυλ εμφάνισης κ.α.

Πρόκειται για μία free βιβλιοθήκη όπου δίνει και τα επί πληρωμή προϊόντα της να χρησιμοποιηθούν δωρεάν αρκεί να αναγράφεται ένα μικρό link στα γραφήματα.

Επίσης η εφαρμογή διαθέτει έναν live editor στον browser επιτρέποντας στους χρήστες της να δημιουργίσουν με ευκολία γραφήματα, εισάγωντας απλά δεδομένα χωρίς να χρειάζεται ιδιαίτερη γνώση προγραμματισμού.

Επιπλέον στην εφαρμογή υπάρχουν χρηματιστηριακά και οικονομικά γραφήματα και διαγράμματα που λειτουργούν σε πραγματικό χρόνο, οπότε μας επιτρέπει να προσθέσουμε νέες γραφικές χωρίς καινούριο κώδικα.

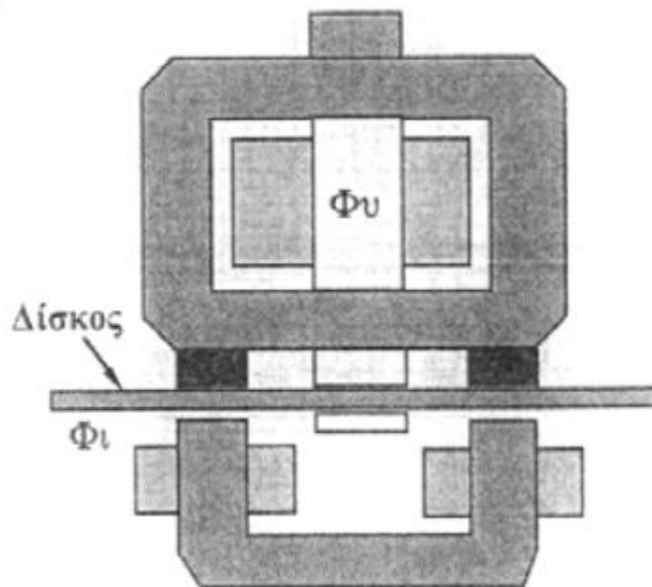
## ΚΕΦΑΛΑΙΟ 3 : Υλικό Μέρος Συστήματος

Σε αυτό το κεφάλαιο θα αναλύσουμε όλα τα υλικά που χρησιμοποιήσαμε για την υλοποίηση του συστήματος μας. Συγκεκριμένα χρησιμοποιήσαμε έναν Αναλογικό μετρητή της Δ.Ε.Η. , το Arduino Due, ένα Ethernet Shield (για το arduino), τον αισθητήρα TCRT5000, το Bizy Type F plug ( ή αλλιώς πρίζα της Meazon) και το Meazon Gateway Advanced ( ή αλλιώς BeagleBone της Meazon).

### 3.1 Αναλογικός Μετρητής Ενέργειας

Ο αναλογικός μετρητής ενέργειας που χρησιμοποιήθηκε για την διεξαγωγή της παρούσας εργασίας είναι ένας επαγωγικός μετρητής. Αποτελείται από δύο σταθερούς ηλεκτρομαγνήτες οι οποίοι είναι ο αμπερομετρικός και ο βολτομετρικός. Αυτοί οι ηλεκτρομαγνήτες επενεργούν πάνω στον δίσκο που βλέπουμε εμείς έξω από το ρολόι ο οποίος δίσκος αυτός είναι αλουμινένιος.

Από τον αμπερομετρικό ηλεκτρομαγνήτη δημιουργείται μια ροή  $\Phi_i$  ανάλογη προς την ένταση που απορροφάται από τον καταναλωτή, ενώ από τον βολτομετρικό ηλεκτρομαγνήτη δημιουργείται μια ροή  $\Phi_u$  ανάλογη προς την τάση που εφαρμόζουμε στον καταναλωτή. Οι δυο μαγνητικές ροές ( $\Phi_i$  και  $\Phi_u$ ) λόγω της κατασκευής τους, βρίσκονται σε φασική απόκλιση μεταξύ τους και δημιουργούν ένα συνιστάμενο στρεφόμενο μαγνητικό πεδίο.



Εικόνα 3.1: Σχέδιο Αναλογικού Μετρητή

$\Phi_u$  = μαγνητική ροή συστήματος τάσης.  
 $\Phi_i$  = μαγνητική ροή συστήματος έντασης.

Έτσι αναπτύσσονται στο δίσκο δινορεύματα, πάνω στα οποία ασκούνται δυνάμεις Laplace και έχουμε ως αποτέλεσμα την περιστροφή.

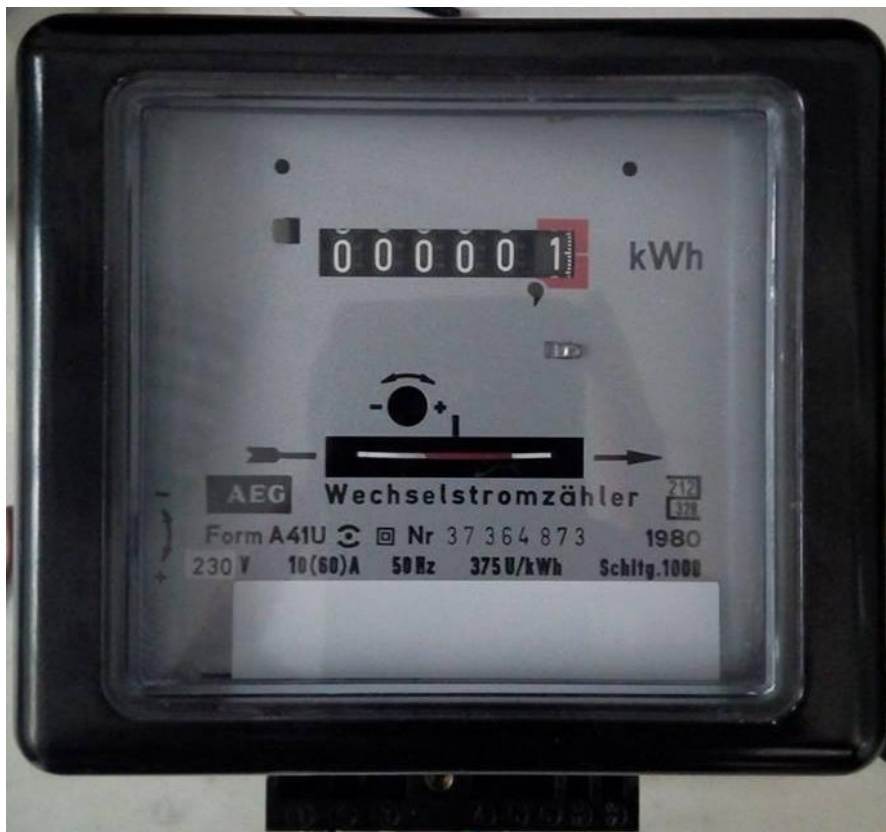
Στο δίσκο του οργάνου ασκείται μία μικρή ροπή και όταν δεν έχουμε συνδεδεμένη κατανάλωση στο όργανο. Για να αποφύγουμε αυτή τη μικρή ροπή στρέψης που αναπτύσσεται, τοποθετούμε στο όργανο ένα μικρό μόνιμο μαγνήτη και ένα μικρό σιδερένιο κομμάτι στον άξονα.

Καθώς ο δίσκος γυρνάει ανάλογα με τις στροφές που κάνει έχουμε κατανάλωση ηλεκτρικού ρεύματος (πχ. 375 στροφές είναι 1kWh).

Ο αλουμινένιος δίσκος έχει ένα σημείο του βαμμένο (συνήθως μαύρο) αυτό χρησιμοποιούμε για να μετρήσουμε τις στροφές που κάνει και αυτό χρησιμοποιεί και ο ανακλαστικός αισθητήρας. Όταν περνάει αυτό το σημείο από μπροστά ο αισθητήρας μας διαβάζει διαφορετική τιμή απ' ότι συνήθως (μεγαλύτερη απ' όταν είναι στο ασημένιο κομμάτι) .

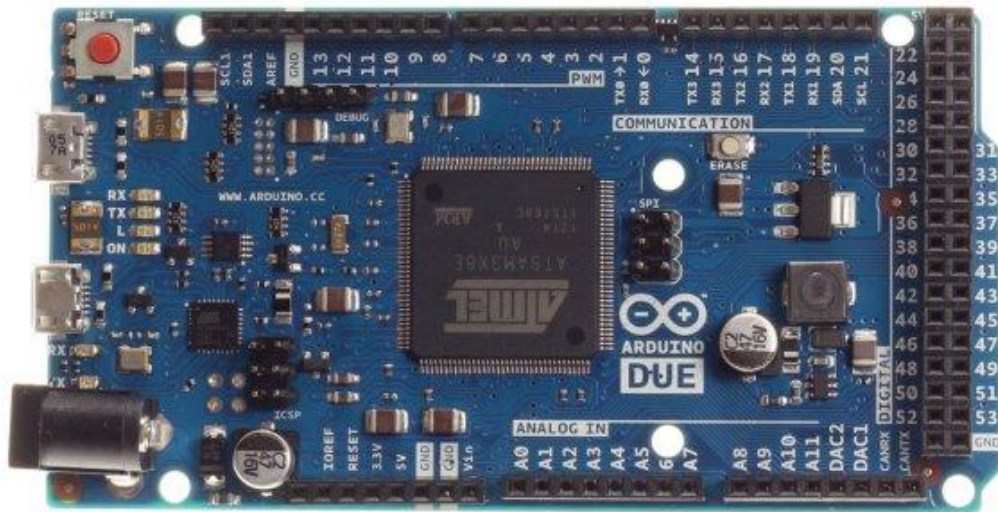
Στην παρούσα εργασία χρησιμοποιήθηκε ο μονοφασικός μετρητής της AEG. Τα χαρακτηριστικά του οποίου είναι :

- Τάση Λειτουργίας: 230V
- Συχνότητα Λειτουργίας : 50Hz
- Μέγιστο Ρεύμα: 60A
- 375U/kWh



Εικόνα 3.2: Αναλογικός Μετρητής

## 3.2 Arduino DUE



Εικόνα 3.3: Arduino DUE

Για την υλοποίηση του συστήματος επιλέχθηκε η πλακέτα Arduino DUE λόγω της υψηλής ταχύτητας του επεξεργαστή. Το Arduino DUE είναι ένας μικροελεγκτής με ενσωματωμένο τον επεξεργαστή Atmel SAM3X8E ARM Cortex-M3. Είναι η πρώτη πλακέτα Arduino που χρησιμοποιεί 32-bit ARM μικροεπεξεργαστή.

### 3.2.1 Τεχνικά χαρακτηριστικά Arduino DUE

Πίνακας 3.1: Τεχνικά χαρακτηριστικά Arduino DUE

Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-16V
Digital I/O Pins	54 (of which 12 provide PWM output)
Analog Input Pins	12
Analog Output Pins	2 (DAC)
Total DC Output Current on all I/O lines	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512 KB all available for the user applications

SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz
Length	101.52 mm
Width	53.3 mm
Weight	36 g

Σε αντίθεση με τις περισσότερα Arduino, η πλακέτα Arduino DUE λειτουργεί με 3.3V. Η μέγιστη τάση που οι I / O ακίδες μπορούν να αντέξουν είναι 3.3V. Εφαρμόζοντας τάσεις υψηλότερες από 3.3V σε καθένα από τα I / O pins θα μπορούσε να βλάψει την ολη την πλακέτα.

Η πλακέτα περιέχει όλα όσα χρειάζονται για τις ανάγκες του μικροελεγκτή. Μπορούμε να τροφοδοτήσουμε την πλακέτα ή σε έναν υπολογιστή με ένα καλώδιο micro-USB ή με ένα προσαρμογέα AC-σε-DC ή με μια μπαταρία. Το Due είναι συμβατό με όλα τα Arduino Shields που λειτουργούν σε 3.3V και είναι συμβατά με το Arduino pinout 1,0.

### 3.2.3 Τροφοδοσία Arduino

Το Arduino Due μπορεί να τροφοδοτηθεί μέσω σύνδεσης USB ή με εξωτερικό τροφοδοτικό ρεύματος. Η πηγή τροφοδοσίας επιλέγεται αυτόματα. Η εξωτερική (non-USB) τροφοδοσία μπορεί να προέρχεται ή από ένα AC-to-DC adapter ή από μπαταρία. Ο adapter μπορεί να συνδεθεί με ένα 2.1mm center-positive βύσμα στην υποδοχή ρεύματος της πλακέτας.

Τα pin τροφοδοσίας είναι τα ακόλουθα:

- **Vin**: Η εισερχόμενη τάση στην πλακέτα Arduino όταν χρησιμοποιούμε εξωτερική πηγή τροφοδοσίας
- **5V**: Το pin αυτό εξάγει μία ρυθμιζόμενη τάση 5V από τον ρυθμιστή στην πλακέτα. Η πλακέτα μπορεί να τροφοδοτηθεί είτε από το DC jack τροφοδοσίας(7 - 12V), είτε μέσω USB σύνδεση(5V), είτε από την ακίδα VIN της πλακέτας(7 - 12V). Δίνοντας τροφοδοσία μέσω των pin 5V ή 3.3V παρακάμπτοντας τον ρυθμιστή, μπορεί προκληθεί ζημιά στην πλακέτα οπότε δεν συνίσταται αυτό.
- 3V3. Μια τάση τροφοδοσίας 3.3Volt δημιουργείται από τον ρυθμιστή που βρίσκεται πάνω στην πλακέτα.. Η μέγιστη κατανάλωση ρεύματος είναι 800mA. Αυτός ο ρυθμιστής παρέχει επίσης και την τροφοδοσία στο μικροελεγκτή SAM3X.
- GND. Ακίδες γείωσης.
- IOREF. Αυτή η ακίδα στην πλακέτα Arduino, μας παρέχει την τάση αναφοράς με την οποία λειτουργεί ο μικροελεγκτής. Μία σωστά ρυθμισμένη ασπίδα διαβάζει την τάση του pin IOREF και επιλέγει την κατάλληλη πηγή τροφοδοσίας ή ενεργοποιεί τους μεταφραστές τάσης στις εξόδους για να δουλέψει με 5V ή 3.3V

### 3.2.4 ΜΝΗΜΗ

Ο SAM3X έχει 512KB (2 μνήμες από 256KB η κάθε μία) μνήμη flash για να αποθηκεύει κώδικα. Ο bootloader είναι προκατασκευασμένος σε εργοστάσιο από την Amtel και αποθηκεύεται σε μία ξεχωριστή μνήμη ROM. Η διαθέσιμη SRAM είναι 96KB σε δύο γειτονικές τράπεζες των 64KB και 32KB. Όλη η διαθέσιμη μνήμη (Flash, RAM and ROM) μπορεί να προσπελαστεί άμεσα από τα διαμερίσματα διευθύνσεων χώρου.

Είναι δυνατόν να διαγράψει η μνήμη Flash από το SAM3X με το κουμπί διαγραφής που βρίσκεται πάνω στην πλακέτα. Αυτό θα αφαιρέσει το τρέχον φορτωμένο sketch από την MCU. Για διαγραφή, πατάμε και κρατάμε για μερικά δευτερόλεπτα το κουμπί διαγραφής ενώ η πλακέτα είναι ενεργοποιημένη.

### 3.2.5 ΕΙΣΟΔΟΙ & ΕΞΟΔΟΙ

- **Digital I/O: τα pin από το 0 μέχρι το 53.** Κάθε ένα από αυτά τα 54 ψηφιακά pin στο Due μπορούν να χρησιμοποιηθούν είτε ως είσοδος είτε ως έξοδος, χρησιμοποιώντας τις `pinMode()`, `digitalWrite()`, και `digitalRead()` συναρτήσεις. Λειτουργούν στα 3,3V. Κάθε pin μπορεί να παρέχει (πηγή) ρεύμα 3 mA ή 15 mA, ανάλογα με το pin, ή να συλλέγει (συλλέκτης) ρεύμα 6 mA ή 9 mA, ανάλογα με το pin. Έχουν επίσης μια εσωτερική αντίσταση pull-up (αποσυνδέετε από προεπιλογή) 100 KΩ. Επιπλέον, μερικά pin έχουν εξειδικευμένες λειτουργίες:

-**Serial: 0 (RX) and 1 (TX)**

-**Serial 1: 19 (RX) and 18 (TX)**

-**Serial 2: 17 (RX) and 16 (TX)**

-**Serial 3: 15 (RX) and 14 (TX)**

Χρησιμοποιούνται για τη λήψη (RX) και την διαβίβαση σειριακών δεδομένων (TX) TTL (με επίπεδο 3,3 V). Τα pin 0 και 1 συνδέονται με τις αντίστοιχα pin του ATmega16U2 USB-to-TTL Serial chip.

- **PWM: τα pins από το 2 έως το 13.** Παρέχουν 8-bit PWM έξοδο με την συνάρτηση `analogWrite()`. Η ανάλυση του PWM μπορεί να αλλάξει με τη συνάρτηση `analogWriteResolution()`.

- **SPI: SPI header (ICSP header on other Arduino boards).** Αυτά τα pin υποστηρίζουν την SPI επικοινωνία, χρησιμοποιώντας τη βιβλιοθήκη SPI.

- **CAN: CANRX and CANTX.** Αυτά τα pin υποστηρίζουν το πρωτόκολλο επικοινωνίας CAN, αλλά ακόμη δεν υποστηρίζονται από Arduino APIs.

- **"L" LED: 13** . Υπάρχει ένα ενσωματωμένο LED που συνδέεται με το ψηφιακό pin 13. Όταν το pin είναι HIGH, είναι LED είναι αναμμένο, όταν το pin είναι LOW, το LED είναι off. Είναι επίσης δυνατόν το LED να κάνει dim επειδή το η ψηφιακή ακίδα 13 είναι και μια έξοδος PWM.

- **TWI 1: 20 (SDA) and 21 (SCL)**

- **TWI 2: SDA1 and SCL1.** Υποστηρίζει TWI επικοινωνία χρησιμοποιώντας τη βιβλιοθήκη `Wire`. Οι SDA1 και SCL1 μπορούν να ελεγχθούν με τη χρήση της κλάσης `Wire1` που παρέχεται από τη βιβλιοθήκη `Wire`. Ενώ οι SDA και SCL έχουν εσωτερικές αντιστάσεις



pullup, οι SDA1 και SCL1 δεν έχουν. Είναι απαραίτητο να προστεθούν δύο αντιστάτες pullup για τις SDA1 και SCL1 γραμμές για να χρησιμοποιηθεί η Wire1.

- **Analog Inputs: Τα pin από το A0 έως το A11.** Το Due έχει 12 αναλογικές εισόδους, οι οποίες μπορούν να παρέχουν 12 bit ανάλυσης (δηλαδή 4096 διαφορετικές τιμές). Από προεπιλογή, η ανάλυση των μετρήσεων ορίζεται σε 10 bits, για συμβατότητα με άλλες πλακέτες Arduino. Μπορείς να αλλάξεις την ανάλυση του ADC με την συνάρτηση `analogReadResolution()`. Τα αναλογικά pin εισόδου στο Due μετράνε από το GND μέχρι την μέγιστη τιμή των 3.3V. Εφαρμόζοντας πάνω από 3.3V στα pin του Due θα προκληθεί ζημιά στο chip SAM3X. Η συνάρτηση `analogReference()` αγνοείται στο Due.

Το pin AREF είναι συνδεδεμένο με το αναλογικό pin αναφοράς SAM3X μέσω μιας γέφυρας αντίστασης. Για να χρησιμοποιηθεί το pin AREF, η αντίσταση BR1 πρέπει να καταργηθεί από το PCB.

- **DAC1 and DAC2**. Αυτά τα pin παρέχουν πραγματικές αναλογικές εξόδους με ανάλυση 12-bit (4096 επίπεδα) με τη συνάρτηση `analogWrite()`. Αυτά τα pin μπορούν να χρησιμοποιηθούν για να δημιουργήσουν μία έξοδο ήχου με τη χρήση της βιβλιοθήκης Audio. Να σημειωθεί ότι αυτό το DAC έχει φάσμα εξόδου στην πραγματικότητα μόνο από 0,55 V έως 2,75V.

Άλλα pin στην έξοδο:

- **AREF**. Τάση αναφοράς για τις αναλογικές εισόδους. Χρησιμοποιείται με `analogReference()`.

- **Reset**. Φέρνοντας αυτό το pin LOW κάνουμε επαναφορά τον μικροελεγκτή. Συνήθως χρησιμοποιείται για να προστεθεί ένα κουμπί reset στα shields που μπλοκάρουν αυτό που βρίσκεται στη πλακέτα.

### Arduino Due pin mapping table:

Πίνακας 3.2: Arduino DUE pin mapping table

Due Pin Number	SAM3X Pin Name	Mapped Pin Name	Max Output Current (mA)	Max Current Sink (mA)
0	PA8	RX0	3	6
1	PA9	TX0	15	9
2	PB25	Digital Pin 2	3	6
3	PC28	Digital Pin 3	15	9
4	connected to both PA29 and PC26	Digital Pin 4	15	9
5	PC25	Digital Pin 5	15	9
6	PC24	Digital Pin 6	15	9
7	PC23	Digital Pin 7	15	9

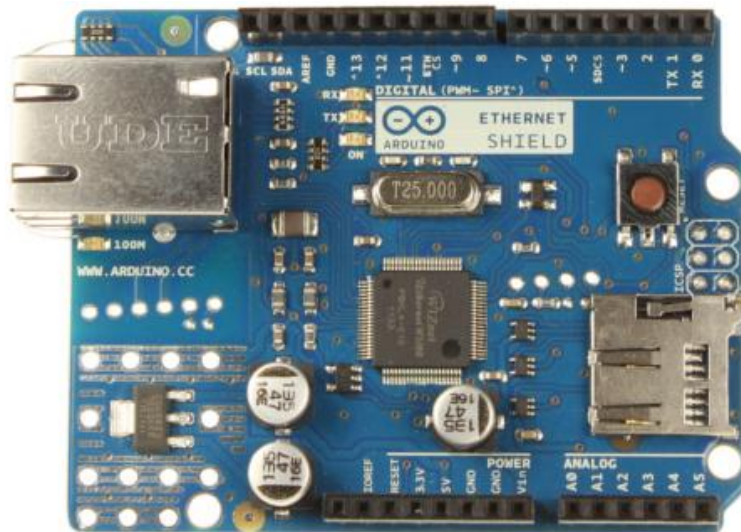
8	PC22	Digital Pin 8	15	9
9	PC21	Digital Pin 9	15	9
10	connected to both PA28 and PC29	Digital Pin 10	15	9
11	PD7	Digital Pin 11	15	9
12	PD8	Digital Pin 12	15	9
13	PB27	Digital Pin 13 / Amber LED "L"	3	6
14	PD4	TX3	15	9
15	PD5	RX3	15	9
16	PA13	TX2	3	6
17	PA12	RX2	3	6
18	PA11	TX1	3	6
19	PA10	RX1	3	6
20	PB12	SDA	3	6
21	PB13	SCL	3	6
22	PB26	Digital Pin 22	3	6
23	PA14	Digital Pin 23	15	9
24	PA15	Digital Pin 24	15	9
25	PD0	Digital Pin 25	15	9
26	PD1	Digital pin 26	15	9
27	PD2	Digital Pin 27	15	9
28	PD3	Digital Pin 28	15	9
29	PD6	Digital Pin 29	15	9
30	PD9	Digital Pin 30	15	9
31	PA7	Digital Pin 31	15	9
32	PD10	Digital Pin 32	15	9
33	PC1	Digital Pin 33	15	9
34	PC2	Digital Pin 34	15	9
35	PC3	Digital Pin 35	15	9
36	PC4	Digital Pin 36	15	9
37	PC5	Digital Pin 37	15	9
38	PC6	Digital Pin 38	15	9
39	PC7	Digital Pin 39	15	9
40	PC8	Digital Pin 40	15	9
41	PC9	Digital Pin 41	15	9
42	PA19	Digital Pin 42	15	9
43	PA20	Digital Pin 43	3	6

44	PC19	Digital Pin 44	15	9
45	PC18	Digital Pin 45	15	9
46	PC17	Digital Pin 46	15	9
47	PC16	Digital Pin 47	15	9
48	PC15	Digital Pin 48	15	9
49	PC14	Digital Pin 49	15	9
50	PC13	Digital Pin 50	15	9
51	PC12	Digital Pin 51	15	9
52	PB21	Digital Pin 52	3	6
53	PB14	Digital Pin 53	15	9
54	PA16	Analog In 0	3	6
55	PA24	Analog In 1	3	6
56	PA23	Analog In 2	3	6
57	PA22	Analog In 3	3	6
58	PA6	Analog In 4	3	6
59	PA4	Analog In 5	3	6
60	PA3	Analog In 6	3	6
61	PA2	Analog In 7	3	6
62	PB17	Analog In 8	3	6
63	PB18	Analog In 9	3	6
64	PB19	Analog In 10	3	6
65	PB20	Analog In 11	3	6
66	PB15	DAC0	3	6
67	PB16	DAC1	3	6
68	PA1	CANRX	3	6
69	PA0	CANTX	15	9
70	PA17	SDA1	3	6
71	PA18	SCL2	15	9
72	PC30	LED "RX"	15	9
73	PA21	LED "TX"	3	6
74	PA25	(MISO)	15	9
75	PA26	(MOSI)	15	9
76	PA27	(SCLK)	15	9
77	PA28	(NPCS0)	15	9
78	PB23	(unconnected)	15	9
USB	PB11	ID	15	9
USB	PB10	VBOF	15	9

### 3.2.6 USB Προστασία από Υπερένταση

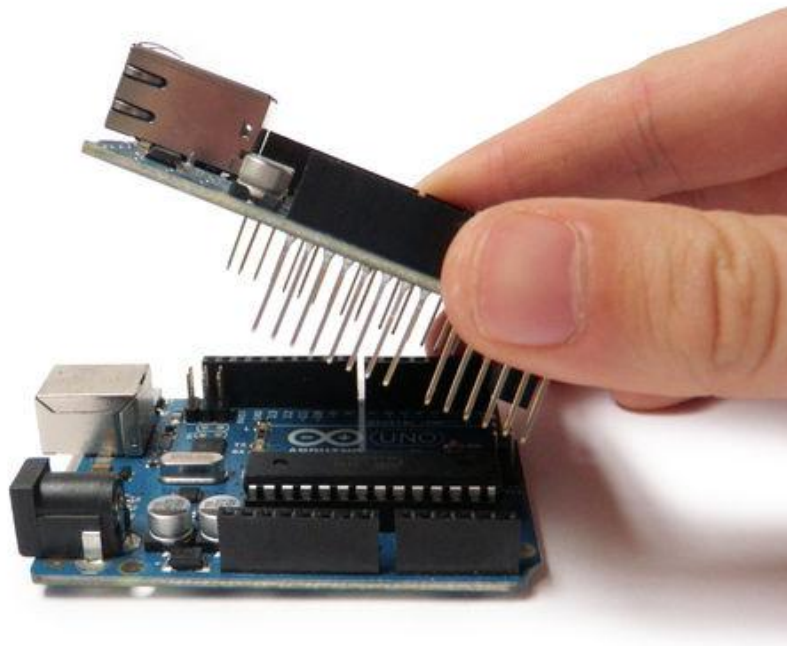
Το Arduino Due έχει δυνατότητα επαναφοράς polyfuse που προστατεύει τις θύρες USB του υπολογιστή μας από βραχυκυκλώματα και υπερεντάσεις. Παρόλο που οι περισσότεροι υπολογιστές παρέχουν τη δική τους εσωτερική προστασία, η ασφάλεια αυτή παρέχει ένα επιπλέον επίπεδο προστασίας. Εάν εφαρμοστούν περισσότερα από 500 mA στη θύρα USB, η ασφάλεια θα χαλάσει τη σύνδεση μέχρι το βραχυκύκλωμα ή η υπερφόρτωση αφαιρεθούν.

### 3.3 Ethernet Shield.



Εικόνα 3.4: Ethernet Shield

Τοποθετώντας πάνω στο Arduino το Ethernet Shield, τοποθετώντας ένα καλώδιο RJ45 (καλώδιο Ethernet) και γράφοντας λίγες γραμμές κώδικα μπορούμε να συνδέσουμε την πλακέτα μας στο διαδίκτυο, ώστε να στέλνουμε αλλά και να δεχόμαστε πληροφορίες.



Εικόνα 3.5: Τοποθέτηση Ethernet Shield

Το Ethernet Shield είναι βασισμένο στο Ethernet chip Wiznet W5100. Το W5100 παρέχει δίκτυο (IP) συμβατό με TCP και UDP. Υποστηρίζει μέχρι και τέσσερις ταυτόχρονες συνδέσεις socket. Απαραίτητη προϋπόθεση για να συνδεθούμε στο Internet, πέρα από τις λίγες επιπλέον γραμμές τον κώδικά μας, είναι και η χρήση της βιβλιοθήκης Ethernet. Το Ethernet Shield συνδέεται πάνω στο Arduino χρησιμοποιώντας μακριά σύρματα-ακίδες που εξέρχουν από το Shield και τοποθετούνται στα αντίστοιχα Pin του Arduino. Αυτός ο τρόπος επιτρέπει στη διάταξη των pin να παραμένει σχεδόν άθικτη και δίνει τη δυνατότητα να προσθέσουμε και να στοιβάξουμε κι άλλα Shields από πάνω.

Το Arduino Ethernet Shield έχει μια πρότυπη RJ-45 σύνδεση με ένα ενσωματωμένο μετασχηματιστή για τροφοδοσία μέσω Ethernet.

Επιπλέον υπάρχει ενσωματωμένη θέση για κάρτα μνήμης micro-SD, η οποία μπορεί να χρησιμοποιηθεί για αποθήκευση αρχείων για την εξυπηρέτηση μέσω του δικτύου. Είναι συμβατό με όλες τις Arduino/Genuino πλακέτες. Ο ενσωματωμένος αναγνώστης microSD μπορεί να γίνει προσβάσιμος μόνο με τη βιβλιοθήκη SD. Όταν εργαζόμαστε με αυτή τη βιβλιοθήκη το SS βρίσκεται στο pin 4.

Το Arduino επικοινωνεί και με το W5100 και με την κάρτα SD χρησιμοποιώντας το SPI bus (μέσω της ICSP κεφαλής). Αυτά βρίσκονται στα ψηφιακά pin 10,11,12 και 13 στο Uno και στα pin 50,51,52 στο Mega. Και στις δύο πλακέτες το pin 10 χρησιμοποιείται για να επιλέξει τον W5100 και το pin 4 για την SD κάρτα. Αυτά τα pin επομένως δεν μπορούν να χρησιμοποιηθούν για γενικές εισοδοί ή έξοδοι. Στο Mega, το υλικό SS pin 53 δεν χρησιμοποιείται ούτε για να επιλέξει τον W5100 ούτε την κάρτα SD, αλλά πρέπει να είναι πάντα δηλωμένο ως έξοδος διαφορετικά το περιβάλλον SPI δεν θα λειτουργεί.

Επίσης, επειδή και η κάρτα SD και ο W5100 χρησιμοποιούν τον διάυλο SPI, μόνο ένα από τα δύο μπορεί να λειτουργεί κάθε φορά. Εάν στον κώδικά μας χρησιμοποιούμε και τα δύο αυτά περιφερειακά θα πρέπει να χρησιμοποιήσουμε τις κατάλληλες βιβλιοθήκες. Εάν δεν χρησιμοποιούμε κάποιο από τα δυο αυτά περιφερειακά στο πρόγραμμά μας ωστόσο, πρέπει

ρητά να το απενεργοποιήσουμε. Για να γίνει αυτό, στην περίπτωση του W5100, θέτουμε το pin 10 HIGH ως έξοδο, ενώ στην περίπτωση της SD, θέτουμε το pin 4 ως HIGH έξοδο.

Το Ethernet shield περιέχει επίσης και reset button με το οποίο όταν το πατήσουμε κάνουμε reset και στο W5100 και στο Arduino.

Το shield περιέχει και κάποια LEDs ενδείξεων:

- PWR: Υποδεικνύει ότι η πλακέτα και το shield έχουν τροφοδοσία.
- LINK: Υποδεικνύει παρουσία δικτύου και αναβοσβήνει όταν το shield στέλνει ή λαμβάνει δεδομένα.
- FULLD: Υποδεικνύει ότι η σύνδεση είναι αμφίδρομη στο ίδιο κανάλι (full duplex).
- 100M: Υποδεικνύει ότι έχουμε σύνδεση δικτύου 100Mb/s (σε αντίθεση με 10 Mb / s).
- RX: Αναβοσβήνει όταν το shield λαμβάνει δεδομένα.
- TX: Αναβοσβήνει όταν το shield στέλνει δεδομένα.
- COLL: Αναβοσβήνει όταν ανιχνεύονται συγκρούσεις δικτύου.

### 3.4 Αισθητήρας TCRT5000:



Εικόνα 3.6: Αισθητήρας TCRT500

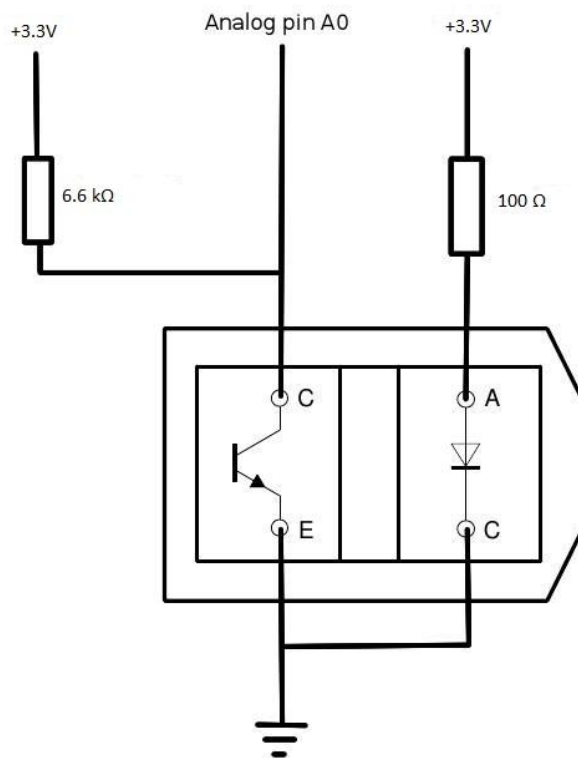
Ο TCRT5000 είναι ένας αισθητήρας ανάκλασης ο οποίος αποτελείται από έναν υπέρυθρο πομπό και ένα φωτοτρανζίστορ (δέκτης) σε κοινό κάλυμμα-συσκευασία έτσι ώστε να μπλοκάρει το ορατό φως (δηλαδή το φως της ημέρας δεν τον επηρεάζει). Είναι μικρός σε

μέγεθος (10.2 mm μήκος, 5.8mm πλάτος και 8mm ύψος ) και το μήκος κύματος του εκπομπού είναι 950nm.

Ο πομπός στέλνει και από την τιμή που λαμβάνει ο δέκτης καταλαβαίνουμε αν βρισκόμαστε στην σημαδεμένη περιοχή του δίσκου του ρολογιού ή όχι. Όσο πιο μεγάλη είναι η τιμή που επιστρέφει τόσο πιο σκούρα είναι η επιφάνεια που “διαβάζει”. Δηλαδή αντιλαμβάνεται αν η επιφάνεια που “διαβάζει” είναι ανοιχτόχρωμη ή σκουρόχρωμη δεν καταλαβαίνει συγκεκριμένα το χρώμα της.

Η σχετική απόσταση ονομαστικής λειτουργίας (“διαβάσματος”) είναι από 0,2 mm ως 15 mm.

Η συνδεσμολογία που υλοποιήθηκε φαίνεται στο παρακάτω σχήμα:



Εικόνα 3.7: Σχέδιο σύνδεσης αισθητήρα TCRT500

Παράδειγμα προγραμματισμού:

```
1  /*
2  AnalogReadSerial
3  Reads an analog input on pin 0, prints the result to the serial monitor.
4  Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and
5
6  This example code is in the public domain.
7  */
8
9  // the setup routine runs once when you press reset:
10 void setup() {
11   // initialize serial communication at 9600 bits per second:
12   Serial.begin(9600);
13 }
14
15 // the loop routine runs over and over again forever:
16 void loop() {
17   // read the input on analog pin 0:
18   int sensorValue = analogRead(A0);
19   // print out the value you read:
20   Serial.println(sensorValue);
21   delay(1);        // delay in between reads for stability
22 }
```

### 3.5 Bizy Type F Plug



Εικόνα 3.8: Bizy Type F Plug

Με την πρίζα της Meazon μπορούμε να ελέγχουμε συσκευές εξ αποστάσεως και να μετράμε την συνολική και την στιγμιαία κατανάλωση της συσκευής που είναι συνδεδεμένη. Οι συσκευές που μπορούμε να συνδέσουμε πρέπει να είναι μέχρι 16Α.

Οι τεχνικές της προδιαγραφής φαίνονται στον παρακάτω πίνακα:

Πίνακας 3.3: Τεχνικά χαρακτηριστικά Bizy Type F Plug

<b>Operating Voltage / Frequency</b>	100 to 240 VAC / 45 to 65 Hz
<b>Power loss response</b>	Automatic resumption of operation after power loss
<b>Internal relay</b>	16 Amp On/Off control
<b>Power consumption</b>	Less than 0.5 Watt
<b>Electric parameters measured</b>	$I_{rms}$ , $V_{rms}$ , frequency, active Power & Energy, reactive Power & Energy
<b>Ranges of measured parameters</b>	Voltage: 100 to 240 volts AC phase-to-neutral, 45 to 65 Hz Current: 0 to 16 Amps 80% to 120% of normal line voltage
<b>Accuracy of measurements*</b>	<ul style="list-style-type: none"><li>• &lt;1% of reading measurement error (metering device)</li><li>• 0.1% of reading measurement error (current sensors)</li></ul>
<b>Data log record</b>	25 days
<b>Coverage</b>	Up to 50m indoor / mesh topology
<b>Dimensions</b>	43 x 88.5 x 86.5 ( WxHxD ) in mm
<b>Operating environment</b>	<ul style="list-style-type: none"><li>• Temperature: -20° C to 50° C</li><li>• Humidity: 10% to 90% relative humidity (RH), non-condensing</li></ul>



### 3.6 Meazon Gateway Advanced



Εικόνα 3.9: Meazon Gateway Advanced

Η συσκευή Meazon Gateway Advanced (ή Bizzy Gateway Advanced) είναι ένα BeagleBone το οποίο είναι βασισμένο στο Linux. Είναι μια μικρή σε μέγεθος συσκευή και χρησιμοποιείται για συνολική αποθήκευση και διαχείριση των δεδομένων από τους μετρητές και τους αισθητήρες της Meazon, πάνω από Ethernet ή GPRS.

Τα χαρακτηριστικά του φαίνονται στον παρακάτω πίνακα:

Πίνακας 3.4: Τεχνικά χαρακτηριστικά Meazon Gateway

	Feature
Processor	1 GHz ARM Cortex-A8
Memory	512MB DDR3
USB 2.0 ports:	USB 2.0 type A host port. Dedicated single mini-USB 2.0 client port (no additional 2-port hub)
Onboard storage	4 GB 8-bit embedded MMC on-board flash version, microSD card 3.3 V Supported
Onboard network	10/100 RJ45
GPRS/GSM (optional)	u-Blox Sara
ZigBee	Texas Instruments CC2531
Real Time Clock	DS1307
Power ratings	210-460 mA @5 V Depending On Activity and Processor Speed
Power source	Mini USB
Size	88 x 72 x 30

Εκτός από το πρότυπο Ethernet για σύνδεση στο Internet, το Gateway μπορεί να συνδεθεί απευθείας στον υπολογιστή αξιοποιώντας το πρωτόκολλο IP over USB.

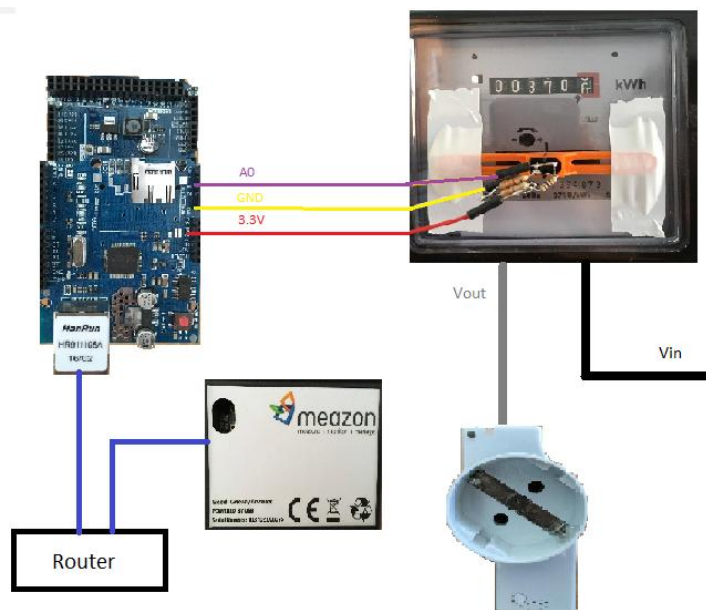
### 3.7 Τοπολογία Διασυνδέσεις

Η συνδεσμολογία που έχουμε υλοποιήσει είναι ως εξής. Ο αναλογικός μετρητής έχει δυο καλώδια, ένα καλώδιο εισόδου (που έρχεται από την Δ.Ε.Η.) και ένα καλώδιο εξόδου ( το

οποίο πάει στην εγκατάσταση που είναι τοποθετημένος ο μετρητής) Έχουμε τοποθετήσει τον αισθητήρα πάνω στο αναλογικό ρολόι της Δ.Ε.Η. σε τέτοια θέση ώστε να μπορεί να “βλέπει” τον δίσκο που γυρνάει. Το Arduino έχει συνδεδεμένο πάνω του το Ethernet Shield και συνδέουμε και τον αισθητήρα. Στο καλώδιο εξόδου του μετρητή έχουμε συνδέσει τον πρίζα της Meazon και πάνω στην πρίζα τοποθετούμε το φορτίο μας.

Το Arduino στέλνει τις μετρήσεις του, μέσω Ethernet σύνδεσης στον server μας και η πρίζα της Meazon στέλνει τις μετρήσεις της στο Gateway ασύρματα μέσω ZigBee. Το Gateway όπως και το Arduino στέλνει τις μετρήσεις που πήρε από την πρίζα μέσω Ethernet σύνδεσης.

Έτσι έχουμε όλες τις μετρήσεις στον server μας.



Εικόνα3.10: Σχέδιο κυκλώματος



Εικόνα 3.11: Υλοποιημένο κύκλωμα σε λειτουργία

## ΚΕΦΑΛΑΙΟ 4: Λογισμικό μέρος του συστήματος

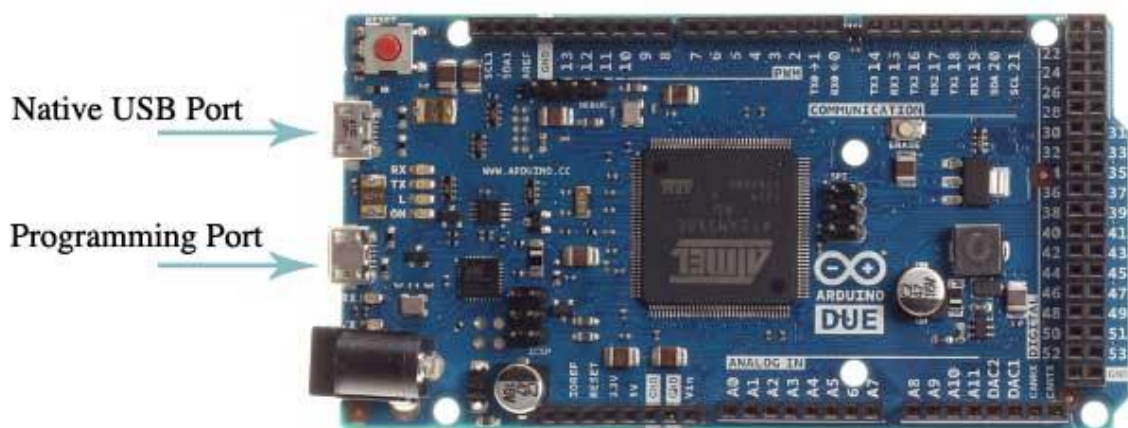
Σε αυτό το κεφάλαιο θα γίνει η επεξήγηση όλων όσων αφορούν το λογισμικό μέρος του συστήματος. Συγκεκριμένα θα αναφερθούμε στον προγραμματισμό που έγινε στις συσκευές που χρησιμοποιήθηκαν καθώς και στους κώδικες για τον ιστοχώρο που δημιουργήθηκε.

### 4.1 Προγραμματισμός Arduino

Στην παρούσα υποενότητα θα περιγράψουμε την διαδικασία για τον προγραμματισμό του Arduino. Το Due μπορεί να προγραμματιστεί με το λογισμικό Arduino (IDE).

Το upload των sketch στο SAM3X είναι διαφορετικό από ό, τι στους μικροελεγκτές AVR που βρίσκονται σε άλλες πλακέτες Arduino, επειδή η μνήμη flash θα πρέπει να διαγραφεί πριν προγραμματιστεί εκ-νέου.

Το upload στο chip διαχειρίζεται από την ROM στο SAM3X, το οποίο λειτουργεί μόνο όταν η μνήμη flash του chip είναι άδεια.



Εικόνα 4.1: Micro USB θύρες Arduino DUE

Μπορούν να χρησιμοποιηθούν και οι δύο θύρες USB για τον προγραμματισμό της πλακέτας, αν και συνιστάται να χρησιμοποιείται η Programming Port εξαιτίας του τρόπου που χειρίζεται το chip

- **Programming Port:** Για να χρησιμοποιήσουμε αυτή τη θύρα, επιλέγουμε ως πλακέτα στο Arduino IDE, "Arduino Due (Programming Port)". Συνδέουμε την Programming Port του Due στον υπολογιστή μας. Η **Programming Port** χρησιμοποιεί το 16U2 ως chip USB-to-serial

και συνδέεται με την πρώτη UART στο SAM3X (RX0 και TX0). Η 16U2 έχει δύο pin που συνδέονται με τα pin Reset και Erase του SAM3X. Άνοιγοντας και κλείνοντας τη Programming Port που συνδέεται στα 1200bps ενεργοποιείται η διαδικασία «hard erase» του chip SAM3X, και ενεργοποιεί τα pin Erase and Reset του SAM3X πριν από την επικοινωνία με την UART. Αυτή είναι η συνιστώμενη θύρα για τον προγραμματισμό του Due, επειδή η “hard erase” διαδικασία είναι πιο αξιόπιστη από την “soft erase” που εμφανίζεται στην Native Port, και θα πρέπει να λειτουργεί ακόμη και αν η κύρια MCU (microcontroller unit) έχει καταρρεύσει.

- **Native Port:** Για να χρησιμοποιήσουμε αυτή τη θύρα, επιλέγουμε ως πλακέτα στο Arduino IDE, “Arduino Due (Native USB Port)”. Η Native Port USB συνδέεται απευθείας με το SAM3X. Συνδέουμε τη Φυσική θύρα USB του Due στον υπολογιστή σας. Το άνοιγμα και το κλείσιμο της Native Port στα 1200bps ενεργοποιεί μια διαδικασία ‘soft erase’: η μνήμη flash διαγράφεται και η πλακέτα κάνει επανεκκίνηση με τον bootloader. Εάν η MCU (microcontroller unit) για κάποιο λόγο καταρρεύσει είναι πιθανό η διαδικασία “soft erase” να μην λειτουργήσει εξ ολοκλήρου στο λογισμικό στο SAM3X. Ανοίγοντας και κλείνοντας τη φυσική θύρα με διαφορετικό baudrate δεν θα κάνει επανεκκίνηση στο SAM3X.

Στην δικιά μας περίπτωση έχουμε χρησιμοποιήσει την Programming Port

### 4.1.1 Εισαγωγή βιβλιοθηκών

Αρχικά το πρόγραμμα μας το ξεκινάμε με την εισαγωγή των βιβλιοθηκών. Οι βιβλιοθήκες στον κώδικα μας αφορούν την σειριακή επικοινωνία μεταξύ περιφερειακών συσκευών και την βιβλιοθήκη για το Ethernet Shield. Για να εισάγουμε βιβλιοθήκες στο Arduino γράφουμε:

```
#include <όνομα_βιβλιοθήκης.h>
```

Συγκεκριμένα σε εμάς:

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

### 4.1.2 Δήλωση - αρχικοποίηση μεταβλητών και ρυθμίσεις Ethernet

Αφού εισάγουμε τις βιβλιοθήκες μετά δηλώνουμε ή αρχικοποιούμε τις μεταβλητές που θα χρησιμοποιήσουμε στο πρόγραμμα μας. Μετά δηλώνουμε την IP διεύθυνση του υπολογιστή στον οποίο είναι εγκατεστημένος ο server μας και στην συνέχεια προσθέτουμε τις κατάλληλες εντολές έτσι ώστε το Arduino – Ethernet Shield να παίρνει δυναμικά την IP διεύθυνση του.

```
unsigned long counter=0; //oi strofes tou diskou twra
unsigned long counterLast=0; //oi prohgomenes strofes
int previous=0;
int white=0;
int red=0;
int printDelay = 0;
int sensorValue = 0;
int delaycnt = 2; // o xronos deigmatolhpsias
float kWh=0;
```

```

float W=0;
unsigned long timePulse=0;
unsigned long timeLast=0;
unsigned long duration=0;
unsigned long dduration=0;
int var=0;

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // MAC of the ethernet
card shield
char ip [] = "192.168.1.101"; // IP Address of the PC or Mac where the
localhost is.

EthernetClient client;

int interval = 5000; // Wait between dumps

```

### 4.1.3 Συνάρτηση void setup()

Αφού ολοκληρώθηκαν οι εισαγωγές, οι δηλώσεις και οι αρχικοποιήσεις τώρα έχει σειρά η συνάρτηση void setup() η οποία όπως έχουμε προαναφέρει είναι μια από τις βασικές συναρτήσεις για το πρόγραμμα μας. Μέσα στην void setup() οι εντολές που έχουμε γράψει είναι για να ξεκινήσει να λειτουργεί η σειριακή θύρα και “προετοιμάζουμε” την βιβλιοθήκη Ethernet και τις ρυθμίσεις δικτύου. Επίσης του λέμε να μας εμφανίσει τις ρυθμίσεις που έκανε σε σχέση με το Ethernet.

```

void setup() {

  Serial.begin(9600);
  Ethernet.begin(mac);
  delay(1000);

  Serial.println("Metrhths Vasili");
  Serial.println("-----\n");
  Serial.print("IP Address      : ");
  Serial.println(Ethernet.localIP());
  Serial.print("Subnet Mask      : ");
  Serial.println(Ethernet.subnetMask());
  Serial.print("Default Gateway IP: ");
  Serial.println(Ethernet.gatewayIP());
  Serial.print("DNS Server IP    : ");
  Serial.println(Ethernet.dnsServerIP());
}

```

### 4.1.4 Συνάρτηση void loop()

Η συνάρτηση void loop() είναι η δεύτερη βασική συνάρτηση του προγράμματος μας και τρέχει συνέχεια μέχρι να βγάλουμε την τροφοδοσία από το Arduino.

Για την καλύτερη κατανόηση του κώδικα που βρίσκεται μέσα στην void loop θα υπενθυμίσουμε αρχικά κάποια πράγματα για την λειτουργία του αισθητήρα TCRT5000. Ο συγκεκριμένος αισθητήρας είναι ένας αισθητήρας ανάκλασης ο οποίος ανάλογα με την τιμή του “διαβάζει” καταλαβαίνει σε ποια περιοχή του δίσκου βρισκόμαστε. Όσο

είμαστε στην λευκή περιοχή η τιμή που θα διαβάζει θα είναι μικρότερη από 410 όταν όμως έρθει η κόκκινη περιοχή του δίσκου τότε η τιμή που θα "διαβάσει" θα είναι μεγαλύτερη από 410, έτσι καταλαβαίνει τις στροφές που κάνει ο δίσκος και μπορούμε να υπολογίσουμε την συνολική κατανάλωση του φορτίου που είναι συνδεδεμένο στον μετρητή. Επειδή όμως μας ενδιαφέρει και η στιγμιαία κατανάλωση του φορτίου και όσο πιο μεγάλη κατανάλωση έχουμε τόσο πιο γρήγορα γυρνάει ο δίσκος, υπολογίζουμε και την διάρκεια της κάθε περιστροφής.

Λόγο του γεγονότος ότι ο δίσκος του μετρητή δεν είναι ομοιόμορφος αλλά έχει ατέλειες κάποιες φορές μας αύξανε τον μετρητή εσφαλμένα.

Για να ξεπεραστεί αυτό το πρόβλημα προστέθηκαν στον κώδικα μας δυο μεταβλητές: η white (για την ασημένια περιοχή του δίσκου) και η red(για την κόκκινη περιοχή). Και τώρα για να αυξήσει τον μετρητή (counter) πρέπει να έχει αυξήσει την red τουλάχιστον 12 φορές και την white τουλάχιστον 10 φορές. Δηλαδή αυξάνει τον μετρητή όταν έχει περάσει η κόκκινη περιοχή ολόκληρη από μπροστά του και έχει μεταβεί στην ασημένια περιοχή. Στην ουσία με την μεταβλητή counter μετράμε τις στροφές του δίσκου. Οι τιμές 12 και 10 επιλέχθηκαν πειραματικά αλλά μπορούν να αποδειχθούν με βάση την παρακάτω θεωρία:

Ο αναλογικός μας δίσκος έχει ακτίνα περίπου ίση με 3,5cm, οπότε η περίμετρος του είναι  $2\pi \cdot 3,5\text{cm} = 21,98\text{cm}$ .

Η κόκκινη περιοχή του είναι περίπου ίση με 1,2cm.

Αν υποθέσουμε ότι ο μετρητής της Δ.Ε.Η. έχει τάση 230V και ρεύμα 60A σε μια ώρα θα έχει μετρήσει  $230 \cdot 60 = 13,8\text{kWh}$ .

Επειδή είναι κατασκευασμένος να μετρά 1 kWh κάθε 375 στροφές:  $375 \cdot 13,8 = 5175$  στρ/ώρα ή 1,4375 στρ/sec άρα για μια στροφή χρειάζεται  $1/1,4375 = 0,696$  sec.

Οπότε ο χρόνος που θα είναι η κόκκινη περιοχή του δίσκου μπροστά στον αισθητήρα μας θα είναι για  $(1,2\text{cm} \cdot 0,696\text{sec}) / 21,98\text{cm} = 0,04\text{sec}$  και αφού η δειγματοληψία μας είναι κάθε 2ms ο αισθητήρας μας θα μετρήσει στο κόκκινο  $0,04\text{sec} / 2\text{ms} = 18,99 = 18$  σημεία υπό ιδανικές συνθήκες οπότε εμείς βάλαμε έτσι την τιμή 12 για το κόκκινο.

Με τον ίδιο τρόπο βρίσκουμε τα σημεία την λευκής περιοχής τα οποία βγαίνουν  $347,82 = 347$  σημεία, αλλά η λευκή περιοχή δεν είναι τόσο σημαντική όσο η κόκκινη και απλά θέσαμε στην μεταβλητή white ενδεικτικά την τιμή 10 έτσι ώστε να έχει περάσει ολόκληρη η κόκκινη περιοχή και δεν έχει σταματήσει ο δίσκος στο μεταίχμιο μεταξύ κόκκινης και ασημένιας (λευκής) περιοχής.

Αφού είδαμε ότι ο αισθητήρας μας μπορεί να μετρήσει σωστά κάναμε κάποιες τεχνητές παρεμβολές για να δούμε την συμπεριφορά του. Η πρώτη παρεμβολή που κάναμε ήταν με ένα φακό για να δούμε αν όντως δεν επηρεάζεται από το ορατό φως, και πράγματι δεν επηρεαζόταν, για την ακρίβεια δεν τον αντιλαμβανόταν καθόλου.



Εικόνα 4.2 Χώρος δοκιμών του συστήματος

Η δεύτερη παρεμβολή που κάναμε ήταν να καλύψουμε ολόκληρο το τζάμι του μετρητή για να μην μπαίνει από πουθενά φως αλλά ο αισθητήρας μας συνέχιζε να μετράει σωστά. Επίσης δοκιμάσαμε τον αισθητήρα και με ανοιχτές τις περσίδες του δωματίου αλλά και κλειστές αλλά και πάλι αισθητήρας μετρούσε σωστά.

Η τρίτη παρεμβολή που κάναμε ήταν με ένα τηλεκοντρόλ επειδή ο αισθητήρας μας είναι υπέρυθρος. Σε αυτή την περίπτωση ο αισθητήρας μας αντιλαμβανόταν την παρεμβολή αλλά συνέχιζε να μετράει σωστά. Ο κώδικας που υλοποιεί τα παραπάνω φαίνεται παρακάτω:

```
void loop() {  
  
    timePulse = millis();  
    printDelay++;  
    // read the input on analog pin 0:  
    sensorValue = analogRead(A0);  
  
    if (sensorValue<=410){
```





```

        client.println();
        client.stop();
        printDelay=0;
    }
else {
    // you didn't get a connection to the server:
    Serial.println("--> connection failed/n");
}

}

```

Όλη η παραπάνω διαδικασία επαναλαμβάνονταν κάθε 2 ms, οπότε προσθέτουμε και αυτή την εντολή και κλείνουμε το bracket της void loop():

```

delay(delaycnt);
}

```

## 4.2 Προγραμματισμός BeagleBone Meazon

Στην υποενότητα αυτή θα περιγράψουμε την διαδικασία για τον προγραμματισμό του BeagleBone της εταιρίας meazon ή αλλιώς Meazon Gateway Advanced.

Πριν από τον προγραμματισμό του ας δούμε κάποια χαρακτηριστικά του.

Το λειτουργικό σύστημα που έχει είναι: Ubuntu 14.04 (trusty).

Διαθέτει έναν Mosquitto στην διεύθυνση 192.168.7.2:1883. Ο Mosquitto είναι ένας ανοιχτού κώδικα (EPL / EDL licensed) message broker που υλοποιεί τις εκδόσεις πρωτοκόλλου MQTT 3.1 και 3.1.1.

Το MQTT παρέχει μια αποτελεσματική μέθοδο διεξαγωγής μηνυμάτων, χρησιμοποιώντας ένα publish / subscribe μοντέλο, βελτιστοποιημένη για το "Internet of Things"(IoT), ως προς την ταχύτητα και την κατανάλωση πόρων (υπολογιστικής ισχύος, ποσότητα δεδομένων), όπως για παράδειγμα σε αισθητήρες χαμηλής ισχύος ή κινητές συσκευές, όπως κινητά τηλέφωνα, τα ενσωματωμένα σε υπολογιστές ή μικροελεγκτές, όπως το Arduino.

Επίσης διαθέτει και έναν ssh (Secure Shell) server στην port 22. Το Secure Shell ή αλλιώς SSH είναι ένα δικτυακό πρωτόκολλο που ορίζει έναν τρόπο για ασφαλείς, κρυπτογραφημένες συνδέσεις μεταξύ δύο μηχανημάτων. Οι κρυπτογραφημένες αυτές συνδέσεις είναι δυνατόν να επιτυγχάνονται και μέσω ενός επισφαλούς, εν γένει, δικτύου, όπως είναι το Internet. Το SSH συνήθως χρησιμοποιείται για την απομακρυσμένη, πλήρη πρόσβαση στη γραμμή εντολών διαφόρων Unix boxes — αλλά όχι μόνο. Συχνά, π.χ., επιστρατεύεται για ασφαλείς μεταφορές αρχείων, περιήγηση στο web μέσω των λεγόμενων SSH tunnels κ.ο.κ

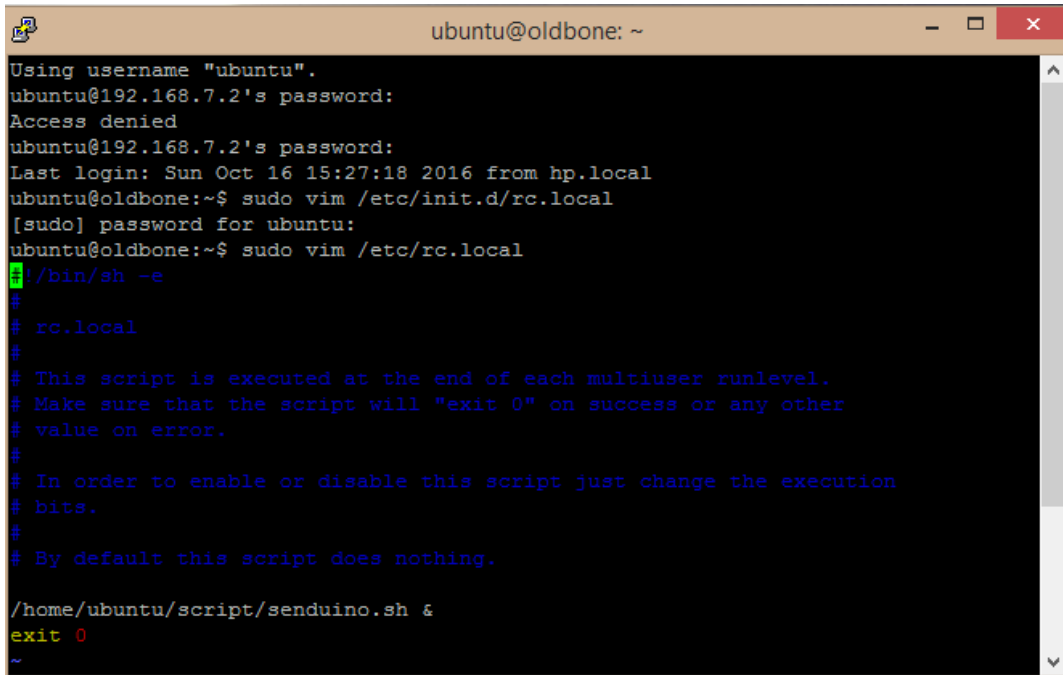
Οπότε η πρίζα Meazon παίρνει τις μετρήσεις, τις στέλνει στο Gateway μέσω ZigBee πρωτοκόλλου και το Gateway στέλνει τις μετρήσεις στον apache server.

Η αρχική ρύθμιση που έγινε στο Gateway ήταν να ξεκινάει και το senduino.sh κατά την εκκίνηση του. Για να πραγματοποιηθεί αυτό μέσω του putty γράφουμε την εντολή:

```
sudo vim /etc/rc.local
```

και κάνοντας edit το αρχείο του προσθέσαμε στο τέλος την παρακάτω γραμμή:

```
/home/ubuntu/script/senduino.sh
```



```
ubuntu@oldbone: ~
Using username "ubuntu".
ubuntu@192.168.7.2's password:
Access denied
ubuntu@192.168.7.2's password:
Last login: Sun Oct 16 15:27:18 2016 from hp.local
ubuntu@oldbone:~$ sudo vim /etc/init.d/rc.local
[sudo] password for ubuntu:
ubuntu@oldbone:~$ sudo vim /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

/home/ubuntu/script/senduino.sh &
exit 0
~
```

Εικόνα 4.3: Προγραμματισμός Gateway μέσω PuTTY

Το script senduino.sh για να στέλνονται οι μετρήσεις στον xampp server μας είναι το παρακάτω:

```
#!/bin/bash
```

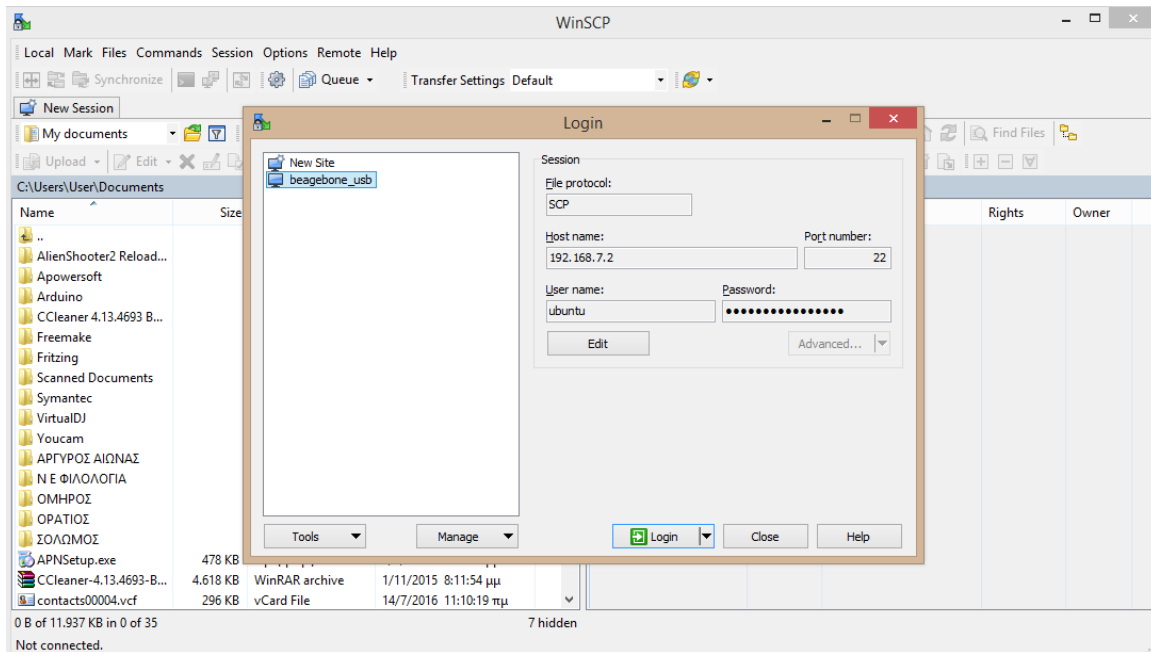
```
COORD=$(/opt/zigpy/tools/select devices 'nwk=0' | tail -n1 | cut -d'|' -f1)
DEV1="124B0002CC8F3E"
DEST="10.0.1.84:80"
```

```
echo "COORD is $COORD"
```

```
mosquitto_sub -v -t "meas/$COORD/$DEV1/cnrgA/+" -t
"meas/$COORD/$DEV1/pwrA/+" |
sed -s -u -e 's/^.*cnrgA.*\ /kWh_B=/' -e 's/^.*pwrA.*\ /W_B=/' |
xargs -I {} curl -G "http://$DEST/arduino/add_data.php" -d "{}"
```

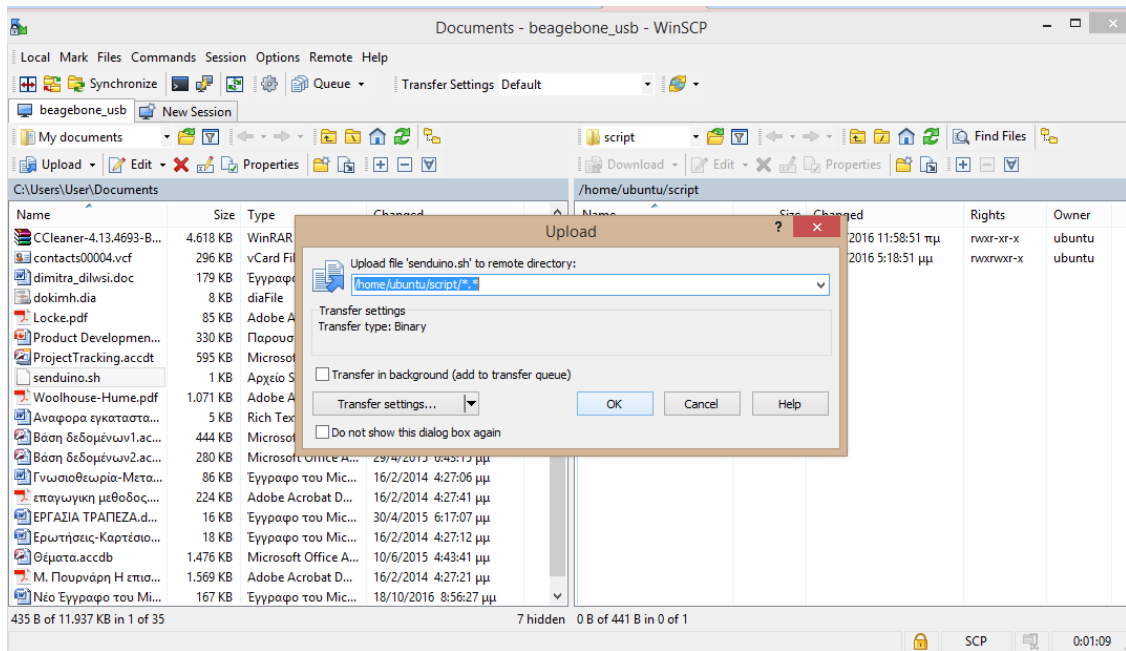
Για να περάσουμε το αρχείο senduino.sh μέσα στο Gateway πρέπει να εκτελέσουμε τα παρακάτω βήματα:

1. Αφού έχουμε αποσυνδέσει τον υπολογιστή μας από το Διαδίκτυο, συνδέουμε το Gateway μέσω USB καλωδίου στον υπολογιστή μας.
2. Μέσω της γραμμής εντολών (cmd) στον υπολογιστή μας μπορούμε να δούμε την IP διεύθυνση του υπολογιστή μας. Η IP που μας εμφανίζει είναι η 192.168.7.1 οπότε του Gateway η IP είναι η 192.168.7.2
3. Ανοίγουμε το WinSCP, επιλέγουμε SCP βάζουμε την IP 192.168.7.2 και Port Number: 22, δίνουμε username και password και πατάμε Login.



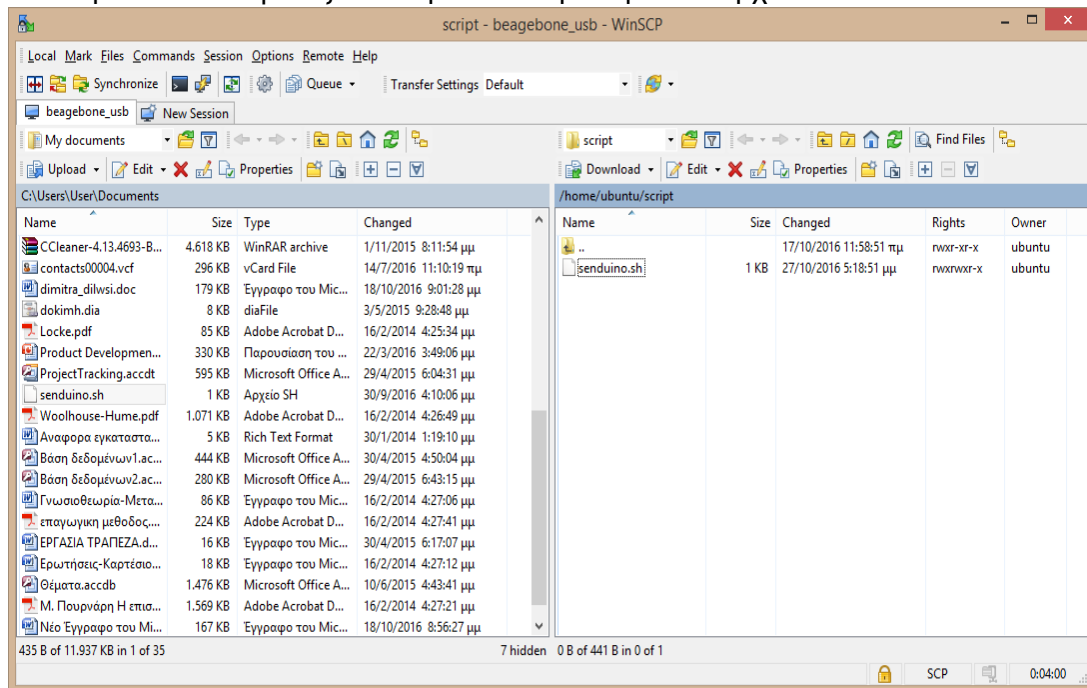
Εικόνα 4.4: Ρυθμίσεις WinSCP

4. Πατάμε Upload και βρίσκουμε το αρχείο senduino.sh εκεί που το έχουμε αποθηκεύσει και μετά πατάμε OK.



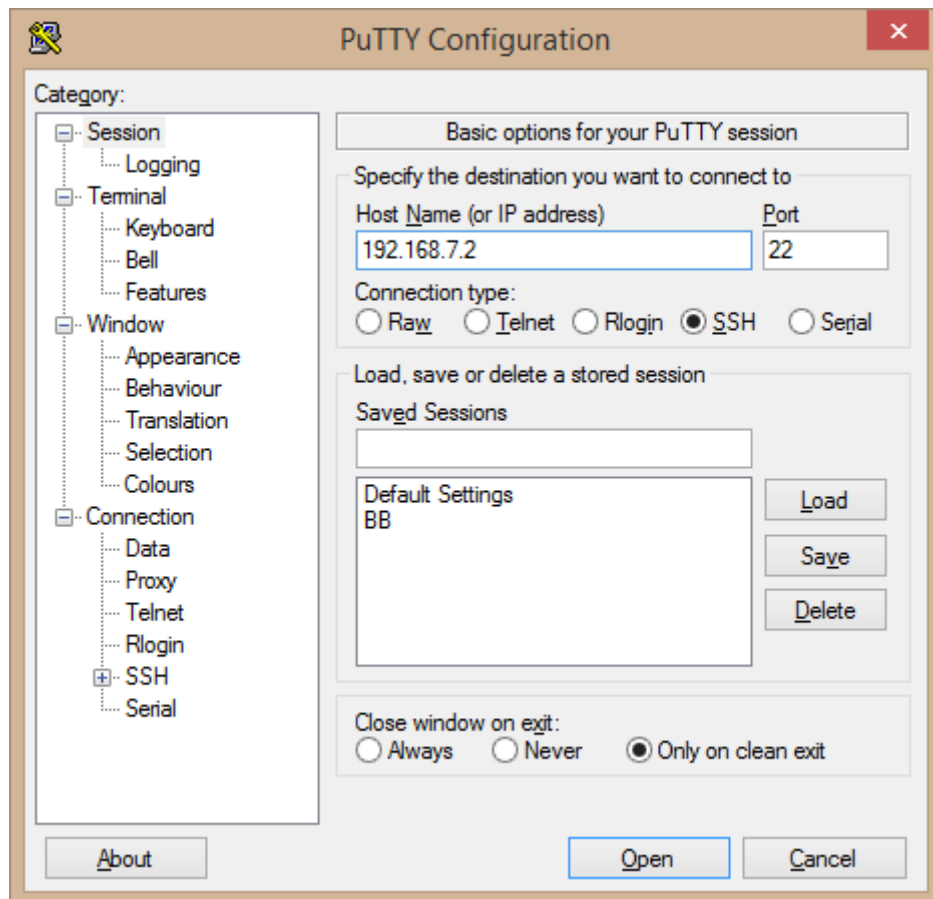
Εικόνα 4.5 Upload senduino.sh

## 5. Τώρα πλέον στην δεξιά πλευρά του παραθύρου υπάρχει το senduino.sh



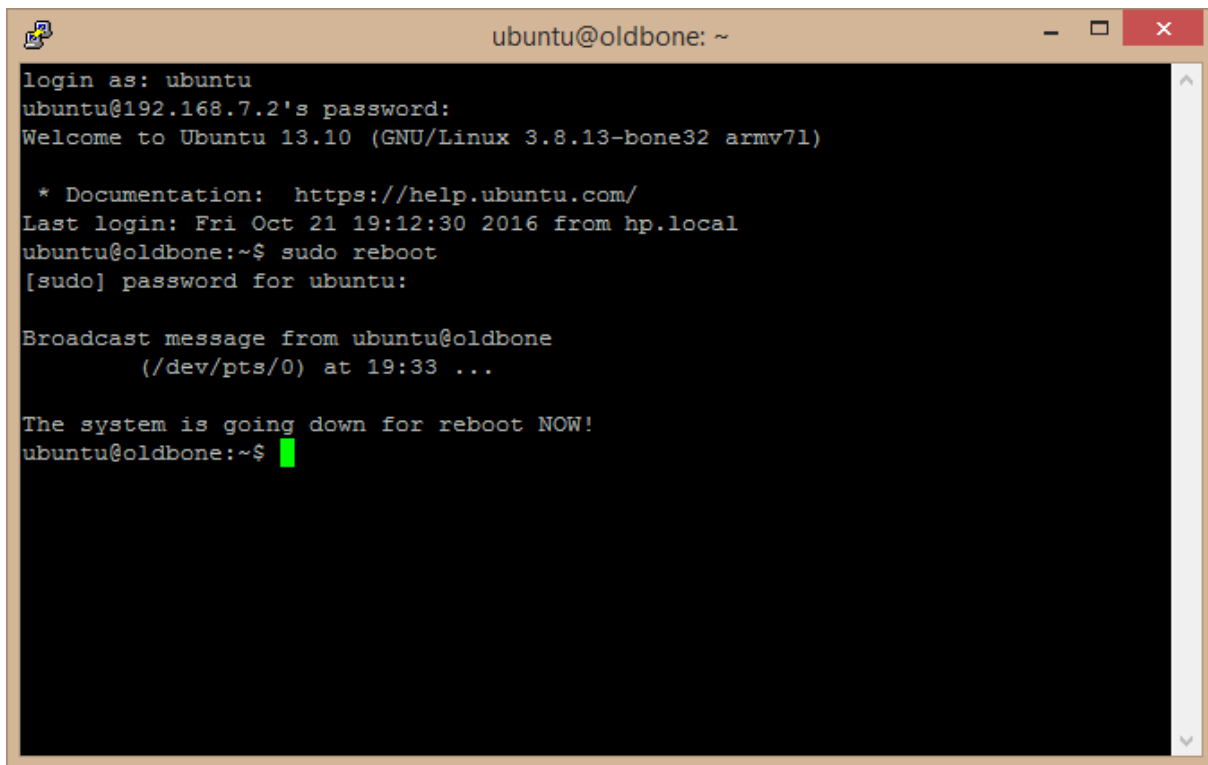
Εικόνα 4.6: Αρχεία που έχουν γίνει upload στο gateway

## 6. Τώρα πάμε και ανοίγουμε το PuTTY. Πληκτρολογούμε την IP 192.168.7.2 και Port: 22 και πατάμε Open.



Εικόνα 4.7: Ρυθμίσεις για εκκίνηση PuTTY

7. Στο καινούριο παράθυρο που μας ανοίγει πληκτρολογούμε username και password και γράφουμε την εντολή `sudo reboot`. Πληκτρολογούμε ξανά το password μας και το σύστημα μας θα κάνει επανεκκίνηση και μετά είναι έτοιμο προς χρήση.

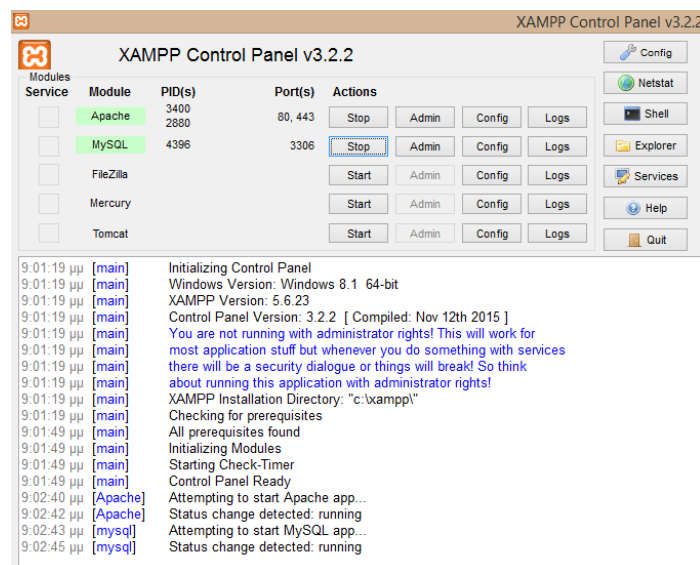


Εικόνα 4.8: Reboot Gateway μέσω PuTTY

## 4.3 Βάση Δεδομένων

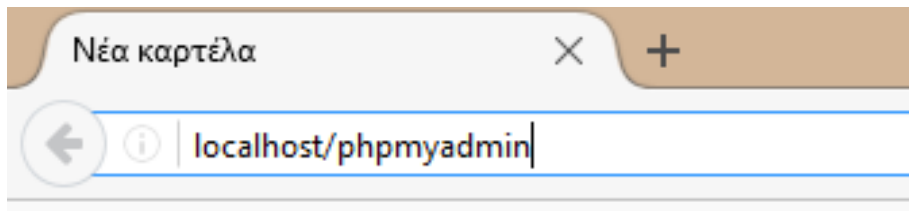
### 4.3.1 Δημιουργία Βάσης Δεδομένων

Για να δημιουργήσουμε την βάση δεδομένων στην οποία θα αποθηκεύονται όλες οι μετρήσεις μας, αφού εγκαταστήσαμε και τρέξαμε τοxampp:



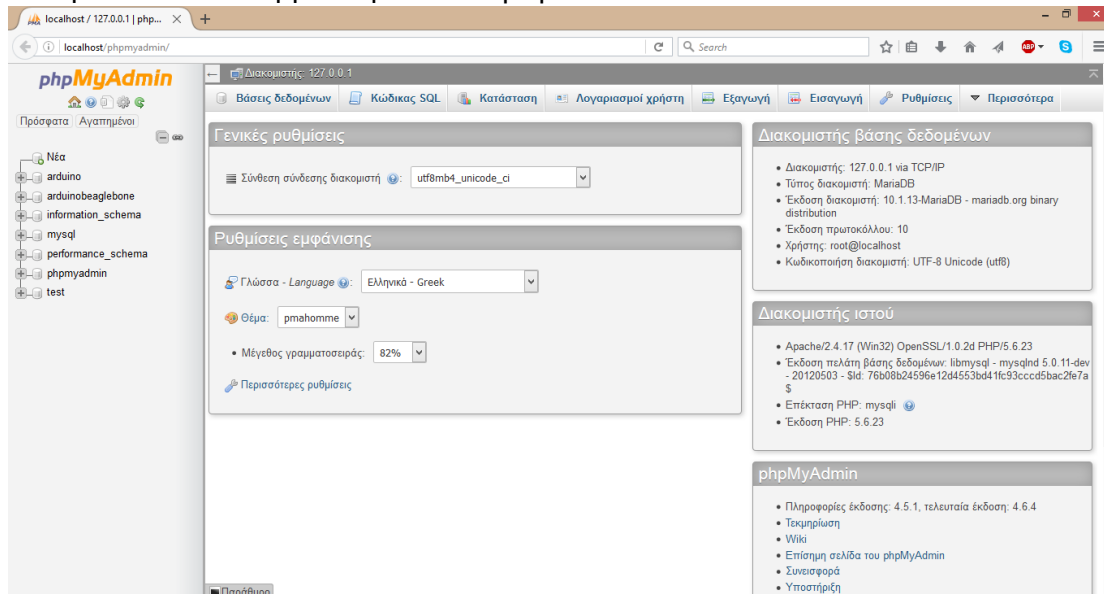
Εικόνα 4.9: Περιβάλλον XAMPP

Ανοίγουμε τον φυλλομετρητή(browser) που χρησιμοποιούμε και γράφουμε:  
localhost/phpmyadmin



Εικόνα 4.10: Μετάβαση στο phpmyadmin

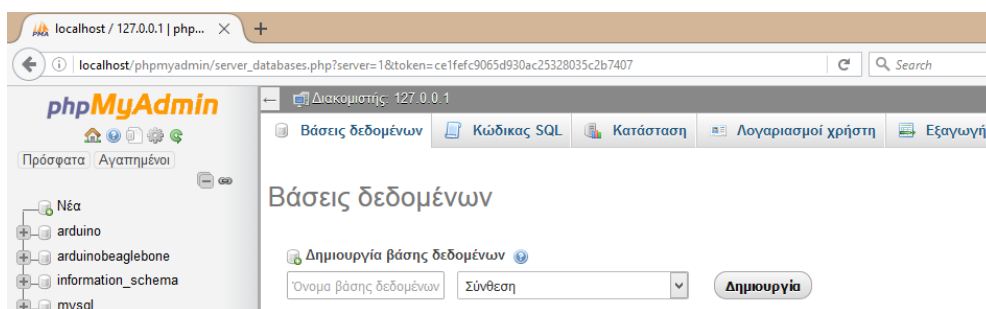
Πατάμε Enter και θα βρεθούμε σε αυτή την σελίδα:



Εικόνα 4.11: Περιβάλλον phpmyadmin

Τώρα στο αριστερό μέρος της σελίδας πάμε κάνουμε κλικ εκεί που λέει “ Νέα ” .

Θα μας πάει στην παρακάτω εικόνα και θα γράψουμε το όνομα που επιθυμούμε για την βάση μας και μετά πατάμε δημιουργία.



Εικόνα 4.12: Δημιουργία Βάσης Δεδομένων

Στην δικιά μας βάση δώσαμε το όνομα “arduinobeaglebone” και τώρα πάμε να δημιουργήσουμε έναν πίνακα τον οποίο θα τον ονομάσουμε “metrhseis” και θα έχει τις εξής

στήλες: id, event, counter, kWh, W, kWh\_B και W\_B. Για να τον δημιουργήσουμε θα γράψουμε τις παρακάτω εντολές:

```

1 CREATE TABLE `metrhseis` (
2
3 `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY COMMENT 'unique ID',
4 `event` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT 'Event Date and Time',
5 `counter` VARCHAR( 10 ) NOT NULL COMMENT 'kykloi diskoy',
6 `kWh` VARCHAR( 10 ) NOT NULL COMMENT 'synolikh katanalwsh',
7 `W` VARCHAR( 10 ) NOT NULL COMMENT 'stigmiaia katanalwsh',
8 `kWh_B` VARCHAR( 10 ) NOT NULL COMMENT 'synolikh katanalwsh BeaglBone',
9 `W_B` VARCHAR ( 10 ) NOT NULL COMMENT 'stigmiaia katanalwsh BeagleBone'
10
11 ) ENGINE = InnoDB;

```

Εικόνα 4.13: Δημιουργία πίνακα και στηλών πίνακα στην βάση δεδομένων

Η δομή των στηλών του πίνακα φαίνεται στον παρακάτω πίνακα:

Πίνακας 4.1 Δομή στηλών πίνακα

#	Όνομα	Τύπος	Κενό	Προεπιλογή	Πρόσθετα
1	id	int(11)	Όχι	Καμία	AUTO_INCREMENT
2	event	timestamp	Όχι	CURRENT_TIMESTAMP	
3	counter	varchar(10)	Όχι	Καμία	
4	kWh	float	Ναι	0	
5	W	float	Ναι	0	
6	kWh_B	float	Ναι	0	
7	W_B	float	Ναι	0	

- id:** Είναι το αναγνωριστικό κάθε εγγραφής. Σε κάθε καινούργια εγγραφή αυξάνεται κατά ένα
- event:** Καταγραφή ημερομηνίας και ώρας για κάθε γραμμή του πίνακα (δηλαδή για κάθε μέτρηση)
- counter:** Αποθηκεύει τις στροφές του δίσκου. Οπότε είναι συνεχώς αυξανόμενο
- kWh:** Αποθηκεύει τις kWh που έχει υπολογίσει το Arduino
- W:** Αποθηκεύει τα Watt που έχει υπολογίσει το Arduino
- kWh\_B:** Αποθηκεύει τις kWh που έχει υπολογίσει το BeagleBone
- W:** Αποθηκεύει τα Watt που έχει υπολογίσει το BeagleBone



### 4.3.2 Σύνδεση βάσης δεδομένων με Arduino και BeagleBone

Για να μπορεί η βάση δεδομένων που δημιουργήσαμε να επικοινωνεί με το Arduino και το BeagleBone δημιουργήσαμε κάποια αρχεία php.

#### Αρχείο dbconnect.php

Το αρχείο αυτό χρησιμοποιείται για να μπορέσουμε να συνδεθούμε στην βάση δεδομένων μας.

```
<?php
$MyUsername = "root"; // enter your username for mysql
$MyPassword = ""; // enter your password for mysql
$MyHostname = "localhost"; // this is usually "localhost" unless your database resides on
a different server

$dbh = mysql_pconnect($MyHostname , $MyUsername, $MyPassword);
$selectd = mysql_select_db("arduinobeaglebone",$dbh); //syndesh me thn vash dedomenwn
arduinobeaglebone
?>
```

#### Αρχείο add\_data.php

Με το αρχείο αυτό αποθηκεύουμε τις μετρήσεις μας από το Arduino και από το BeagleBone στην βάση δεδομένων. Για να συνδεθούμε στην βάση δεδομένων χρησιμοποιούμε το αρχείο dbconnect.php που δημιουργήσαμε ακριβώς από πάνω.

```
<?php
// Connect to MySQL
include("dbconnect.php");
$this_minute = date("Y-m-d H:i:00");
$keys = ["kWh", "W", "kWh_B", "W_B", "counter"];

$keystr = "event";
$valuestr = ''.$this_minute.''';
$key_values = 'event="" . $this_minute . ''';

$SQL_QUERY = "SELECT 1 FROM arduinobeaglebone.metrhseis WHERE event="" . $this minute .
''";
$result = mysql_result(mysql_query($SQL_QUERY), 0);

foreach ($keys as $key) {
    if(isset($_GET[$key])) {
        $keystr .= ',' . $key;
        $valuestr .= ',' . $_GET[$key] . ''';
        $key_values .= ',' . $key . '=' . $_GET[$key] . ''';
    }
}

$SQL_INSERT = 'INSERT INTO arduinobeaglebone.metrhseis (' . $keystr . ') VALUES (' .
$valuestr . ')';
$SQL_UPDATE = 'UPDATE arduinobeaglebone.metrhseis SET ' . $key_values . ' WHERE event="" .
$this_minute . ''';

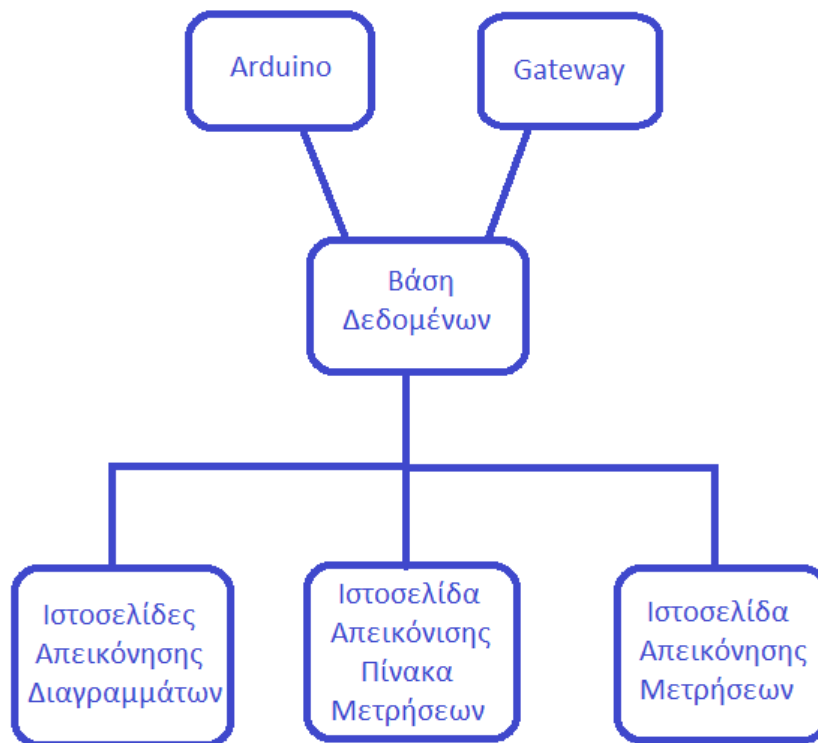
echo $result . '\n';
if($result) {
    echo $SQL_UPDATE;
    mysql_query($SQL_UPDATE);
} else {
```

```
    echo $SQL_INSERT;
    mysql_query($SQL_INSERT);
}

// Go to the review_data.php (optional)
header("Location: review_data.php");
?>
```

## 4.4 Ιστοσελίδα που υλοποιήθηκε και οι λειτουργίες της

Για να την καλύτερη κατανόηση της ιστοσελίδας που δημιουργήσαμε, σχεδιάσαμε το παρακάτω σχήμα το οποίο αναπαριστά τον τρόπο λειτουργίας της ιστοσελίδας.



Εικόνα 4.14: Δομή λογισμικού μέρους του συστήματος

Με λίγα λόγια, πηγαίνουν οι μετρήσεις μας στην βάση δεδομένων με τον τρόπο τον οποίο είπαμε στο προηγούμενο κεφάλαιο και από την βάση δεδομένων παίρνουμε τις τιμές που χρειαζόμαστε και δημιουργούμε την ιστοσελίδα απεικόνισης, τον πίνακα μετρήσεων και της ιστοσελίδες που περιέχουν τα διαγράμματα.

### 4.4.1 Αρχική σελίδα

Για πιο εύκολη περιήγηση στον ιστοχώρο, δημιουργήσαμε μια αρχική σελίδα η οποία έχει συγκεντρωμένες όλες του τις δυνατότητες-επιλογές.

Παρακάτω φαίνεται ο κώδικας της αρχικής σελίδας ο οποίος είναι αποθηκευμένος στο φάκελο htdocs του τοπικού server με όνομα home.html .

```
<!DOCTYPE html>
<html>
  <style>
    h1 {
      width:1400;
      margin-top:30px;
      margin-right:85px;
      margin-bottom:40px;
      margin-left:85px;
      font-weight: bold;
      font-style:normal;
      text-decoration: none;
      background-color:#0080ff;
      color:white;
      text-shadow: 2px 2px 4px #000000;
      //border-radius: 0px;
      border-top-left-radius: 35px;
      border-top-right-radius: 35px;
      padding-top:20px ;
      padding-bottom:20px ;
      font-family: Arial;
      text-align:center;
    }
    body {
      //background-color:black;
      background-color:#b7c1b0;
      text-align: center;
    }
  </style>
  <head>
    <title>Energy Meter</title>
  </head>
  <body>
    <h1>Energy Meter</h1>
    <div> 
    </div>
    <div> </div>
    <div> </div>
    <h1>Menu</h1>
    <div style="margin:50px 0px 0px 0px;">
      <a href="charts/navigation1.html" style="text-decoration:
none;">
        
        </a>
        <a href="charts/navigation2.html" style="text-decoration:
none;">
          
          </a>
          <a href="charts/navigation3.html" style="text-decoration:
none;">
```

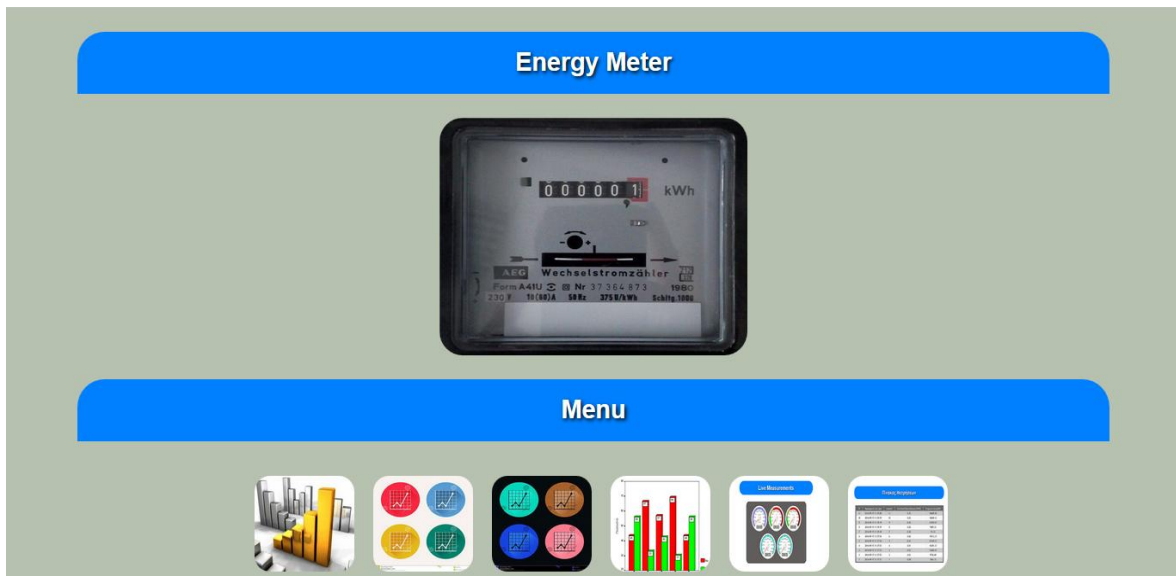
```

        
    </a>
    <a href="charts/navigation4.html" style="text-decoration:
none;">
        
    </a>
    <a href="konter.php" style="text-decoration: none;">
        
    </a>
    <a href="review_data.php" style="text-decoration: none;">
        
    </a>

</div>
</body>
</html>

```

Στην αρχική σελίδα υπάρχουν 6 κουμπιά τα οποία με το πάτημα τους σε μεταφέρουν στην αντίστοιχη σελίδα.



Εικόνα 4.15: Αρχική σελίδα ιστοχώρου

Ο χρήστης μπορεί να δει που αντιστοιχεί το κάθε κουμπί απλά πηγαίνοντας τον κέρσορα του ποντικιού πάνω στο κουμπί που θέλει.



Εικόνα 4.16: Περιγραφή κουμπιού

#### 4.4.2 Σελίδα απεικόνισης των μετρήσεων

Σε αυτή την σελίδα μπορούμε να βλέπουμε τις μετρήσεις μας από το Arduino και από το BeagleBone. Περιλαμβάνει πέντε μετρητές οι οποίοι απεικονίζουν με αναλογικό αλλά και ψηφιακό τρόπο τις μετρήσεις. Επειδή όταν έχουμε πολύ μεγάλες κλίμακες οι μικρές μετρήσεις θα ήταν δύσκολο να αναγνωστούν για αυτό τον λόγο απεικονίζουν τις μετρήσεις και με ψηφιακό τρόπο.

Έχουμε την δυνατότητα να δώσουμε εμείς τα χρώματα της κλίμακας, την μέγιστη/ελάχιστη τιμή της καθώς και τα βήματα της. Στους μετρητές counter, kWh και kWh\_B που απεικονίζουν τις περιστροφές του δίσκου, την συνολική κατανάλωση από το Arduino και την συνολική κατανάλωση από το BeagleBone αντίστοιχα, η κλίμακα είναι ένα χρώμα επειδή οι συγκεκριμένοι μετρητές συνεχώς αυξάνονται. Ενώ στους μετρητές W και W\_B που απεικονίζουν την στιγμιαία κατανάλωση από το Arduino και την στιγμιαία κατανάλωση από το BeagleBone αντίστοιχα αλλάζει χρώμα η κλίμακα τους επειδή αυξομειώνονται ανάλογα με το φορτίο που είναι συνδεδεμένο. Και επειδή οι δύο αυτοί μετρητές μετρούν το ίδιο μέγεθος έχουν ίδια κλίμακα και ίδιο χρώμα στην κλίμακα.

0-2000 W πράσινο χρώμα: ελάχιστη κατανάλωση  
2000-4000 W κίτρινο χρώμα: κανονική κατανάλωση  
4000-6500 W κόκκινο χρώμα: αυξημένη κατανάλωση  
6500-10000 W μπορντό χρώμα: μέγιστη κατανάλωση

Επίσης υπάρχει και ένα κουμπί για να μεταφερόμαστε στην αρχική σελίδα. Ο χρήστης μπορεί να μεταβεί σε αυτή την σελίδα μέσω της αρχικής σελίδας πατώντας το κατάλληλο κουμπί.



Εικόνα 4.18: Κουμπί μετάβασης στην ιστοσελίδα απεικόνισης μετρήσεων

Το αρχείο που περιέχει τον κώδικα της σελίδας απεικόνισης ονομάζεται `konter.php` και είναι αποθηκευμένο στο φάκελο `htdocs` του τοπικού `server`. Ο κώδικας της σελίδας φαίνεται παρακάτω:

<?php

```

// Έναρξη σύνδεσης MySQL χρησιμοποιώντας το αρχείο dbconnect.php
include('dbconnect.php');
$page = $ SERVER['PHP SELF'];
$sec = "60"; //ο xronos pou kanei refresh thn selida apeikonishs
?>
<html>
<head>
<meta http-equiv="refresh" content="<?php echo $sec?>;URL='<?php echo $page?>'>
<title>Live Measurments</title>
<script>

var counter = 0;
var kWh =0;
var W = 0;
var kWh B = 0;
var W_B = 0;

eval(function(p,a,c,k,e,r){e=function(c){return(c<a?'':e(parseInt(c/a)))+(c=c%a)>35?String.fromCharCode(c+29):c.toString(36)};if(!''.replace(/^/,String)){while(c-->){r[e(c)]=k[c]||e(c);k=function(e){return r[e]};e=function(){return'\w+';c=1};while(c-->){if(k[c])p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c]);return p}('t
W=v(f){W.2t.3T(A);A.B={Z:2u,19:1q,1h:1q,1J:U,1b:D,K:0,V:['\0','\20','\40','\2A','\2B','\D
\'],2r:10,2C:M,1E:U,2q:{2D:3,2E:2},2H:M,1c:{2j:10,2m:3y,2O:'3v'},J:{2P:'\4h',V:'\31',2r:
'\#3k',1J:'\#37',1E:'\#37',3e:'\#31',1t:{2s:'1e(3Y, 3d, 3d, 1)'\},3c:'1e(1Y, 5I, 5E,
.9)'\}},1o:[{1n:20,1D:2A,1F:'\#3j'},{1n:2A,1D:2B,1F:'\#36'},{1n:2B,1D:D,1F:'\#5n'}]};t
g=0,1p=A,N=0,1S=0,1G=U;A.5d=v(a){N=f.lc?g:a;t b=(f.lb-f.K)/D;1S=a>f.lb?1S=f.lb+b:a<f.K?f.K-
b:a;g=a;f.lc?31():A.lg();C A};A.3m=v(a){N=g+a;A.lg();C A};A.4T=v(){g=N=1S=A.B.K;A.lg();C
A};A.4R=v(){C g};A.13=v(){v 2k(a,b){Q(t i 4P b){z(1H b[i]="'1W"&&! (4O.4y.2V.4p(b[i])==='1W
4n]'\)&&i!='\Z')}{z(1H
a[i]!'="1W")}{a[i]=}}2k(a[i],b[i])O{a[i]=b[i]}}};2k(A.B,f);A.B.K=1R(A.B.K);A.B.lb=1R(A.B.lb);f
=A.B;N=g=f.K;z(!f.Z){4m 4j("4g 4d 4b 46 44 41 3Z 3W W 1W!");t
j=f.Z.5K?f.Z.2R.5v(f.Z),q=j.3u('\2d'),1i,1y,1A,14,17,u,1d;v
2M(){j.19=f.19;j.1h=f.1h;1i=j.4s(M);1d=1i.3u('\2d');1y=j.19;1A=j.1h;14=1y/2;17=1A/2;u=14<17?1
4:17;1i.2J=U;1d.3P(14,17);1d.G();q.3P(14,17);q.G();2M();A.4Z=v(a){2k(A.B,a);2M();A.lg();C
A};t k={4q:v(p){C p},4e:v(p){C E.1L(p,2)},4c:v(p){C E.1L(p,5)},3v:v(p){C 1-
E.1O(E.5C(p))},5k:v(p){C 1-(v(p){Q(t a=0,b=1;1;a=b,b/=2){z(p>=(7-4*a)/11){C-E.1L((11-6*a-
11*p)/4,2)+E.1L(b,2)}(1-p)},4S:v(p){C 1-(v(p){t x=1.5;C E.1L(2,10*(p-
1))*E.1T(20*E.1a*x/3*p)}(1-p)});t l=2u;v 3S(d){t e=2v 3R;l=2x(v){t a=2v 3R-
e,1M=a/d.2m;z(1M>1){1M=1}t b=1H d.2g="v"?d.2g:k[d.2g];t
c=b(1M);d.3Q(c);z(1M==1){2b(1)};d.2j||10};v 3l(){1&&2b(1);t b=(1S-
N),1n=N,29=f.lc;3S({2j:29.2j,2m:29.2m,2g:29.20,3Q:v(a){N=1R(1n)+b*a;1p.lg()}});q.5l="30";A.lg
=v(){z(1i.2J){1d.3M(-14,-17,1y,1A);1d.G();t
a=q;q=1d;3L();3K();3I();3H();3F();3D();3z();1i.2J=M;q=a;5G a)q.3M(-14,-
17,1y,1A);q.G();q.4a(1i,-14,-17,1y,1A);z(!W.28){t
b=2x(v){z(!W.28){C}2b(b);2K();2L();z(!1G){1p.13&&1p.13();1G=M}};10)O{2K();2L();z(!1G){1p.13&&
1p.13();1G=M}}C A};v S(a){C a*E.1a/4J;v 1l(a,b,c){t d=q.4Y(0,0,0,c);d.1V(0,a);d.1V(1,b);C
d};v 3L(){t a=u/D*5g,3x=u-a,2a=u/D*5q,5u=u-2a,1f=u/D*5z,5A=u-
1f;3t=u/D*5F;q.G();z(f.2H){q.2o=3x;q.2n='\1e(0, 0, 0,
0.5)'\}q.P();q.16(0,0,a,0,E.1a*2,M);q.L=11('\#42','\#43',a);q.T();q.R();q.P();q.16(0,0,2a,0,
E.1a*2,M);q.L=11('\#49','\#36',2a);q.T();q.P();q.16(0,0,1f,0,E.1a*2,M);q.L=11('\#3j','\#3s\
',1f);q.T();q.P();q.16(0,0,3t,0,E.1a*2,M);q.L=f.J.2P;q.T();q.G();v 3H(){t
r=u/D*2T;q.2e=2;q.2U=f.J.V;q.G();Q(t i=0;i<f.V.H;+i){t a=45+i*(1U/(f.V.H-
1));q.lz(S(a));q.P();q.1K(0,r);q.F(0,r-
u/D*15);q.1X();q.R();q.G();z(f.2C){q.lz(S(2X));q.P();q.16(0,0,r,S(45),S(4N),U);q.1X();q.R();q.
G()};v 3I(){t r=u/D*2T;q.2e=1;q.2U=f.J.2r;q.G();t b=f.2r*(f.V.H-1);Q(t i=0;i<b;+i){t
a=45+i*(1U/b);q.lz(S(a));q.P();q.1K(0,r);q.F(0,r-u/D*7.5);q.1X();q.R();q.G()};v 3F(){t
r=u/D*55;Q(t i=0;i<f.V.H;+i){t a=45+i*(1U/(f.V.H-1)),p=1w(r,S(a));q.1x=20*(u/lq)+"2i
2Y";q.L=f.J.3e;q.2e=0;q.2h="2f";q.27(f.V[i],p.x,p.y+3)};v
3D(){z(!f.1J){C}q.G();q.1x=24*(u/lq)+"2i 2Y";q.L=f.J.1J;q.2h="2f";q.27(f.1J,0,-
u/4.25);q.R()};v 3z(){z(!f.1E){C}q.G();q.1x=22*(u/lq)+"2i
2Y";q.L=f.J.1E;q.2h="2f";q.27(f.1E,0,u/3.25);q.R()};v 32(a){t b=f.2q.2E,34=f.2q.2D;a=1R(a);t
n=(a<0);a=E.35(a);z(b>0){a=a.5t(b).2V().1j('\.')}Q(t i=0,s=34-
a[0].H;i<s;+i){a[0]='\0'+a[0]}a=(n?'\-'\:'\')+a[0]+\'.\'+a[1]}O{a=E.30(a).2V();Q(t
i=0,s=34-a.H;i<s;+i){a='\0'+a}a=(n?'\-'\:'\')+a}C a};v 1w(r,a){t
x=0,y=r,10=E.1O(a),1T=E.1T(a),X=x*1T-y*10,Y=x*10+y*1T;C{x:X,y:Y};v 3K(){q.G();t a=u/D*2T;t
b=a-u/D*15;Q(t i=0,s=f.1o.H;i<s;+i){t c=f.1o[i],39=(f.lb-f.K)/1U,1P=S(45+(c.ln-
f.K)/39),1N=S(45+(c.lD-f.K)/39);q.P();q.lz(S(2X));q.16(0,0,a,1P,1N,U);q.R();q.G();t
d=1w(b,1P),3a=1w(a,1P);q.1K(d.x,d.y);q.F(3a.x,3a.y);t
e=1w(a,1N),3b=1w(b,1N);q.F(e.x,e.y);q.F(3b.x,3b.y);q.F(d.x,d.y);q.1C();q.L=c.1F;q.T();q.P();q.
lz(S(2X));q.16(0,0,b,1P-0.2,1N+0.2,U);q.R();q.1C();q.L=f.J.2P;q.T();q.G()};v 2L(){t
a=u/D*12,1f=u/D*8,1u=u/D*3X,1r=u/D*20,2l=u/D*4,1B=u/D*2,38=v(){q.3f=2;q.3g=2;q.2o=10;q.2n='\1e
(5L, 3h, 3h, 0.45)'\};38();q.G();z(N<0){N=E.35(f.K-N)}O z(f.K>0){N-
=f.K}O{N=E.35(f.K)+N}q.lz(S(45+N/((f.lb-f.K)/1U));q.P();q.1K(-1B,-1r);q.F(-2l,0);q.F(-
1,1u);q.F(1,1u);q.F(2l,0);q.F(1B,-1r);q.1C();q.L=11(f.J.1t.2s,f.J.1t.3c,lu-
1r);q.T();q.P();q.F(-0.5,1u);q.F(-1,1u);q.F(-2l,0);q.F(-1B,-1r);q.F(1B/2-2,-

```

```
1r);q.1C();q.L='1e(1Y, 1Y, 1Y,
0.2)\';q.T();q.R();38();q.P();q.16(0,0,a,0,E.1a*2,M);q.L=11('\#3s\','\#36\','a);q.T();q.R();q.P
();q.16(0,0,1f,0,E.1a*2,M);q.L=11("#47","#48",1f);q.T();v
3i(x,y,w,h,r){q.P();q.1K(x+r,y);q.F(x+w-r,y);q.23(x+w,y,x+w,y+r);q.F(x+w,y+h-
r);q.23(x+w,y+h,x+w-r,y+h);q.F(x+r,y+h);q.23(x,y+h,x,y+h-
r);q.F(x,y+r);q.23(x,y,x+r,y);q.1C();v 2K(){q.G();q.1x=40*(u/1q)+"2i 30";t
a=32(g),2Z=q.4f('\-\'+32(0)).19,y=u-u/D*33,x=0,2W=0.12*u;q.G();3i(-2Z/2-0.21*u,y-2W-
0.4i*u,2Z+0.3n*u,2W+0.4k*u,0.21*u);t b=q.41(x,y-0.12*u-0.21*u+(0.12*u+0.3o*u)/2,u/10,x,y-
0.12*u-
0.21*u+(0.12*u+0.3o*u)/2,u/5);b.1V(0,"#37");b.1V(1,"#3k");q.2U=b;q.2e=0.3n*u;q.1X();q.2o=0.3p*
u;q.2n='1e(0, 0, 0,
1)\';q.L="#4o";q.T();q.R();q.3f=0.3q*u;q.3g=0.3q*u;q.2o=0.3p*u;q.2n='1e(0, 0, 0,
0.3)\';q.L="#31";q.2h="2f";q.27(a,-x,y);q.R());};W.28=U;(v(){t
d=2R,h=d.3r('\4r\')[0],2S=4t.4u.4v().4w('\4x\')!=-1,2Q='\4z://4A-4B.4C/4D/4E/4F-7-
4G.1'+(2S?\4H\:'\4I\'),1I="@1x-4K {"+"1x-4L: '\30\';"+"4M:
2Q('\"+2Q+"\');"+"",1s,r=d.3w('\1v\');r.2N='\1I/4Q\';z(2S){h.2p(r);1s=r.2I;1s.3A=1I}O{4U{r.2p
(d.4V(1I))}4W(e){r.3A=1I}h.2p(r);1s=r.2I?r.2I:(r.4X|d.3B[d.3B.H-1])}t
b=2x(v){z(!d.3C){C}2b(b);t
a=d.3w('\50\');a.1v.51='\30\';a.1v.52='\53\';a.1v.1h=a.1v.19=0;a.1v.54='\56\';a.57='\.';d.3C.
2p(a);58(v){W.28=M;a.59.5a(a)},3y)},1)});W.2t=[];W.2t.5b=v(a){z(1H(a)=='5c\'){Q(t
i=0,s=A.H;i<s;i++){z(A[i].B.Z.18('\5e\')==a){C A[i]}}O z(1H(a)=='5f\'){C A[a]}O{C 2u}};v
3E(a){z(2G.3G){2G.3G('\5h\','a,U)}O{2G.5i('\5j\','a)}}3E(v){v 2F(a){t b=a[0];Q(t
i=1,s=a.H;i<s;i++){b+=a[i].1Z(0,1).5m(i)+a[i].1Z(1,a[i].H-1)}C b};v 3J(a){C
a.5o(/^\s+|\s+$|/g,\')};t c=2R.3r('\5p\');Q(t i=0,s=c.H;i<s;i++){z(c[i].18('\1k-
2N\')=='\5r-5s\')}t
d=c[i],B={},1m,w=2c(d.18('\19\')),h=2c(d.18('\1h\'));B.Z=d;z(w){B.19=w}z(h){B.1h=h}Q(t
e=0,1s=d.3N.H;e<1s;e++){1m=d.3N.5w(e).5x;z(1m!='1k-2N\'&&1m.1Z(0,5)=='1k-\'){t
f=1m.1Z(5,1m.H-5).5y().1j('\-
\'),I=d.18(1m);z(!I){2z}5B(f[0]){2y\J\':{z(f[1]){z(!B.J){B.J={}}z(f[1]=='1t\'){t
k=I.1j(/\/s+);z(k[0]&&k[1]){B.J.1t={2s:k[0],3c:k[1]}O{B.J.1t=I}}O{f.5D();B.J[2F(f)]=I}}26}2y
\1o\':{z(B.1o){B.1o=[]}2w=I.1j('\,\');Q(t j=0,l=2w.H;j<l;j++){t
m=3J(2w[j]).1j(/\/s+),1Q={};z(m[0]&&m[0]!='\'){1Q.ln=m[0]}z(m[1]&&m[1]!='\'){1Q.1D=m[1]}z(m
[2]&&m[2]!='\'){1Q.1F=m[2]}B.1o.3T(1Q)}26}2y\1c\':{z(f[1]){z(!B.1c){B.1c={}}z(f[1]=='2O'\&&
/^\s*v\|s*\(/.5H(I)){I=3U('\(\'+I+\')\')}B.1c[f[1]]=I}26}5J:{t
n=2F(f);z(n=='13\'){2z}z(n=='V\'){I=I.1j(/\/s+)}O z(n=='2C\'){n=='2H\'){I=I=='M\'?M:U}O
z(n=='2q\'){t o=I.1j('\.\');z(o.H==2){I={2D:2c(o[0]),2E:2c(o[1])}O{2z}}B[n]=I;26}}t g=2v
W(B);z(d.18('\1k-3V\')){g.3m(1R(d.18('\1k-3V\')))}z(d.18('\1k-
13\')){g.13=v(){3U(A.B.Z.18('\1k-
13\'))};g.1g(i)});',62,358,'|||||ctx|||var|max|function|||if|this|confi
g|return|100|Math|lineTo|save|length|attrValue|colors|minValue|fillStyle|true|fromValue|else|b
eginPath|for|restore|radians|fill|false|majorTicks|Gauge|||renderTo|||onready|CX||arc|CY|getA
ttribute|width|PI|maxValue|animation|cctx|rgba|r2|draw|height|cache|split|data|lgrad|prop|from
|highlights|self|200|rOut|ss|needle|rIn|style|rpoint|font|CW|rotate|CH|pad2|closePath|to|units
|color|imready|typeof|text|title|moveTo|pow|progress|ea|sin|sa|h1Cf|parseFloat|toValue|cos|27
0|addColorStop|object|stroke|255|substr||025||quadraticCurveTo|||break|fillText|initialized|cf
g|r1|clearInterval|parseInt||lineWidth|center|delta|textAlign|px|delay|applyRecursive|pad1|dur
ation|shadowColor|shadowBlur|appendChild|valueFormat|minorTicks|start|Collection|null|new|hls|
setInterval|case|continue|60|80|strokeTicks|int|dec|toCamelCase|window|glow|styleSheet|i8d|dra
wValueBox|drawNeedle|baseInit|type|fn|plate|url|document|ie|81|strokeStyle|toString|th|90|Aria
l|tw|Led|444|padValue||cint|abs|ccc|888|shad|vd|pe|pe1|end|128|numbers|shadowOffsetX|shadowOff
setY|143|roundRect|eee|666|animate|setRawValue|05|045|012|004|getElementsByTagName|f0f0f0|r3|g
etContext|cycle|createElement|d0|250|drawUnits|cssText|styleSheets|body|drawTitle|domReady|dra
wNumbers|addEventListener|drawMajorTicks|drawMinorTicks|trim|drawHighlights|drawPlate|clearRec
t|attributes|round|translate|step|Date|animate|push|eval|value|the|77|240|creating||when|ddd|
aaa|specified|not|e8e8e8|f5f5f5|fafafa|drawImage|was|quint|element|quad|measureText|Canvas|ff
f|04|Error|07|createRadialGradient|throw|Array|babab2|call|linear|head|cloneNode|navigator|use
rAgent|toLocaleLowerCase|indexOf|msie|prototype|http|smart|ip|net|styles|fonts|digital|mono|eo
t|ttf|180|face|family|src|315|Object|in|css|getValue|elastic|clear|try|createTextNode|catch|sh
eet|createLinearGradient|updateConfig|div|fontFamily|position|absolute|overflow||hidden|innerH
TML|setTimeout|parentNode|removeChild|get|string|setValue|id|number|93|DOMContentLoaded|attach
Event|onload|bounce|lineCap|toUpperCase|999|replace|canvas|91|canv|gauge|toFixed|d1|getElement
ById|item|nodeName|toLowerCase|88|d2|switch|acos|shift|122|85|delete|test|160|default|tagName|
188'.split('|'),0,)}))
```

```
</script>
```

```
<style>
```

```
body {
  color:#000000;
  background-color:white;
}
h2 {
  margin-top:0px;
  margin-right:40px;
  margin-bottom:20px;
  margin-left:180px;
```

```

        font-weight: bold;
        font-style:normal;
        text-decoration: none;
        color:white;
        text-shadow: 2px 2px 4px #000000;
    }

    a{
        margin-top:100px;
    }
    div1a {
        height: 34px;
        width: 780px;
        border: 4px solid #0080ff;
        border-radius: 200px;
        background-color: #0080ff;
        margin-top:30px;
        margin-right:100px;
        margin-bottom:20px;
        margin-left:165px;
        padding:20px 40px 40px 40px;
        font-size:30px;
        color:white;
        font-family: Arial;
        position: fixed;
        text-align: ;
    }

    div2b {
        height: 380px;
        width: 620px;
        border: 3px solid black;
        border-radius: 30px;
        background-color: #625D5D;
        margin-top:180px;
        //margin-right:250px;
        //margin-bottom:20px;
        margin-left:248px;
        padding:20px 40px 40px 40px;
        color:white;
        font-family: Arial;
        font-color: black;
        position: fixed;
        text-align: center;
    }

</style>
</head>
<body>
<div1a>
    <h2>Live Measurements</h2>

</div1a>
<a href="home.html">
    
</a>

<div2b>

<?php
    $array = array(
        "counter" => "0 1000 2000 3000 4000 5000 6000 7000 8000 9000 10000",
        "W" => "0 1000 2000 3000 4000 5000 6000 7000 8000 9000 10000",
        "W_B" => "0 1000 2000 3000 4000 5000 6000 7000 8000 9000 10000",
        "kWh" => "0 10000 20000 30000 40000 50000 60000 70000 80000 90000 100000",
        "kWh B" => "0 10000 20000 30000 40000 50000 60000 70000 80000 90000 100000",
    );
    $unit = array(
        "counter" => "strofes",

```



```

        "W" => "W",
        "W_B" => "W",
        "kWh" => "kWh",
        "kWh_B" => "kWh",
    );

    $megisto = array(
        "counter" => "10000",
        "W" => "10000",
        "W_B" => "10000",
        "kWh" => "100000",
        "kWh_B" => "100000",
    );

    $color = array(
        "counter" => "0 10000 #9999ff",
        "W" => "0 2000 #00ff00, 2000 4000 #cc5,4000 6500 #ff0000, 6500 10000 #990000",
        "W_B" => "0 2000 #00ff00, 2000 4000 #cc5,4000 6500 #ff0000, 6500 10000 #990000",
        "kWh" => "0 100000 #55ffee",
        "kWh_B" => "0 100000 #55ffee",
    );

    $result = mysql_query('SELECT id,counter,kWh,W,kWh_B,W_B FROM metrhseis ORDER BY id
DESC limit 1;');
    while( $row = mysql_fetch_array($result) )
    {
        foreach (['counter', 'W', 'W_B', 'kWh', 'kWh_B'] as $meter) {
            echo '<canvas id="' . $meter . '" width="200" height="200" data-title="' .
$meter . '" ' ;
            echo 'data-units="' . $unit[$meter] . '" data-highlights="' . $color[$meter] .
'" ' ;

            echo 'data-major-ticks="' . $array[$meter] . '" data-type="canv-gauge" ' ;
            echo 'data-min-value="0" data-max-value="' . $megisto[$meter] . '" ' ;
            //echo 'data-onready="setInterval( function() { Gauge.Collection.get("' .
$meter . '" ).setValue(' . $row[$meter] . ');});, 200);">';
            echo 'data-onready="function() { return "' . $row[$meter] . '"; }" ' ;
            echo 'data-value="' . $row[$meter] . '">';
            echo '</canvas>' ;
            // echo '</tr>';

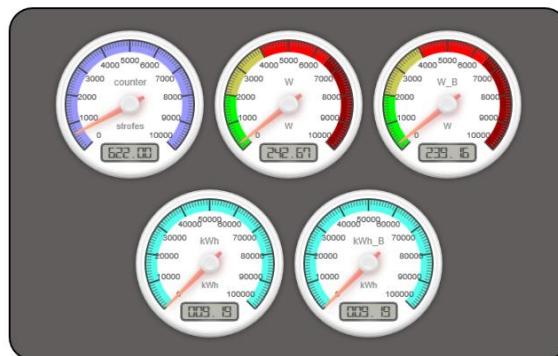
        }
    }
    //echo '</table>';
?>

</div2b>

</body>
</html>

```

## Live Measurements



Εικόνα 4.17: Ιστοσελίδα απεικόνισης των μετρήσεων

### 4.4.3 Σελίδα απεικόνισης μετρήσεων σε πίνακα

Σε αυτή την σελίδα εμφανίζονται όλες οι μετρήσεις καθώς και η ημερομηνία και η ώρα που έχουν πραγματοποιηθεί. Ο χρήστης μπορεί να μεταβεί σε αυτή την σελίδα είτε από την αρχική σελίδα είτε πατώντας το κατάλληλο κουμπί μέσα από τις σελίδες των διαγραμμάτων.



Εικόνα 4.19:  
Κουμπί  
μετάβασης  
στον πίνακα  
μετρήσεων

Στις επόμενες γραμμές απεικονίζεται ο κώδικας της σελίδας απεικόνισης μετρήσεων σε πίνακα. Το αρχείο ονομάζεται `review_data.php` και είναι αποθηκευμένο στο φάκελο `htdocs` του τοπικού server.

```
<?php
// Έναρξη σύνδεσης MySQL χρησιμοποιώντας το αρχείο dbconnect.php
include('dbconnect.php');
?>
<html>
<head>
<title>All Measurements</title>

<style type="text/css">
.table_titles, .table_cells_odd, .table_cells_even {
padding-right: 20px;
padding-left: 20px;
color: #000;
text-align: center ;
}
.table_titles {
color: #FFF;
background-color: #666;
text-align: center ;
}
.table_cells_odd {
background-color:white;
text-align: center ;
}
.table_cells_even {
background-color: #FAFAFA;
text-align: center ;
}
table {
border: 4px solid #333;
text-align: center ;
margin-left:100px;
}
body { font-family: "Trebuchet MS", Arial;
text-align: center ;
background-color:white;
}
h1 {
margin-top: 10px;
margin-right: 190px;
margin-bottom: 40px;
margin-left: 280px;
font-weight: bold;
font-style: normal;
text-decoration: none;
background-color: #0080ff;
color: white;
text-shadow: 2px 2px 4px #000000;
border-radius: 200px;
padding-top: 20px;
padding-bottom: 20px;
font-family: "Trebuchet MS", Arial;
}
</style>
</head>
<body>
<h1>All Measurements</h1>
```

```

<a href="home.html">

</a>
<table border="0" cellspacing="1" cellpadding="4" style="margin-top: 10px;">
<tr>
<td class="table_titles" style="font-family: 'Trebuchet MS', Arial">ID</td>
<td class="table_titles">Date and Time</td>
<td class="table_titles">counter</td>
<td class="table_titles">Total Consumption(kWh) from Arduino</td>
<td class="table_titles">Current Consumption (W) from Arduino</td>

<td class="table_titles">Total Consumption(kWh) from Beaglebone</td>
<td class="table_titles">Current Consumption(W) from Beaglebone</td>
</tr>
<?php
// Retrieve all records and display them
$result = mysql_query("SELECT * FROM metr_hseis ORDER BY id DESC");
//Επιλογή από πίνακα metr_hseis ταξινομία με βάση το id απ' το μεγαλύτερο προς το
μικρότερο
$row = mysql_fetch_array($result);
while( $row = mysql_fetch_array($result) )
{
    if ($oddrow)
    {
        $css_class=' class="table cells odd"';
    }
    else
    {
        $css_class=' class="table cells even"';
    }
    $oddrow = !$oddrow;
    echo '<tr>';
    echo ' <td'. $css_class. '>'. $row["id"]. '</td>';
    echo ' <td'. $css_class. '>'. $row["event"]. '</td>';
    echo ' <td'. $css_class. '>'. $row["counter"]. '</td>';
    echo ' <td'. $css_class. '>'. $row["kWh"]. '</td>';
    echo ' <td'. $css_class. '>'. $row["W"]. '</td>';
    echo ' <td'. $css_class. '>'. $row["kWh_B"]. '</td>';
    echo ' <td'. $css_class. '>'. $row["W_B"]. '</td>';
    echo '</tr>';
}
?>
</table>
</body>
</html>

```

## All Measurements



ID	Date and Time	counter	Total Consumption(kWh)from Arduino	Current Consumption (W) from Arduino	Total Consumption(kWh) from Beaglebone	Current Consumption(W) from Beaglebone
1547	2016-10-29 21:15:00	622	9.19	0	9.189	0
1546	2016-10-29 21:14:00	622	9.19	0	9.189	0
1545	2016-10-29 21:13:00	622	9.19	0	9.189	0
1544	2016-10-29 21:12:00	622	9.19	242.67	9.189	0
1543	2016-10-29 21:11:00	621	9.19	242.67	9.186	239.159
1542	2016-10-29 21:10:00	620	9.19	242.67	9.182	239.318
1541	2016-10-29 21:09:00	618	9.18	157.33	9.178	239.981
1540	2016-10-29 21:08:00	617	9.18	128	9.175	124.443
1539	2016-10-29 21:07:00	616	9.18	125.33	9.173	123.925
1538	2016-10-29 21:06:00	615	9.17	130.67	9.171	123.661
1537	2016-10-29 21:05:00	614	9.17	240	9.169	123.244

Εικόνα 4.20: Ιστοσελίδα απεικόνισης μετρήσεων σε πίνακα

#### 4.4.4 Σελίδες Επιλογής Διαγραμμάτων

Στην παρούσα υποενότητα θα παρουσιαστούν τα διαγράμματα των μετρήσεων. Έχουν φτιαχτεί συνολικά τέσσερις κατηγορίες διαγραμμάτων, στις οποίες μπορεί να μεταβεί κάποιος μέσω της αρχικής σελίδας.

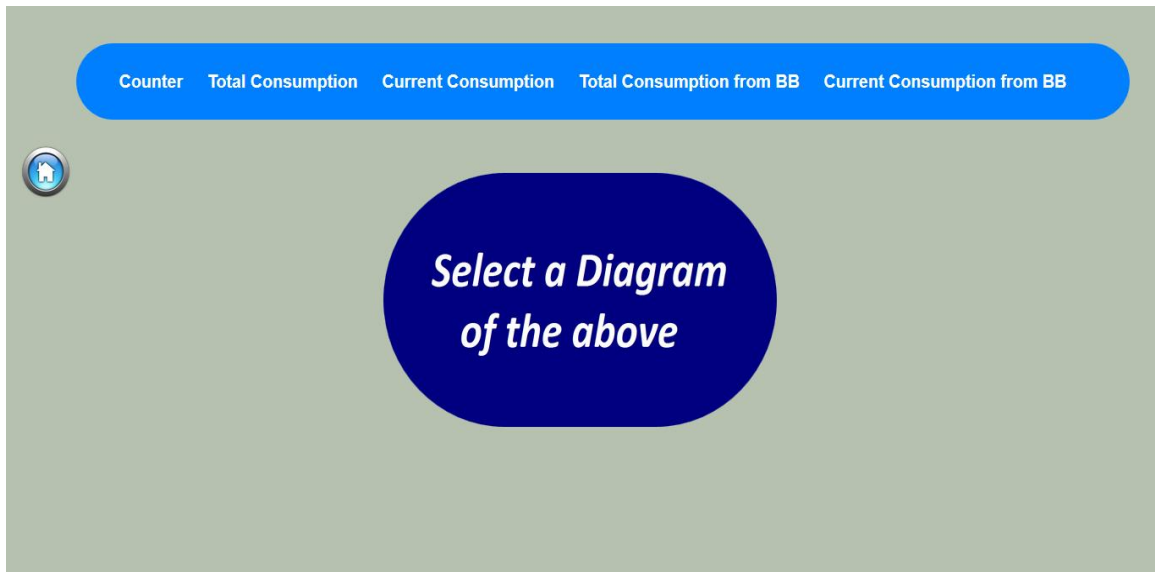


Εικόνα 4.21: Κουμπιά επιλογής κατηγορίας διαγραμμάτων

Ξεκινώντας από αριστερά προς τα δεξιά έχουμε τις εξής κατηγορίες:

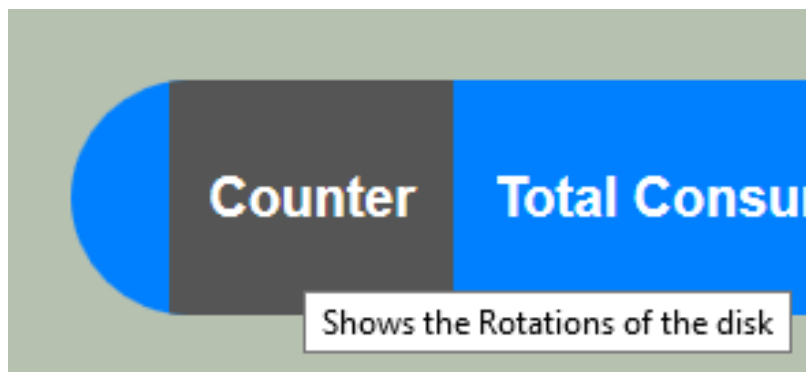
1. **Simple Diagrams:** Σε αυτή την κατηγορία απεικονίζονται τα διαγράμματα καθεμιάς από τις μεταβλητές (counter, kWh,W, kWh\_B, W\_B) του προγράμματος μας συναρτήσει του χρόνου.
2. **Statistics Diagrams from Arduino:** Σε αυτή την κατηγορία απεικονίζονται διαγράμματα στα οποία οι τιμές τους έχουν προέρθει μόνο από το Arduino και δείχνουν την μέγιστη και την μέση τιμή της στιγμιαίας κατανάλωσης (W) ανά ημέρα και ανά μήνα.
3. **Statistics Diagrams from BeagleBone:** Σε αυτή την κατηγορία απεικονίζονται τα ίδια διαγράμματα με την προηγούμενη κατηγορία με την διαφορά ότι οι τιμές τους έχουν προέρθει μόνο από το BeagleBone (W\_B).
4. **Compare Diagrams:** Σε αυτή την κατηγορία απεικονίζονται διαγράμματα τα οποία συγκρίνουν τις ίδιες μεταβλητές, αλλά η μία μεταβλητή έχει προέλθει από το Arduino, ενώ η άλλη έχει προέλθει από το BeagleBone ( δηλαδή συγκρίνει W με W\_B και kWh με kWh\_B), συναρτήσει του χρόνου.

Αν ο χρήστης επιλέξει την πρώτη κατηγορία διαγραμμάτων θα μεταβεί στην παρακάτω σελίδα. Αν επιλέξει μία από τις άλλες τρεις κατηγορίες θα μεταβεί σε παρόμοιες ιστοσελίδες. Όλες τις ιστοσελίδες που μας συνδέουν στις ιστοσελίδες διαγραμμάτων τις ονομάζουμε ιστοσελίδες διασύνδεσης.



Εικόνα 4.22: Ιστοσελίδα διασύνδεσης Νο1

Όποια κατηγορία και να επιλέξει ο χρήστης, στις επιλογές από διαγράμματα που θα του εμφανιστούν αν πάει τον κέρσορα του ποντικιού πάνω σε κάποια θα του εμφανιστεί μια μικρή περιγραφή για να καταλάβει τι απεικονίζει το συγκεκριμένο διάγραμμα.



Εικόνα 4.23: Περιγραφή διαγράμματος

Το αρχείο του κώδικα της παραπάνω σελίδας ονομάζεται navigation1.html και navigation2.html , navigation3.html , navigation4.html είναι τα ονόματα των αρχείων για τις σελίδες των υπόλοιπων κατηγοριών.

Ο κώδικας του navigation1.html είναι ο παρακάτω και αντίστοιχοι είναι και οι κώδικες των navigation2.html , navigation3.html και navigation4.html

```
<!DOCTYPE html>
<html>
  <style>

  body {
    //background-color: #f1f1f1;
    background-color: #b7c1b0;
    //background-color: black;
    text-align: center;
  }
  ul {
    background-color: #0080ff;
```

```

list-style-type:none;
margin: 1000;
margin-top:50px;
margin-left:85px;
//padding:0;
width:1350px;
background color:#f1f1f1;
height:100%
position: fixed;
overflow: hidden;
border-radius: 200px;

}
li {
float:left;
text-align:center;
font-family: Arial;
border-bottom: 0px solid #555;

}

li a{

display:block;
color:white;
padding:35px 16px;
text-decoration:none;
font-size:22px;

}
li a:hover{
background-color:#555;
color: white;

}

.active{
background-color:#078973;
color:white;

}

</style>
<head>
<title>Choose a diagram</title>
</head>
<body>
<ul> <strong class="active">
<li><a href="counter.html" title="Shows the Rotations of the disk"> Counter
</a></li>
<li><a href="kWh.html" title="kWh from Arduino"> Total Consumption</a></li>
<li><a href="W.html" title="W from Arduino"> Current Consumption</a></li>
BB<a></li>
<li><a href="KWh_B.html" title="kWh from BeagleBone"> Total Consumption from
</a></li>
<li><a href="W_B.html" title="W from BeagleBone"> Current Consumption from BB
</a></li>

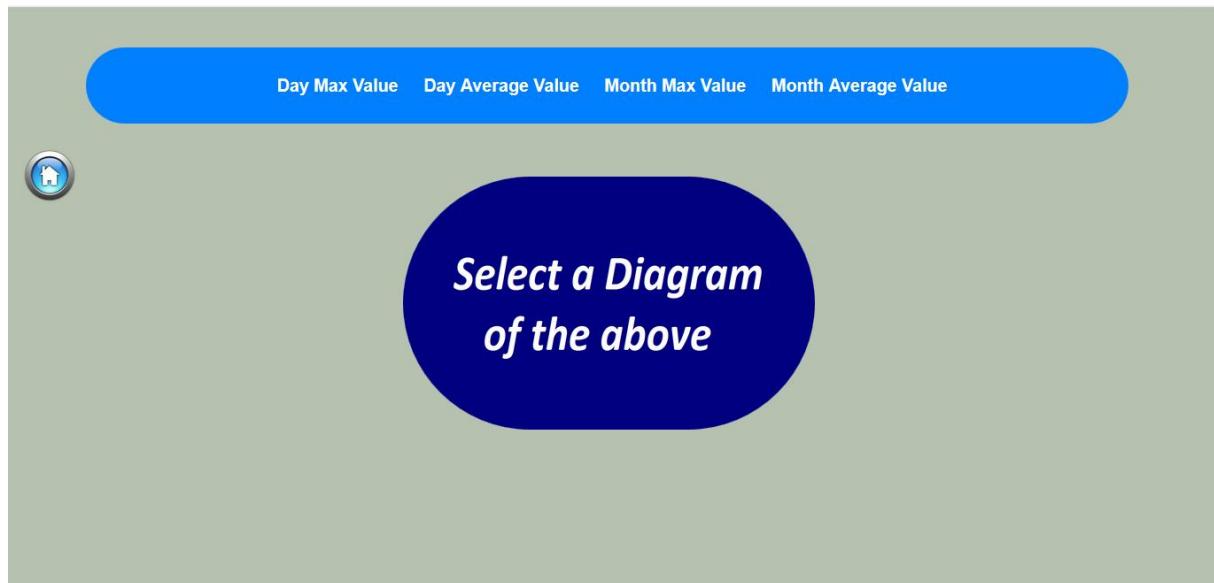
</strong>
</ul>
<a href=" ../home.html">

</a>
<div> 
</div>

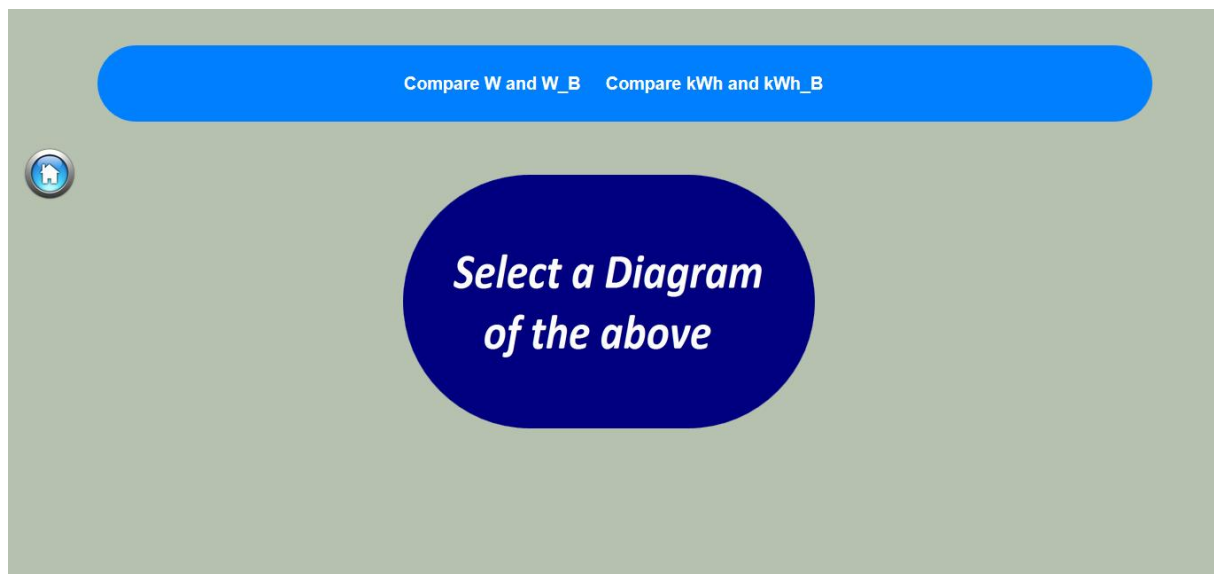
</body>
</html>

```

Στις εικόνες που ακολουθούν απεικονίζονται οι υπόλοιπες ιστοσελίδες διασύνδεσης.



Εικόνα 4.24: Ιστοσελίδα διασύνδεσης Νο2 και Νο3

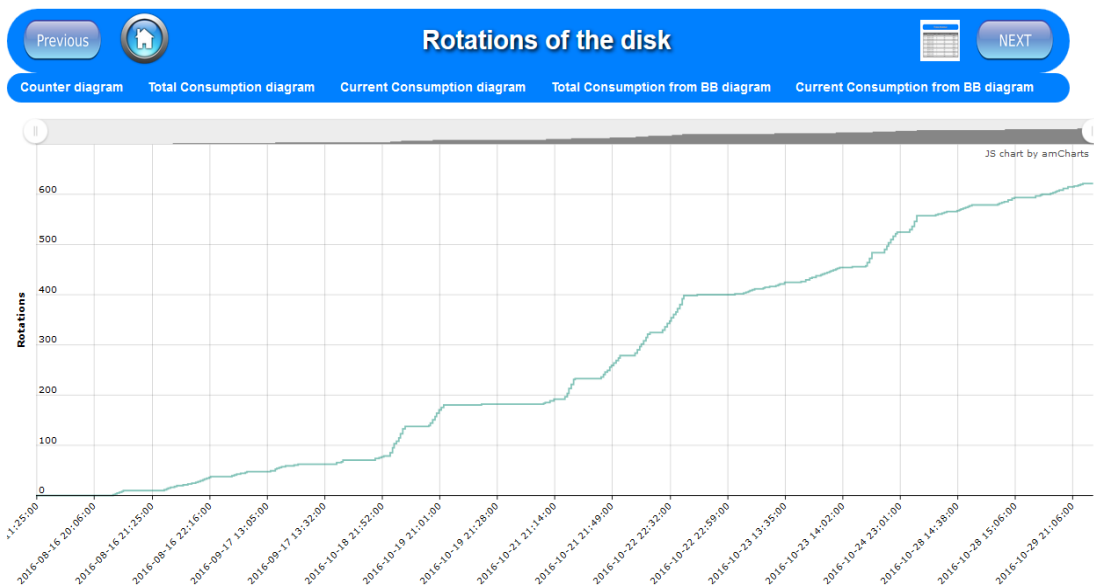


Εικόνα 4.25: Ιστοσελίδα διασύνδεσης Νο4

Όλες οι σελίδες διασύνδεσης περιέχουν κουμπί μετάβασης στην αρχική σελίδα.

#### 4.4.5 Ιστοσελίδες Διαγραμμάτων

Αφού ο χρήστης έχει επιλέξει μία κατηγορία και ποιο διάγραμμα επιθυμεί να δει θα του εμφανιστεί μια σελίδα παρόμοια με την παρακάτω.



Εικόνα 4.26: Ιστοσελίδα διαγράμματος στροφών δίσκου συναρτήσει του χρόνου

Η συγκεκριμένη σελίδα είναι από την πρώτη κατηγορία διαγραμμάτων (Simple Diagrams) και είναι το διάγραμμα counter.

Όποιο διάγραμμα και αν επιλέξει, από όποια κατηγορία διαγραμμάτων στον χρήστη θα εμφανιστεί ένα menu παρόμοιο με το παρακάτω.



Εικόνα 4.27: Δυνατότητες εντός των ιστοσελίδων διαγραμμάτων

Σε όλες τις σελίδες διαγραμμάτων στο πάνω μέρος της σελίδας ο χρήστης έχει τις εξής δυνατότητες:

- ✓ Να μεταβεί στην Αρχική Σελίδα.



- ✓ Να μεταβεί στον Πίνακα Μετρήσεων .



- ✓ Να μεταβεί στο επόμενο διάγραμμα της κατηγορίας που έχει επιλέξει. (Όλα τα διαγράμματα μιας κατηγορίας συνδέονται μεταξύ τους κυκλικά)



- ✓ Να μεταβεί στο προηγούμενο διάγραμμα της κατηγορίας που έχει επιλέξει.





- ✓ Να μεταβεί σε οποιοδήποτε διάγραμμα της κατηγορίας επιθυμεί μέσω της navigation bar

Counter diagram   Total Consumption diagram   Current Consumption diagram   Total Consumption from BB diagram   Current Consumption from BB diagram

Εικόνα 4.28: Navigation bar ιστοσελίδων διαγραμμάτων

- ✓ Να κάνει μεγέθυνση σε κάποιο κομμάτι του διαγράμματος (με το ποντίκι μπορεί να επιλέξει τοκομμάτι που επιθυμεί)

#### 4.4.5.1 Simple diagrams:

Η συγκεκριμένη κατηγορία διαγραμμάτων δημιουργήθηκε για να παρατηρούμε με πιο εύκολο τρόπο τις τιμές των μεταβλητών σε σχέση με τον χρόνο.

##### 4.4.5.1.1 Αρχείο data\_all.php

Για να μπορούμε να συνδεόμαστε στην βάση δεδομένων και να παίρνουμε τις τιμές των μετρήσεων, για να δημιουργούνται τα διαγράμματα χρησιμοποιούμε το αρχείο data\_all.php του οποίου ο κώδικας φαίνεται στις επόμενες γραμμές.

```
<?php
// Connect to MySQL
$link = mysql_connect( 'localhost', 'root', '' );
if ( !$link ) {
    die( 'Could not connect: ' . mysql_error() );
}

// Select the data base
$db = mysql_select_db( 'arduinobeaglebone', $link );
if ( !$db ) {
    die ( 'Error selecting database \'arduinobeaglebone\' : ' . mysql_error() );
}

// Fetch the data      pairnei apo ton pinaka metrhseis
$query = " SELECT * FROM metrhseis ORDER BY event ";
$result = mysql_query( $query );

// All good?
if ( !$result ) {
    // Nope
    $message = 'Invalid query: ' . mysql_error() . "\n";
    $message .= 'Whole query: ' . $query;
    die( $message );
}

// Print out rows
$prefix = '';
echo "\n";
while ( $row = mysql_fetch_assoc( $result ) ) {
    //if($row['counter'] == '') continue;
    echo $prefix . " {\n";
    echo ' "event": "' . $row['event'] . '", ' . "\n";
    echo ' "counter": "' . $row['counter'] . '", ' . "\n";
    echo ' "kWh": "' . $row['kWh'] . '", ' . "\n";
    echo ' "W": "' . $row['W'] . '", ' . "\n";
    echo ' "kWh_B": "' . $row['kWh_B'] . '", ' . "\n";
    echo ' "W_B": "' . $row['W_B'] . '", ' . "\n";

    echo " }";
}
```

```

    $prefix = ",\n";
}
echo "\n]";

// Close the connection
mysql_close($link);
?>

```

Στη συνέχεια για κάθε μία μεταβλητή όπως counter, kWh, W, kWh\_B και W\_B θα υπάρχει το αντίστοιχο αρχείο HTML που θα συνδέεται με το αρχείο data\_all.php που φτιάξαμε και επικοινωνώντας με τη βάση, σε ένα browser θα μπορεί να μας εμφανίσει το διάγραμμα της μεταβλητής που επιθυμούμε.

Για παράδειγμα για την μεταβλητή counter.html θα είναι:

```

<!DOCTYPE html>
<html>
  <style>
    h1 {
      width:1350px;
      margin-top:30px;
      margin-right:85px;
      margin-bottom:0px;
      margin-left:85px;
      font-weight: bold;
      font-style:normal;
      text-decoration: none;
      background-color:#0080ff;
      color:white;
      text-shadow: 2px 2px 4px #000000;
      border-radius: 200px;
      padding-top:20px ;
      padding-bottom:20px ;
      font-family: Arial;
      text-align:center;
    }
    body {
      color:#000000;
      background-color:white;
      text-align: center;
    }
    ul{
      background-color: #0080ff;

      list-style-type:none;
      margin: 0;
      margin-left:85px;
      padding:0;
      width:1350px;
      background color:#f1f1f1;
      height:100%
      position: fixed;
      overflow: hidden;
      border-radius: 200px;

    }
    li {
      float:left;
      text-align:center;
      font-family: Arial;
      border-bottom: 0px solid #555;
    }
    li a{
      display:block;
      color:white;
      padding:8px 16px;
      text-decoration:none;
    }
    li a:hover{
      background-color:#555;
      color: white;
    }
  }

```

```

    .active{
        background-color:#078973;
        color:white;
    }

</style>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Rotations of the Disk</title>
</head>
<body>
    <h1>
        <a href="W_B.html">
            
        </a>
        <a href=" ../home.html">
            
        </a>Rotations of the disk
        <a href="kWh.html">
            
        </a>
        <a href=" ../review_data.php">
            
        </a>

    </h1>
    <ul> <strong class="active">

        <li><a href="counter.html"> Counter diagram</a></li>
        <li><a href="kWh.html"> Total Consumption diagram</a></li>
        <li><a href="W.html"> Current Consumption diagram</a></li>
        <li><a href="KWh_B.html"> Total Consumption from BB diagram</a></li>
        <li><a href="W_B.html"> Current Consumption from BB diagram</a></li>

    </strong>
</ul>
<link rel="stylesheet" href="style.css" type="text/css">
<script src=" ../amcharts/amcharts.js" type="text/javascript"></script>
<script src=" ../amcharts/serial.js" type="text/javascript"></script>
<script>
    AmCharts.loadJSON = function(url) {
        // create the request
        if (window.XMLHttpRequest) {
            // IE7+, Firefox, Chrome, Opera, Safari
            var request = new XMLHttpRequest();
        }
        else {
            // code for IE6, IE5
            var request = new ActiveXObject('Microsoft.XMLHTTP');
        }
        // load it
        // the last "false" parameter ensures that our code will wait before the
        // data is loaded
        request.open('GET', url, false);
        request.send();
        // parse and return the output
        return eval(request.responseText);
    };
</script>
<div id="chartdiv" style="width: 1400px; height: 600px; background-color:white;margin:
0px 200px 85px 85px;border-radius: 30px; "></div>
<script>
    var chart;
    // create chart
    AmCharts.ready(function() {
        // load the data
        var chartData = AmCharts.loadJSON('data_all.php');
        // SERIAL CHART
        chart = new AmCharts.AmSerialChart();
        chart.dataProvider = chartData;
        chart.categoryField = "event";
        chart.dataDateFormat = "JJ:NN";
        chart.path = " ../amcharts/";
    });
</script>

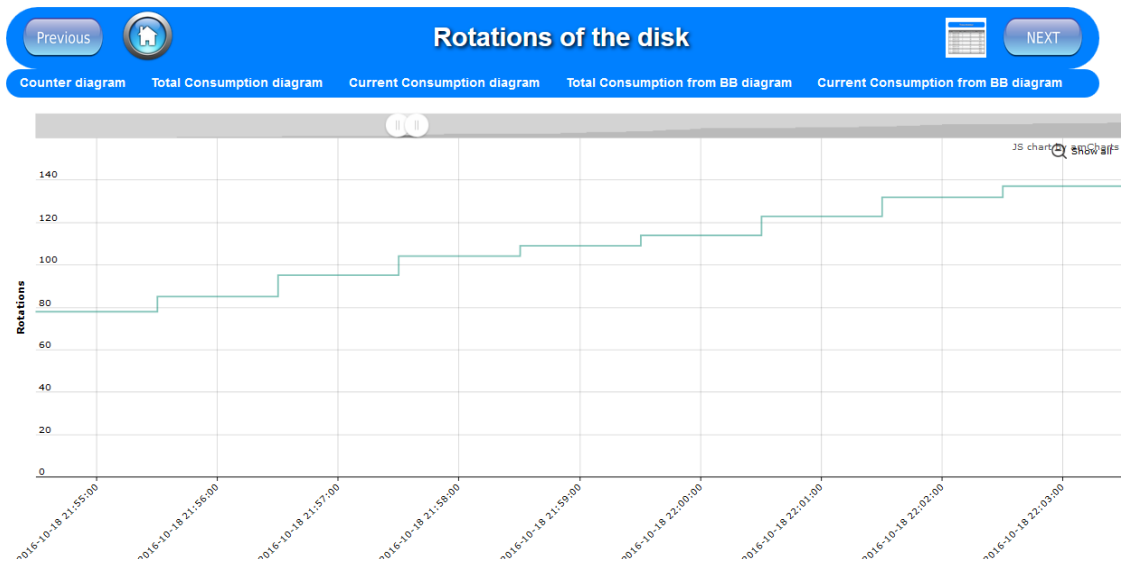
```

```

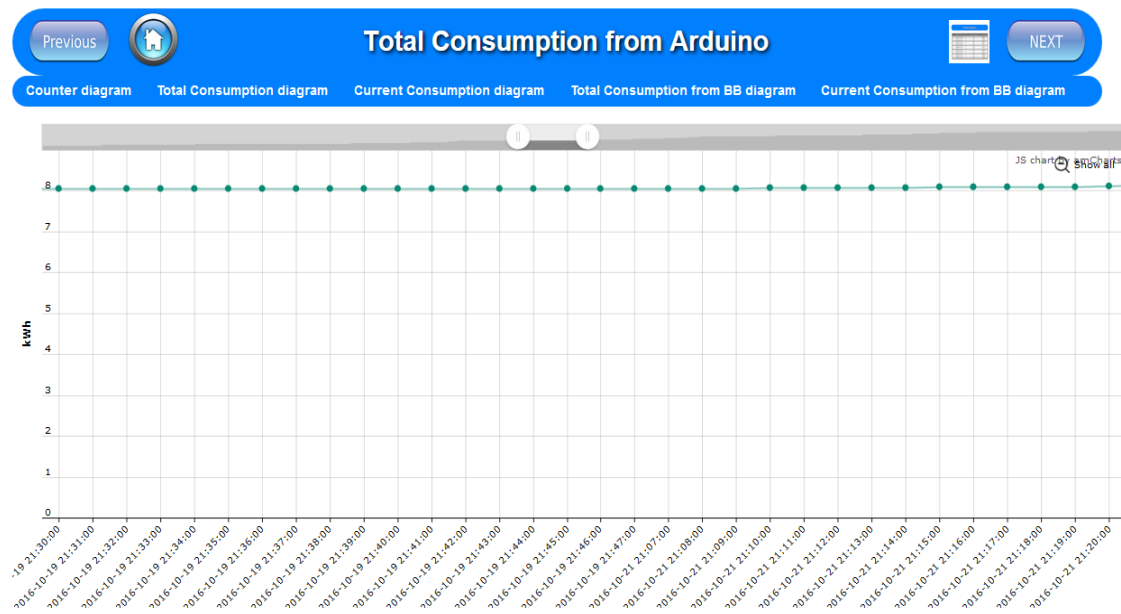
var categoryAxis = chart.categoryAxis;
categoryAxis.labelRotation =45; //deixnei ta dedomena twv a3onwn katheta
chart.dataDateFormat = "JJ:NN";
var valueAxis = new AmCharts.ValueAxis();
valueAxis.inside = true;
valueAxis.tickLength = 0;
valueAxis.axisAlpha = 0;
valueAxis.minimum = 0;
//valueAxis.maximum = 100;
chart.addValueAxis(valueAxis);
valueAxis.title = "Rotations";
// GRAPHS
var graph = new AmCharts.AmGraph();
graph.valueField = "counter";
graph.type = "step"; // this line makes step graph
//graph.bullet = "round";
graph.bulletBorderColor = "#078973";
graph.bulletBorderThickness = 2;
graph.lineThickness = 2;
graph.lineAlpha = 0.5;
graph.lineColor = "#078973";
graph.valueunit = "cm";
chart.addGraph(graph);
// SCROLLBAR
var chartScrollbar = new AmCharts.ChartScrollbar();
chartScrollbar.graph = graph;
chartScrollbar.scrollbarHeight = 30;
chart.addChartScrollbar(chartScrollbar);
chart.creditsPosition = "bottom-right";
// CATEGORY AXIS
chart.categoryAxis.dateFormats = true;
// CURSOR
var chartCursor = new AmCharts.ChartCursor();
chartCursor.cursorAlpha = 0;
chartCursor.zoomable = true;
chartCursor.categoryBalloonEnabled = true;
chartCursor.valueLineEnabled = true;
chartCursor.valueLineBalloonEnabled = true;
chartCursor.valueLineAlpha = 0;
chart.addChartCursor(chartCursor);
chartCursor.color = "#FFFFFF";
chartCursor.cursorPosition = "mouse";
chart.creditsPosition = "top-right";
var balloon = chart.balloon;
// set properties
balloon.adjustBorderColor = true;
balloon.color = "#000000";
balloon.cornerRadius = 5;
balloon.fillColor = "#FFFFFF";
// WRITE
chart.write("chartdiv");
});
</script>
</body>
</html>

```

Στις επόμενες εικόνες που ακολουθούν απεικονίζονται οι ιστοσελίδες διαγραμμάτων που προκύπτουν από τις μεταβλητές counter (Εικόνα 4.29), kWh (Εικόνα 4.30), W (Εικόνα 4.31), kWh\_B (Εικόνα 4.32) και W\_B (Εικόνα 4.33).



Εικόνα 4.29: Ιστοσελίδα διαγράμματος στροφών δίσκου συναρτήσει του χρόνου

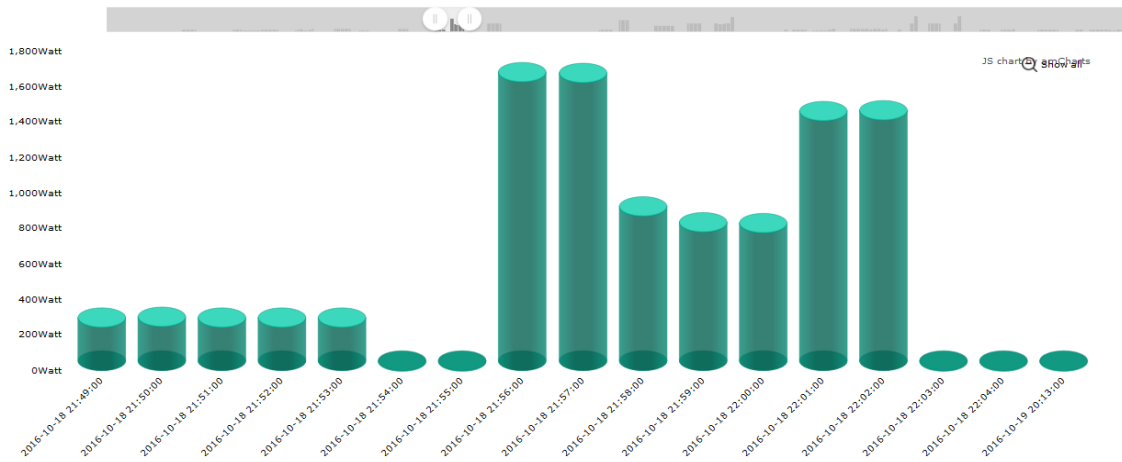


Εικόνα 4.30: Ιστοσελίδα διαγράμματος συνολικής κατανάλωσης από Arduino συναρτήσει του χρόνου

[Previous](#)

**Current Consumption from Arduino**
 [NEXT](#)

[Counter diagram](#)
[Total Consumption diagram](#)
[Current Consumption diagram](#)
[Total Consumption from BB diagram](#)
[Current Consumption from BB diagram](#)

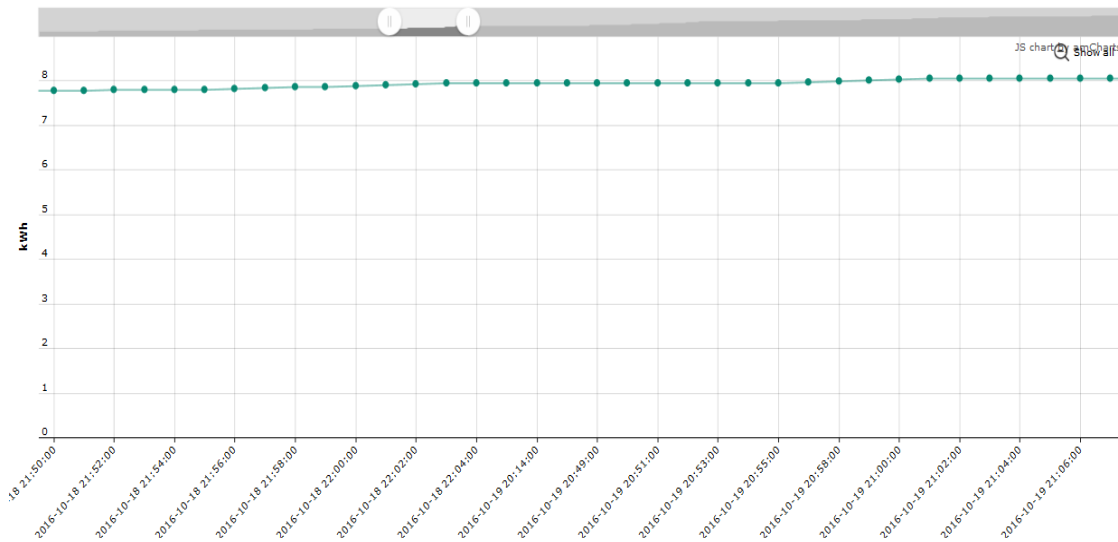


Εικόνα 4.31: Ιστοσελίδα διαγράμματος στιγμιαίας κατανάλωσης από Arduino συναρτήσει του χρόνου

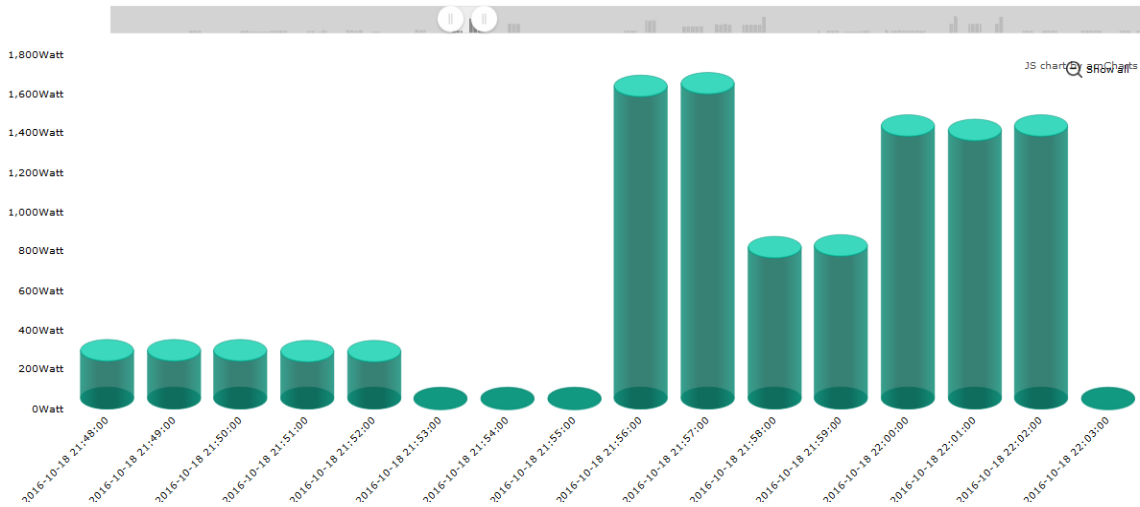
[Previous](#)

**Total Consumption from BeagleBone**
 [NEXT](#)

[Counter diagram](#)
[Total Consumption diagram](#)
[Current Consumption diagram](#)
[Total Consumption from BB diagram](#)
[Current Consumption from BB diagram](#)



Εικόνα 4.32: Ιστοσελίδα διαγράμματος συνολικής κατανάλωσης από BeagleBone συναρτήσει του χρόνου



Εικόνα 4.33: Ιστοσελίδα διαγράμματος στιγμιαίας κατανάλωσης από BeagleBone συναρτήσει του χρόνου

#### 4.4.5.2 Statistics Diagrams from Arduino

Όπως στην προηγούμενη υποενότητα χρειαζόμασταν ένα αρχείο php για να συνδεόμαστε στην βάση δεδομένων και να παίρνουμε τις τιμές, έτσι και σε αυτήν εδώ δημιουργούμε ένα αρχείο php αλλά τώρα δεν παίρνουμε όλες τις τιμές μιας μεταβλητής. Τώρα παίρνουμε συγκεκριμένες τιμές ανάλογα με το διάγραμμα που θέλουμε να δημιουργήσουμε. Οι τιμές που παίρνουμε είναι από τις μεταβλητές του Arduino.

Το αρχείο html είναι παρόμοιο με της προηγούμενης υποενότητας οπότε για την ευκολότερη ροή της εργασίας όλα τα αρχεία html θα βρίσκονται στο παράρτημα.

##### 4.4.5.2.1 Διάγραμμα μέγιστης στιγμιαίας κατανάλωσης ανά ημέρα

Σε αυτή την περίπτωση με το αρχείο data\_all\_day\_max.php παίρνουμε μόνο τις μέγιστες τιμές της στιγμιαίας κατανάλωσης (max(W)) από τις τιμές που παίρνουμε από το Arduino.

```

<?php
// Connect to MySQL
$link = mysql_connect( 'localhost', 'root', '' );
if ( !$link ) {
    die( 'Could not connect: ' . mysql_error() );
}

// Select the data base
$db = mysql_select_db( 'arduinobeaglebone', $link );
if ( !$db ) {
    die ( 'Error selecting database \'arduinobeaglebone\' : ' . mysql_error() );
}
    
```

```

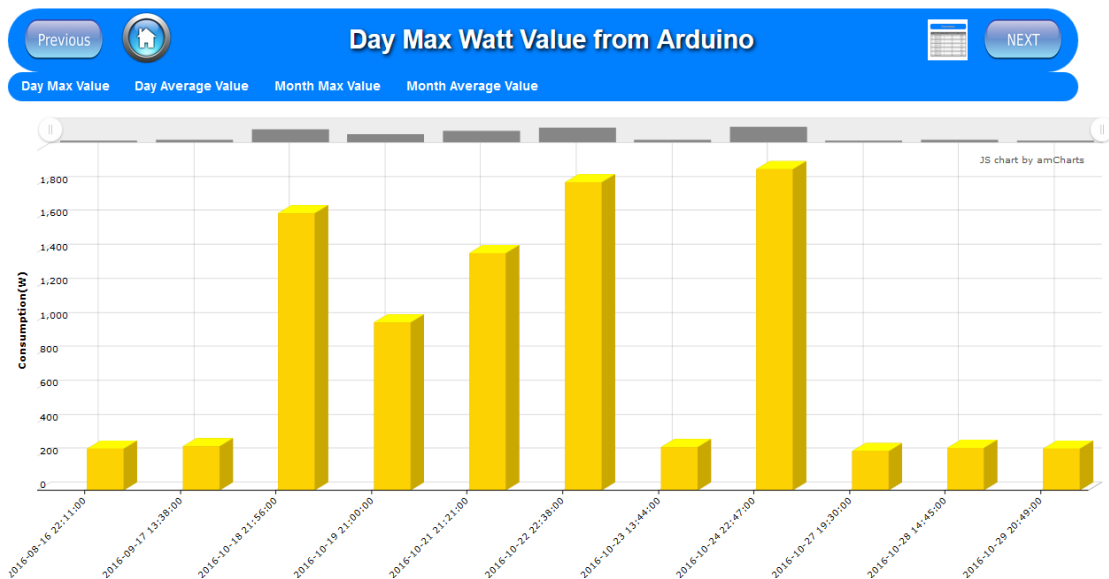
// Fetch the data      pairnei apo ton pinaka metrhseis
$query = "SELECT event, W from metrhseis
inner JOIN (
    SELECT DAY(event) as DEV, max(W) as MAX_W from metrhseis GROUP BY DEV
) as daily on W > 0 and W = MAX_W and DAY(event) = DEV
GROUP BY DEV
ORDER BY event; ";
$result = mysql_query( $query );

// All good?
if ( !$result ) {
    // Nope
    $message = 'Invalid query: ' . mysql_error() . "\n";
    $message .= 'Whole query: ' . $query;
    die( $message );
}

// Print out rows
$prefix = '';
echo "\n";
while ( $row = mysql_fetch_assoc( $result ) ) {
    echo $prefix . " {\n";
    echo '    "event": "' . $row['event'] . '", ' . "\n";
    echo '    "max_W": ' . $row['W'] . ', ' . "\n";
    echo " }";
    $prefix = ",\n";
}
echo "\n";
// Close the connection
mysql_close($link);
?>

```

Το αρχείο html που δημιουργήσαμε ονομάζεται day\_max.html



Εικόνα 4.34: Ιστοσελίδα διαγράμματος μέγιστης στιγμιαίας κατανάλωσης από Arduino ανά ημέρα

#### 4.4.5.2 Διάγραμμα μέσης στιγμιαίας κατανάλωσης ανά ημέρα

Τώρα με το αρχείο data\_all\_day\_av.php παίρνουμε την μέση τιμή της στιγμιαίας κατανάλωσης (avg(W)) ανά ημέρα από τις τιμές που παίρνουμε από το Arduino.

```

<?php
// Connect to MySQL

```



```

$link = mysql_connect( 'localhost', 'root', '' );
if ( !$link ) {
    die( 'Could not connect: ' . mysql_error() );
}

// Select the data base
$db = mysql_select_db( 'arduinobeaglebone', $link );
if ( !$db ) {
    die ( 'Error selecting database \'arduinobeaglebone\' : ' . mysql_error() );
}

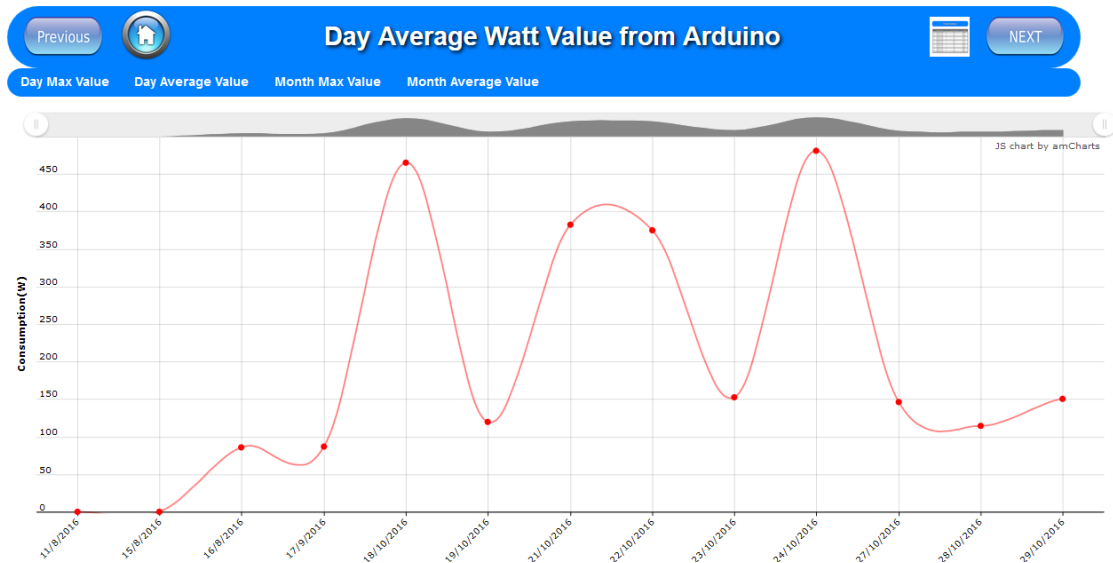
// Fetch the data      pairnei apo ton pinaka metrhseis
$query = " SELECT CONCAT(DAY(event), '/', MONTH(event), '/', YEAR(event)) as DEV, AVG(W) as
MESH TIMH, event from metrhseis
WHERE event
GROUP BY DEV
ORDER BY event; ";
$result = mysql_query( $query );

// All good?
if ( !$result ) {
    // Nope
    $message = 'Invalid query: ' . mysql_error() . "\n";
    $message .= 'Whole query: ' . $query;
    die( $message );
}

// Print out rows
$prefix = '';
echo "[\n";
while ( $row = mysql_fetch_assoc( $result ) ) {
    echo $prefix . " {\n";
    echo '   "event": "' . $row['DEV'] . ', ' . "\n";
    echo '   "AVG_W": ' . $row['MESH_TIMH'] . ', ' . "\n";
    echo " }";
    $prefix = ",\n";
}
echo "\n]";
// Close the connection
mysql_close($link);
?>

```

To html αρχείο ονομάζεται day\_av.html



Εικόνα 4.35 Ιστοσελίδα διαγράμματος μέσης στιγμιαίας κατανάλωσης από Arduino ανά ημέρα

### 4.4.5.2.3 Διάγραμμα μέγιστης στιγμιαίας κατανάλωσης ανά μήνα

Με το αρχείο data\_all\_month\_max.php παίρνουμε την μέγιστη τιμή της στιγμιαίας κατανάλωσης (max(W)) ανά μήνα από τις τιμές που παίρνουμε από το Arduino.

```
<?php
// Connect to MySQL
$link = mysql_connect( 'localhost', 'root', '' );
if ( !$link ) {
    die( 'Could not connect: ' . mysql_error() );
}

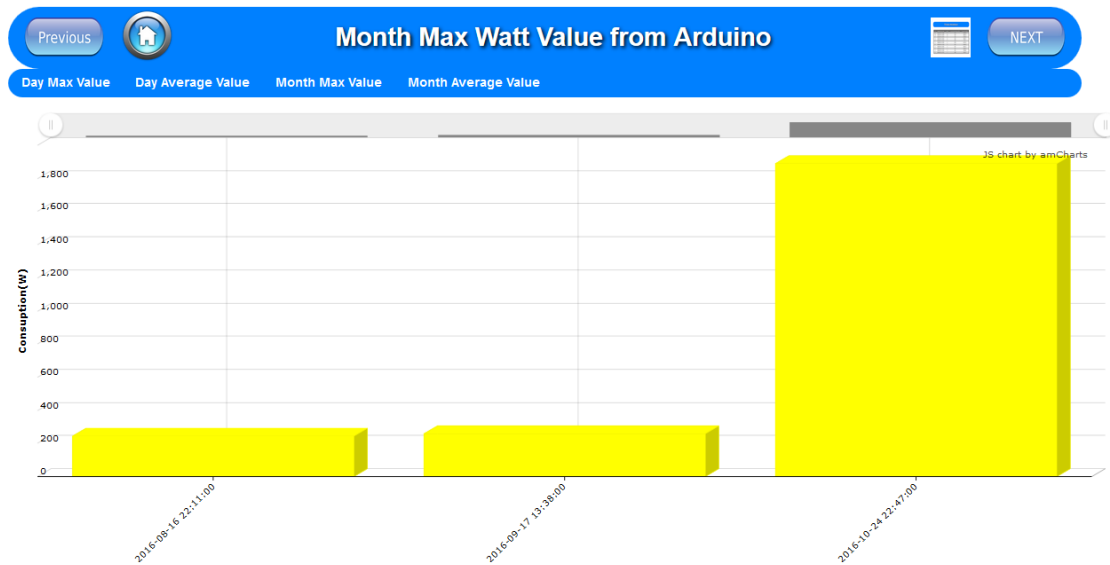
// Select the data base
$db = mysql_select_db( 'arduinobeaglebone', $link );
if ( !$db ) {
    die ( 'Error selecting database \'arduinobeaglebone\' : ' . mysql_error() );
}

// Fetch the data          pairnei apo ton pinaka metrhseis
$query = " SELECT event, W from metrhseis
inner JOIN (
    SELECT MONTH(event) as MEV, max(W) as MAX_W from metrhseis GROUP BY MEV
) as monthly on W > 0 and W = MAX_W and MONTH(event) = MEV
GROUP BY MEV
ORDER BY event;          ";
$result = mysql_query( $query );

// All good?
if ( !$result ) {
    // Nope
    $message = 'Invalid query: ' . mysql_error() . "\n";
    $message .= 'Whole query: ' . $query;
    die( $message );
}

// Print out rows
$prefix = '';
echo "\n";
while ( $row = mysql_fetch_assoc( $result ) ) {
    //if($row['counter'] == '') continue;
    echo $prefix . " {\n";
    echo '   "event": "' . $row['event'] . ',' . "\n";
    echo '   "max_w": ' . $row['W'] . ',' . "\n";
    echo " }";
    $prefix = ",\n";
}
echo "\n";
// Close the connection
mysql_close($link);
?>
```

Το html αρχείο ονομάζεται month\_max.html



Εικόνα 4.36: Ιστοσελίδα διαγράμματος μέγιστης στιγμιαίας κατανάλωσης από Arduino ανά μήνα

#### 4.4.5.2.4 Διάγραμμα μέσης στιγμιαίας κατανάλωσης ανά μήνα

Τώρα με το αρχείο `data_all_month_av.php` παίρνουμε την μέση τιμή της στιγμιαίας κατανάλωσης ( $avg(W)$ ) ανά μήνα από τις τιμές που παίρνουμε από το Arduino.

```
<?php
// Connect to MySQL
$link = mysql_connect( 'localhost', 'root', '' );
if ( !$link ) {
    die( 'Could not connect: ' . mysql_error() );
}

// Select the data base
$db = mysql_select_db( 'arduinobeaglebone', $link );
if ( !$db ) {
    die ( 'Error selecting database \'arduinobeaglebone\' : ' . mysql_error() );
}

// Fetch the data
$query = " SELECT CONCAT(YEAR(event), '-', MONTH(event)) as MEV, AVG(W) as MESH TIMH, event
from metrhseis
WHERE event
GROUP BY MEV
ORDER BY event; ";
$result = mysql_query( $query );

// All good?
if ( !$result ) {
    // Nope
    $message = 'Invalid query: ' . mysql_error() . "\n";
    $message .= 'Whole query: ' . $query;
    die( $message );
}

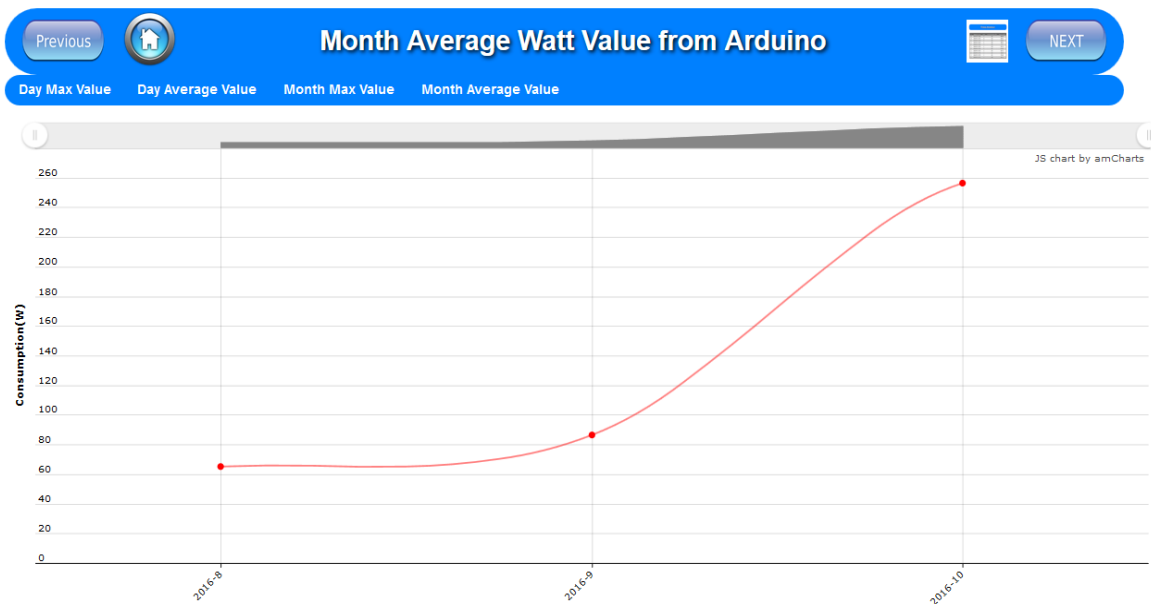
// Print out rows
$prefix = '';
echo "\n";
while ( $row = mysql_fetch_assoc( $result ) ) {
    //if($row['counter'] == '') continue;
    echo $prefix . " {\n";
    echo "  \"event\": " . $row['MEV'] . ", " . "\n";
```

```

echo ' "AVG_W": ' . $row['MESH_TIMH'] . ',' . "\n";
echo " }";
$prefix = ",\n";
}
echo "\n]";
// Close the connection
mysql_close($link);
?>

```

Το αρχείο html ονομάζεται month\_av.html

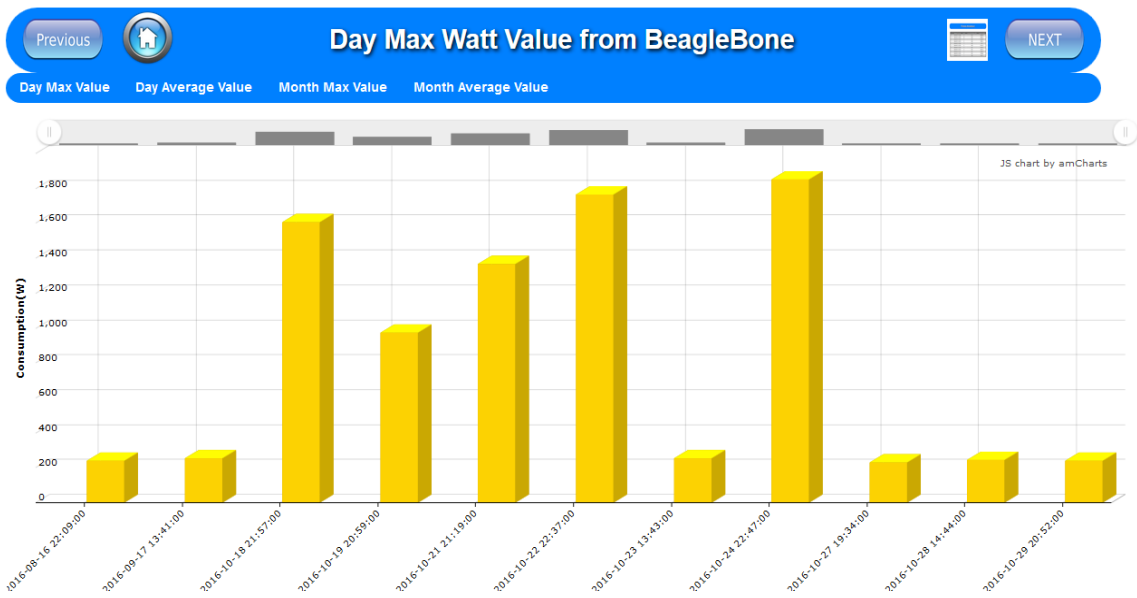


Εικόνα 4.37: Ιστοσελίδα διαγράμματος μέσης στιγμιαίας κατανάλωσης από Arduino ανά μήνα

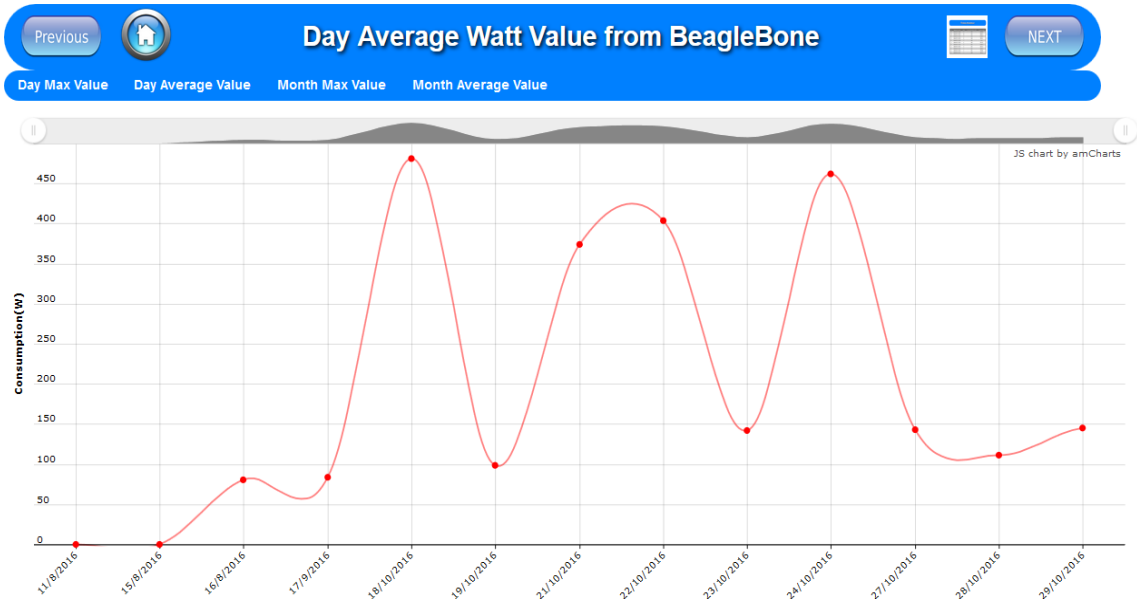
#### 4.4.5.3 Statistics Diagrams from BeagleBone

Υπενθυμίζουμε ότι τα αρχεία data\_all\_day\_max.php , data\_all\_day\_av.php , data\_all\_month\_max.php και data\_all\_month\_av.php έχουν δημιουργηθεί για να συνδέουν τις ιστοσελίδες διαγραμμάτων με την βάση δεδομένων για τις τιμές που παίρνουμε από το Arduino, δηλαδή για τα διαγράμματα της δεύτερης κατηγορίας (Statistics Diagrams from Arduino).

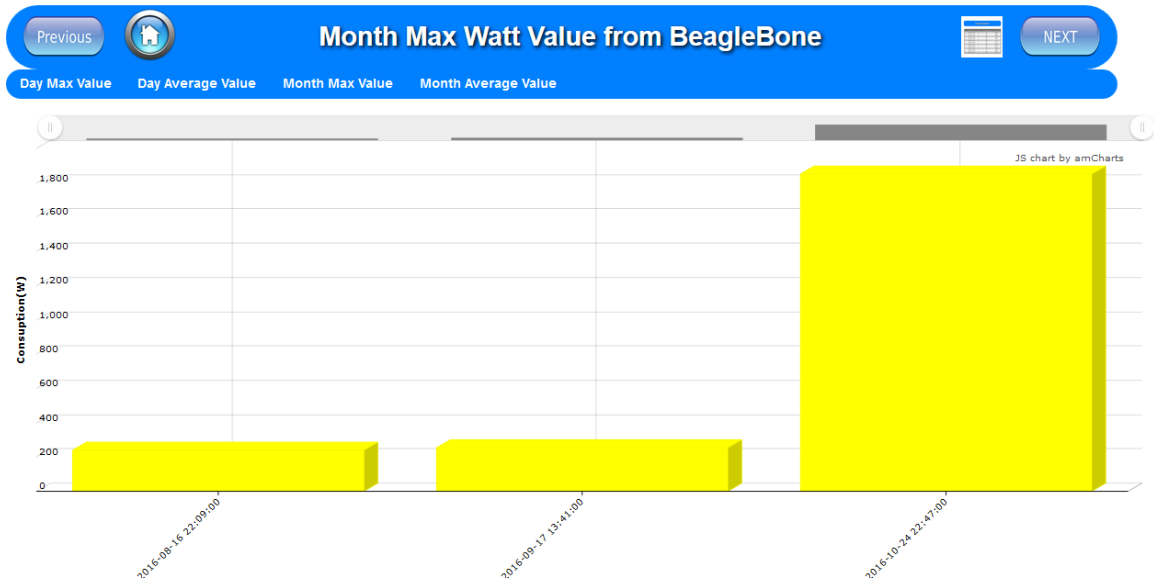
Με τον ίδιο τρόπο δημιουργούμε τα αρχεία data\_all\_day\_max\_BB.php , data\_all\_day\_av\_BB.php , data\_all\_month\_max\_BB.php και data\_all\_month\_av\_BB.php μόνο που τώρα επιλέγουμε την τιμή της στιγμιαίας κατανάλωσης που παίρνουμε από το BeagleBone (W\_B). Αντίστοιχα δημιουργούμε και τις ιστοσελίδες day\_max\_BB.html , day\_av\_BB.html , month\_max\_BB.html και month\_av\_BB.html , οι οποίες ιστοσελίδες απεικονίζουν τα διαγράμματα της τρίτης κατηγορίας (Statistics Diagrams from BeagleBone).



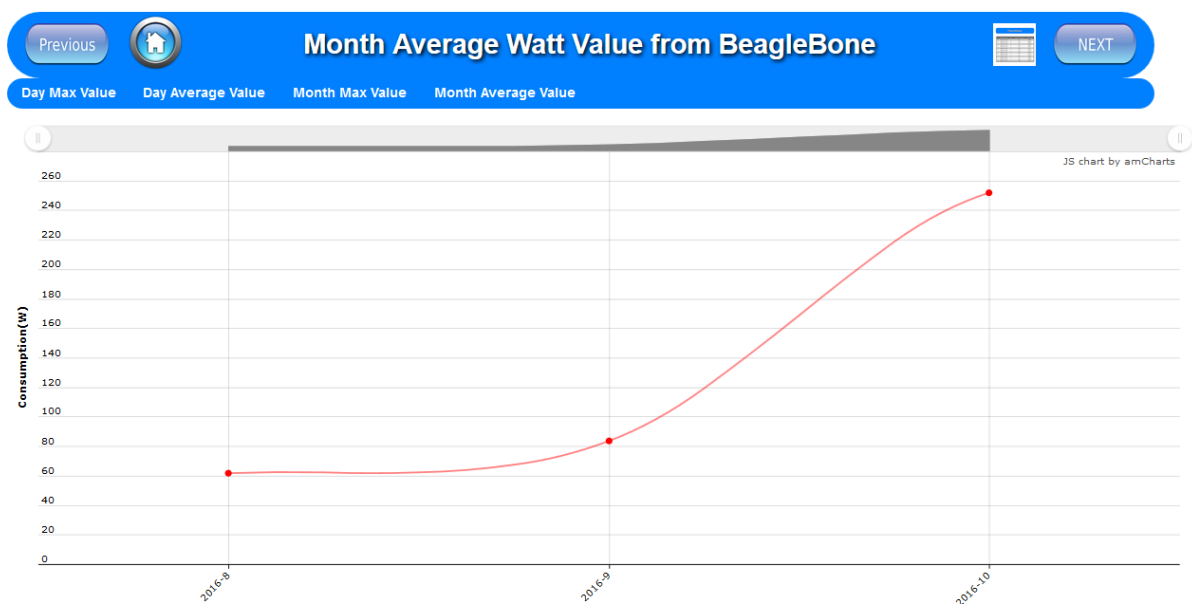
Εικόνα 4.38: Ιστοσελίδα διαγράμματος μέγιστης στιγμιαίας κατανάλωσης από BeagleBone ανα ημέρα



Εικόνα 4.39 Ιστοσελίδα διαγράμματος μέσης στιγμιαίας κατανάλωσης από BeagleBone ανά ημέρα



Εικόνα 4.40 Ιστοσελίδα διαγράμματος μέγιστης στιγμιαίας κατανάλωσης από BeagleBone ανά μήνα



Εικόνα 4.41: Ιστοσελίδα διαγράμματος μέσης στιγμιαίας κατανάλωσης από BeagleBone ανά μήνα

#### 4.4.5.4 Compare diagrams:

Για να εξετάσουμε το σύστημα μας (Arduino) ότι μετράει σωστά χρησιμοποιήσαμε την πρίζα της Meazon έτσι ώστε να συγκρίνουμε τις τιμές που παίρνουμε από το Arduino με τις αντίστοιχες τιμές από το BeagleBone. Στην υποενότητα αυτή θα κάνουμε ακριβώς αυτή την σύγκριση. Για να δημιουργηθούν αυτά τα διάγραμμα, χρησιμοποιήσαμε το αρχείο data\_all.php που περιγράψαμε σε προηγούμενη υποενότητα.

##### 4.4.5.4.1 Διάγραμμα σύγκρισης στιγμιαίας κατανάλωσης από Arduino και BeagleBone

Για την απεικόνιση αυτής της ιστοσελίδας διαγράμματος δημιουργήσαμε το αρχείο WandW\_B.html :

```
<!DOCTYPE>
<html>
  <style>
    h1 {
      width:1350px;
      margin-top:30px;
      margin-right:85px;
      margin-bottom:0px;
      margin-left:85px;
      font-weight: bold;
      font-style:normal;
      text-decoration: none;
      background-color:#0080ff;
      color:white;
      text-shadow: 2px 2px 4px #000000;
      border-radius: 200px;
      padding-top:20px ;
      padding-bottom:20px ;
      font-family: Arial;
      text-align:center;
    }
    body {
      color:#000000;
      background-color:white;
      text-align: center;
    }
    ul{
      background-color: #0080ff;

      list-style-type:none;
      margin: 0;
      margin-left:85px;
      padding:0;
      width:1350px;
      background color:#f1f1f1;
      height:100%
      position: fixed;
      overflow: hidden;
      border-radius: 200px;
    }
    li {
      float:left;
      text-align:center;
      font-family: Arial;
      border-bottom: 0px solid #555;
    }
    li a{
      display:block;
      color:white;
      padding:8px 16px;
      text-decoration:none;
    }
    li a:hover{
```

```

        background-color:#555;
        color: white;
    }
    .active{
        background-color:#078973;
        color:white;
    }
</style>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>W and W_B</title>
</head>
<body>
    <h1>
        <a href="kWhandkWh_B.html">
            
            </a>
            <a href=" ../home.html">
                
                </a>W and W_B
            <a href="kWhandkWh_B.html">
                
                </a>
                <a href=" ../review_data.php">
                    
                    </a>
            </h1>
            <ul> <strong class="active">

                <li><a href="WandW_B.html"> Compare W and W_B</a></li>
                <li><a href="kWhandkWh_B.html"> Compare kWh and kWh_B</a></li>

            </strong>
            </ul>
            <link rel="stylesheet" href="style.css" type="text/css">
            <script src=" ../amcharts/amcharts.js" type="text/javascript"></script>
            <script src=" ../amcharts/serial.js" type="text/javascript"></script>
            <script>
                AmCharts.loadJSON = function(url) {
                    // create the request
                    if (window.XMLHttpRequest) {
                        // IE7+, Firefox, Chrome, Opera, Safari
                        var request = new XMLHttpRequest();
                    }
                    else {
                        // code for IE6, IE5
                        var request = new ActiveXObject('Microsoft.XMLHTTP');
                    }
                    // load it
                    // the last "false" parameter ensures that our code will wait before the
                    // data is loaded
                    request.open('GET', url, false);
                    request.send();
                    // parse and return the output
                    return eval(request.responseText);
                };
            </script>
            <div id="chartdiv" style="width: 1400px; height: 600px; background-color:white;margin:
0px 200px 85px 85px;border-radius: 30px; "></div>
            <script>
                var chart;
                AmCharts.ready(function () {
                    // SERIAL CHART
                    var chartData = AmCharts.loadJSON('data_all.php');
                    // SERIAL CHART
                    chart = new AmCharts.AmSerialChart();
                    chart.dataProvider = chartData;
                    chart.path = " ../amcharts/";
                    chart.categoryField = "event";
                    chart.dataDateFormat = "JJ:NN:SS";
                    chart.marginTop = 0;
                    // AXES
                    // category axis

```



```

chart.categoryAxis.dateFormats = true;
// as we have data of different units, we create two different value axes
// Duration value axis
var durationAxis = new AmCharts.ValueAxis();

durationAxis.title = "W";
durationAxis.gridAlpha = 0;
durationAxis.axisAlpha = 0;
durationAxis.inside = false;
// the following line makes this value axis to convert values to duration
// it tells the axis what duration unit it should use. mm - minute, hh -
hour...

chart.addValueAxis(durationAxis);
// Distance value axis
var distanceAxis = new AmCharts.ValueAxis();

distanceAxis.title = "W";
distanceAxis.gridAlpha = 0;
distanceAxis.axisAlpha = 0;
distanceAxis.position = "right";
distanceAxis.inside = false;
distanceAxis.unit = "watt";
chart.addValueAxis(distanceAxis);
// GRAPHS
// duration graph
var durationGraph = new AmCharts.AmGraph();

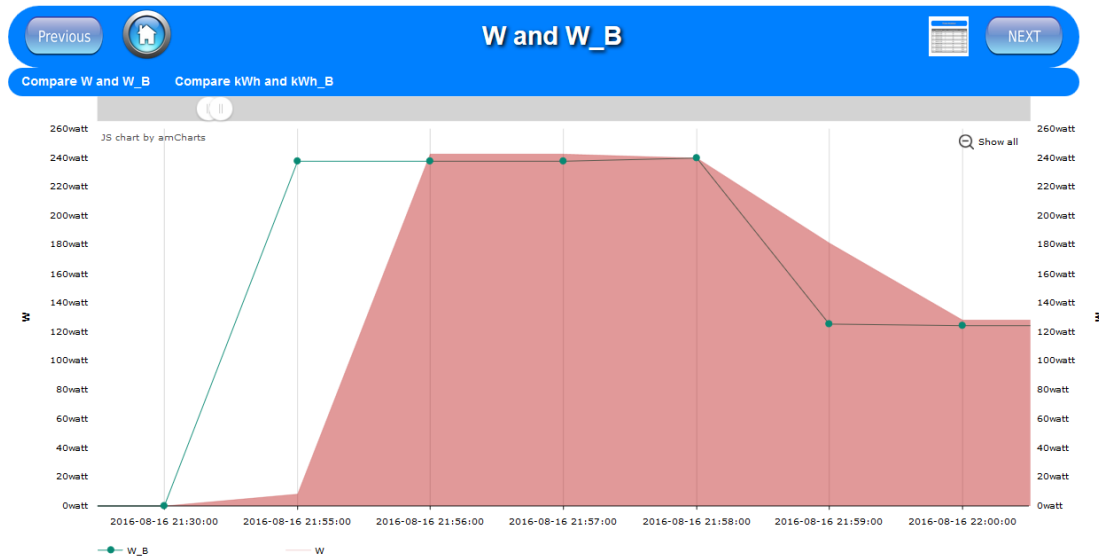
durationGraph.title = "W_B";
durationGraph.valueField = "W_B";
durationGraph.type = "line";
durationGraph.valueAxis = durationAxis; // indicate which axis should be used
durationAxis.labelRotation = 90; //deixnei ta dedomena tw n a3onwn katheta
durationGraph.lineColor = "#078973";
durationGraph.balloonText = "[[value]]";
durationGraph.lineThickness = 1;
durationGraph.legendValueText = "[[value]]";
durationGraph.bullet = "round";
durationGraph.bulletBorderColor = "#078973";
durationGraph.bulletBorderAlpha = 1;
durationGraph.bulletBorderThickness = 1;
chart.addGraph(durationGraph);
// distance graph
var distanceGraph = new AmCharts.AmGraph();
distanceGraph.valueField = "W";
distanceGraph.title = "W";
//distanceGraph.type = "column";
distanceGraph.type = "line";
distanceGraph.fillAlphas = 0.4;
distanceGraph.valueAxis = distanceAxis;
durationAxis.labelRotation = 90; //deixnei ta dedomena tw n a3onwn katheta
durationAxis.unit = "watt";
distanceGraph.lineColor = "#B30000";
distanceGraph.lineAlpha = 0.2;
chart.addGraph(distanceGraph);
// CURSOR
var chartCursor = new AmCharts.ChartCursor();
chartCursor.zoomable = true;
chartCursor.categoryBalloonDateFormat = "JJ";
chartCursor.cursorAlpha = 0;
chartCursor.valueLineEnabled = true;
chartCursor.valueLineBalloonEnabled = true;
chartCursor.valueLineAxis = distanceAxis;
chart.addChartCursor(chartCursor);
var chartScrollbar = new AmCharts.ChartScrollbar();
chartScrollbar.scrollbarHeight = 30;
chartScrollbar.offset = 10;
chart.addChartScrollbar(chartScrollbar);
// LEGEND
var legend = new AmCharts.AmLegend();
legend.bulletType = "round";
legend.equalWidths = false;
legend.valueWidth = 120;
legend.color = "#000000";
legend.useGraphSettings = true;
chart.addLegend(legend);
// WRITE
chart.write("chartdiv");

```

```

    });
  </script>
</body>
</html>

```



Εικόνα 4.42: Ιστοσελίδα διαγράμματος σύγκρισης στιγμιαίας κατανάλωσης από Arduino και BeagleBone

#### 4.4.5.4.2 Διάγραμμα σύγκρισης συνολικής κατανάλωσης από Arduino και BeagleBone

Για την απεικόνιση αυτής της ιστοσελίδας διαγράμματος δημιουργήσαμε το αρχείο kWhandkWh\_B.html :

```

<!DOCTYPE>
<html>
  <style>
    h1 {
      width:1350px;
      margin-top:30px;
      margin-right:85px;
      margin-bottom:0px;
      margin-left:85px;
      font-weight: bold;
      font-style:normal;
      text-decoration: none;
      background-color:#0080ff;
      color:white;
      text-shadow: 2px 2px 4px #000000;
      border-radius: 200px;
      padding-top:20px ;
      padding-bottom:20px ;
      font-family: Arial;
      text-align:center;
    }
    body {
      color:#000000;
      background-color:white;
      text-align: center;
    }
  </style>

```

```

    ul{

background-color: #0080ff;

list-style-type:none;
margin: 0;
margin-left:85px;
padding:0;
width:1350px;
background color:#f1f1f1;
height:100%
position: fixed;
overflow: hidden;
border-radius: 200px;

    }

li {
float:left;
text-align:center;
font-family: Arial;
border-bottom: 0px solid #555;
}

li a{
display:block;
color:white;
padding:8px 16px;
text-decoration:none;
}

li a:hover{
background-color:#555;
color: white;
}

.active{
background-color:#078973;
color:white;
}
</style>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>kWh and kWh_B </title>
</head>
<body>
<h1>
<a href="WandW_B.html">

</a>
<a href=" ../home.html">

</a>Compare kWh and kWh_B
<a href="WandW_B.html">

</a>
<a href=" ../review_data.php">

</a>
</h1>
<ul> <strong class="active">

<li><a href="WandW_B.html"> Compare W and W_B</a></li>
<li><a href="kWhandkWh_B.html"> Compare kWh and kWh_B</a></li>

</strong>
</ul>
<link rel="stylesheet" href="style.css" type="text/css">
<script src=" ../amcharts/amcharts.js" type="text/javascript"></script>
<script src=" ../amcharts/serial.js" type="text/javascript"></script>
<script>
AmCharts.loadJSON = function(url) {
// create the request
if (window.XMLHttpRequest) {
// IE7+, Firefox, Chrome, Opera, Safari
var request = new XMLHttpRequest();

```

```

    }
    else {
        // code for IE6, IE5
        var request = new ActiveXObject('Microsoft.XMLHTTP');
    }
    // load it
    // the last "false" parameter ensures that our code will wait before the
    // data is loaded
    request.open('GET', url, false);
    request.send();
    // parse and return the output
    return eval(request.responseText);
};
</script>
<div id="chartdiv" style="width: 1400px; height: 600px; background-color:white;margin:
0px 200px 85px 85px;border-radius: 30px; "></div>
<script>
    var chart;
    AmCharts.ready(function () {
        // SERIAL CHART
        var chartData = AmCharts.loadJSON('data_all.php');
        // SERIAL CHART
        chart = new AmCharts.AmSerialChart();
        chart.dataProvider = chartData;
        chart.path = "../amcharts/";
        chart.categoryField = "event";
        chart.dataDateFormat = "JJ:NN:SS";
        chart.marginTop = 0;
        // AXES
        // category axis
        chart.categoryAxis.dateFormats = true;
        // as we have data of different units, we create two different value axes
        // Duration value axis
        var durationAxis = new AmCharts.ValueAxis();
        durationAxis.title = "kWh_B";
        durationAxis.gridAlpha = 0;
        durationAxis.axisAlpha = 0;
        durationAxis.inside = false;
        // the following line makes this value axis to convert values to duration
        // it tells the axis what duration unit it should use. mm - minute, hh -
hour...

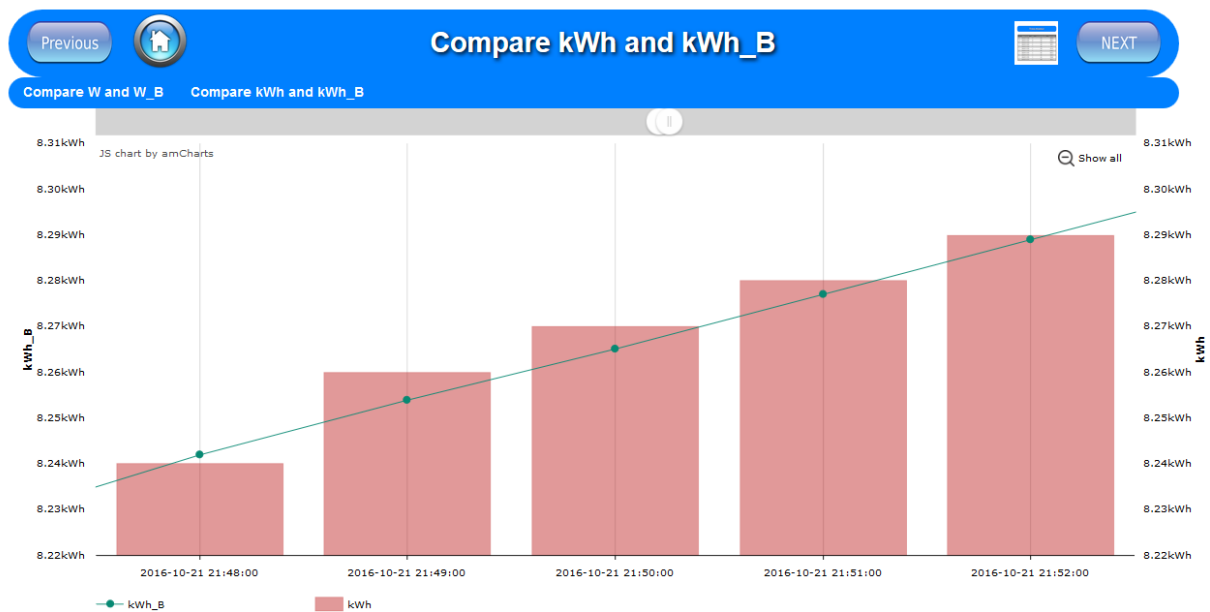
        chart.addValueAxis(durationAxis);
        // Distance value axis
        var distanceAxis = new AmCharts.ValueAxis();
        distanceAxis.title = "kWh";
        distanceAxis.gridAlpha = 0;
        distanceAxis.axisAlpha = 0;
        distanceAxis.position = "right";
        distanceAxis.inside = false;
        distanceAxis.unit = "kWh";
        chart.addValueAxis(distanceAxis);
        // GRAPHS
        // duration graph
        var durationGraph = new AmCharts.AmGraph();
        durationGraph.title = "kWh_B";
        durationGraph.valueField = "kWh_B";
        durationGraph.type = "line";
        durationGraph.valueAxis = durationAxis; // indicate which axis should be used
        durationGraph.lineColor = "#078973";
        durationGraph.balloonText = "[[value]]";
        durationGraph.lineThickness = 1;
        durationGraph.legendValueText = "[[value]]";
        durationGraph.bullet = "round";
        durationGraph.bulletBorderColor = "#078973";
        durationGraph.bulletBorderAlpha = 1;
        durationGraph.bulletBorderThickness = 1;
        chart.addGraph(durationGraph);
        // distance graph
        var distanceGraph = new AmCharts.AmGraph();
        distanceGraph.valueField = "kWh";
        distanceGraph.title = "kWh";
        distanceGraph.type = "column";
        distanceGraph.fillAlphas = 0.4;
        distanceGraph.valueAxis = distanceAxis;
        durationAxis.unit = "kWh";
        distanceGraph.lineColor = "#B30000";
        distanceGraph.lineAlpha = 0.2;
    });

```

```

chart.addGraph(distanceGraph);
// CURSOR
var chartCursor = new AmCharts.ChartCursor();
chartCursor.zoomable = true;
chartCursor.categoryBalloonDateFormat = "JJ";
chartCursor.cursorAlpha = 0;
chartCursor.valueLineEnabled = true;
chartCursor.valueLineBalloonEnabled = true;
chartCursor.valueLineAxis = distanceAxis;
chart.addChartCursor(chartCursor);
var chartScrollbar = new AmCharts.ChartScrollbar();
chartScrollbar.scrollbarHeight = 30;
chartScrollbar.offset = 10;
chart.addChartScrollbar(chartScrollbar);
// LEGEND
var legend = new AmCharts.AmLegend();
legend.bulletType = "round";
legend.equalWidths = false;
legend.valueWidth = 120;
legend.color = "#000000";
legend.useGraphSettings = true;
chart.addLegend(legend);
// WRITE
chart.write("chartdiv");
});
</script>
</body>
</html>

```



Εικόνα 4.43: Ιστοσελίδα διαγράμματος σύγκρισης συνολικής κατανάλωσης από Arduino και BeagleBone

#### 4.4.5.4.3 Συμπεράσματα συγκριτικών διαγραμμάτων

Παρατηρώντας τα παραπάνω διαγράμματα βλέπουμε ότι οι τιμές από το arduino έχουν μια μικρή καθυστέρηση σε σχέση με τις τιμές από το BeagleBone. Αυτό συμβαίνει επειδή η μέτρηση του Arduino βασίζεται στον δίσκο του αναλογικού ρολογιού που περιστρέφεται. Οπότε όσο πιο μικρή είναι η ταχύτητα περιστροφής του δίσκου, τόσο πιο αργά θα κάνει ο

δίσκος για να ολοκληρώσει μια περιστροφή με συνέπεια να καθυστερήσει και το arduino να “αντιληφθεί” την κατανάλωση.

Αυτό το πρόβλημα όμως ελαττώνεται καθώς αυξάνεται το φορτίο στον αναλογικό μετρητή γιατί, καθώς αυξάνεται το φορτίο αυξάνεται και η ταχύτητα περιστροφής του δίσκου και κατά συνέπεια να αυξάνεται και η ακρίβεια του Arduino.

## Κεφάλαιο 5: Συμπεράσματα και Μελλοντικές Επεκτάσεις

Το αντικείμενο της παρούσας πτυχιακής εργασίας ήταν η κατασκευή ενός συστήματος εποπτείας με το οποίο από το αναλογικό ρολόι της Δ.Ε.Η. μπορούμε να υπολογίσουμε την συνολική και την στιγμιαία κατανάλωση της εγκατάστασης που είναι τοποθετημένος ο μετρητής της Δ.Ε.Η..

Ο σκοπός αυτού του συστήματος ήταν η απομακρυσμένη καταγραφή της κατανάλωσης μιας εγκατάστασης χωρίς να απαιτείται η ανθρώπινη παρουσία (όσο αυτό είναι εφικτό)(προσωπικό της Δ.Ε.Η. να κάνει την μέτρηση) και επίσης η απομακρυσμένη παρακολούθηση της κατανάλωσης από τον χρήστη της εγκατάστασης, ώστε να μειώσει τις σπατάλες ενέργειας . Με αυτό τον τρόπο κάνει καλό στο περιβάλλον επειδή δεν το επιβαρύνει χωρίς λόγο αλλά και στον ίδιο γιατί περιορίζει τα έξοδα του, χωρίς όμως να περιορίζει τις ανέσεις του.

Για την υλοποίηση αυτού του συστήματος χρησιμοποιήθηκε η αναπτυξιακή πλατφόρμα Arduino, όπου στα κεφάλαια της εργασίας εξηγήσαμε τι είναι το Arduino, τα χαρακτηριστικά του και τον τρόπο με τον οποίο το προγραμματίσουμε. Αναλύσαμε όλα τα υλικά που χρησιμοποιήσαμε (Μονοφασικό μετρητή, Ethernet Shield, TCRT5000, Bizy Type F Plug, Meazon Gateway Advanced)

Για να δούμε ότι όντως μετράμε σωστά προσθέσαμε στο κύκλωμα μας τον μετρητή Bizy Type F Plug της εταιρίας MEAZON και κάναμε σύγκριση των τιμών τους.

Όλες οι μετρήσεις που έγιναν καταγράφηκαν σε βάση δεδομένων που δημιουργήσαμε και δημιουργήθηκε επίσης και ένας δυναμικός ιστοχώρος στον οποίο απεικονίζονται όλες οι μετρήσεις σε πίνακα, κάποια απλά διαγράμματα, κάποια συγκριτικά διαγράμματα και κάποια στατιστικά διαγράμματα.

Επίσης ας μην ξεχνάμε το πιο σημαντικό χαρακτηριστικό του συστήματος που υλοποιήσαμε, το οποίο είναι το μικρό κόστος κατασκευής. Στην ουσία με ένα Arduino, ένα Ethernet Shield και έναν αισθητήρα ανάκλασης δημιουργήσαμε τον δικό μας μετρητή ηλεκτρικής ενέργειας.

Τέλος θα ήθελα να αναφέρω κάποιες μελλοντικές βελτιώσεις και επεκτάσεις που θα μπορούσαν να εφαρμοστούν στο σύστημα μας:

- Αρχικά θα μπορούσαμε να χρησιμοποιήσουμε Wifi Shield αντί για Ethernet Shield έτσι ώστε να αποφύγουμε όσο γίνεται την ταλαιπωρία των καλωδίων.
- Ένας άλλος τρόπος για την αποφυγή των καλωδίων θα ήταν αν τροφοδοτούσαμε το σύστημα μας μέσω μιας μπαταρίας.
- Επίσης θα ήταν δυνατό να δημιουργήσουμε εφαρμογή για το κινητό μας τηλέφωνο και να βλέπουμε και από εκεί την συμπεριφορά της κατανάλωσης που έχουμε.
- Επιπλέον για να είμαστε ακόμη πιο σίγουροι ότι το σύστημα μας μετράει σωστά εκτός από τον δίσκο του μετρητή να παρατηρούμε και τους αριθμούς μέσω μιας κάμερας.

## Βιβλιογραφία

### Βιβλία:

Π. Παπάζογλου και Σπ. Π. Λιωνής, Ανάπτυξη εφαρμογών με το Arduino, Εκδόσεις Τζιόλα

### Διαδίκτυο:

Arduino (Ανάκτηση Σεπτέμβριος 2016):

<https://el.wikipedia.org/wiki/Arduino>

<https://www.arduino.cc/>

<https://t->

[h.wikispaces.com/file/view/%CE%95%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE%B3%CE%AE%CF%83%CF%84%CE%BFArduino.pdf](https://t-wikispaces.com/file/view/%CE%95%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE%B3%CE%AE%CF%83%CF%84%CE%BFArduino.pdf)

<https://deltahacker.gr/arduino-intro/>

Ethernet Shield (Ανάκτηση Σεπτέμβριος 2016):

<http://www.instructables.com/id/Arduino-Ethernet-Shield-Tutorial/>

<https://www.arduino.cc/en/Main/ArduinoEthernetShield>

Αισθητήρα (Ανάκτηση Σεπτέμβριος 2016):

<http://www.vishay.com/docs/83760/tcrt5000.pdf>

<http://www.crcibernetica.com/reflective-infrared-optical-sensor-tcrt5000/>

Αναλογικός μετρητής ενέργειας (Ανάκτηση Σεπτέμβριος 2016):

[http://www.ee.teihal.gr/lessons/metrology/private/uploads/ERG\\_ASK11\\_METRHHS.pdf](http://www.ee.teihal.gr/lessons/metrology/private/uploads/ERG_ASK11_METRHHS.pdf)

Χρώματα δεκαεξαδικό κωδικό (Ανάκτηση Σεπτέμβριος 2016):

<https://www.w3.org/TR/css3-color/#svg-color>

<http://il6.picdn.net/shutterstock/videos/4902233/thumb/1.jpg>

HTML (Ανάκτηση Σεπτέμβριος 2016):

[https://el.wikipedia.org/wiki/HTML#.CE.A0.CE.B1.CF.81.CE.B1.CE.B4.CE.B5.CE.AF.CE.B3.CE.BC.CE.B1.CF.84.CE.B1\\_.CF.83.CF.84.CE.BF.CE.B9.CF.87.CE.B5.CE.AF.CF.89.CE.BD](https://el.wikipedia.org/wiki/HTML#.CE.A0.CE.B1.CF.81.CE.B1.CE.B4.CE.B5.CE.AF.CE.B3.CE.BC.CE.B1.CF.84.CE.B1_.CF.83.CF.84.CE.BF.CE.B9.CF.87.CE.B5.CE.AF.CF.89.CE.BD)

<https://www.codecademy.com/learn>

CSS (Ανάκτηση Σεπτέμβριος 2016):

<https://el.wikipedia.org/wiki/CSS>

<http://www.wlearn.gr/index.php/home-css-83>



JS (Ανάκτηση Σεπτέμβριος 2016):

<https://el.wikipedia.org/wiki/JavaScript>

[https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B5%CF%81%CE%BC%CE%B7%CE%BD%CE%B5%CF%85%CE%BC%CE%AD%CE%BD%CE%B7\\_%CE%B3%CE%BB%CF%8E%CF%83%CF%83%CE%B1](https://el.wikipedia.org/wiki/%CE%94%CE%B9%CE%B5%CF%81%CE%BC%CE%B7%CE%BD%CE%B5%CF%85%CE%BC%CE%AD%CE%BD%CE%B7_%CE%B3%CE%BB%CF%8E%CF%83%CF%83%CE%B1)

<http://www.tweaking4all.com/hardware/arduino/arduino-ethernet-data-push/>

PHP (Ανάκτηση Σεπτέμβριος 2016):

[http://www.ip.gr/General/%CE%A4%CE%B9\\_%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9\\_%CE%B7\\_PHP-63.html](http://www.ip.gr/General/%CE%A4%CE%B9_%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9_%CE%B7_PHP-63.html)

MySQL (Ανάκτηση Σεπτέμβριος 2016):

[http://webseminars.ee.auth.gr/presentations/11\\_MySQL\\_beginner.pdf](http://webseminars.ee.auth.gr/presentations/11_MySQL_beginner.pdf)

<https://el.wikipedia.org/wiki/MySQL>

[https://arch.icte.uowm.gr/mdasyg/pdfs/avgeris\\_thesis.pdf](https://arch.icte.uowm.gr/mdasyg/pdfs/avgeris_thesis.pdf)

xampp (Ανάκτηση Σεπτέμβριος 2016):

<https://el.wikipedia.org/wiki/XAMPP>

WinSCP (Ανάκτηση Οκτώμβριος 2016):

<https://en.wikipedia.org/wiki/WinSCP>

PuTTY (Ανάκτηση Οκτώμβριος 2016):

[http://wiki.dmst.aueb.gr/doku.php?id=unix\\_software](http://wiki.dmst.aueb.gr/doku.php?id=unix_software)

Ssh(Ανάκτηση Οκτώμβριος 2016) :

[https://en.wikipedia.org/wiki/Secure\\_Shell](https://en.wikipedia.org/wiki/Secure_Shell)

meazon (Ανάκτηση Οκτώμβριος 2016):

[http://meazon.com/wp-content/uploads/2015/07/Meazon\\_Bizy\\_Gateway\\_Advanced.pdf](http://meazon.com/wp-content/uploads/2015/07/Meazon_Bizy_Gateway_Advanced.pdf)

[http://meazon.com/wp-content/uploads/2015/07/Meazon\\_Bizy\\_TypeF\\_Plug.pdf](http://meazon.com/wp-content/uploads/2015/07/Meazon_Bizy_TypeF_Plug.pdf)

Εξοικονόμηση ενέργειας (Ανάκτηση Οκτώμβριος 2016):

<https://www.eac.com.cy/EL/EAC/NewsAndAnnouncements/Informative%20Leaflets/FINAL%20VERSION%203%20EXOIKONOMHSH%20AHK.pdf>

## Παράρτημα

### Κώδικες

Όλοι οι κώδικες βρίσκονται στον επισυναπτόμενο δίσκο (CD). Ο κώδικας που αφορά το Arduino μπορεί να διαβαστεί με το Arduino IDE. Όλοι οι υπόλοιποι κώδικες αλλά και ο κώδικας του Arduino, μπορούν να διαβαστούν με το Notepad++.