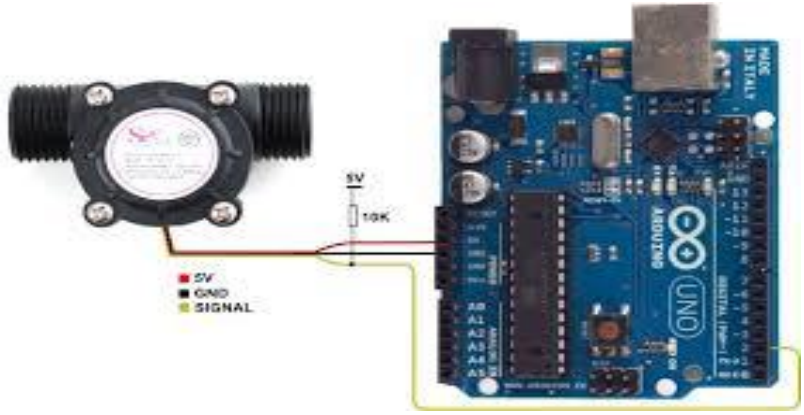
ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ
ΤΕΙ ΗΠΕΙΡΟΥ

ΤΕΙ ΗΠΕΙΡΟΥ
ΣΧΟΛΗ: Τεχνολογικών Εφαρμογών
ΤΜΗΜΑ:Μηχανικών Πληροφορικής Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ:

«ΔΥΝΑΜΙΚΟΣ ΜΕΤΡΗΤΗΣ ΚΑΤΑΝΑΛΩΣΗΣ ΝΕΡΟΥ ΜΕ ΔΥΝΑΤΟΤΗΤΑ ΑΠΟΣΤΟΛΗΣ ΔΕΔΟΜΕΝΩΝ ΣΕ ΒΑΣΗ»



ΣΠΟΥΔΑΣΤΕΣ

ΜΑΝΤΗΣ ΚΩΝ/ΝΟΣ

ΜΠΕΛΝΤΕΜΙΡΗΣ ΣΕΡΑΦΕΙΜ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ

ΧΑΡΙΛΟΓΗΣ ΒΑΣΙΛΕΙΟΣ

Δηλώνουμε υπεύθυνα ότι το παρόν κείμενο αποτελεί προϊόν προσωπικής μελέτης και εργασίας και πως όλες οι πηγές που χρησιμοποιήθηκαν για τη συγγραφή της δηλώνονται σαφώς είτε στις παραπομπές είτε στη βιβλιογραφία. Γνωρίζουμε πως η λογοκλοπή αποτελεί σοβαρότατο παράπτωμα και είμαστε ενήμεροι για την επέλευση των νομίμων συνεπειών»

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 3: Εγχειρίδιο τοποθέτησης μετρητή

ΚΕΦΑΛΑΙΟ 4: Συμπεράσματα

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Π Ε Ρ Ι Λ Η Ψ Η

Η παρούσα πτυχιακή εργασία επικεντρώνετε στην κατασκευή ενός δυναμικού μετρητή κατανάλωσης νερού με την δυνατότητα αποστολής των δεδομένων σε βάση, έτσι δίνεται επιλογή ελέγχου της κατανάλωσης από απόσταση και ανά πάσα στιγμή.

Η κατασκευή δημιουργήθηκε με βάση την πλακέτα ArduinoUno, η οποία είναι ιδανική για την κατασκευή τέτοιου είδους συστημάτων, είναι ευκολόχρηστη, μπορείς εύκολα να αναπτύξεις τον προγραμματισμό της και με μεγάλες δυνατότητες εξέλιξης με σχετικά χαμηλό κόστος. Προγραμματίστηκε σε γλώσσα C++ και το περιβάλλον χρήστη κυρίως σε PHP με την χρήση του Codeigniter.

Τα κεφάλαια παρακάτω αναπτύσσονται με τέτοιο τρόπο ώστε να γίνει πλήρως κατανοητή η λειτουργία, ο προγραμματισμός, ο σκοπός και η χρήση της κατασκευής. Στο πρώτο κεφάλαιο αναλύονται οι έννοιες των εργαλείων που χρησιμοποιήθηκαν, στο δεύτερο ο τρόπος που χρησιμοποιήθηκαν και ο ακριβής κώδικας προγραμματισμού και στα τελευταία ο τρόπος εγκατάστασης καθώς και η επίδραση του αποτελέσματος.

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

Παρακάτω αναλύονται και αναπτύσσονται τα κύρια μέρη για την υλοποίηση και τον προγραμματισμό του Δυναμικού Μετρητή Κατανάλωσης Νερού, με σκοπό την κατανόηση των εννοιών ώστε γίνει ευκολότερα αντιληπτή και κατανοητή η λειτουργία της κατασκευής .

1.1 Ορισμός του Arduino

Το **Arduino** είναι ένας μικροελεκτής, μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή ,εισόδουςκαι εξόδους από τις οποίες λαμβάνονται και στέλνονται δεδομένα και η οποία μπορεί να προγραμματιστεί με τη γλώσσα προγραμματισμού C++και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++ . Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων, ενώ είναι εύκολο και ευέλικτο στην χρήση του.
Μπορεί κάποιος να συνδέσει πάνω του οτιδήποτε, αισθητήρες, οθόνες, μετρητές, φώτα, κινητήρες και να υλοποιήσει ότι σκέφτηκε για να καλύψει τις ανάγκες του και να το προγραμματίσει τον τρόπο λειτουργίας του.Κατασκευάζεται στην Ιταλία από την εταιρεία SmartProjects και κάποιες πλακέτες από την εταιρεία SparkFunElectronics.[1]

Εικόνα 1.Πλακέτα ArduinoUno

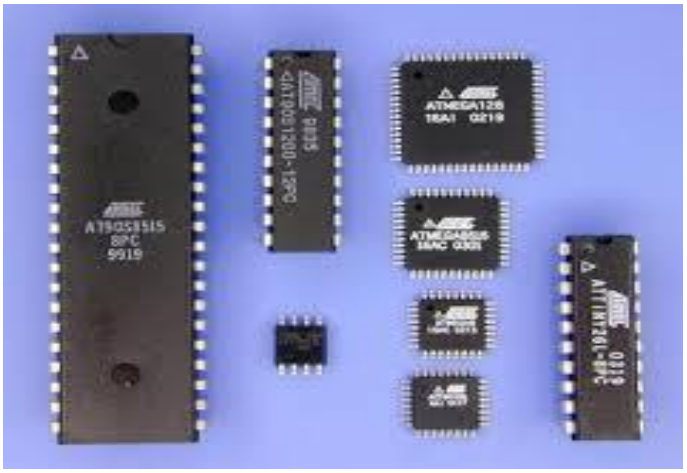
1.2 Η ιστορία του Arduino

Η αιτία για την δημιουργία του Arduino υπήρξε μια πρόκληση για το πως θα διδαχθεί σε μαθητές να κατασκευάζουν ηλεκτρονικά συστήματα γρήγορα.Το 2002 έναςπρογραμματιστής ο MassimoBanzi επιλέχθηκε ως αναπληρωτής καθηγητής για να αναλάβει αυτήν την πρόκληση.
Το 2005 δημιουργήθηκε ένα σχέδιο προκειμένου να φτιαχτεί μια συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, της οποίας η ιδιαιτερότητα θα ήταν το χαμηλό κόστος κατασκευής από τα άλλα συστήματα που υπήρχαν. Οι ιδρυτές του MassimoBanzi και DavidCueartielles ονόμασαν το σχέδιο από το Arduin της Ivrea, μια κωμόπολη της Ιταλίας και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο.Κάπως έτσι μπορούμε να πούμε πως από την ημέρα που εμφανίσθηκε το Arduino η τεχνολογία έχει γίνει πιο προσιτή.[2]

Εικόνα 2.ArduinoLogo

1.3 Τα μέρη του Arduino

Μια πλακέτα Arduino αποτελείται από έναν μικροελεγκτή AtmelAVR και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωσή του σε άλλα κυκλώματα.



Εικόνα 3.AtmelAVRMicrocontrollers

Όλες οι πλακέτες περιλαμβάνουν έναν γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλατωντή 16MHz. Ο προγραμματισμός του επιτυγχάνεται μέσω USB καλωδίου.. Η πλακέτα του Arduinoδιαθέτει τα περισσότερα microcontrollerI/Opins για χρήση από όλα τα άλλα κυκλώματα .[3]

Εικόνα 4.ArduinoParts

1.4 Οι εκδόσεις του Arduino

Αρχικά πρέπει να αναφέρουμε, πως πολύς κόσμος όταν αναφέρεται στο Arduino το μυαλό του πάει απευθείας στο ArduinoUNO, το οποίο είναι ευρέως διαδομένο καθώς καλύπτει πολλές ανάγκες σε σχέση με την ανάπτυξη συστημάτων και είναι από τις οικονομικότερες πλακέτες που έχουν υλοποιηθεί. Πρέπει να το ξεχωρίσουμε στο μυαλό μας καθώς το UNO είναι μια από τις πολλές πλακέτες που έχουν κατασκευαστεί.. Συνολικά έχουν κατασκευαστεί 16 πλακέτες Arduino οι οποίες αναφέρονται παρακάτω: [4]

1. Το Serial Arduino
2. Το Arduino Extreme
3. Το Arduino Mini
4. Το Arduino Nano
5. Το LilyPad Arduino
6. Το Arduino NG
7. Το Arduino NG plus
8. Το Arduino Bluetooth
9. Το Arduino Diecimila
10. Το Arduino Duemilanove
11. Το Arduino Mega
12. Το Arduino Uno
13. Το Arduino Mega 2560
14. Το Arduino Leonardo
15. Το Arduino Esplora
16. Το Arduino Due

1.5 Γιατί επιλέξαμε Arduino

Το βασικό κριτήριο της επιλογής μας για το Arduino είναι το ότι γνωρίζαμε εξ΄αρχής πως θα έχουμε εύκολα πρόσβαση σε πολλές πληροφορίες , επειδή είναι ανοικτού κώδικα και υποστηρίζεται από μια μεγάλη κοινότητα χρηστών , καθώς και η προσιτή τιμή του. Κατά την διάρκεια της υλοποίησης της εργασίας μας παρατηρήσαμε πόσο ευκολόχρηστο ,επεκτάσιμο και εξελίξιμο είναι. Μπορεί δηλαδή πολύ εύκολα κάποιος με ένα απλό καλώδιο USB να το προγραμματίσει συνδέοντάς το στον υπολογιστή του και αργότερα μπορεί εύκολα να επεκτείνει

τηναρχική του ιδέα και να την εξελίξει, να την διαμορφώσει προσθέτοντας και άλλα κομμάτια στα ήδη υπάρχοντα.

1.6 Γλώσσα προγραμματισμού

Γλώσσα προγραμματισμού λέγεται μια τεχνητή γλώσσα που μπορεί να χρησιμοποιηθεί για τον έλεγχο μιας μηχανής, συνήθως ενός υπολογιστή. Οι γλώσσες προγραμματισμού (όπως άλλωστε και οι ανθρώπινες γλώσσες) ορίζονται από ένα σύνολο συντακτικών και εννοιολογικών κανόνων, που ορίζουν τη δομή και το νόημα, αντίστοιχα, των προτάσεων της γλώσσας.

Υπάρχουν χιλιάδες διαφορετικές γλώσσες προγραμματισμού, και κάθε χρόνο δημιουργούνται περισσότερες. Κάθε γλώσσα προγραμματισμού έχει το δικό της σύνολο τυπικών κανόνων που αφορούν το συντακτικό, το λεξιλόγιο και το νόημα της. Για πολλές γλώσσες που χρησιμοποιούνται ευρέως και έχουν χρησιμοποιηθεί για αρκετό χρονικό διάστημα (π.χ. C, C++, Java, Scheme), υπάρχουν ειδικοί φορείς τυποποίησης, οι οποίοι μέσα από τακτές συναντήσεις δημιουργούν, τροποποιούν ή επεκτείνουν τις τυπικέςπροδιαγραφές που διέπουν τη χρήση μιας γλώσσας προγραμματισμού. Άλλες γλώσσες δεν περιγράφονται σε κάποιο επίσημο πρότυπο αλλά ορίζονται μόνο με βάση κάποια υλοποίησή τους , όπως η Python που περιγράφεται από την υλοποίηση CPython.

Γλώσσες Προγραμματισμού:

Ada	Haskell	Pascal
Algol	Java	Perl
Applescr ipt	JavaScript	PHP
AWK	Lisp	Prolog
BASIC	Logo	Python
C	Lua	Ruby
C++	Lucid	Scala
C#	Mathematica	Scheme
Cilk	Matlab	Simula
Clojure	Miranda	Smalltalk
COBOL	ML	SQL
Datalog	OBJ / Σύστημα Maude	Tcl
Erlang	Objective-C	Visual Basic
Forth	OCaml	

1.7 Σχετικά με τι γλώσσες που χρησιμοποιήθηκαν για την υλοποίηση της κατασκευής

Στην εργασία μας θα συναντήσουμε δυο γλώσσες προγραμματισμού, την C++ και την PHP. Η ανάπτυξη του απαραίτητου κώδικα ως προς τον προγραμματισμό του Arduino έγινε σε C++. Η C++ είναι μια γενικού σκοπού γλώσσα προγραμματισμού Η/Υ. Θεωρείται μέσου επιπέδου γλώσσα, καθώς περιλαμβάνει έναν συνδυασμό χαρακτηριστικών απόγλώσσες υψηλού και χαμηλού επιπέδου. Είναι μια μεταγλωττιζόμενη γλώσσα πολλαπλών παραδειγμάτων, με τύπους. Υποστηρίζει δομημένο, αντικειμενοστραφή και γενικό προγραμματισμό.

Η γλώσσα αναπτύχθηκε από τον Μπιάρνε Στρούστρουπ το 1979 στα εργαστήρια Bell της AT&T, ως βελτίωση της ήδη υπάρχουσας γλώσσας προγραμματισμού C, και αρχικά ονομάστηκε "C with Classes", δηλαδή C με Κλάσεις. Μετονομάστηκε σε C++ το 1983. Οι βελτιώσεις ξεκίνησαν με την προσθήκη κλάσεων και ακολούθησαν, μεταξύ άλλων, εικονικές συναρτήσεις, υπερφόρτωση τελεστών, πολλαπλή κληρονομικότητα, πρότυπα κ.α.Η γλώσσα ορίστηκε παγκοσμίως, το 1998, με το πρότυπο ISO/IEC 14882:1998. Η τρέχουσα έκδοση αυτού του προτύπου είναι αυτή του 2003, η ISO/IEC 14882:2003. Μια καινούρια έκδοση είναι υπό ανάπτυξη, γνωστή ανεπίσημα με την ονομασία C++0x. [8]

Το **περιβάλλον του διαχειριστή**, στο οποίο μπορούμε να δούμε την κατανάλωση και το κόστος της κατανάλωσης των χρηστών αναπτύχθηκε σε PHP.Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ

αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0.

Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP. [7]Παρακάτω θα αναλύσουμε το XAMPP, το Codeigniter και το SublimeTextEditor τα οποία χρησιμοποιήθηκαν για την δημιουργία του περιβάλλοντος του διαχειρηστή.[5] [6]

1.8Ορισμός του XAMPP

Απαραίτητο για την σωστή λειτουργία του μετρητή είναι να αποθηκεύονται σε μια βάση δεδομένων οι μετρήσεις τις κατανάλωσης, ώστε να υπάρχει ένα ιστορικό και να μπορεί να γίνει έλεγχος της κατανάλωσης ανά πάσα στιγμή.Αυτή την βάση δεδομένων στην παρούσα εργασία μας την προσφέρει το XAMPP.

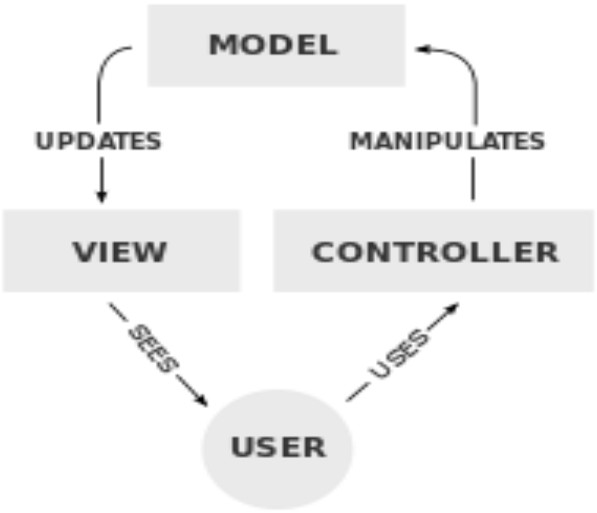
Το **XAMPP** είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σεγλώσσες προγραμματισμού PHP και Perl. Προϋποθέτει μόνο τα λογισμικά συμπίεσης αρχείων zip, tar, 7z ή exe κατά την διάρκεια της εγκατάστασης. Το XAMPP έχει δυνατότητα αναβάθμισης σε νέες εκδόσεις του εξυπηρετητή ιστοσελίδων http Apache, της βάσης δεδομένων MySQL, της γλώσσας PHP και Perl. Το XAMPP συμπεριλαμβάνει επίσης τα πακέτα OpenSSL και το phpMyAdmin.

Επίσημα οι σχεδιαστές του XAMPP προόριζαν το λογισμικό ως εργαλείο ανάπτυξης και δοκιμής ιστοσελίδων τοπικά στον υπολογιστή χωρίς να απαιτείται 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 «wampp». Για την βάση δεδομένων MySQL υπάρχει ο χρήστης «root» χωρίς κωδικό πρόσβασης. [9]

Εικόνα 5.XAMPPLogo

1.9ΟρισμόςτουCodeigniter

ToCodeigniter είναι ένα ανοιχτού κώδικα PHPframework το οποίο χρησιμοποιείται για την κατασκευή δυναμικών ιστοσελίδων. Η πρώτη δημόσια έκδοση κυκλοφόρησε στις 28 Φεβρουαρίου 2006 .Είναι βασισμένο στο πρότυπο ανάπτυξης Model-View-Controller. ToModel και το View είναι προαιρετικά, ο Controller όμως είναι απαραίτητος.



Εικόνα 6.MVC

Η κάθε συνιστώσα αυτού του προτύπου ανάπτυξης έχει τον ρόλο της:

- Ο Controller που στέλνει εντολές και στις δυο υπόλοιπες συνιστώσες και μπορεί να επηρεάσει την κατάστασή τους.
- Το Model
- Το View όπου είναι υπεύθυνο γι' αυτό που βλέπει ο χρήστης στον υπολογιστή του

Το πρότυπο MVC που προαναφέρθηκε μας βοηθά να κρατάμε τον κώδικα σε μικρότερο μέγεθος,να τον σπάμε σε μέρη μικρότερα τα οποία μπορούν να καλούνται εύκολα μέσα στον κώδικα και να φορτώνονται γρήγορα. Όποιο πρόβλημα και αν αντιμετωπίσεις κατά την ανάπτυξη του κώδικα σου , υπάρχει από πίσω μια μεγάλη ενεργή κοινότητα στην οποία μπορείς να απευθυνθείς για βοήθεια.Το Documentation είναι από τα πληρέστερα και πιο καλογραμμένα που μπορεί να συνοδεύει ένα framework. [10] [11]

1.10SublimeTextEditorγια την ανάπτυξη κώδικα

Οι κειμενογράφοι για κώδικα είναι εργαλείο που χρησιμοποιούν οι προγραμματιστές καθώς δημιουργούν προγράμματα. Υπάρχουν πάρα πολλοί και με διαφορές κειμενογράφοι , όλοι

όμως έχουν τον ίδιο σκοπό, την ανάπτυξη πηγαίου κώδικα. Ένας κειμενογράφος προσφέρει ταχύτητα στην ανάπτυξη του κώδικα, αυτοσυμπληρώνει λέξεις ή εντολές που έχουν ξαναχρησιμοποιηθεί, αποφεύγονται ευκολότερα πιθανά λάθη καθώς εμφανίζονται στην οθόνη με διαφορετικούς χρωματισμούς οι εντολές, οι δεσμευμένες λέξεις, οι μεταβλητές και έτσι ο προγραμματιστής μπορεί ευκολότερα να εντοπίσει διαφορές και λάθη.

Μερικοί γνωστοί TextEditors είναι :

- Amaya
- Arachnophilia
- Bespin
- Bluefish
- Sublime

Στην παρούσα εργασία θα χρησιμοποιηθεί ο SublimeTextEditor.

Εικόνα 7.SublimeLogo

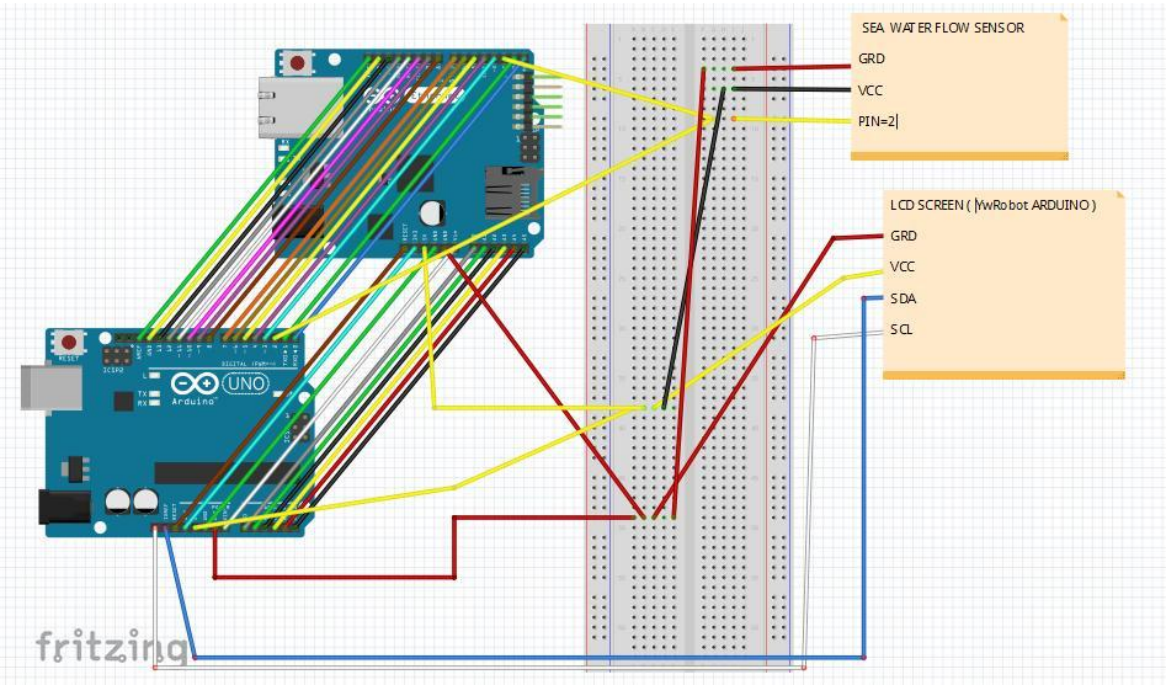
ΚΕΦΑΛΑΙΟ 2: ΑΝΑΛΥΣΗΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΔΥΝΑΜΙΚΟΥ ΜΕΤΡΗΤΗ ΚΑΤΑΝΑΛΩΣΗΣ ΝΕΡΟΥ

2.1 Προγράμματα και Εφαρμογές που χρησιμοποιήθηκαν

Στο πρώτο κεφάλαιο αναφέρονται και αναλύονται όλα τα βασικά προγράμματα και εφαρμογές χρησιμοποιήθηκαν. Παρακάτω γίνεται μια συνοπτική αναφορά για όλα όσα χρησιμοποιήθηκαν για την ανάπτυξη της κατασκευής.

- **Fritzing**

Αρχικά ξεκινήσαμε την υλοποίηση χρησιμοποιώντας το Frizting. ToFritzing είναι ένα πρόγραμμα προσομοίωσης και σχεδίασης μικροελεκτών.Σε αυτό κάναμε ένα σχέδιο της κατασκευής μας και αργότερα προσομοιώσαμε τον κώδικά μας για να δούμε αν λειτουργεί και αν έχει αποτελέσματα. [12]



Εικόνα 8.Σχέδιο στο Fritzing

- **XAMPP**
- **Codeigniter**
- **Sublime Text Editor**

2.2 Υλικά και Εξαρτήματα

Για το πρακτικό κομμάτι της παρούσας εργασίας η πρώτη κίνηση ήταν να ξεκαθαρίσουμε τι υλικά θα μας ήταν απαραίτητα και να τα παραγγείλουμε, αναφέρονται και αναλύονται παρακάτω όλα τα υλικά και τα εξαρτήματα που χρησιμοποιήθηκαν.

- Arduino Uno πλακέτα
- Ethernet Πλακέτα
- LCD Οθόνη
- Καλώδιο Ethernet
- Καλώδιο USB
- Μετρητής Κατανάλωσης Νερού

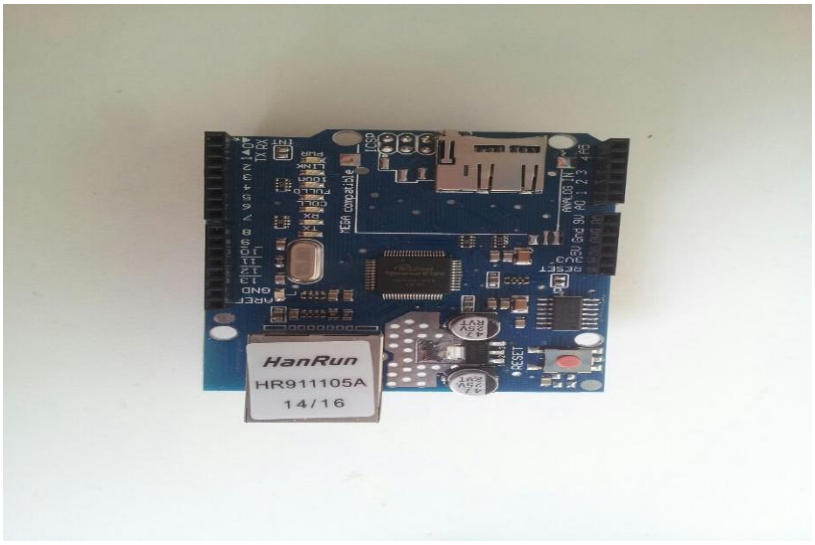
Η έκδοση Arduino που χρησιμοποιήσαμε είναι η UNO, η οποία απεικονίζεται παρακάτω:

Εικόνα 9.Πλακέτα ArduinoUNO

Αυτή η πλακέτα είναι στην ουσία το σώμα της εφαρμογής, πάνω στην οποία συνδέθηκαν όλα τα υπόλοιπα εξαρτήματα της κατασκευής.

Επόμενη κίνηση ήταν η συναρμολόγηση των ξεχωριστών κομματιών.

Πάνω από την πλακέτα του Arduino τοποθετήθηκε η πλακέτα Ethernet.



Εικόνα 10.Πλακέτα Ethernet

Με την τοποθέτηση της πλακέταςEthernet επιτυγχάνεται η σύνδεση του Arduino μας με το διαδίκτυο.Η ταχύτητα σύνδεσης που μας παρέχει είναι 10/100Mb.Πάνω της υπάρχουν LEDs τα οποία σου δείχνουν την κατάσταση της λειτουργίας της κάθε στιγμή.

Τα LEDs που θα συναντήσουμε καθώς και το τι αντιπροσωπεύει το καθένα είναι τα εξής :

- PWR, ανάβει όταν η πλακέτα τροφοδοτείται με ρεύμα
- LINK, ανάβει όταν συνδέεται με ένα δίκτυο και αναβοσβήνει όταν μεταδίδονται ή στέλνονται δεδομένα.
- FULLD, ανάβει υποδεικνύοντάς μας πως η σύνδεση είναιFullDuplex
- 100M, μας δείχνει ότι υπάρχει σύνδεση με το δίκτυο
- RX, αναβοσβήνει όταν λαμβάνονται δεδομένα
- TX, αναβοσβήνει όταν στέλνονται δεδομένα
- COLL, αναβοσβήνει όταν υπάρχουν προβλήματα με το δίκτυο. [13]

Έπειτα τοποθετήθηκε η οθόνη LCDη οποία μας προβάλλει εκείνη τη στιγμή τα δεδομένα που στέλνονται στην βάση δεδομένων μας.

Εικόνα 11.LCDMonitor.

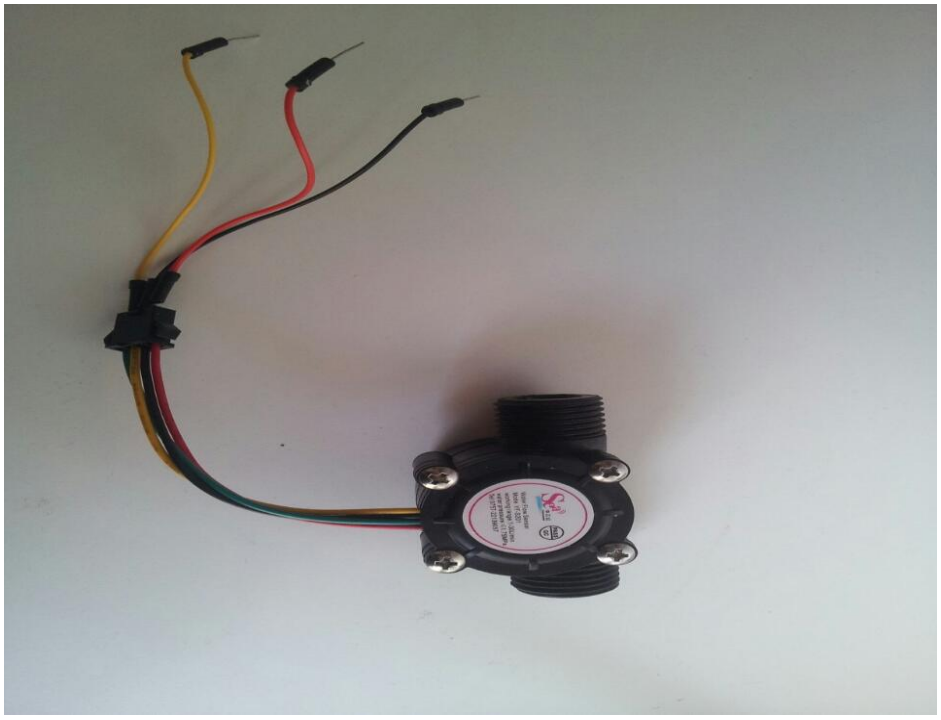
Το Ethernet καλώδιο το χρησιμοποιούμε για να συνδέσουμε την κατασκευή με το δίκτυο, μέσω του καλωδίου επιτυγχάνεται η μεταφορά των δεδομένων.

Εικόνα 12.Ethernet Καλώδιο Cat 5

Με το καλώδιο USB συνδέουμε την κατασκευή με τον υπολογιστή ώστε να μπορέσουμε να την προγραμματίσουμε κατάλληλα ώστε να καλύπτει τις ανάγκες μας.

Εικόνα 13.USB Καλώδιο

Το τελευταίο κομμάτι της κατασκευής ήταν η τοποθέτηση το μετρητή κατανάλωση νερού. Το υλικό κατασκευής του είναι Fibernylon και δεν επηρεάζει την ποιότητα του νερού. Ο μετρητής χρησιμοποιείται για την καταγραφή της ποσότητας του νερού που περνάει από αυτόν σε λίτρα. Έτσι εφόσον έχουμε αναπτύξει τον κώδικα ώστε να αναγνωρίζει και να λαμβάνει τα δεδομένα από τον μετρητή καθώς αυτός διαρρέεται από νερό, μέσω της Ethernet πλακέτας αποστέλλονται τα δεδομένα στην βάση δεδομένων μας και εκεί είναι διαθέσιμα για προβολή.

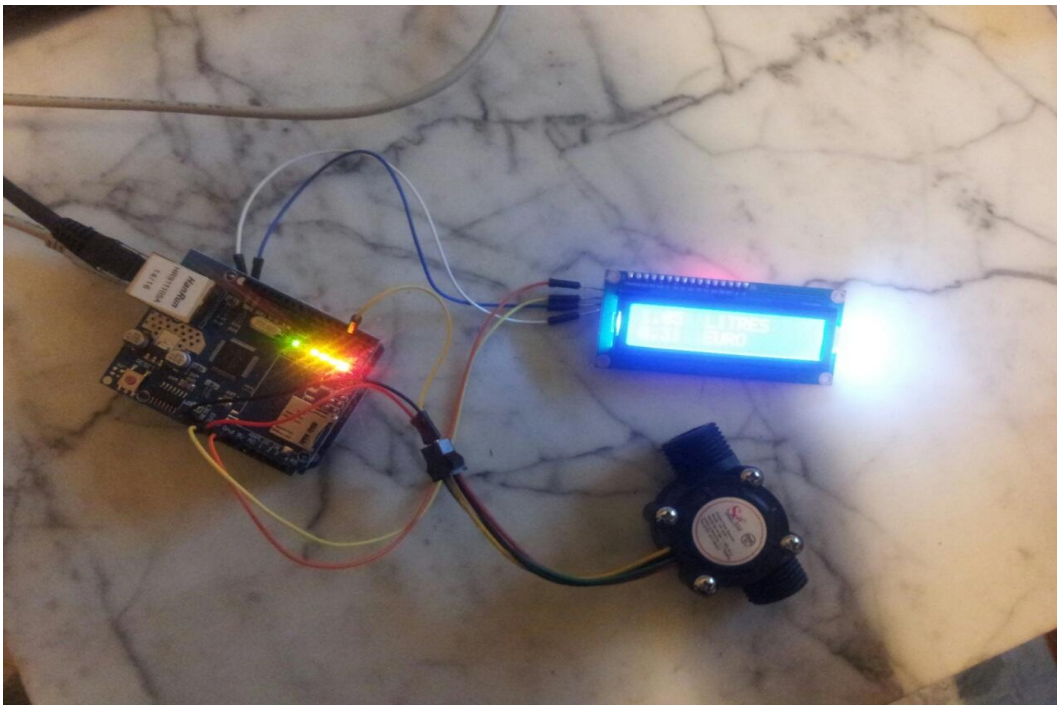


Εικόνα 14.Μετρητής Κατανάλωσης Νερού

Τα χαρακτηριστικά του μετρητή μας είναι τα εξής:[14]

Τάση Λειτουργίας	5V-24V
Μέγιστη Ένταση	15 mA（DC 5V）
Βάρος	43g
Διάμετρος	20mm
Ροή Νερού	1-30 L/min
Θερμοκρασία Λειτουργίας	0℃-80℃
Θερμοκρασία Υγρού	<120℃
Υγρασία	35%-90%RH
Πίεση	Κάτω από 1.2Mpa
Θερμοκρασία	-25℃−+80℃

Αφού συναρμολογήθηκαν όλα τα κομμάτια , ο μετρητής πήρε την παρακάτω μορφή :



Εικόνα 15.WaterCounter

Μετά δημιουργήσαμε μια απλή συσκευασία για αισθητικούς και πρακτικούς λόγους και ολοκληρωμένος στην τελική του μορφή είναι όπως παρουσιάζεται στην παρακάτω εικόνα

Εικόνα 16.WaterCounter στην τελική μορφή

2.3Ανάλυση ενεργειών και βημάτων συσχέτισης προγραμμάτων και υλικών

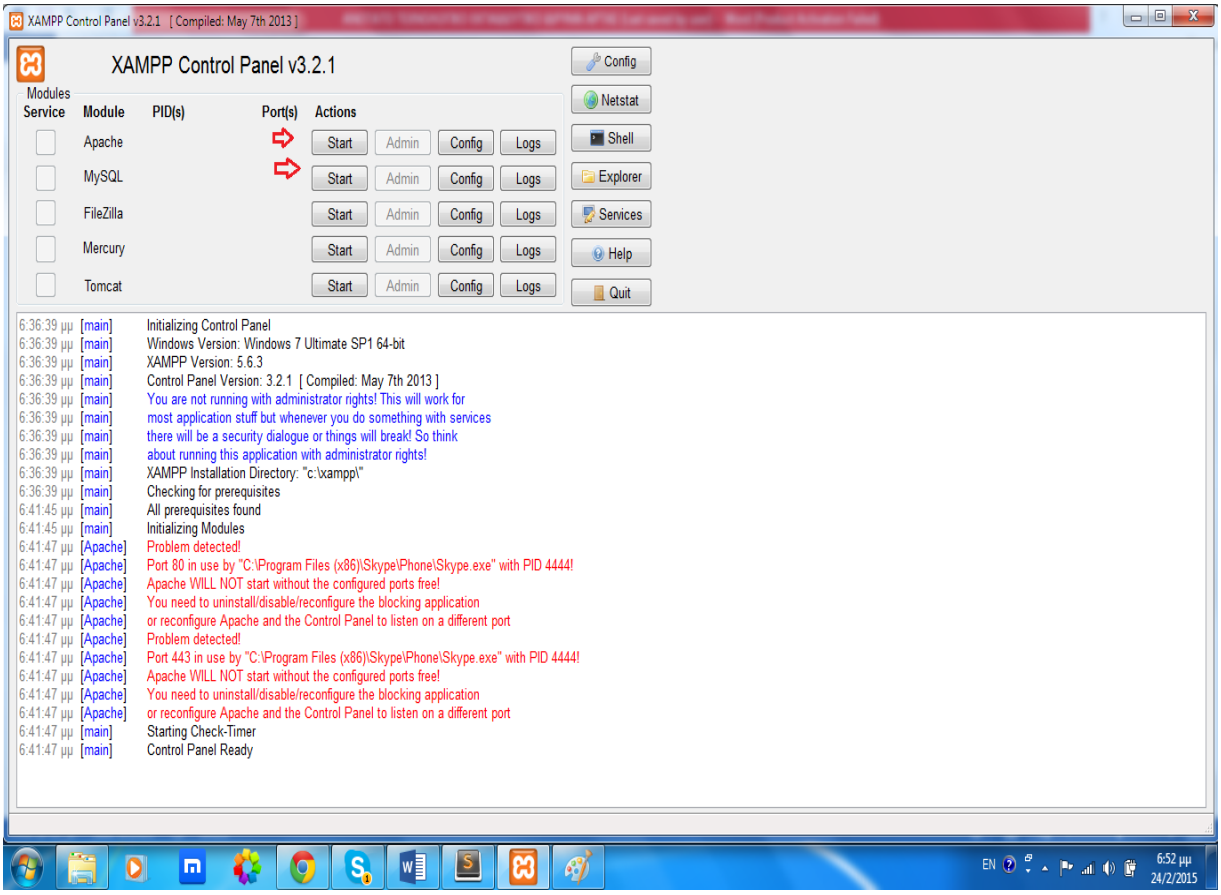
Η λειτουργία του δυναμικού μετρητή κατανάλωσης νερού , όπως προδίδει και ο τίτλος, είναι να συλλέγει την ποσότητα της κατανάλωσης του νερού μέσω του μετρητή, να τα στέλνει δυναμικά σε μία βάση δεδομένων και τέλος εμείς να μπορούμε να ελέγχουμε την κατανάλωση του χρήστη μέσω ενός περιβάλλοντος στο οποίο θα μπορούμε να βλέπουμε απ’ακριβώς την ώρα καταγραφής της μέτρησης, την κατανάλωση, αλλά και το κόστος αυτής.

Για να φτάσουμε στο τελικό ακολουθήσαμε κάποια βήματα τα οποία αναλύονται παρακάτω με την σειρά την οποία έγιναν.

Αρχικά ξεκινήσαμε δημιουργώντας την βάση δεδομένων, η οποία χρειάζεται για να αποθηκεύονται εκεί η μετρήσεις και μετέπειτα να τραβάμε αυτά τα δεδομένα των μετρήσεων από την βάση και να τα προβάλλουμε στο περιβάλλον του χρήστη.

Για την δημιουργία της βάσης μας χρησιμοποιήσαμε το XAMPP(υποκεφάλαιο 1.7), το κατεβάσαμε δωρεάν από το διαδίκτυο και το εγκαταστήσαμε στον υπολογιστή μας, όπως ακριβώς γίνεται εγκατάσταση οποιουδήποτε προγράμματος και στην αρχική του προγράμματος πατάμε το κουμπί START που βρίσκεται δεξιά από το πλαίσιο APACHE και MYSQL, όπως

υποδεικνύουν τα κόκκινα βελάκια παρακάτω. Κάνουμε αυτές τις κινήσεις για να χρησιμοποιήσουμε τον υπολογιστή μας θεωρητικά ως Server και να μπορούμε να τρέξουμε την εργασία μας.

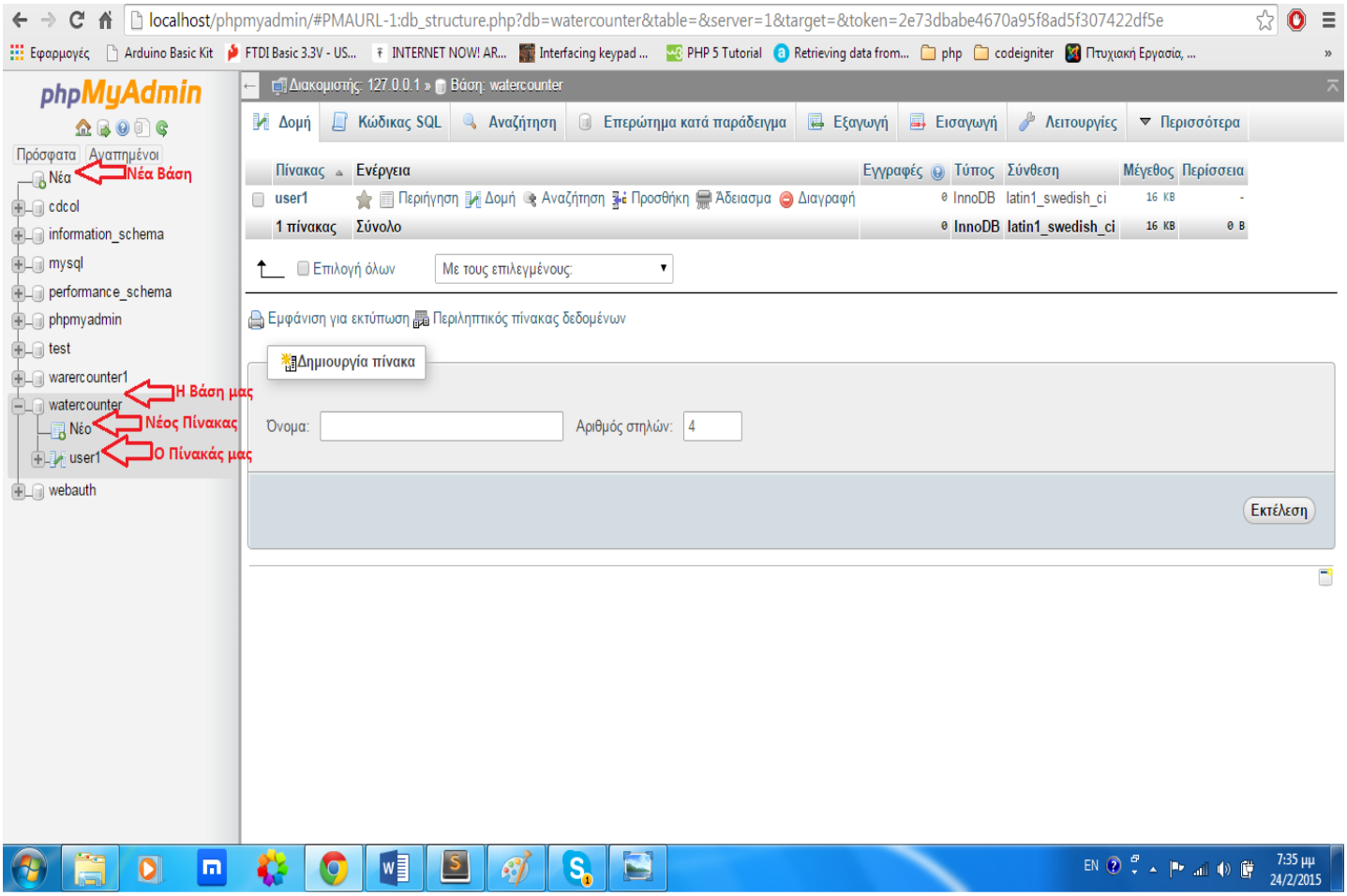


Εικόνα 17.Περιβάλλον XAMPP

Έπειτα ανοίγουμε τον εξυπηρετητή μας και πληκτρολογούμε http://localhost/xampp/. Σε αυτό το σημείο επιλέγουμε κάτω αριστερά το πλαίσιο που αναγράφει <**phpMyAdmin**> στην κατηγορία tools.

Επιλέγοντάς το μεταβαίνουμε στο σημείο όπου δημιουργούμε την βάση που θέλουμε και την διαχειριζόμαστε διαμορφώνοντάς την ανάλογα με τις ανάγκες μας.

Την βάση στην παρούσα εργασία την ονομάσαμε watercounter και μέσα της δημιουργήσαμε έναν πίνακα με όνομα user1 και δυο τιμές. Η πρώτη τιμή είναι η katanalwsh τύπου float όπου εδώ θα αποθηκεύεται η ποσότητα της κατανάλωσης του χρήστη και η δεύτερη η datetime τύπου timestamp η οποία θα μας δείχνει την ώρα της μέτρησης. Στην φωτογραφία που έπεται τα κόκκινα βελάκια και τα σχόλια δείχνουν τα παραπάνω βήματα.



Εικόνα 18.Βάση Δεδομένων

Όπως εξηγήσαμε και παραπάνω στον πίνακα της βάσης θα καταγράφονται τα δεδομένα από τις μετρήσεις τις οποίες θα παίρνει ο μετρητής ανά τακτά χρονικά διαστήματα.

Επόμενο βήμα ήταν να αναπτύξουμε τον κώδικα βάση του οποίου ο μετρητής θα προγραμματιζόταν ώστε να μπορεί να περνάει τα δεδομένα του στην βάση δεδομένων.

Ο κώδικας του Arduino που στον οποίο βασίζεται η λειτουργία όλης της πλακέτας, αναπτύχθηκε σε γλώσσα C++. Παρακάτω παραθέτουμε τον κώδικα του Arduino που χρησιμοποιήσαμε στην τελική του μορφή.

```
#include <SPI.h>
#include <Ethernet.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27,16,2);

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
EthernetClient client;
volatileintstrofes;                    //μετράειτιςστροφέςτουαισθητήρα
String counter;
intmetrisi;
intestitiras = 2;                              //τοpintουαισθητήρα
floatTotalliters;
float liters;
int time;
voidsensorfan ()     //αυξάνει τις στροφές
{
strofes++;
}

void setup()
{
lcd.init();                                          // initialize the lcd
lcd.backlight();
lcd.home();
Ethernet.begin(mac);
pinMode(estitiras, INPUT);//ξεκινάει τον αισθητήρα στοpin 2
Serial.begin(19200); //This is the setup function where the serial port is initialised,
attachInterrupt(0, sensorfan, RISING); //ξεκινάειτοinterupt
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());
time = 0;
counter="";
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("IP Address");
lcd.setCursor(0, 1);
lcd.print(Ethernet.localIP());
}


void loop ()
{
sei(); //Enables interrupts
// midenisilitronmetaapoena mina
//time= millis();
//if (time >2628000000 ){
//Totalliters=0.0;
//}
strofes = 0;      //μηδενίζει τις στροφές για να ξεκινήσει να υπολογίζει
//  sei();
delay (10000);      //Wait 10 second
//  cli();
metrisi =(strofes / 7.5); //(Pulse frequency x 60) / 7.5Q, = flow rate in L/hour
liters =(float) strofes / 450;
Totalliters += liters;
counter = "count=" + String(Totalliters);
if (client.connect("192.168.1.160",80)) { // server address
```

```
Serial.println("Client connected.");

client.println("POST /ci/index.php/WaterCounter/add HTTP/1.1");

client.println("Host: 192.168.1.160"); // server address

client.println("Content-Type: application/x-www-form-urlencoded");

client.print("Content-Length: ");

client.println(counter.length());

client.println();


client.println(counter);
//Serial.println(counter);
//
//Serial.println(metrisi);

}
else {

Serial.println("Client connection failed.");

}
delay(1000);                          //περίμενε 1s για να γίνει τοPOST
if (client.connected()){
client.stop();
```

```
}
Serial.print (liters);
Serial.print (" liters\r\n");
Serial.print (Totalliters);
Serial.print (" total liters\r\n");
Serial.print (metrisi, DEC);
Serial.print (" L/MIN\r\n");
Serial.print(strofes);
Serial.print (" fan \r\n");
delay(1000);
//  sei();
lcd.clear();
lcd.setCursor(0, 0);
lcd.print((int)strofes*60/7.5);
lcd.setCursor(6, 0);
lcd.print("L/H");
lcd.setCursor(0, 1);
lcd.print(liters);
lcd.setCursor(6, 1);
lcd.print("L");
delay(5000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print((float)Totalliters,2);
lcd.setCursor(6, 0);
lcd.print("LITRES");
lcd.setCursor(0, 1);
lcd.print((float)Totalliters*0.3,2);
lcd.setCursor(6, 1);
lcd.print("EURO");
```

```
cli(); //Disableinterrupts
delay(42000);
}
```

. Με τον παραπάνω κώδικα προγραμματίστηκε όλη η κατασκευή και μας εξασφαλίζειόλη την λειτουργία της,δηλαδή συλλογή δεδομένων καθώς ο μετρητής διαρέεται από νερό, αποστολή δεδομένων στην βάση και τα μηνύματα της LCDοθονης.

Αφού τελειώσαμε με τον προγραμματισμό της πλακέτας και πλεον μπορούσαμε να στέλνουμε τα δεδομένα στην βάση δεδομένων μας, το μόνο που έμενε ήταν να δημιουργήσουμε το περιβάλλον στο οποίο θα εκθέτουμε αυτά τα δεδομένα, δηλάδη παρακάτω θα αναλύσουμε τις κινήσεις που έγιναν για να μπορούμε να τραβάμε τα δεδομένα από την βάση.

Για αυτό το κομμάτι χρησιμοποιήσαμε το Codeigniter, το κατεβάσαμε και αυτό από το διαδίκτυο, κατεβαίνει με την μορφή συμπιεσμένου αρχείου που περιέχει διάφορους φακέλους. Αυτούς τους φακέλους τους τοποθετήσαμε στον φάκελο XAMPP/htdocs που δημιουργήθηκε με την εγκατάσταση του XAMPP

Το Codeigniterλειτουργεί βασισμένο στο μοντέλο MVC(Model-View-Controller) όπως ακριβώς αναφερθήκαμε στο κεφάλαιο 1. Έτσι και εμείς ακολουθήσαμε τους κανόνες και δημιουργήσαμε ένα αρχείο για το Model, ένα για το Viewκαι ένα για τον Controller, τα οποία με το τρόπο λειτουργίας του Codeignitersυνεργάζονται και μας βοηθούν πολύ ώστε να πάρουμε το τελικό αποτέλεσμα, να δούμε στην οθόνη μας τα δεδομένα της βάσης μας.

Αρχικα φτιάξαμε το αρχείο του Controller, το οποίο ονομάσαμε waterCOUNTER.php.

```
<?php

classwaterCOUNTER extends CI_controller
{

function __construct()
{
```

```
parent::__construct();

}

function user()
{
$this->load->model('WaterCounterM');
$data['query'] = $this->WaterCounterM->getuser();
$this->load->view('index.php', $data);
}
functionrealtime()
{
$this->load->model('WaterCounterM');
$data['query'] = $this->WaterCounterM->getuserall();
$this->load->view('index.php', $data);
}

function add()
{
$count=$this->input->post('count');
$this->load->database();

$this->db->set('katanalwsh', $count);
$this->db->insert('user1');
$this->load->model('WaterCounterM');
$data['query'] =$this->WaterCounterM->getuser();
$this->load->view('index.php', $data);
        }
}
?>
```

Αποτελείται από τρεις συναρτήσεις(function),η κάθε μια από αυτές έχει την δική της λειτουργία.

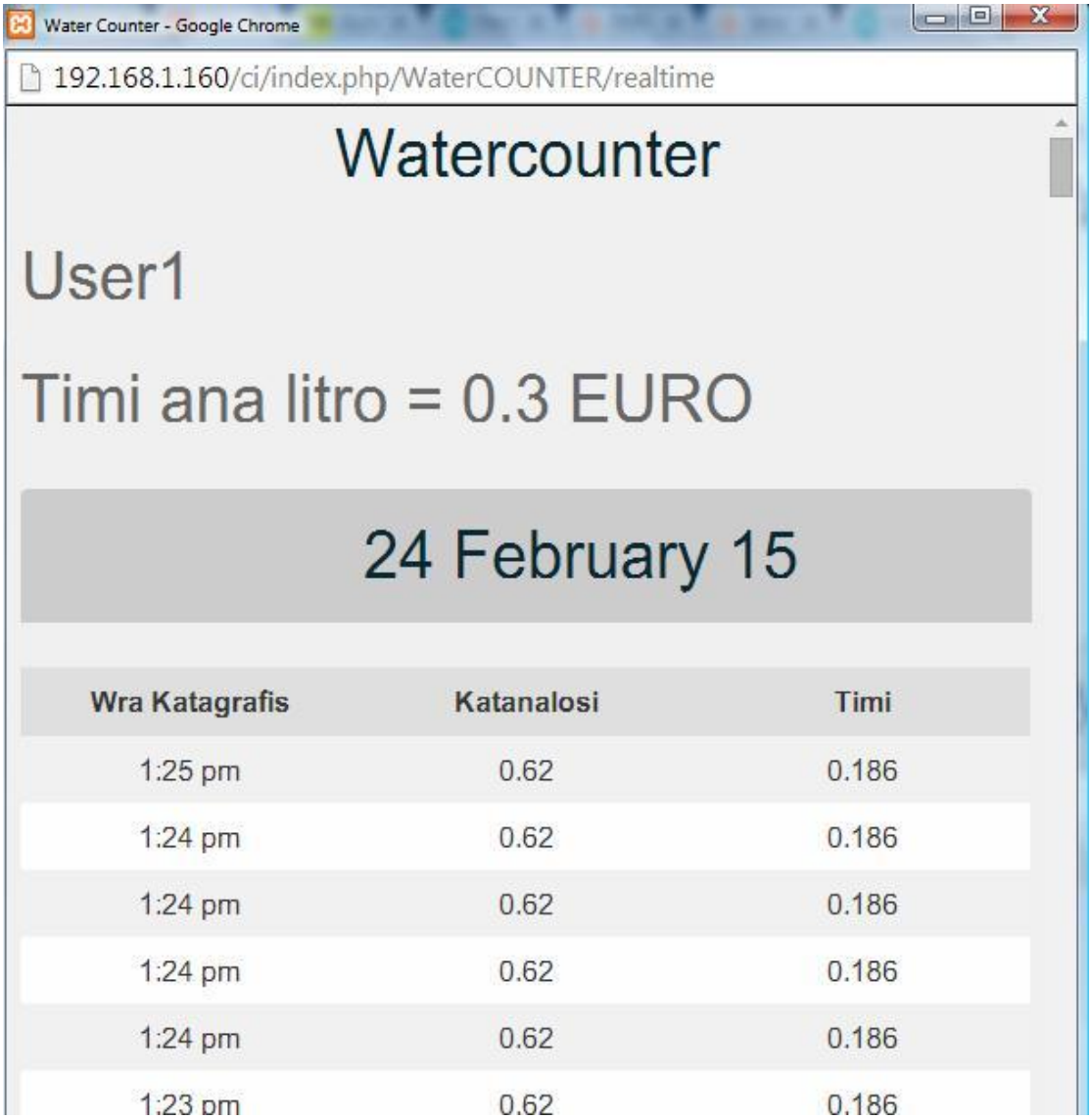
Η πρώτη συνάρτηση **functionuser()**,αυτό πουτης ορίσαμε να κάνει είναι απλά να καλεί το αρχείο του Model και συγκεκριμένα να παίρνει και να βάζει το αποτέλεσμα της συνάρτησης **getuser()**του Model στην μεταβλητή **$data['query']**και μετά να καλεί το αρχείο του View δίνοντάς του ως είσοδο την μεταβλητή αυτή. Τρέχοντας τα παραπάνω το αποτέλεσμα που παίρνουμε στην οθόνη μας είναι:

Εικόνα 19.Τελευταία Μέτρηση

Παίρνουμε δηλαδή ως αποτέλεσμα την τελευταία μέτρηση του χρήστη, την ώρα, την ποσότητα της κατανάλωσης και το κόστος αυτής.

Στην επόμενη συνάρτηση του Controller**functionrealtime()** γίνεται ακριβώς το ίδιο με μόνη διαφορά ότι αυτήν την φορά καλούμε άλλη συνάρτηση του Model, την συνάρτηση **getuserall().** Αυτό διαφοροποιεί και το αποτέλεσμα που θα έχουμε στην

οθόνη μας, αυτήν την φορά δεν θα βλέπουμε μόνο την τελευταία μέτρηση της κατανάλωσης του χρήστη.



Εικόνα 20.Κατανάλωση σε Πραγματικό Χρόνο

Με την τελευταία συνάρτηση του Controller**functionadd()** αυτό που επιτυγχάνουμε είναι να ανανεώνουμε τα δεδομένα της βάσης μας όταν πραγματοποιείται νέα μέτρηση.

Επόμενο βήμα ήταν να δημιουργήσουμε το αρχείο το Model το οποίο κάνει όλη την δουλεία στην ουσία κάθε φορά που καλείται σε οποιοδήποτε σημείο του Controller. Τοαρχείο του Modelτο ονομάσαμε **waterCounterM.php**και δημιουργήσαμε μέσα του δυο συναρτήσεις.

```
classwaterCounterM extends ci_model
<?php
{

function __construct()
{
parent::__construct();

}
public function getuser()
{

$this->load->database();
$this->db->from('user1');
$this->db->order_by('datetime', 'desc');
$this->db->limit('1');
$query=$this->db->get();
return $query->result();
}
public function getuserall()
{
$this->load->database();
$this->db->from('user1');
$this->db->order_by('datetime', 'desc');
$query=$this->db->get();
return $query->result();
}
}
?>
```

Η συνάρτηση **functiongetuser()**φορτώνει την βάση δεδομένων, τακτοποιεί τα περιεχόμενα της **datetime**του πίνακα που βρίσκεται στην βάση δεδομένων κατά το νεότερο και επιστρέφει την τελευταία καταγραφή. Δεν την εμφανίζει πουθενά απλά δουλειά της είναι να κρατά το αποτέλεσμα και να το δίνει οπουδήποτε ζητηθεί στον Controller. Είναι δουλειά του Controllerόπως προαναφέραμε να στείλει αυτό το αποτέλεσμα στο View ώστε να εμφνιστεί στην οθόνη μας [Εικόνα 18].

Η επόμενη συνάρτηση **getuserall()** κάνει το ίδιο με την προηγούμενη μόνο που επιστρέφει ως αποτέλεσμα όλες τις καταγραφές από την βάση δεδομένων [Εικόνα 21].

Το τελευταίο κομμάτι είναι ο κώδικας που περιέχεται στο αρχείο **index.php**του **View.**Για αυτό το μέρος της εργασίας έχουμε δημιουργήσει και ακόμα ένα αρχείο που είναι υπεύθυνο για τα γραφικά που βλέπουμε στο περιβάλλον που δημιουργήσαμε στο View, το **waterstyle.css.**

Στο index.php υπάρχουν και κάποια κομμάτια γραμμένα σε htmlγλώσσα, όπως βλέπουμε παρακάτω.

```
<html>
<head>
<link href="/ci/assets/css/waterstyle.css" rel="stylesheet" type="text/css" />

<scriptsrc="/ci/assets/js/jquery-1.10.2.js"></script>
<scriptsrc="/ci/assets/js/jquery-ui-1.10.4.custom.js"></script>

<title>Water Counter</title>
</head>
<body>

<?php

$previmerominia= date('y-m-d') ;
$realtime = "/ci/index.php/WaterCOUNTER/user";
$realtime1 = "/ci/index.php/watercounter/user";
```

```
$currentpage = $_SERVER['REQUEST_URI'];
if ($realtime==$currentpage || $realtime1==$currentpage )
{
?>

<h1>Watercounter</h1>
<div class="header">
<div class="name">
<h2>User1</h2>
</div>
</div>
<!--iptou server 192.168.1.160 -->
<atarget='_blank'  href='http://192.168.1.160/ci/index.php/WaterCOUNTER/realtime'
onclick=  "javascript:window.open(this.href,  "  ,
'menubar=no,toolbar=no,resizable=yes,scrollbars=yes,height=600,width=600');
returnfalse;"class='realtime'></a>


<?php }
else
{

?>
<script type="text/javascript">
setTimeout(function () { location.reload(true); }, 5000);
</script>
<h1>Watercounter</h1>
<div class="name">
<h2>User1</h2>
</div>
<?php
}
```

```
?>


<h2>Timianalitro = <?php $timi = 0.3; echo "$timi";  ?> EURO </h2>
<div id="accordion" >

<?php
foreach ($query as $row)
{
$imerominia = $row->datetime;
$timiSinolo = $row->katanalwsh * $timi;

if ($previmerominia !== date('d F y',strtotime($imerominia)) )
{

echo "</table>";
echo "<h1>",date('d F y',strtotime($imerominia)),"</h1>";
echo "<table>";


$previmerominia = date('d F y',strtotime($imerominia));
?>
<tr>
<th>WraKatagrafis</th>
<th>Katanalosi</th>
<th>Timi</th>
</tr>
<?php


if ($previmerominia == date('d F y',strtotime($imerominia)))
{
```

```
echo "<tr>";
echo "<td>", date('g(idea) a',strtotime($imerominia)),"</td>";

echo "<td>$row->katanalwsh</td>";
echo "<td>$timiSinolo</td>";
echo "</tr>";

}

}
else {

if ($previmerominia == date('d F y',strtotime($imerominia))) {


?>

<?php
echo "<tr>";
echo "<td>", date('g(idea) a',strtotime($imerominia)),"</td>";
echo "<td>$row->katanalwsh</td>";
echo "<td>$timiSinolo</td>";
echo "</tr>";
}

}

$previmerominia=  date('d F y',strtotime($imerominia));
}
echo "</table>";
?>
```

```
</div>
</body>
<script>
$(function() {
$( "#accordion" ).accordion({
collapsible: true,
});
});
</script>
</html>
```

Το**waterstyle.css**είναιγραμμένοσε**css**γλώσσα,

```
body{font-family:Arial,Helvetica,sans-serif;background-color:#F0F0F0;font-weight:
300;line-height: 1.42em; font-size:12px;}

h1{
text-align: center;
font-size:3em;
font-weight: 300;
line-height:1em;
color:#072936;}

h2{
text-align: left;
font-size:3em;
font-weight: 300;
line-height:1em;
color:#666;}
```

```
table {

color: #333;
font-family: Helvetica, Arial, sans-serif;
width:100%;
border-collapse:collapse;
border-spacing: 0;
max-height: 3em;
}

th {

padding: 0.5em 1em 0.5em 1em;
background: #DFDFDF;
width: 3em;

}

td {

padding: 0.5em 1em 0.5em 1em;
background: #FAFAFA;
text-align: center;
width: 3em;



}
tr:nth-child(even) td { background: #F1F1F1; }

tr:nth-child(odd) td { background: #FEFEFE; }

trtd:hover { background: #666; color: #FFF; }
```

```
td, th {
border: 1px solid transparent;
height: 0.1em;
transition: all 0.6s;
width: 6em;
}



a[target="_blank"] {

display:inline-block;



}
.realtime {content: url(/ci/assets/images/realtime.png);
width:8em;
height:8em;

-webkit-transition-property: -webkit-transform;
-webkit-transition-duration: 1s;

-moz-transition-property: -moz-transform;
-moz-transition-duration: 1s;
}
.realtime:hover {
-webkit-animation-name: rotate;
-webkit-animation-duration: 1s;
-webkit-animation-iteration-count: infinite;
-webkit-animation-timing-function: linear;
```

```
-moz-animation-name: rotate;
-moz-animation-duration: 1s;
-moz-animation-iteration-count: infinite;
-moz-animation-timing-function: linear;
}
@-webkit-keyframes rotate {
from {-webkit-transform: rotate(0deg);}
to {-webkit-transform: rotate(360deg);}
}

@-moz-keyframes rotate {
from {-moz-transform: rotate(0deg);}
to {-moz-transform: rotate(360deg);}

}




.header {
clear: both;
}
.header .name {
float: left;
}
.header .realtime{
float: left;
margin-left: 10px;
}
```

```
.ui-accordion .ui-accordion-header {
display: block;
background:#CCC;
cursor: pointer;
position: relative;
margin-top: 2px;
padding: .5em .5em .5em .7em;
min-height: 0; / support: IE7 /
}
.ui-accordion .ui-accordion-icons {
padding-left: 2.2em;
}
.ui-accordion .ui-accordion-noicons {
padding-left: .7em;
}
.ui-accordion .ui-accordion-icons .ui-accordion-icons {
padding-left: 2.2em;
}
.ui-accordion .ui-accordion-header .ui-accordion-header-icon {
position: absolute;
left: .5em;
top: 50%;
margin-top: -8px;
}
.ui-accordion .ui-accordion-content {
border-top: 0;
overflow: auto;

}
.ui-button {
display: inline-block;
position: relative;
```

```
padding: 0;
line-height: normal;
margin-right: .1em;
cursor: pointer;
vertical-align: middle;
text-align: center;
overflow: visible; / removes extra width in IE /
}
.ui-button,
.ui-button:link,
.ui-button:visited,
.ui-button:hover,
.ui-button:active {
text-decoration: none;
}

/ Corner radius /
.ui-corner-all,
.ui-corner-top,
.ui-corner-left,
.ui-corner-tl {
border-top-left-radius: 4px;
}
.ui-corner-all,
.ui-corner-top,
.ui-corner-right,
.ui-corner-tr {
border-top-right-radius: 4px;
}
.ui-corner-all,
.ui-corner-bottom,
.ui-corner-left,
```

```
.ui-corner-bl {
border-bottom-left-radius: 4px;
}
.ui-corner-all,
.ui-corner-bottom,
.ui-corner-right,
.ui-corner-br {
border-bottom-right-radius: 4px;
}
```

Στο το **waterstyle.css** με τον κώδικα που βλέπουμε κάνουμε πιο όμορφο το περιβάλλον οπού βλέπουμε τα αποτελέσματα, με τον παραπάνω κώδικα έχουν φτιαχτεί τα εφέ της σελίδας μας, όπως η αλλαγή του χρώματος όταν περνάει ο δείκτης του ποντικιού πάνω από ένα στοιχείο .

ΚΕΦΑΛΑΙΟ 3: Εγχειρίδιο τοποθέτησης μετρητή

3.1 Εγκατάσταση του μετρητή

Η τοποθέτηση του μετρητή στην εγκατάστασή μας είναι απλή, βασίζεται στην ίδια λογική με τα αναλογικά ρολόγια μέτρησης που χρησιμοποιούνται σήμερα. Δηλαδή το νερό πρέπει να περνάει από τον αισθητήρα ώστε να μπορεί να καταγράφεται.

Στον μετρητή της παρούσας εργασίας κάτω από τον αισθητήρα ροής νερού υπάρχει ένα βελάκι το οποίο μας δείχνει από ποια κατεύθυνση πρέπει να εισέρχεται το νερό. Μπορούμε να τοποθετήσουμε τον μετρητή στην θέση του παλιού αναλογικού ρολογιού κατά την σωστή φορά και απαραίτητα να τον τροφοδοτήσουμε με ρεύμα καθώς να φροντίσουμε και για την σύνδεσή του με το διαδίκτυο.

ΚΕΦΑΛΑΙΟ 4: ΣΥΜΠΕΡΑΣΜΑΤΑ

4.1 Στόχος και η εξέλιξη.

Σκοπός της παρούσας πτυχιακής εργασίας ήταν η κατασκευή ενός δυναμικού μετρητή κατανάλωσης νερού με την ιδιαιτερότητα να μπορεί να στέλνει την καταγραφή των μετρήσεων μέσω διαδικτύου σε μια βάση δεδομένων από όπου θα μπορούν να ελέγχονται ανά πάσα στιγμή.

Αν και η τεχνολογία έχει προχωρήσει πάρα πολύ, οι χώρες και οι εταιρείες δεν έχουν εκσυγχρονίσει τα συστήματα τους στην καταγραφή κατανάλωσης ενεργειών. Αυτό είναι και κάτι που μας κάνει να πιστεύουμε πως στο εγγύς μέλλον είναι σίγουρο πως σταδιακά όλες οι εταιρείες θα προχωρήσουν σε αλλαγή του κλασικού τρόπου των μετρήσεων τους και θα περάσουν στην χρήση συστημάτων σαν αυτό που αναπτύχθηκε στην παρούσα πτυχιακή εργασία. Πλέον όλοι έχουμε πρόσβαση στο internet στο χώρο μας και αυτό είναι που κάνει ακόμα πιο εύκολη την εφαρμογή της κατασκευής μας.

Με την τοποθέτηση του δυναμικού μετρητή κατανάλωσης νερού αποφεύγονται καταρχήν ώρες εργασίας υπαλλήλων οι οποίοι πηγαίνουν σήμερα από σπίτι σε σπίτι για να καταγράψουν τις τιμές των ρολογιών ώστε να μπορεί να εκδοθεί λογαριασμός πληρωμής, αυτό πρακτικά σημαίνει για τις εταιρείες λιγότερα έξοδα και περισσότερες παραγωγικές ώρες για τους εργαζόμενους.

Επίσης στο μυαλό μας υπάρχει και η εξέλιξη της κατασκευής αργότερα, οπού στο επόμενο στάδιο θα ωφελείται και ο καταναλωτής άμεσα.

Το επόμενο βήμα θα είναι η δημιουργία περιβάλλοντος όπου όλοι οι καταναλωτές θα έχουν τον δικό τους λογαριασμό, θα βλέπουν την κατανάλωση τους και το κόστος αυτής από τον υπολογιστή τους. Επίσης ενδιαφέρουσα και ίσως απαραίτητη, μιας και όλοι μας έχουμε στην ζωή μας τηλέφωνα τα οποία έχουν μεγάλες δυνατότητες και τα έχουμε πάντα μαζί μας, θα ήταν η δημιουργία εφαρμογής για κινητά όπου άμεσα με την πρόσβαση στο διαδίκτυο ο χρήστης θα μπορούσε να μπει στο προσωπικό του περιβάλλον και να ελέγχει και από εκεί την κατανάλωσή του με την δυνατότητα να μπορεί να πληρώσει άμεσα από το κινητό του αν το επιθυμεί, το κόστος της κατανάλωσης του έως εκείνη την στιγμή.

ΚΕΦΑΛΑΙΟ 5: ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Arduino, 2015. *ArduinoEthernetShield*. Διαθέσιμο στο διαδικτυακό τόπο: http://arduino.cc/en/Main/ArduinoEthernetShield. [2]
2. Arduino, 2015. *ArduinoEthernetShield*. Διαθέσιμοστοδιαδικτυακότόπο: http://arduino.cc/en/Main/ArduinoEthernetShield. [13]
3. Arduino, 2015. *Introduction to the Arduino*. Διαθέσιμοστοδιαδικτυακότόπο: http://www.arduino.cc/en/Reference/Board. [3]
4. Arduino, 2015. *WhatisArduino?*Διαθέσιμο στο διαδικτυακό τόπο: http://arduino.cc/en/Guide/Introduction. [1]
5. Braden S. Blanchette.*"Wireless Water Flow Meter Network in the Great Bay*" [ΔιπλωματικήΕργασία], University of New Hampshire - Main Campus, bsz28@wildcats.unh.edu.
6. CodeIgniter, 2014. *Model View Controller*. Διαθέσιμοστοδιαδικτυακότόπο: https://ellislab.com/codeIgniter/user-guide/overview/mvc.html [11]
7. CodeIgniter, 2014. *WhoisCodeIgniterFor*?.Διαθέσιμοστοδιαδικτυακότόπο: https://ellislab.com/codeIgniter/user-guide/ [10]
8. ComputerHope, 2015. C++. Διαθέσιμοστοδιαδικτυακότόπο: http://www.computerhope.com/jargon/c/cplus.htm [8]
9. ComputerHope, 2015. *PHP*. Διαθέσιμοστοδιαδικτυακότόπο: http://www.computerhope.com/jargon/p/php.htm. [7]
10. Computer Hope, 2015. *Programming Language*. Διαθέσιμοστοδιαδικτυακότόπο: http://www.computerhope.com/jargon/p/proglang.htm.[6]
11. Fritzing , (χ.χ). *Fritzingelectronicmadeeasy*. Διαθέσιμοστοδιαδικτυακότόπο: http://fritzing.org/home/ [12]
12. GOKARAJU RANGARAJU. "WATERFLOWGAUGEUSINGARDUINO" ,[ΔιπλωματικήΕργασία],INSTITUTE OF ENGINEERING &TECHNOLOGY.
13. Plotly, 2015. *Arduino + Water Flow Sensor*.Διαθέσιμοστοδιαδικτυακότόπο: https://plot.ly/arduino/water-flow-tutorial/

14. RobotShop. Διαθέσιμοστοδιαδικτυακότόπο: http://www.robotshop.com/media/files/pdf/manual-pow110d3b.pdf [14]
15. Udemy blog, 2015. *Xampp Tutorial: How to use Xampp to run your own web server*. Διαθέσιμοστοδιαδικτυακότόπο: https://blog.udemy.com/xampp-tutorial/. [9]
16. Webopedia, 2015. *Programminglanguage*. Διαθέσιμοστοδιαδικτυακότόπο: http://www.webopedia.com/TERM/P/programming_language.html. [5]
17. Wikipedia, 2015.*List of Arduino boards and compatible systems* .Διαθέσιμοστοδιαδικτυακότόπο: http://en.wikipedia.org/wiki/List_of_Arduino_boards_and_compatible_systems. [4]