

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ (ΤΕΙ)**

**ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ**

**(ΠΡΩΗΝ ΤΜΗΜΑ ΕΜΠΟΡΙΑΣ ΚΑΙ ΔΙΑΦΗΜΙΣΗΣ)**

**ΤΜΗΜΑ ΔΙΟΙΚΗΣΗ ΕΠΙΧΕΙΡΗΣΕΩΝ (ΠΑΤΡΑ)**

**ΤΜΗΜΑ (Πάτρα)**

*Τίτλος Εργασίας:* Χρήση Arduino για την διαδικτυακή παρακολούθηση  
θερμοκρασίας νερού

**Πτυχιακή Εργασία του**

Ζαρούλια Βασίλειου

*Επιβλέπων :* Στάμος Κωνσταντίνος

**ΠΑΤΡΑ, ..... 2015**

Χρήση Arduino για την διαδικτυακή παρακολούθηση θερμοκρασίας νερού.  
Ζαρούλιας Βασίλειος

## Ευχαριστίες

Copyright © Ζαρούλιας Βασίλειος, 2015.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του ΤΕΙ Πατών.

Χρήση Arduino για την διαδικτυακή παρακολούθηση θερμοκρασίας νερού.  
Ζαρούλιας Βασίλειος

*Περίληψη*

Η χρήση αισθητήρων σε συνδυασμό με ποικίλες περιφερειακές συσκευές, στοχεύουν στη συνεχή εποπτεία και παραγωγή πληροφοριών προς ενημέρωση του ενδιαφερόμενου χρήστη, άμεσα και με ακρίβεια. Η συγκεκριμένη εργασία έχει ως στόχο την κατασκευή συσκευής σε συνδυασμό υλικού και λογισμικού που παρακολουθεί την θερμοκρασία υγρών και μεταφέρει την πληροφορία σε κάποιον υπολογιστή.

Επιλέχθηκε το συγκεκριμένο σύστημα να προσαρμοστεί στη λήψη δεδομένων παρατήρησης θερμοκρασίας του νερού και της διακύμανσης αυτού καθώς και να δημιουργηθεί μια βάση που φιλτράρει τα αποτελέσματα των μετρήσεων τα οποία εμφανίζονται στην ιστοσελίδα που έχει κατασκευαστεί γι αυτόν τον σκοπό. Με τη χρήση μικροεπεξεργαστή, αισθητήρα και σταθμού εργασίας, επιτεύχθηκε η λήψη δεδομένων στον υπολογιστή – δέκτη, τοποθετήθηκαν τα δεδομένα σε βάση δεδομένων και εμφανίστηκαν στην ιστοσελίδα. Παρέχεται επίσης στους χρήστες της ιστοσελίδας η υπηρεσία λήψης μέτρησης την στιγμή που οι ίδιοι επιθυμούν .

Σκοπός του project είναι να έχει εφαρμογή σε οποιοδήποτε υγρό στοιχείο όπως: ενυδρεία, λίμνες, ποτάμια, υδραγωγεία κ.α. ώστε να μπορεί να ενημερώσει τον οποιονδήποτε χρήστη για τη διακύμανση της θερμοκρασίας.

**Λέξεις Κλειδιά:** *Arduino, Νερό, Θερμοκρασία, διακύμανση θερμοκρασίας, υγρό, Joomla, X-Ctu, Αισθητήρας.*

Χρήση Arduino για την διαδικτυακή παρακολούθηση θερμοκρασίας νερού.  
Ζαρούλιας Βασίλειος

## Πίνακας Περιεχομένων

---

|   |    |
|---|----|
| Πίνακας Περιεχομένων .....  | 5  |
| Λίστα Σχημάτων .....  | 8  |
| 1 Νερό & μέτρηση θερμοκρασίας του. ....                               | 9  |
| 1.1 Νερό.....   | 9  |
| 1.2 Που μπορεί να εφαρμοστεί η εφαρμογή; .....                        | 10 |
| 2 Hardware Υλοποίησής. ....   | 11 |
| 2.1 Arduino .....   | 11 |
| 2.1.1 Πλεονεκτήματα Arduino.....                                      | 12 |
| 2.1.2 Γνωριμία με την πλακέτα. ....                                   | 13 |
| 2.1.3 Τροφοδοσία.....   | 14 |
| 2.1.4 Κουμπιά και Led's.....  | 16 |
| 2.1.5 Arduino IDE και σύνδεση με τον υπολογιστή. ....                 | 16 |
| 2.2 Αισθητήρας θερμοκρασίας DS18B20.....                              | 17 |
| 2.3 Επικοινωνία του αισθητήρα με τον Arduino. ....                    | 18 |
| 2.4 Γλώσσα προγραμματισμού .....                                      | 19 |
| 3 Software .....  | 19 |
| 3.1 Λογισμικό..Arduino. ....  | 19 |
| 3.1.1 Εγκατάσταση λογισμικού Arduino .....                            | 20 |
| 3.2 Εγκατάσταση του Joomla. ....                                      | 22 |
| 4 Υλοποίηση Εφαρμογής. ....   | 25 |
| 4.1 Συνδεσμολογία.....  | 26 |
| 4.2 Μέτρηση θερμοκρασίας με θερμόμετρο DS18B20. ....                  | 26 |
| 4.2.1 Κώδικας.....  | 27 |
| 4.3 Δημιουργία ιστοσελίδας μέσω Joomla. ....                          | 30 |
| 4.3.1 Τι είναι το Joomla. ....  | 30 |
| 4.3.2 Δημιουργία Component. ....                                      | 32 |
| 4.3.3 Blank.php. ....   | 34 |
| 4.3.4 Δημιουργία controller.php. ....                                 | 35 |
| 4.3.5 Κώδικας:.....   | 38 |
| 4.3.6 Δημιουργία του εκτελέσιμου αρχείου (installation package). .... | 41 |
| 4.3.7 Δημιουργία αρχείου γλώσσας μετάφρασης. ....                     | 41 |
| 5 Προβλήματα & Τρόποι αντιμετώπισης. ....                             | 44 |
| 6 Βελτιστοποιήσεις. ....  | 47 |

|                          |                               |           |
|--------------------------|-------------------------------|-----------|
| 6.1                      | Υλικό (Hardware).....         | 47        |
| 6.2                      | Λογισμικό (Software). ....    | 48        |
| 7                        | Παράρτημα .....               | 50        |
| 7.1                      | Κώδικας Arduino .....         | 50        |
| 7.2                      | Κώδικες Component.....        | 52        |
| 7.2.1                    | Controller.php.....           | 52        |
| 7.2.2                    | Default.php .....             | 56        |
| 7.2.3                    | View - Default.php .....      | 59        |
| 7.3                      | Προδιαγραφές Arduino.....     | 62        |
| 7.4                      | Προδιαγραφές Αισθητήρα.....   | 62        |
| 7.5                      | Προδιαγραφές Joomla 3.0. .... | 63        |
| <b>BIBΛΙΟΓΡΑΦΙΑ.....</b> |                               | <b>64</b> |



## Λίστα Σχημάτων

---

|  |    |
|--|----|
| Εικόνα 1: Αρχιτεκτονική Arduino.....           | 15 |
| Εικόνα 2: Θερμόμετρο DS18B20.....              | 17 |
| Εικόνα 3: Σύνδεση Arduino & DS18B20. ....      | 18 |
| Εικόνα 4: IDE Arduino. ....                    | 19 |
| Εικόνα 5: Λειτουργικά για Arduino. ....        | 20 |
| Εικόνα 6: Arduino IDE Sketch. ....             | 21 |
| Εικόνα 11: Joomla. ....                        | 22 |
| Εικόνα 12: Υλοποιημένη εφαρμογή εργασίας. .... | 25 |
| Εικόνα 13: Σελίδα διαχείρισης Joomla. ....     | 30 |

## 1 Νερό & μέτρηση θερμοκρασίας του.

### 1.1 Νερό.

Το νερό είναι υγρό, διαυγές, άχρωμο σε λεπτά στρώματα. Η καθαρή ουσία είναι άγευστη, ενώ το καλό πόσιμο νερό έχει γεύση, που οφείλεται στα διαλυμένα άλατα και αέρια.

Το νερό υπάρχει και στις τρεις κοινές καταστάσεις της ύλης στη Γη: στερεή, υγρή και αέρια, ενώ συνήθως με μορφή πάγου υπάρχει και σε άλλους πλανήτες και δορυφόρους του ηλιακού συστήματος, αλλά και έξω από αυτό.

Πιο συγκεκριμένα, το 96,5% του νερού της Γης βρίσκεται στους ωκεανούς και τις θάλασσες, το 1,7% στα υπόλοιπα επιφανειακά νερά (λίμνες, ποτάμια, έλη, κ.τ.λ.). Μόνο το 2,5% του νερού της Γης είναι «γλυκό» και το 98,8% του πόσιμου νερού βρίσκεται στα παγοκαλύμματα και στα υπόγεια ύδατα.

Το νερό στη Γη κινείται συνεχόμενα μέσω του «κύκλου του νερού» (μια φυσική ανακύκλωση) που περιλαμβάνει την εξάτμιση (κυρίως των θαλασσών), τη μεταφορά της υγρασίας, τη συμπύκνωση, την κατακρήμνιση (με βροχή, χιόνι, χαλάζι, κ.ά.) και την αποστράγγιση με την οποία το μεγαλύτερο ποσοστό επιστρέφει στις θάλασσες. Το μόριο του νερού αποτελείται από δύο άτομα υδρογόνου (H) και ένα άτομο οξυγόνου (O). Έχει χημικό τύπο  $\text{H}_2\text{O}$  αλλά σε μερικές περιπτώσεις χρησιμοποιούνται και οι χημικοί τύποι HOH και σπανιότερα  $\text{OH}_2$ .

## 1.2 Που μπορεί να εφαρμοστεί η εφαρμογή;

Η συγκεκριμένη εφαρμογή μπορεί να εφαρμοστεί σε πολλούς τομείς όπως:

Γενικότερη εξερεύνηση του βυθού: Με την βοήθεια του Arduino ο χρήστης μπορεί να έχει μία πλήρη εικόνα του βυθού της θάλασσας για να παρακολουθεί τις μεταβολές θερμοκρασίας, PH, στάθμης του νερού και την κινητικότητα της υδρόβιας ζωής.

Επεξεργασία για την ύδρευση μιας πόλης: Με αυτόν τον τρόπο ο χρήστης θα προλαμβάνει πλημμύρες, μολύνσεις και καταστροφές υπόγειας ύδρευσης μιας πόλης.

Μελέτη για την υδρόβια ζωή: Ο χρήστης μπορεί να βγάλει κάποιους γενικούς κανόνες για την ζωή και την κινητικότητα των ψαριών και τις διατροφικές τους συνήθειες.

Ανάπτυξη υδρόβιας τροφής: Ο χρήστης μπορεί να έχει μία πλήρη γνώση αν υπάρχει έλλειψη τροφής και ανάγκη ανατροφοδότησης και μελλοντικής ανάγκης για αλλαγή εξωτερικών συνθηκών για καλλιέργεια νέων τροφών.

Παρακολούθηση σεισμικών φαινομένων: Ο χρήστης με το Arduino μπορεί να έχει πλήρη εικόνα της κατάστασης ενός μελλοντικού σεισμού μέσω αισθητήρων και με ειδοποίηση να προλαμβάνουν μέτρα για την προφύλαξη των ανθρώπων σε οποιαδήποτε περιοχή.

Πρόληψη βλαβών σε σωλήνες-φρεάτια και σε περιοχές που έχουν πληγεί από πυρκαγιά: . Οι χρήστες με ειδικές εφαρμογές και με την βοήθεια του Arduino και της Smart Dust θα μπορεί να παρακολουθεί και την υδρόβια κατάσταση των σωλήνων και μιας περιοχής που ενδεχομένως έχει πληγεί από πυρκαγιά.

## 2 Hardware Υλοποίησής.

---



### 2.1 Arduino

Το Arduino είναι μια πλατφόρμα ανοικτού κώδικα βασισμένη στο σύνδεσμο hardware και software. Προορίζεται για όσους έχουν μικρή εμπειρία σε hardware και έχουν βασικές γνώσεις προγραμματισμού για την υλοποίηση διαδραστικών αντικειμένων ή περιβαλλόντων. Ουσιαστικά είναι ένα ηλεκτρονικό κύκλωμα που η καρδιά του έχει τον μικροελεγκτή Atmega της Atmel ,του οποίου τα σχέδια για το hardware και software ,διανέμονται δωρεάν για να μπορεί να κατασκευαστεί από τον καθένα.

Γι' αυτό το λόγο, προκύπτει και ο ασυνήθιστος για hardware χαρακτηρισμός 'πλατφόρμα προτυποποίησης ανοικτού κώδικα'. Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας ηλεκτρονικός υπολογιστής διότι η πλατφόρμα επιτρέπει τον συνδυασμό πολλών μονάδων εισόδου-εξόδου από τον χρήστη.

Οπότε με τον κατάλληλο προγραμματισμό και συνδυασμό των περιφερειακών (καλώδια, αντιστάσεις) στις μονάδες εισόδου του hardware και τις σωστές εντολές στις εξόδους (LEDs, ρότορες) μπορεί να επιτευχθεί το θεμιτό αποτέλεσμα. Το χαρακτηριστικό αυτό είναι και ένας από τους λόγους που το Arduino διαπρέπει στη ρομποτική.

### 2.1.1 Πλεονεκτήματα Arduino.

Παρόλο που τα σχέδια των πλακετών της εταιρείας διανέμονται δωρεάν και ελεύθερα, υπάρχουν πολλοί χρήστες που δεν έχουν τις γνώσεις να κατασκευάσουν ηλεκτρονικό κύκλωμα. Γι' αυτό το λόγο υπάρχουν έτοιμες πλακέτες σε διάφορες εκδόσεις (*UNO*, *LEONARDO*, *DUE*, *MILANO*, *NOVE* κτλ). Οι πλακέτες αυτές κοστίζουν λιγότερο από 50 ευρώ ενώ μπορεί να συμπεριληφθεί και *starter kit* με *breadboard*, καλώδια και άλλα περιφερειακά. Επίσης δεν πρέπει να περάσουν απαρατήρητα α) η τεράστια κοινότητα που βρίσκεται στο παρασκήνιο και β) το εκτενές *documentation* με την μεγάλου μεγέθους γνωσιακή βάση που υπάρχει *online*.

Το Arduino βασίζεται στον *ATmega328*, έναν *8-bit RISC* μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Ο *ATmega328* διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 2Kb μνήμης *SRAM*, δηλαδή η ωφέλιμη μνήμη που μπορούν να χρησιμοποιούν τα προγράμματά του χρήστη για την αποθήκευση μεταβλητών, πινάκων κ.λπ. κατά το *runtime*.
- 1Kb μνήμης *EEPROM* η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά σας κατά το *runtime*. Σε αντίθεση με

την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή *reset* οπότε είναι το ανάλογο του σκληρού δίσκου.

- 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των δικών μας προγραμμάτων. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή μας. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή *reset*.

### 2.1.2 Γνωριμία με την πλακέτα.

Το Arduino επικοινωνεί μέσω σειριακού interface. Η καρδιά του όμως, όπως προαναφέρθηκε είναι ο μικροελεγκτής AT-mega της Atmel που υποστηρίζει σειριακή επικοινωνία, όπου από τον Arduino προωθείται μέσα από τον ελεγκτή SERIAL-OVER –USB ώστε να μπορεί να συνδεθεί με τον υπολογιστή με USB.

Η συγκεκριμένη σύνδεση χρησιμοποιείται για να παρέχει ρεύμα στη πλακέτα αν δεν χρησιμοποιεί το power junk με άλλη πηγή τροφοδοσίας από τον υπολογιστή. Επίσης χρησιμοποιείται για τη μεταφορά προγραμμάτων προς τον Arduino από τον υπολογιστή αλλά και για την αμφίδρομη επικοινωνία μεταξύ των δυο.

Υπάρχουν δεκατέσσερα θηλυκά Pin (0-13) που λειτουργούν σαν ψηφιακές είσοδοι - έξοδοι. Λειτουργούν στα 5 V και το καθένα μπορεί να παρέχει ή να δεχτεί μέχρι 40mA . Κάποια από τα δεκατέσσερα αυτά pin έχουν και δεύτερη λειτουργία:

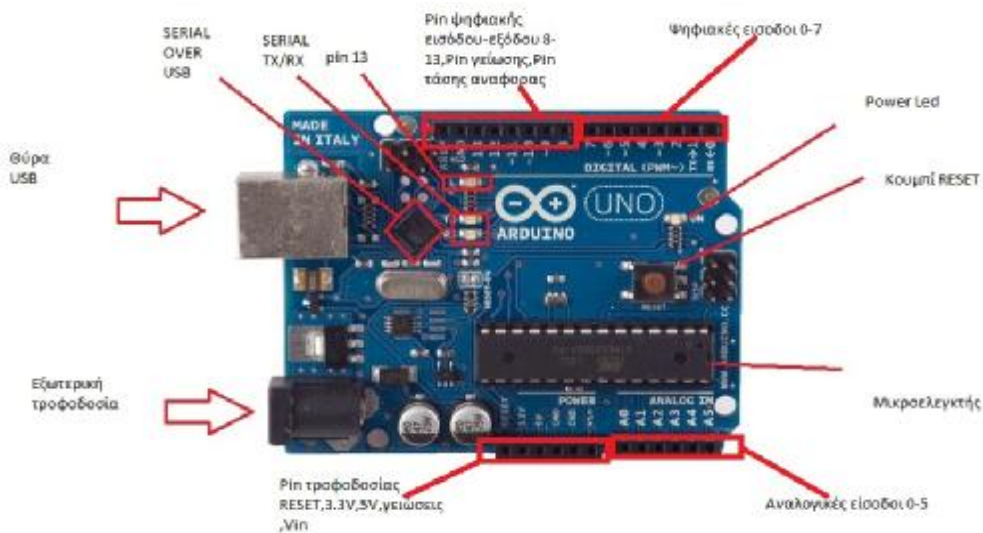
- PIN (0,1) Λειτουργούν ως RX ,TX της σειριακής όταν το πρόγραμμα ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγω χάρη το πρόγραμμα σας στέλνει δεδομένα, αυτά προωθούνται στην θύρα USB μέσω του ελεγκτή αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. ένα δεύτερο Arduino).
- (2,3) Εξωτερικά interrupt 0 και 1 αντίστοιχα
- PIN (5,6,9,10,11) Ψευδοαναλογικές έξοδοι με σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι, μπορεί να συνδεθεί ένα LED σε κάποιο από αυτά τα pin και να ελεγχθεί πλήρως η φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να υπάρχει απλά η δυνατότητα αναμμένο - σβηστό που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι. Στην απέναντι πλευρά με τη σήμανση ANALOG IN συναντάται μια ακόμα σειρά από 6 θηλυκά pin (0-5) που λειτουργούν σαν αναλογικές είσοδοι κάνοντας χρήση του ADC(analog to digital controller). Επίσης το καθένα από τα 6 pin με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin όπως της απέναντι πλευράς και να μετονομαστεί από (0-5) σε (14-19) αντίστοιχα.

### 2.1.3 Τροφοδοσία.

Η εξωτερική τροφοδοσία πρέπει να είναι μεταξύ 7-12 volt και μπορεί να προέρχεται από μια μπαταρία ή από ένα κοινό μετασχηματιστή DC που παρέχεται μέσω ενός φισ (2,1 mm) θετικός πόλος στο κέντρο «καρφίτσα». Αν και ο πιο απλός τρόπος είναι μέσω

της USB σύνδεσης. Δίπλα από τα 6 pin ANALOG IN υπάρχει μια τέταρτη συστοιχία πάλι των 6 pin με τη σήμανση POWER:

- RESET (επανεκκίνηση του Arduino στη γείωση)
- 3,3V (τροφοδοτεί με 3,3 Volt που προέρχονται από τον SERIAL-OVER-USB τα εξαρτήματα μας)
- 5V (τροφοδοτεί με 5V τα εξαρτήματα μας )
- 4 και 5 pins (γειώσεις)
- Vin (έχει διπλό ρόλο: σε συνδυασμό με την γείωση δίπλα του, λειτουργεί σαν μέθοδος τροφοδοσίας του Arduino .Αν όμως χρησιμοποιείται το φιν για τροφοδοσία, τότε αυτό λειτουργεί με πλήρη τάση, δηλαδή αυτή της εξωτερικής τροφοδοσίας χωρίς την επέμβαση του ρυθμιστή τάσης που θα την τοποθετούσε στα 5V)



Εικόνα 1: Αρχιτεκτονική Arduino.



#### 2.1.4 Κουμπιά και Led's.

- Κουμπί reset και led power με προφανή λειτουργία.
- Led's RX-TX όπου λειτουργούν ως ένδειξη λειτουργίας του σειριακού interface
- Led με σήμανση L (κυρίως λειτουργεί σαν ενσωματωμένο led που βρίσκεται στο breadboard. Έχοντας συνδέσει το pin 13 σε αυτό μπορεί να χρησιμοποιηθεί σαν led και δίνοντας του την τιμή High ανάβει.)

#### 2.1.5 Arduino IDE και σύνδεση με τον υπολογιστή.

Για την διαχείριση του Arduino από τον υπολογιστή μας υπάρχει το Arduino IDE το οποίο μπορεί να προμηθευτεί από το επίσημο site και υπάρχει σε διάφορες εκδόσεις για όλα τα λειτουργικά συστήματα.

Το Arduino IDE είναι βασισμένο σε Java και συγκεκριμένα παρέχει:

- [1] Ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων (τα οποία ονομάζονται sketch στην ορολογία του Arduino)
- [2] Αρκετά έτοιμα παραδείγματα
- [3] Μερικές έτοιμες βιβλιοθήκες για προέκταση της γλώσσας
- [4] Τον compiler για την μεταγλώττιση των sketch του χρήστη
- [5] Ένα serial monitor που παρακολουθεί τις επικοινωνίες της σειριακής θύρας (USB), αναλαμβάνει να στέλνει

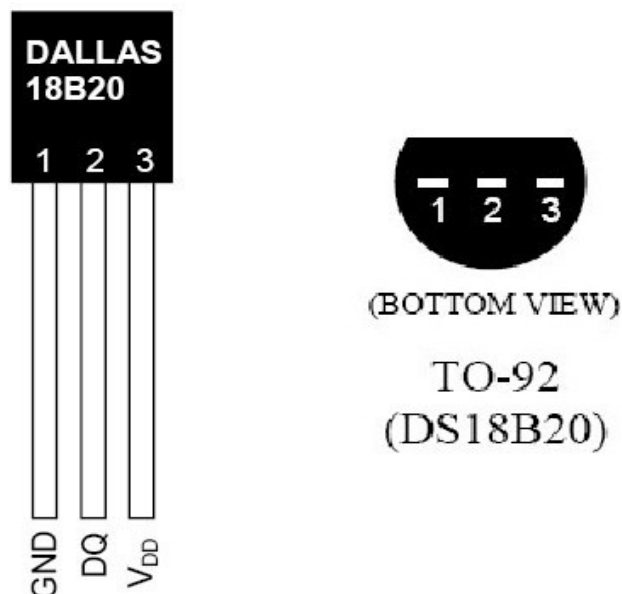
αλφαριθμητικά της επιλογής του χρήστη στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για το debugging των sketch

[6]Επιλογή να γίνει ανέβασμα του μεταγλωττισμένου sketch στο Arduino.

Για τα δύο τελευταία χαρακτηριστικά βέβαια, το Arduino πρέπει να έχει συνδεθεί σε μια από τις θύρες USB του υπολογιστή και λόγω του ελεγκτή Serial-over-USB, θα πρέπει να αναγνωριστεί από το λειτουργικό σύστημα ως εικονική σειριακή θύρα.

## 2.2 Αισθητήρας θερμοκρασίας DS18B20.

Τα θερμόμετρα DS18B20 κατασκευάζονται από την εταιρεία Maxim Integrated. Παρέχουν μετρήσεις θερμοκρασίας 9 ως 12 bit σε βαθμούς κελσίου και έχουν λειτουργία alarm με προγραμματιζόμενα μέγιστα/ελάχιστα trigger points. Έχουν τρεις ακροδέκτες, έναν για τάση, έναν για γείωση και έναν ακροδέκτη επικοινωνίας με τον μικροελεγκτή, που αποκαλείται συνήθως “One Wire”, λόγω της δυνατότητας μετάδοσης δεδομένων μέσω μόνο ενός καλωδίου.

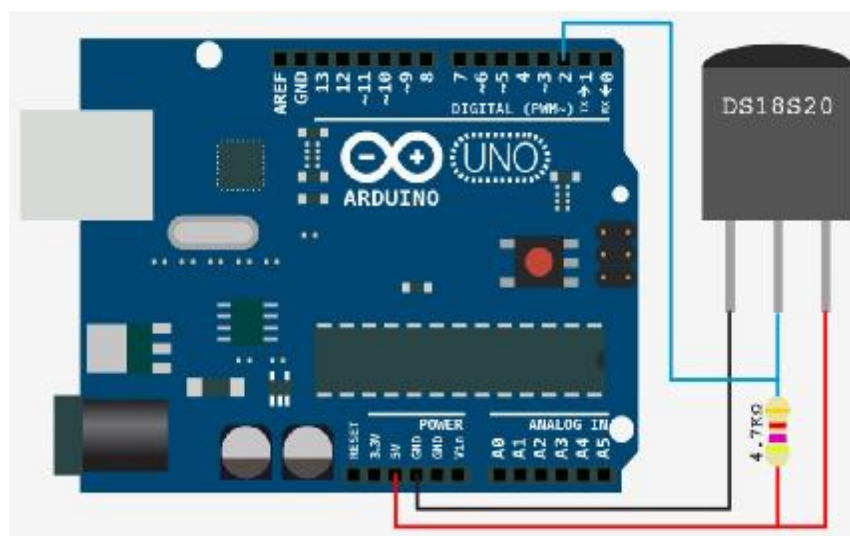


Εικόνα 2: Θερμόμετρο DS18B20.

Τα θερμόμετρα αυτά μπορούν να μετρήσουν θερμοκρασίες από  $-55^{\circ}\text{C}$  μέχρι  $125^{\circ}\text{C}$  με ακρίβεια  $\pm 0.5^{\circ}\text{C}$  τουλάχιστον στην κλίμακα από  $-10^{\circ}\text{C}$  ως  $85^{\circ}\text{C}$ . Ένα πολύ σημαντικό χαρακτηριστικό τους είναι ότι μπορούν να τροφοδοτηθούν ακόμα και από τον ακροδέκτη επικοινωνίας, αποκλείοντας την ανάγκη για ακροδέκτη τροφοδοσίας. Κάθε θερμόμετρο έχει έναν ιδιαίτερο σειριακό κωδικό 64ων bit, που επιτρέπει την πολλαπλή λειτουργία θερμομέτρων μέσω ενός μόνο διαύλου.

### 2.3 Επικοινωνία του αισθητήρα με τον Arduino.

Η σύνδεση του θερμομέτρου στον μικροελεγκτή γίνεται συνδέοντας το One-Wire (τον ακροδέκτη δεδομένων DQ του θερμομέτρου) σε ένα από τα digital pins. Για τη λειτουργία του χρειάζεται η σύνδεση του ακροδέκτη One-Wire με τον ακροδέκτη τροφοδοσίας μέσω ενός pull-up resistor  $4.7\text{k}\Omega$ . Παρακάτω φαίνεται ένα σχηματικό από το site bildr.org.



Εικόνα 3: Σύνδεση Arduino & DS18B20.

## 2.4 Γλώσσα προγραμματισμού

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

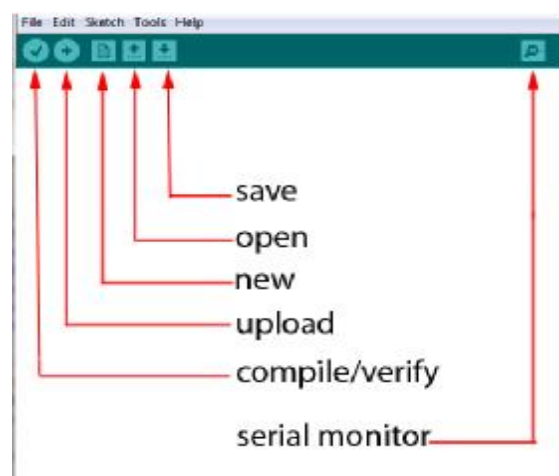
Πέρα από τις βασικές εντολές και συναρτήσεις που ήδη είναι γνωστές από τη C, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για την διαχείριση του ειδικού hardware του.

## 3 Software

---

### 3.1 Λογισμικό Arduino.

Το IDE του Arduino είναι γραμμένο σε Java και μπορεί να τρέξει σε πολλαπλές πλατφόρμες.



Εικόνα 4: IDE Arduino.

Περιλαμβάνει επεξεργαστή κώδικα και μεταγλωττιστή και έχει την ικανότητα να φορτώνει εύκολα το πρόγραμμα μέσω σειριακής θύρας από τον υπολογιστή στην πλακέτα. Το περιβάλλον ανάπτυξης είναι βασισμένο στην Processing, ένα περιβάλλον ανάπτυξης σχεδιασμένο να εισαγάγει στον προγραμματισμό νέους χρήστες μη εξοικειωμένους με την ανάπτυξη λογισμικού.

### 3.1.1 Εγκατάσταση λογισμικού Arduino

Ο χρήστης επισκέπτεται την επίσημη σελίδα [www.arduino.cc](http://www.arduino.cc), επιλέγει το λειτουργικό σύστημα που διαθέτει, στην συνέχεια κατεβάζει την αντίστοιχη έκδοση λογισμικού και την εγκαθιστά.

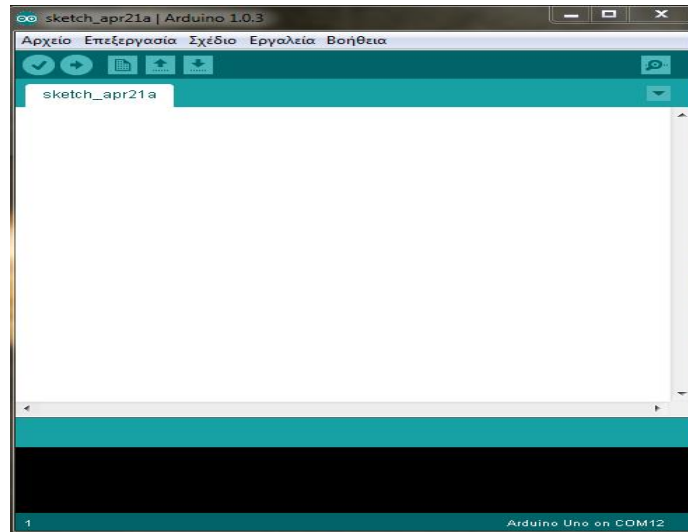
#### Download

Arduino 1.0.4 ([release notes](#)), hosted by [Google Code](#):

- + [Windows](#)
- + [Mac OS X](#)
- + [Linux: 32 bit, 64 bit](#)
- + [source](#)

Εικόνα 5: Λειτουργικά για Arduino.

Ανοίγοντας από την επιφάνεια εργασίας το αρχείο με όνομα `arduino.exe` ανοίγει το πρόγραμμα `arduino`:



Εικόνα 6: Arduino IDE Sketch.

Στην συνέχεια ο χρήστης γράφει το κώδικα που επιθυμεί και το αποθηκεύει. Συνδέοντας τον Arduino στον υπολογιστή, επιλέγει από το πρόγραμμα την πόρτα (COM) όπου έγινε η σύνδεση και εκτελείται η επιλογή Φόρτωση για να εισαχθεί ο κώδικάς στο Arduino. Όταν ολοκληρωθεί η διαδικασία φόρτωσης και εφόσον ο χρήστης δεν έχει λάβει κάποιο μήνυμα λάθους στον κώδικα, ανοίγει την σειριακή οθόνη που διαθέτει το πρόγραμμα και λαμβάνει τα αποτελέσματα του προγράμματος του.

### 3.2 Εγκατάσταση του Joomla.



Εικόνα 7: Joomla.

Για την υλοποίηση της θα χρειαστεί να εγκατασταθεί αρχικά ένα πακέτο προγραμμάτων, το οποίο περιέχει τον εξυπηρετητή ιστοσελίδων http Apache, βάση δεδομένων MySQL και έναν διερμηνέα για κώδικα, γραμμένο σε γλώσσες προγραμματισμού PHP και Perl.

Στην συγκεκριμένη περίπτωση έχει επιλεγεί το XAMPP, ένα ελεύθερο λογισμικό το οποίο περιέχει έναν εξυπηρετητή ιστοσελίδων ο οποίος μπορεί να εξυπηρετεί και δυναμικές ιστοσελίδες τεχνολογίας PHP/MySQL. Το XAMPP είναι ανεξάρτητο πλατφόρμας και τρέχει σε Microsoft Windows, Linux, Solaris, and Mac OS X και χρησιμοποιείται ως πλατφόρμα για την σχεδίαση και ανάπτυξη ιστοσελίδων με τεχνολογίες όπως PHP, JSP και Servlets.

- **Apache Web Server:** Πρόκειται για έναν εξυπηρετητή (server) του παγκόσμιου Ιστού (Web).

Ο Apache εγκαθίσταται σε έναν υπολογιστή ο οποίος μπορεί να χρησιμοποιεί διάφορα λειτουργικά συστήματα όπως Linux, Unix, Microsoft Windows, GNU, FreeBSD, Solaris, Novell NetWare, Mac OS X, OS/2, TPF. Ο ρόλος του Apache είναι να

αναμένει αιτήσεις από διάφορα προγράμματα – χρήστες όπως για παράδειγμα ένας φυλλομετρητής ενός χρήστη και στη συνέχεια να εξυπηρετεί αυτές τις αιτήσεις προσφέροντας τις σελίδες που ζητούν είτε απευθείας μέσω μιας ηλεκτρονικής διεύθυνσης, είτε μέσω ενός συνδέσμου. Ο τρόπος με τον οποίο ο Apache εξυπηρετεί αυτές τις αιτήσεις, είναι σύμφωνος με τα πρότυπα που ορίζει το πρωτόκολλο HTTP (Hypertext Transfer Protocol).

- **MySQL:** Είναι ένα πολύ γρήγορο και δυνατό σύστημα διαχείρισης βάσεων δεδομένων.

Μια βάση δεδομένων επιτρέπει την αποθήκευση, την αναζήτηση, την ταξινόμηση και την ανάκληση των δεδομένων αποτελεσματικά. Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα για να μπορούν να εργαστούν πολλοί χρήστες ταυτόχρονα και με ταχύτητα, διασφαλίζοντας ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση σε αυτά.

- **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 ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά GNU/Linux, Microsoft Windows, Mac OS X υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP, είτε με την χρήση ενός πρόσθετου (mod\_php) είτε με την αποστολή του κώδικα προς εκτέλεση σε εξωτερική διεργασία CGI ή FCGI είτε με την έλευση της php5.4 που υποστηρίζεται



η εκτέλεση σε πολυάσχολους ιστοχώρους, FastCGI Process Manager (FPM).

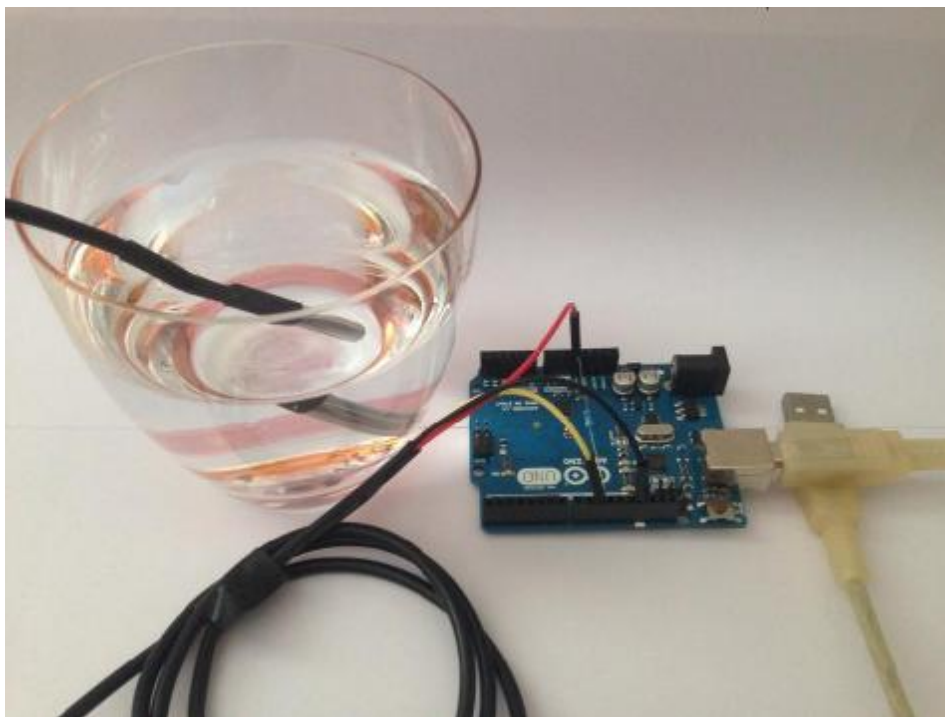
Μετά την εγκατάσταση εκτελείται από τον χρήστη το πρόγραμμα XAMPP control panel και επιλέγεται η ενεργοποίηση των Apache και Mysql. Στην συνέχεια εφόσον έχουν ενεργοποιηθεί οι παραπάνω επιλογές, μέσω φυλλομετρητή πληκτρολογείται η διεύθυνση <http://localhost/phpmyadmin> όπου κατευθύνει τον χρήστη στο περιβάλλον διαχείρισης των βάσεων δεδομένων. Ο χρήστης δημιουργεί μια βάση δεδομένων και δηλώνει κωδικοποίηση utf8\_unicode. Αμέσως μετά δημιουργείται ένας χρήστης, της βάσης που μόλις δημιουργήθηκε και ορίζονται σε αυτόν πλήρη δικαιώματα επάνω στην βάση.

Το δεύτερο βήμα για την εγκατάσταση, είναι να κατέβουν τα αρχεία εγκατάστασης του Joomla και τα αρχεία ελληνικής γλώσσας για το δημόσιο τμήμα και τη διαχείριση, από την επίσημη ελληνική ιστοσελίδα του Joomla. Στον φάκελο C:/xampp/htdocs που δημιούργησε η εγκατάσταση του XAMPP, δημιουργείται ένας υποφάκελος (π.χ. mysite) όπου εκεί δίνεται η δυνατότητα να αποσυμπιεστεί το αρχείο εγκατάστασης.

Ανοίγοντας τον φυλλομετρητή και πληκτρολογώντας την διεύθυνση: <http://localhost/mysite>, αυτομάτως εκτελείται η διαδικασία εγκατάστασης. Εφόσον ολοκληρωθεί η διαδικασία εγκατάστασης του Joomla, παρέχεται η δυνατότητα δημιουργίας της ιστοσελίδας ανάλογα με της απαιτήσεις και της ανάγκες του χρήστη.

## 4 Υλοποίηση Εφαρμογής.

---



Εικόνα 8: Υλοποιημένη εφαρμογή εργασίας.

Στο εικόνα απεικονίζεται η συνδεσμολογία των συσκευών. Πιο συγκεκριμένα ως αφετηρία είναι ο αισθητήρας ο οποίος βυθισμένος μέσα σε υγρό, ανάλογα με το ερέθισμα παράγει μια ψηφιακή τιμή όπου μέσα από το digital port που έχει ορίσει ο χρήστης κατά την αρχική συνδεσμολογία, στέλνεται από τον Arduino στην θύρα USB του υπολογιστή όπου ο χρήστης επιθυμεί να λαμβάνει της μετρήσεις. Οι μετρήσεις πιο συγκεκριμένα στέλνονται σειριακά στον υπολογιστή, μέσω του καταλλήλου προγραμματισμού διαβάζονται

απευθείας από την πόρτα όπου λαμβάνονται, τοποθετούνται στην βάση δεδομένων της ιστοσελίδας και τέλος απεικονίζονται σε αυτήν.

#### **4.1 Συνδεσμολογία.**

Για την υλοποίηση της εφαρμογής έχει χρησιμοποιηθεί ο εξής εξοπλισμός: Arduino Uno, ο αισθητήρας θερμοκρασίας καθώς και καλώδιο USB για την τροφοδοσία του Arduino και την λήψη των μετρήσεων.

Αρχικά συνδέεται ο Arduino με τροφοδοσία ρεύματος μέσω καλωδίου USB σε αντίστοιχη θύρα του εκάστοτε υπολογιστή όπου θα λαμβάνονται οι μετρήσεις. Εφόσον έχει φορτωθεί στον Arduino ο κώδικας της εφαρμογής, συνεχίζεται η συναρμολόγηση.

Στο Arduino συνδέεται ο αισθητήρας θερμοκρασίας στις ανάλογες πόρτες: 5 Volt το κόκκινο καλώδιο, GND (Γείωση) μαύρο καλώδιο και στην ψηφιακή πόρτα 10 το κίτρινο καλώδιο από όπου στέλνονται τα δεδομένα από τις μεταβολές της θερμοκρασίας του νερού από τον αισθητήρα.

#### **4.2 Μέτρηση θερμοκρασίας με θερμόμετρο DS18B20.**

Για να επιτευχθεί μέτρηση θερμοκρασίας με την χρήση θερμόμετρου DS18B20, είναι απαραίτητη η χρήση της βιβλιοθήκης OneWire. Η βιβλιοθήκη αυτή έχει βασιστεί σε sample code που παρέχεται από την Dallas Semiconductors. Το κάθε θερμόμετρο έχει μοναδικό σειριακό κωδικό 64bit, που καθιστά δυνατή τη σύνδεση πολλαπλών θερμομέτρων σε έναν μόνο δίαυλο. Σε κάθε επανάληψη του κώδικα, καλείται μια διεργασία η οποία απαριθμεί τα συνδεδεμένα στον “one wire” δίαυλο καλώδια. Η μέτρηση των θερμοκρασιών γίνεται με τη χρήση 9-12 bytes. Η αναγνώριση σφαλμάτων γίνεται μέσω ενός 8 bit CRC. Είναι δυνατόν να χρησιμοποιηθεί και 16bit CRC.

Ο κώδικας που χρησιμοποιείται για τη μέτρηση των θερμοκρασιών είναι μια function με το όνομα `getTemp`, η οποία επιστρέφει τη θερμοκρασία που στέλνει ένα θερμόμετρο σε βαθμούς Κελσίου.

Αρχικά καλείται η function “search”, η οποία ψάχνει για ένα συνδεδεμένο θερμόμετρο και επιστρέφει τη διεύθυνση του θερμομέτρου που θα βρεί. Αν δε βρει θερμόμετρο ή το CRC είναι λανθασμένο ή είναι συνδεδεμένη άλλη συσκευή (το 0x28 αποτελεί το family code των DS18B20 1-wire, ενώ το 0x10 αποτελεί το family code των απλών DS18B20), επιστρέφει την τιμή “-1000”. Αν βρει θερμόμετρο συνδεδεμένο, ανοίγει επικοινωνία μαζί του και του ζητά τη θερμοκρασία.

Συγκεκριμένα στέλνει την εντολή “Convert T” (0x44), η οποία κάνει το DS18B20 να ξεκινήσει τη μετατροπή θερμοκρασίας και να στείλει στον μικροελεγκτή το conversion status. Στη συνέχεια, με την εντολή “Read Scratchpad” (0xBE), το θερμόμετρο αποστέλλει μέχρι 9 bytes δεδομένων, συμπεριλαμβανομένου του CRC byte. Τα bytes αυτά αποθηκεύονται σε έναν πίνακα `data[]`. Τέλος, τα δεδομένα αυτά μετατρέπονται σε float αριθμό και τα επιστρέφονται. Στον παρακάτω κώδικα, το όρισμα `ds` αποτελεί ένα instance του OneWire.

#### 4.2.1 Κώδικας.

```
#include <OneWire.h>

int DS18S20_Pin = 10; //DS18S20 Signal pin on digital 2

//Temperature chip i/o

OneWire ds(DS18S20_Pin); // on digital pin 2

void setup(void) {

  Serial.begin(9600);

}

void loop(void) {

  float temperature = getTemp();

  Serial.println(temperature);

  delay(1000); //just here to slow down the output so it is easier to read
```

```
}  
  
float getTemp(){  
    //returns the temperature from one DS18S20 in DEG Celsius  
  
    byte data[12];  
    byte addr[8];  
  
    if ( !ds.search(addr) ) {  
        //no more sensors on chain, reset search  
  
        ds.reset_search();  
  
        return -1000;  
    }  
  
    if ( OneWire::crc8( addr, 7) != addr[7] ) {  
        Serial.println("CRC is not valid!");  
  
        return -1000;  
    }  
  
    if ( addr[0] != 0x10 && addr[0] != 0x28 ) {  
        Serial.print("Device is not recognized");  
  
        return -1000;  
    }  
  
    ds.reset();  
  
    ds.select(addr);  
  
    ds.write(0x44,1); // start conversion, with parasite power on at the end  
  
    byte present = ds.reset();  
  
    ds.select(addr);  
  
    ds.write(0xBE); // Read Scratchpad  
  
    for (int i = 0; i < 9; i++) { // we need 9 bytes  
  
        data[i] = ds.read();  
  
    }  
}
```

Χρήση Arduino για την διαδικτυακή παρακολούθηση θερμοκρασίας νερού.  
Ζαρούλιας Βασίλειος

```
ds.reset_search();  
  
byte MSB = data[1];  
  
byte LSB = data[0];  
  
float tempRead = ((MSB << 8) | LSB); //using two's compliment  
  
float TemperatureSum = tempRead / 16;  
  
return TemperatureSum;  
  
}
```

## 4.3 Δημιουργία ιστοσελίδας μέσω Joomla.

### 4.3.1 Τι είναι το Joomla.

Το Joomla είναι ένα ανοιχτού κώδικα σύστημα διαχείρισης περιεχομένου το οποίο έχει χαρακτηριστεί από τους δεκάδες χιλιάδες χρήστες του ως το καλύτερο CMS στον κόσμο διότι έχει τεράστιες δυνατότητες αλλά και ταυτόχρονα εξαιρετικά ευέλικτο και φιλικό περιβάλλον εργασίας και διαχείρισης.

Μόλις εγκατασταθεί, ο διαχειριστής του δικτυακού τόπου μπορεί να συνδεθεί με την κονσόλα διαχείρισης και να δημιουργήσει τμήματα, κατηγορίες, άρθρα με περιεχόμενο, δημοσκοπήσεις και άλλα πολλά.



Εικόνα 9: Σελίδα διαχείρισης Joomla.

Όταν δημιουργείται ένα νέο περιεχόμενο, ένας **WYSIWYG (What You See Is What You Get)** επεξεργαστής κειμένου επιτρέπει απλές αλλαγές χωρίς τη γνώση της **HTML**.

Παρόλο ότι δεν είναι απαραίτητο ο χρήστης να έχει γνώση της HTML και CSS, συνιστάται να αρχίσει τη μελέτη των βασικών εντολών της HTML και CSS για την περίπτωση που θα πρέπει να επεξεργαστεί ή να τροποποιήσει κάποιο στοιχείο ώστε να το προσαρμόσει στις ανάγκες που θα δημιουργηθούν. Στις κοινότητες φόρουμ για Joomla είναι διαθέσιμοι χιλιάδες προγραμματιστές και σχεδιαστές για να βοηθήσουν νέους και έμπειρους χρήστες, εφόσον είναι δυνατό.

**Το Joomla είναι μία εφαρμογή,** ανοιχτού κώδικα σύστημα διαχείρισης περιεχομένου, το οποίο έχει συνταχθεί στην γλώσσα προγραμματισμού PHP για τη διαχείριση και τη δημοσίευση περιεχομένου στον ιστό (web) και ενδοδικτύων, (intranets) χρησιμοποιώντας μια MySQL βάση δεδομένων.

Το Joomla περιλαμβάνει χαρακτηριστικά όπως η cache των σελίδων για τη βελτίωση των επιδόσεων, Web ευρετηρίασης (indexing), RSS feeds, εκτυπώσιμες (printable) εκδόσεις των σελίδων, newsflashes, blogs, forums, δημοσκοπήσεις, ημερολόγια, αναζήτηση στην ιστοσελίδα, την ανάλογη γλώσσα άλλα και την δυνατότητα πολυγλωσσικών ιστοσελίδων.

Έχοντας αναλύσει στο κεφάλαιο τρία (3) τον τρόπο εγκατάστασης του Joomla και εφόσον αυτή έχει υλοποιηθεί, μέσω της σελίδας διαχείρισης ο χρήστης μπορεί να ξεκινήσει σιγά σιγά την διαμόρφωση της ιστοσελίδας ανάλογα με τις απαιτήσεις και τις ανάγκες του.

Στην συγκεκριμένη εργασία, η ιστοσελίδα όπου υλοποιήθηκε με την ονομασία «BilZar» έχει δημιουργηθεί μενού τεσσάρων επιλογών. Πιο συγκεκριμένα υπάγονται οι επιλογές:



Αρχική σελίδα: Όπου προβάλλονται εικόνες από τον εξοπλισμό που χρησιμοποιήθηκε για την υλοποίηση της πτυχιακής εργασίας, καθώς και άρθρο όπου αναφέρεται το θέμα και ο σκοπός της εργασίας.

Τρέχον μέτρηση: Όπου προβάλλεται η πιο πρόσφατη μέτρηση pH καθώς η ημερομηνία και η ώρα που ελήφθη η μέτρηση. Επίσης υπάρχει επιλογή όπου ο χρήστης μπορεί να λάβει μέτρηση την στιγμή που επιθυμεί.

Παλαιότερες μετρήσεις: Όπου προβάλλονται από την βάση δεδομένων όλες οι μετρήσεις που έχουν ληφθεί στο παρελθόν.

Σχετικά με εμάς (About us): Όπου αναφέρονται αναλυτικά τα στοιχεία του δημιουργού καθώς και ο τρόπος επικοινωνίας μαζί του.

#### **4.3.2 Δημιουργία Component.**

Το Component είναι εφαρμογή ή ομάδα εφαρμογών αντίστοιχα, όπου παρέχουν στον χρήστη την δυνατότητα της επέκτασης δυνατοτήτων του Joomla.

Το Component που δημιουργήθηκε για την ιστοσελίδα της εφαρμογής, επικεντρώνεται αποκλειστικά στο M.V.C. ( Model - View - Controller ), ως αρχιτεκτονική δημιουργίας του κώδικα.

MVC:

- **Model** : Κώδικας που σχετίζεται με την βάση δεδομένων (διαχείριση των Create-Read-Update-Delete).
- **View**: Κώδικας που σχετίζεται με την δημιουργία διάφορων views (layouts - διαφορετικές απεικονίσεις των δεδομένων).
- **Controller**: Κώδικας που λαμβάνει εντολές και δρομολογεί την ροή του λογισμικού, συνδέοντας τα models και τα views.

Το Component που δημιουργήθηκε, διαχειρίζεται κατηγοριοποιημένα μηνύματα μέτρησης θερμοκρασίας όπου μετά την λήψη, η μέτρηση τοποθετείται στην βάση δεδομένων που αποσκοπεί για αυτόν τον σκοπό.

Όνομα Component: **Com\_black**

Κάθε component δημιουργείται σε έναν ανεξάρτητο φάκελο εκτός Joomla και όταν ολοκληρωθεί η δημιουργία του, τα περιεχόμενά του συμπιέζονται σ' ένα zip για να δημιουργηθεί το εκτελέσιμο αρχείο (installation archive.).

Στην συνέχεια ο χρήστης δημιουργεί ένα φάκελο " **Com\_black** " και εκεί μέσα έναν υποφάκελο "admin" αφού θα αναπτύξει το backend τμήμα του component, δηλαδή το τμήμα που αναλαμβάνει την επικοινωνία ανάμεσα στην βάση δεδομένων και στην διεπαφή χρήστη.

### 4.3.3 Blank.php.

Το εκκινήτήριο αρχείο πρέπει να έχει το όνομα του component.  
Δημιουργούμε το Blank.php

Κώδικας:

```
<?php

// Valid execution check

defined('_JEXEC') or die('Restricted Access');

// Import Joomla Controller Library

jimport( 'joomla.application.component.controller' );

// Include dependancies

require_once JPATH_COMPONENT.'/helpers/php_serial.class.php';

$controller = JController::getInstance( "Blank" );

$controller->execute( JRequest::getCmd('task') );

$controller->redirect();
```

Το εκκινήτήριο αρχείο ενεργοποιεί τον γενικό controller του component, τον εκτελεί με παράμετρο την τιμή της μεταβλητής \$\_REQUEST['task'] και αν έχει οριστεί, εκτελεί το σχετικό redirect σε άλλη σελίδα. Τον γενικό controller θα τον αναζητήσει η εφαρμογή μέσα στο αρχείο controller.php.

#### 4.3.4 Δημιουργία controller.php.

Κώδικας:

```
<?php

// Valid execution check

defined('_JEXEC') or die('Restricted Access');

// Import Joomla Controller Library

jimport( 'joomla.application.component.controller' );

ini_set("display_errors", false);

class BlankController extends JController {

    protected $db;

    public function display($cachable = false, $urlparams = false)
    {

        parent::display($cachable, $urlparams);

    }

    public function sense() {

        date_default_timezone_set("Europe/Athens");

        $this->db = JFactory::getDBO();

        // Command Line to read port from arduino

        $device = "COM27";
```

```
$baud = 9600;

$command = 'powershell "$port= new-Object
System.IO.Ports.SerialPort ".$device.', '.$baud.',None,8,one;
$port.open(); $port.ReadLine(); $port.close()";

$output = shell_exec($command);

if(!$output) {

    $output = shell_exec($command);

}

$timestamp = time(); // Get current system time

$data = array('Μέτρηση ' => $output, 'Ώρα - Ωρα' =>
$timestamp);

// save it

$q = $this->db->getQuery(TRUE);

$q->insert("#__arduino");

$q->set("measurement='{ $output }");

$q->set("timestamp='".date("Y-m-d H:i:s",
$timestamp)."'");

$this->db->setQuery($q);

$result = $this->db->loadResult();

print_r($result);

$data = array('Μέτρηση ' => $output, 'Ώρα - Ωρα' =>
date("d-m-Y H:i:s", $timestamp));

echo json_encode($data);

parent::display($cachable, $urlparams);
```

}

}

Ο γενικός controller όταν πάρει εντολή από το προηγούμενο αρχείο να εκτελεστεί, θα καλέσει αυτόματα την function sense().

Η function αυτή ορίζει την ζώνη ώρας όπου θα αναγράφεται και στις μετρήσεις που θα τοποθετούνται στην βάση δεδομένων. Η επόμενη ενέργεια της είναι να οριστεί η πόρτα (com) όπου θα εισέρχεται η μέτρηση θερμοκρασίας από τον Arduino στον σταθμό (υπολογιστή) καθώς επίσης και η δομή της βασικότερης εντολής, όπου θα διαβάζεται η μέτρηση από την πόρτα του σταθμού (υπολογιστή) μέσω επεξεργαστή εντολών windows (command line).

Στην συνέχεια εκτελείται εντολή λήψης της τρέχουσας ώρας του συστήματος ώστε να αποδίδεται σε κάθε μέτρηση που λαμβάνεται από τον σταθμό και η ακριβής ημερομηνία και ώρα λήψης.

Τέλος εκτελείται η αποθήκευση της λήψης μετρήσεων θερμοκρασίας καθώς και η ημερομηνία και ώρα στην βάση δεδομένων που έχει οριστεί για αυτόν τον σκοπό.

Στην συνέχεια υλοποιείται η ανάγνωση των δεδομένων όπου έχουν καταχωρηθεί στην βάση δεδομένων και με ποιον τρόπο θα απεικονίζονται από αισθητικής πλευράς στην ιστοσελίδα που εφαρμόζεται το Component.

Όπως αναφέρθηκε στο ξεκίνημα του υποκεφαλαίου, σε ότι αφορά τα δεδομένα είναι υπεύθυνο το Model. Πρέπει να οριστεί το Model το οποίο θα επιστρέφει τα δεδομένα στο συγκεκριμένο View που τα ζητάει.

Τα Models στο Joomla αποθηκεύονται σε έναν φάκελο με όνομα "models". Είναι απαραίτητο να υπάρχει ένα model για κάθε view και

το αρχείο που θα το περιέχει πρέπει να είναι ίδιο με το όνομα του View. Εκτελείται λοιπόν η δημιουργία του helloworldlist.html μέσα στον φάκελο "models"

#### 4.3.5 Κώδικας:

```
<?php

// Valid execution check

defined('_JEXEC') or die('Restricted Access');

jimport( "joomla.application.component.modellist" );

class BlankModelDefault extends JModelList {

    public function getItems() {

        $items = parent::getItems();

        return $items;

    }

    public function getListQuery() {

        $q = parent::getListQuery();

        $q->select("*");

        $q->from("#__arduino");

        $q->order('timestamp DESC');

        return $q;

    }

}
```

```
protected function populateState($ordering = null, $direction =
null) {

    parent::populateState($ordering, $direction);

}

}

/*

// Import Joomla Models Library

jimport( 'joomla.application.component.model' );

class BlankModelDefault extends JModel {

    protected $db;

    function __construct() {

        parent::__construct();

        $this->db = JFactory::getDBO();

    }

    function getData() {

        $data = array();

        $q = $this->db->getQuery(TRUE);

        $q->select("*");

        $q->from("#__arduino");

        $q->order('timestamp DESC');

        $this->db->setQuery($q);

        $data = $this->db->loadObjectList();

    }

}
```



```
return $data;  
  
    }  
  
}  
  
*/
```

Όταν το view ζητήσει τα δεδομένα από το model του με `get('Items')`, θα εκτελεστεί αυτόματα η function `getListQuery()` της class `BlankModelDefault` που κληρονομεί τα στοιχεία της `JModelList` class του Joomla. Το όνομα της class αποτελείται πάντα από το όνομα του component ("Blank") + το keyword "Model" + το όνομα του συγκεκριμένου model ("Default").

Η `getListQuery()` στην περίπτωση αυτή, θα εκτελέσει το query `"select * from #__arduino"` και θα επιστρέψει τα αποτελέσματα στο view που έκανε το request.

Όταν το view ζητήσει τα δεδομένα από το model του με `get('Data')`, θα εκτελεστεί αυτόματα η function `getData()` της class `BlankModelDefault` που κληρονομεί τα στοιχεία της `JModel` class του Joomla. Το όνομα της class αποτελείται από το όνομα του component ("Blank") + το keyword "Model". Η `getData()` ελέγχει αν έχει εκτελεστεί η function `getListQuery()` και αν όντως έχει εκτελεστεί τότε εκτελείται το `"loadObjectList()"` όπου με αυτό τον τρόπο το όνομα της κάθε στήλης του πίνακα θα μετατραπεί σε αντικείμενο

Το `#__` είναι αναφορά στο εκάστοτε ισχύον πρόθεμα των πινάκων του Joomla (συνήθως `"jos_"`). Τον πίνακα αυτόν θα τον δημιουργήσει αργότερα το installation package κατά την εγκατάσταση του extension.

#### 4.3.6 Δημιουργία του εκτελέσιμου αρχείου (installation package).

Για την δημιουργία του συμπιεσμένου εκτελέσιμου αρχείου (installation package του component) εκτελούνται τρία βασικά βήματα:

- Δημιουργία Language Definition όπου αναφέρεται η μετάφραση των κωδικοποιημένων αναφορών που χρησιμοποιήθηκε με την JText\_().
- Δημιουργία των SQL αρχείων που θα διαχειριστούν την εγκατάσταση και την απεγκατάσταση του component.
- Τέλος το manifest XML το οποίο θα περιέχει τις κατάλληλες οδηγίες στο Joomla για την σωστή εγκατάσταση.

#### 4.3.7 Δημιουργία αρχείου γλώσσας μετάφρασης.

Ο χρήστης δημιουργεί για την γλώσσα, τον φάκελο "language" μέσα στον φάκελο "admin", και εκεί μέσα έναν υποφάκελο "en-GB". Ως περιεχόμενα θα δημιουργηθεί το αρχείο "en-GB.com\_helloworld.ini" με τα ακόλουθα περιεχόμενα:

COM\_BLANK\_ADMINISTRATION=Blank administration

COM\_BLANK\_CATEGORIES=Categories

COM\_BLANK\_MANAGER=Blank manager

COM\_BLANK\_MESSAGES=Messages

COM\_BLANK\_BLANK\_CATEGORY=category

COM\_BLANK\_BLANK\_DETAILS=Details

COM\_BLANK\_BLANK\_EDITING=Editing

COM\_BLANK\_BLANK\_ERROR\_SOME\_VALUES\_ARE\_UNACCEPTABLE=Same value are unacceptable

COM\_BLANK\_BLANK\_MEASUREMENT\_DESC=Message to be displayed

COM\_BLANK\_BLANK\_MEASUREMENT = Measurement

COM\_BLANK\_BLANKLIST\_ARE\_YOU\_SURE\_YOU\_WANT\_TO\_DELETE\_THESE\_MEASUREMENT =Are you sure you want to delete these measurement?

COM\_BLANK\_BLANKLIST\_MEASUREMENT = Measurement

COM\_BLANK\_BLANKLIST\_MEASUREMENT\_REMOVED=Measurement removed.

COM\_BLANK\_BLANKLIST\_ID=Id

COM\_BLANK\_BLANKLIST\_ONE\_OR\_MORE\_MEASUREMENT\_COULD\_NOT\_BE\_DELETED=One or more measurement could not be deleted: %s

COM\_BLANK\_BLANK\_VIEW\_DEFAULT\_MESSAGE\_DESC=This message will be displayed

COM\_BLANK\_BLANK\_VIEW\_DEFAULT\_MESSAGE\_LABEL=Message

COM\_BLANK\_BLANK\_EDIT\_CANCELLED=Edit cancelled

COM\_BLANK\_BLANK\_GREETING\_SAVED= Measurement save

COM\_BLANK\_BLANK\_ERROR\_SAVING\_GREETING=Error saving record

Διαπιστώνεται πως αν και μέσα στα αρχεία, οι αναφορές στην γλώσσα ήταν με πεζούς χαρακτήρες (πχ `com_blank_categories`), αντιθέτως σε αυτή την διαδικασία πρέπει να είναι με κεφαλαία. Επίσης απαγορεύονται τα κενά οπότε αντικαθίστανται με κάτω παύλα (`_`).

Το Joomla κατά την εκτέλεση, θα αντικαταστήσει όλες τις αναφορές με τις αντίστοιχες μεταφράσεις που θα βρει σ' αυτό το αρχείο. Το όνομα του αρχείου αποτελείται πάντα από τα ISO αρχικά της γλώσσας, μια παύλα, το όνομα του component με πρόθεμα το `"com_"` και την επέκταση `".ini"`. Για δημιουργία αντίστοιχων ελληνικών μεταφράσεων, το αρχείο θα ήταν το `"el GR.com_helloworld.ini"` μέσα στον φάκελο `"el-GR"`.

## 5 Προβλήματα & Τρόποι αντιμετώπισης.

Ο στόχος της πτυχιακής αυτής εργασίας ήταν να επιτευχθεί η περισυλλογή δεδομένων θερμοκρασίας υδάτων και κατά την συλλογή να πραγματοποιηθεί αποστολή αυτών σε υπολογιστή και να απεικονιστούν σε ιστοσελίδα όπου έχει δημιουργηθεί για τον σκοπό αυτό .

Το πρώτο διπλό πρόβλημα που παρουσιάστηκε στο ξεκίνημα της υλοποίησης ήταν αυτό της επιλογής του κατάλληλου στοιχείου του νερού που θα έχριζε σημασίας , ρόλου και λειτουργικότητας, ενώ παράλληλα έπρεπε να ήταν και το κατάλληλο για να μπορέσει να προσαρμοστεί και ο εξοπλισμός. Κατόπιν εκτενούς έρευνας επιλέχθηκε η θερμοκρασία του νερού καθώς και η δυνατότητα παρατήρησης της διακύμανσης της όπου είναι χρήσιμη σε χιλιάδες δραστηριότητες σε παρά πολλούς τομείς. Υπάρχουν κατάλληλα αισθητήρια όργανα στο εμπόριο με μεγάλο εύρος τιμής ανάλογα με την ποιότητα των υλικών, την αξιοπιστία και την ακρίβεια που μπορεί να αποδώσει. Έτσι επιδέχθηκε και το πρώτο κομμάτι του εξοπλισμού.

Το επόμενο βήμα ήταν η επιλογή του κατάλληλου μικροεπεξεργαστή- μικροελεγκτή ώστε να μπορεί να επικοινωνεί με τον αισθητήρα. Η επιλογή του Arduino της Atmel ήταν η καταλληλότερη λόγω του τρόπου προγραμματισμού του (Wiring παραλλαγή C/C++) αλλά και η κατάλληλη δομή του hardware του που υποστηρίζει τη σύνδεση του αισθητήρα και την επικοινωνία με τον σταθμό εργασίας όπου θα στέλνονται η μετρήσεις.

Το κυριότερο πρόβλημα που έπρεπε να αντιμετωπισθεί ήταν αυτό της αποστολής των δεδομένων από Serial monitor του Arduino σε ιστοσελίδα ώστε να έχουν πρόσβαση και απομακρυσμένοι χρήστες . Κατά τη διάρκεια αυτής της διαδικασίας πρόεκυψαν πολλά ερωτήματα, όπως:

- Πως λειτουργεί ο αισθητήρας
- Τι παράγει
- Τι τροφοδοσία χρειάζεται για τη λειτουργία του
- Πως θα συνδεθεί με τον μικροεπεξεργαστή / μικροελεγκτή

Ολοκληρώνοντας την διαδικασία συναρμολόγησης και ξεπερνώντας τα προβλήματα που αναφέρθηκαν, το αμέσως επόμενο πρόβλημα που παρουσιάστηκε και ίσως και το βασικότερο της εργασίας αυτής ήταν η ανακάλυψη του τρόπου προβολής των δεδομένων που λάμβανε ο υπολογιστής. Λόγω του ότι το λογισμικό των συσκευών που χρησιμοποιήθηκαν δεν παρήγαγαν κάποιο εξαγωγή αρχείο με της μέτρησης που να είναι συμβατό με την ιστοσελίδα και να ενημερώνεται σε πραγματικό χρόνο, έπρεπε να ανακαλυφθεί τρόπος όπου τα δεδομένα να διαβάζονται απευθείας από την θύρα (Com) που λαμβάνονται, να τοποθετούνται σε μια βάση δεδομένων και να εκτελείται η απεικόνιση μέσω αυτής. Μετά από αρκετές αποτυχημένες δοκιμές και πειραματισμούς, μη έχοντας το επιθυμητό αποτέλεσμα, την λύση έφερε η κατασκευή εντολής όπου διαβάζει μέσω της γραμμής εντολών (terminal) των Windows την τιμή που είχε λάβει η προκαθορισμένη πόρτα.

Η εντολή: « powershell "\$port= new-Object System.IO.Ports.SerialPort COM(Αριθμός πόρτας),9600,None,8,one; \$port.open(); \$port.ReadLine(); » που προσαρμόστηκε μέσα στο Component της ιστοσελίδας, επίλυσε το πρόβλημα προγραμματίζοντας ο χρήστης την επαναλαμβανόμενη εκτέλεση της ανά συγκεκριμένο χρονικό διάστημα και την ανανέωση της ιστοσελίδας αυτόματα για να πραγματοποιείται η απεικόνιση της πρόσφατης μέτρησης.

## 6 Βελτιστοποιήσεις.

---

### 6.1 Υλικό (Hardware).

Μελλοντικές βελτιώσεις της εφαρμογής προκύπτουν με βάση τα παραπάνω που αναλύθηκαν. Η συσκευή θα μπορούσε να ήταν πιο αποδοτική και πιο ακριβής με κάποιες μετατροπές, προσθήκες ή αλλαγές.

Το πρώτο πράγμα που θα έκανε τη συσκευή αυτόνομη θα ήταν η αλλαγή του τρόπου τροφοδοσίας. Στο συγκεκριμένο project λόγω κόστους ακολουθήθηκε ο εύκολος τρόπος με καλώδιο τύπου USB συνδεδεμένο σε υπολογιστή. Αν δεν είχε επιλεγθεί ο συγκεκριμένος τρόπος τροφοδοσία της πλακέτα θα μπορούσε να χρησιμοποιηθεί μια μπαταρία ξηρού τύπου στα υπερορία που επαρκούν για την λειτουργία των συσκευών (περίπου 10-20 mA) άλλα και μέσα στα όρια τάσης τροφοδοσίας που χρειάζεται ο Arduino για να δουλέψει ομαλά (7-12V). Κατά αυτό το τρόπο ο χρήστης θα είχε στην διάθεση του μια τελείως αυτόνομη συσκευή όπου με ορισμένες επιπλέον προσθήκες (Xbee, buetooth, HiFi) για την επίτευξη ασύρματης αποστολής των μετρήσεων χωρίς να χρειάζεται να είναι συνδεδεμένη με υπολογιστή άλλα ούτε και να είναι συνδεδεμένη στο οικιακό ηλεκτρικό δίκτυο με μετασχηματιστή στο power junk. Επίσης άλλη μια πολύ καλή λύση για τη τροφοδοσία του μικροεπεξεργαστή θα ήταν η χρήση φωτοβολταϊκού συστήματος που παρέχει πολλά πλεονεκτήματα σε σχέση με τις άλλες επιλογές, όπως:



- § Αντιρρυπαντική λειτουργία
- § Αθόρυβη λειτουργία
- § Η ηλιακή ενεργεία είναι άφθονη και ελεύθερη
- § Μεγάλη διάρκεια ζωής
- § Μηδαμινή συντήρηση.

## 6.2 Λογισμικό (Software).

Σε ότι αφορά το λογισμικό μέρος των συσκευών που χρησιμοποιήθηκε κατά την εκπόνηση την πτυχιακής εργασίας, ήταν για την κάθε συσκευή η τρέχον διαθέσιμη έκδοση όπου συνιστούσε ο εκάστοτε κατασκευαστής. Ανά τακτά χρονικά διαστήματα η εκδόσεις αυτές ανανεώνονται και εξελίσσονται. Για τον λόγο αυτό θα πρέπει ο χρήστης να φροντίσει να προμηθεύεται της πιο πρόσφατες εκδόσεις και να προχωρά στην προσαρμογή τους.

Η κατασκευή ιστοσελίδας πραγματοποιήθηκε με την χρήση του διαχειριστικού λογισμικού Joomla 3.0 όπου ήταν και η πιο πρόσφατη έκδοση. Σαφέστατα μπορούν να χρησιμοποιηθούν διαφορετικά διαχειριστικά προγράμματα ή προγράμματα σχεδίασης Web ιστοσελίδων όπου αφορά καθαρά της γνώσης του δημιουργού και τι τον εξυπηρετεί περισσότερο.

Βελτιστοποίηση της ιστοσελίδας θα μπορούσε να αποτελέσει η προσθήκη υπηρεσίας στην καρτέλα “Παλαιότερες \_ Μετρήσεις” όπου θα επιτρέπει στους επισκέπτες να εξάγουν και να αποθηκεύουν στον απομακρυσμένο υπολογιστή τους αρχεία, για παράδειγμα με κατάληξη “.CSV”, όπου θα εμπεριέχουν ιστορικό των παλαιότερων

μετρήσεων που έχουν καταχωρηθεί στην βάση. Η υπηρεσία αυτή θα ήταν χρήσιμη σε ενδιαφερόμενους ερευνητές όπου ήθελαν να τα μελετήσουν. Θα χρησίμευε επίσης και στον ίδιο τον χρήστη που έχει εγκαταστήσει την εφαρμογή σε απομακρυσμένη περιοχή και θα του δινόταν η δυνατότητα να συντηρεί αρχείο των μετρήσεων όπου θα λειτουργούσε ως Backup σε ενδεχόμενη απώλεια δεδομένων της βάσης.

Επίσης στην καρτέλα “Μέτρηση Τώρα”, η διαδικασία λήψης της πιο πρόσφατης μέτρησης που ακολουθείται είναι με το να ανανεώνετε ολόκληρη η τρέχων σελίδα όταν ο επισκέπτης βρίσκετε σε αυτήν. Η συγκεκριμένη διαδικασία θα μπορούσε να πραγματοποιηθεί με την χρήση της τεχνικής AJAX η οποία επιτρέπει σε μια ιστοσελίδα να ανανεώνεται ασύγχρονα (asynchronously) ανταλλάσσοντας στο παρασκήνιο μικρού όγκου δεδομένα με τον server επιτρέποντας να ανανεώνονται μέρη της ιστοσελίδας (ένα div για παράδειγμα), χωρίς να ανανεώνεται ολόκληρη η σελίδα, κάτι τέτοιο θα εξοικονομούσε χρόνο και θα έκανε την σελίδα πιο δυναμική και λιγότερο κουραστική προς τον επισκέπτη.

## 7 Παράρτημα

---

### 7.1 Κώδικας Arduino

```
#include <OneWire.h>

int DS18S20_Pin = 10; //DS18S20 Signal pin on digital 2

//Temperature chip i/o

OneWire ds(DS18S20_Pin); // on digital pin 2

void setup(void) {
  Serial.begin(9600);
}

void loop(void) {
  float temperature = getTemp();
  Serial.println(temperature);
  delay(1000); //just here to slow down the output so it is easier to read
}

float getTemp(){
  //returns the temperature from one DS18S20 in DEG Celsius

  byte data[12];
  byte addr[8];

  if ( !ds.search(addr) ) {
    //no more sensors on chain, reset search

    ds.reset_search();
    return -1000;
  }

  if ( OneWire::crc8( addr, 7) != addr[7] ) {
```

```
Serial.println("CRC is not valid!");

return -1000;

}

if ( addr[0] != 0x10 && addr[0] != 0x28) {

    Serial.print("Device is not recognized");

    return -1000;

}

ds.reset();

ds.select(addr);

ds.write(0x44,1); // start conversion, with parasite power on at the end

byte present = ds.reset();

ds.select(addr);

ds.write(0xBE); // Read Scratchpad

for (int i = 0; i < 9; i++) { // we need 9 bytes

    data[i] = ds.read();

}

ds.reset_search();

byte MSB = data[1];

byte LSB = data[0];

float tempRead = ((MSB << 8) | LSB); //using two's compliment

float TemperatureSum = tempRead / 16;

return TemperatureSum;

}
```

## 7.2 Κώδικες Component

### 7.2.1 Controller.php

```
<?php

// Valid execution check

defined('_JEXEC') or die('Restricted Access');

// Import Joomla Controller Library

jimport( 'joomla.application.component.controller' );

ini_set("display_errors", false);

class BlankController extends JController {

    protected $db;

    public function display($cachable = false, $urlparams = false)
    {

        parent::display($cachable, $urlparams);
    }
}
```

```
}
```

```
public function sense() {  
  
    date_default_timezone_set("Europe/Athens");  
  
    $this->db = JFactory::getDBO();  
  
    // Command Line to read port from arduino  
  
    $device = "COM12";  
  
    $baud = 9600;  
  
    $command = 'powershell "$port= new-Object  
System.IO.Ports.SerialPort "$device.', '$baud.', None, 8, one;  
$port.open(); $port.ReadLine(); $port.close()";'  
  
    $output = shell_exec($command);  
  
    if(!$output) {  
  
        $output = shell_exec($command);  
  
    }  
  
    //$output = trim(preg_replace('/\s\s+/', ' ', $output)); //  
Save command line output  
  
    $timestamp = time(); // Get current system time
```

```
$data = array('Μέτρηση ' => $output, 'Ώρα - Ωρα' =>  
$timestamp);
```

```
// save it
```

```
$q = $this->db->getQuery(TRUE);
```

```
$q->insert("#__arduino");
```

```
$q->set("measurement='{ $output}'");
```

```
$q->set("timestamp='".date("Y-m-d  
$timestamp)."'");
```

```
$this->db->setQuery($q);
```

```
$result = $this->db->loadResult();
```

```
$data = array('measurement' => $output, 'timestamp' =>  
date("d-m-Y H:i:s", $timestamp));
```

```
echo json_encode($data);
```

```
parent::display($cachable, $urlparams);
```

```
}
```

```
public function single() {
```

```
date_default_timezone_set("Europe/Athens");
```

```
$this->db = JFactory::getDBO();
```

```
// Command Line to read port from arduino

$device = "COM27";

$baud = 9600;

$command = 'powershell "$port= new-Object
System.IO.Ports.SerialPort '.$device.','.$baud.',None,8,one;
$port.open(); $port.ReadLine(); $port.close();"';

$output = shell_exec($command);

if(!$output) {

    $output = shell_exec($command);

}

//$output = trim(preg_replace('/\s\s+/', ' ', $output)); //
Save command line output$result = $this->db->loadResult();

$timestamp = time(); // Get current system time

$data = array('Μέτρηση ' => $output, 'Ημ/νία - Ωρα' =>
$timestamp);

// save it

$q = $this->db->getQuery(TRUE);

$q->insert("#__arduino");

$q->set("measurement='{ $output }'");
```



```
        $q->set("timestamp='".date("Y-m-d H:i:s",  
$timestamp)."'");  
  
        $this->db->setQuery($q);  
  
        $this->db->query();  
  
        parent::display($cachable, $urlparams);  
    }  
  
}
```

### 7.2.2 Default.php

```
<?php  
  
// Valid execution check  
  
defined('_JEXEC') or die('Restricted Access');  
  
jimport( "joomla.application.component.modellist" );  
  
class BlankModelDefault extends JModelList {  
  
    public function getItems() {
```

```
        $items = parent::getItems();

        return $items;

    }

    public function getListQuery() {

        $q = parent::getListQuery();

        $q->select("*");
        $q->from("#__arduino");
        $q->order('timestamp DESC');

        return $q;

    }

    protected function populateState($ordering = null, $direction =
null) {

        parent::populateState($ordering, $direction);

    }

}
```

```
/*  
  
// Import Joomla Models Library  
  
jimport( 'joomla.application.component.model' );  
  
class BlankModelDefault extends JModel {  
  
    protected $db;  
  
    function __construct() {  
        parent::__construct();  
  
        $this->db = JFactory::getDBO();  
    }  
  
    function getData() {  
  
        $data = array();  
  
        $q = $this->db->getQuery(TRUE);
```

```
$q->select("*");  
  
$q->from("#__arduino");  
  
$q->order('timestamp DESC');  
  
$this->db->setQuery($q);  
  
$data = $this->db->loadObjectList();  
  
  
return $data;  
  
}  
  
}  
  
*/
```

### 7.2.3 View - Default.php

```
<?php  
  
// Valid execution check  
  
defined('_JEXEC') or die('Restricted Access');  
  
?>  
  
<h1>Μετρήσεις Ph</h1>
```



```
</table>
```

```
<script>
```

```
    setTimeout("location.reload(true);", 60000);
```

```
</script>
```

```
<style>
```

```
    table td { padding: 8px; }
```

```
    input[type=button] { margin:10px 0;}
```

```
    .pagination ul { list-style-type:none; }
```

```
    .pagination ul li { float:left; }
```

```
</style>
```

```
<style>
```

```
table, td, th
```

```
{
```

```
border:1px solid gray;
```

```
}
```

```
th
```

```
{
```

```
background-color:red;
```

```
color:white;
```

```
}
```

```
</style>
```

### 7.3 Προδιαγραφές Arduino.

|                             |  |
|-----------------------------|--|
| Microcontroller             | ATmega328  |
| Operating Voltage           | 5V   |
| Input Voltage (recommended) | 7-12V  |
| Input Voltage (limits)      | 6-20V  |
| Digital I/O Pins            | 14 (of which 6 provide PWM output)                   |
| Analog Input Pins           | 6  |
| DC Current per I/O Pin      | 40 mA  |
| DC Current for 3.3V Pin     | 50 mA  |
| Flash Memory                | 32 KB (ATmega328) of which 0.5 KB used by bootloader |
| SRAM                        | 2 KB (ATmega328)                                     |
| EEPROM                      | 1 KB (ATmega328)                                     |
| Clock Speed                 | 16 MHz   |

### 7.4 Προδιαγραφές Αισθητήρα.

|  |
|--|
| Usable temperature range: -55 to 125°C (-67°F to +257°F)               |
| 9 to 12 bit selectable resolution                                      |
| Uses 1-Wire interface- requires only one digital pin for communication |
| Unique 64 bit ID burned into chip                                      |
| Multiple sensors can share one pin                                     |
| ±0.5°C Accuracy from -10°C to +85°C                                    |
| Temperature-limit alarm system   |
| Query time is less than 750ms  |
| Usable with 3.0V to 5.5V power/data                                    |

## 7.5 Προδιαγραφές Joomla 3.0.

| Software   | Recommended | Minimum | More information  |
|--|-------------|---------|---|
| PHP  | 5.3 +       | 5.2.4 + | <a href="http://www.php.net">http://www.php.net</a>         |
| MySQL  | 5.0.4 +     | 5.0.4 + | <a href="http://www.mysql.com">http://www.mysql.com</a>     |
| <b>Supported Web Servers:</b>                        |             |         |   |
| Apache<br>(with mod_mysql, mod_xml,<br>and mod_zlib) | 2.x +       | 2.x +   | <a href="http://www.apache.org">http://www.apache.org</a>   |
| Nginx  | 1.1         | 1.0     | <a href="http://wiki.nginx.org/">http://wiki.nginx.org/</a> |
| Microsoft IIS  | 7           | 7       | <a href="http://www.iis.net">http://www.iis.net</a>         |



## Βιβλιογραφία

---

- [1] Welling, Thomson, 2005, "Ανάπτυξη Web Εφαρμογών με PHP και MySQL", 3η έκδοση
- [2] Massimo Banzi, 2011, "Getting Started with Arduino", 2nd Edition
- [3] Tom Igoe, 2011, "Making Things Talk Using Sensors, Networks, and Arduino to see, hear, and feel your world", 2nd Edition
- [4] <https://www.sparkfun.com>
- [5] <http://www.arduino.cc>
- [6] <https://www.adafruit.com>
- [7] <http://bildr.org/2011/07/ds18b20-arduino/>
- [8] <http://garagelab.com>
- [9] <http://www.joomla.gr/>
- [10] <http://arudinodisligreek.weebly.com/>
- [11] <http://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/DS18B20.html>
- [12] <http://playground.arduino.cc/Learning/OneWire>
- [13] <http://www.youtube.com>
- [14] <https://sites.google.com/site/wwwitebooksinfoobookslist/>
- [15] <https://www.siteground.com>
- [16] <http://websitesetup.org/build-website-with-joomla/>
- [17] <http://www.hobbytronics.co.uk/ds18b20-arduino>