



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino**

**Χριστίνα Σ. Παγώνη**

**Εισηγητής: Πάρις Μαστροκώστας, Καθηγητής**

**ΑΘΗΝΑ**  
**ΙΟΥΝΙΟΣ 2018**

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

**(Κενό φύλλο)**

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino**  
**Χριστίνα Σ. Παγώνη**  
**A.M. 42063**

**Εισηγητής:**

**Πάρις Μαστροκώστας, Καθηγητής**

**Ημερομηνία εξέτασης ..../..../2018**

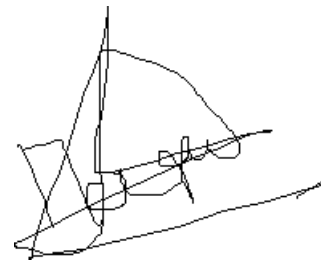
## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Η κάτωθι υπογεγραμμένη Χριστίνα Σ. Παγώνη, του Σωτηρίου, με αριθμό μητρώου 42063 φοιτήτρια του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού δμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»



Παγώνη Χριστίνα  
12 Ιουνίου 2018

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

**(Κενό φύλλο)**

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της χρήσης του Arduino. Τη προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, τον οποίο θα ήθελα να ευχαριστήσω.

Ακόμα θα ήθελα να ευχαριστήσω την οικογένειά μου που θα ήθελε να τελειώσω τις σπουδές μου το συντομότερο δυνατόν.

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

**(Κενό φύλλο)**

## ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία αποσκοπεί στην δημιουργία ενός συστήματος διαδικτυακής παρακολούθησης μετρήσεων με τη χρήση Arduino. Συγκεκριμένα, το arduino υπο θα χρησιμοποιηθεί σε συνδυασμό με τον αισθητήρα θερμοκρασίας DHT11, τον αισθητήρα απόστασης υπερήχων HC-SR04 και το Ethernet Shield. Το Ethernet Shield θα αποτελέσει το μέσω διασύνδεσης του Arduino με το διαδίκτυο ώστε να απεικονίζονται και να παρακολουθούνται οι πειραματικές μετρήσεις. Δηλαδή το arduino θα ρυθμιστεί ώστε να λειτουργεί ως Web Server.

Τα δεδομένα θα αποθηκεύονται και θα απεικονίζονται αριθμητικά γραφικά τόσο αυτά καθαυτά όσο και μετά από επεξεργασία (π.χ. στατιστικές τιμές).

Τα παραδοτέα της εργασίας θα είναι ο τόμος της πτυχιακής εργασίας, το λογισμικό και το hardware.

Το εκπαιδευτικό και γνωστικό όφελος τη εργασίας είναι η εμβάθυνση στο Arduino και τη διαχείριση αισθητήρων, καθώς και η ανάπτυξη της διαδικτυακής εφαρμογής.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Προγραμματισμός Μικροελεγκτών, Arduino

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μικροελεγκτής, DHT11, HC-SR04, Ethernet Shield



## **ABSTRACT**

The present thesis aims at creating an online monitoring system using Arduino. In particular, the arduino uno will be used in combination with the DHT11 temperature sensor, the HR-SR04 ultrasonic distance sensor and the Ethernet Shield. The Ethernet Shield will be the interface of Arduino with the internet to display and monitor experimental measurements. That is, arduino will be configured to run as a Web Server.

The data will be stored and displayed numerically in graphics both as such and after processing (eg statistical values).

Deliveries of the work will be the volume of the thesis, software and hardware. The educational and cognitive benefit of the work is the deepening of Arduino and sensor management, as well as the development of the web application.

SCIENTIFIC AREA: Microcontroller Programming, Arduino

KEYWORDS: Microcontroller, DHT11, HC-SR04, Ethernet Shield

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>1. ΕΙΣΑΓΩΓΗ</b> .....	<b>14</b>
1.1 Περιγραφή του αντικείμενου της πτυχιακής εργασίας .....	14
1.2 Μικροεπεξεργαστής .....	14
1.2.1 Ιστορική αναδρομή .....	14
1.3 Μικροελεγκτής .....	15
1.3.1 Ιστορική αναδρομή .....	15
1.4 Arduino .....	16
1.4.1 Ιστορική αναδρομή .....	16
<b>2. ΠΛΑΤΦΟΡΜΑ ARDUINO</b> .....	<b>18</b>
2.1 Τι είναι ο Arduino .....	18
2.2 Μικροελεγκτής, η καρδιά του Arduino Uno .....	20
2.3 Χαρακτηριστικά του Arduino Uno .....	20
2.4 Τροφοδοσία .....	22
2.5 Εγκατάσταση περιβάλλοντος Arduino IDE .....	23
<b>3. ΑΙΣΘΗΤΗΡΕΣ ΚΑΤΑΓΡΑΦΗΣ</b> .....	<b>26</b>
3.1 Γενικά οι αισθητήρες .....	26
3.2 Αισθητήρας Υπερήχων HC-SR04 .....	27
3.3 Αισθητήρας Θερμοκρασίας DHT11 .....	28
3.3.1 Πώς μετρά ο DHT11 την υγρασία και τη θερμοκρασία .....	30
3.3.2 Απεικόνιση υγρασίας και θερμοκρασίας στο serial monitor .....	31
<b>4. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ARDUINO</b> .....	<b>34</b>
4.1 Η δομή του προγράμματος .....	34
4.2 Ψηφιακές ακίδες (DigitalPins) .....	35
4.3 Αναλογικές ακίδες (AnalogInputPins) .....	36
4.4 Σειριακή οθόνη (Serial monitor) .....	37
<b>5. ΠΕΡΙΦΕΡΙΑΚΕΣ ΣΥΣΚΕΥΕΣ</b> .....	<b>38</b>
5.1 Κάρτα επέκτασης δικτύου (Arduino Ethernet Shield) W5100 .....	38

5.2 Breadboard .....	40
5.3 Jumper wire .....	40
<b>6. ΣΥΝΔΕΣΗ ΠΕΡΙΦΕΡΙΑΚΩΝ – ΑΙΣΘΗΤΗΡΩΝ &amp; ΥΛΟΠΟΙΗΣΗ .....</b>	<b>42</b>
6.1 Κάρτα επέκτασης δικτύου (Ethernet Shield) .....	42
6.2 Αισθητήρας Καταγραφής Θερμοκρασίας Υγρασίας DHT11 .....	44
6.3 Υλοποίηση .....	45
6.3.1 Πρόγραμμα και αξιοποιήσιμες βιβλιοθήκες .....	45
6.3.2 Sketches.....	45
6.3.3 Μενού File .....	46
6.3.4 Edit .....	48
6.3.5 Sketch .....	49
6.3.6 Tools .....	50
6.3.7 Help .....	51
6.3.8 Sketch Book .....	52
6.3.9 Uploading .....	52
6.3.10 Libraries (Βιβλιοθήκες) .....	53
6.3.11 Third – Party Hardware .....	53
6.3.12 Serial Monitor.....	54
6.3.13 Preferences (Προτιμήσεις) .....	54
<b>7. ΔΙΕΠΑΦΗ ΧΡΗΣΤΗ .....</b>	<b>56</b>
7.1 Αρχική Σελίδα .....	56
7.2 Προβολή Στατιστικών .....	58
<b>8. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ .....</b>	<b>60</b>
8.1 Σύνοψη της πτυχιακής εργασίας .....	60
8.2 Προοπτικές .....	61
<b>9. ΠΑΡΑΡΤΗΜΑ Α' .....</b>	<b>61</b>
<b>ΠΑΡΑΡΤΗΜΑ Β' .....</b>	<b>71</b>
<b>ΠΑΡΑΡΤΗΜΑ Γ' .....</b>	<b>74</b>
<b>10. ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>79</b>

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

<b>Εικόνα 1.1:</b> Μικροεπεξεργαστής .....	<b>15</b>
<b>Εικόνα 1.2:</b> Arduino Mini ATmega328 .....	<b>17</b>
<b>Εικόνα 2.1:</b> Arduino Uno .....	<b>18</b>
<b>Εικόνα 2.2:</b> Arduino Uno R3,Duemilanove, Diecimila, Leonardo .....	<b>19</b>
<b>Εικόνα 2.3:</b> Χαρακτηριστικά Arduino Uno .....	<b>21</b>
<b>Εικόνα 2.4:</b> Arduino Installation .....	<b>24</b>
<b>Εικόνα 2.5:</b> Arduino Installation .....	<b>24</b>
<b>Εικόνα 2.6:</b> Arduino Installation .....	<b>25</b>
<b>Εικόνα 3.1:</b> Αισθητήρας HC-SR04 .....	<b>28</b>
<b>Εικόνα 3.2:</b> Αισθητήρας DHT11.....	<b>30</b>
<b>Εικόνα 3.3:</b> DHT11 Pins .....	<b>31</b>
<b>Εικόνα 3.4:</b> Arduino with DHT11 .....	<b>32</b>
<b>Εικόνα 4.1:</b> Δομή προγράμματος .....	<b>35</b>
<b>Εικόνα 5.1:</b> Arduino Ethernet Shield w5100 .....	<b>39</b>
<b>Εικόνα 5.2:</b> Breadboard .....	<b>40</b>
<b>Εικόνα 5.3:</b> Jumper Wires .....	<b>41</b>
<b>Εικόνα 6.1:</b> Ethernet Shlied W5100 with Arduino Uno .....	<b>42</b>
<b>Εικόνα 6.2:</b> Σύνδεση Arduino - W5100 - Breadbord .....	<b>43</b>
<b>Εικόνα 6.3:</b> Εκτύπωση ip διεύθυνσης .....	<b>44</b>
<b>Εικόνα 7.1:</b> Διάγραμμα ροής διαδικασίας εξυπηρετητή διαδικτύου .....	<b>56</b>
<b>Εικόνα 7.2:</b> Αρχική Σελίδα .....	<b>57</b>
<b>Εικόνα 7.3:</b> Σελίδα προβολής Στατιστικών stats.htm .....	<b>59</b>

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

<b>Πίνακας 1:</b> Τεχνικά χαρακτηριστικά DHT11 .....	<b>29</b>
<b>Πίνακας 2:</b> Επεξήγηση κουμπιών της γραμμής εργαλείων .....	<b>46</b>

## ΚΕΦΑΛΑΙΟ 1

### ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο αναλύεται το αντικείμενο της πτυχιακής εργασίας και γίνεται μια ιστορική αναδρομή γύρω από τις μεθόδους που έχουν παρουσιαστεί σε αυτήν την περιοχή.

#### 1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη ενός συστήματος διαδικτυακής παρακολούθησης μετρήσεων με τη χρήση Arduino. Συγκεκριμένα, το arduino uno θα χρησιμοποιηθεί σε συνδυασμό με τον αισθητήρα θερμοκρασίας DHT11, τον αισθητήρα απόστασης υπερήχων HR-SR04 και το Ethernet Shield. Το εκπαιδευτικό και γνωστικό όφελος της εργασίας είναι η εμπάθυνση στο Arduino και τη διαχείριση αισθητήρων, καθώς και η ανάπτυξη της διαδικτυακής εφαρμογής.

Με τη κατασκευή ενός συστήματος που χρησιμοποιεί τη συσκευή ανοιχτού κώδικα arduino, θα καταγραφούν οι τιμές της υγρασίας και της θερμοκρασίας που επικρατούν σε ένα χώρο. Τα δεδομένα θα αποθηκεύονται σε αρχείο σε μια κάρτα μνήμης τύπου microSD, ανά x λεπτά της ώρας, 24 ώρες το 24ωρο.

#### 1.2 Μικροεπεξεργαστής

##### 1.2.1 Ιστορική αναδρομή

Η επιθυμία των κατασκευαστικών συστημάτων να δημιουργήσουν συστήματα με περισσότερες δυνατότητες και με μικρότερο μέγεθος, οδήγησε στην ανάγκη για ενσωμάτωση όλων των λειτουργιών ενός υπολογιστή σε ένα ή μερικά ολοκληρωμένα κυκλώματα. Η ανάπτυξη της τεχνολογίας που χρησιμοποιήθηκε για την κατασκευή των κυκλωμάτων αυτών είχε σαν αποτέλεσμα την ανάπτυξη του μικροεπεξεργαστή. Οι μικροεπεξεργαστές είναι υπεύθυνοι για την έμπνευση και τη δημιουργία μερικών από τις μεγαλύτερες καινοτομίες στα συστήματα υπολογιστών. Αυτές οι καινοτομίες περιλαμβάνουν τους embedded μικροελεγκτές, τους προσωπικούς υπολογιστές, τους σύγχρονους σταθμούς εργασίας, συσκευές χειρός και κινητές συσκευές (όπως τους επεξεργαστές των κινητών τηλεφώνων), servers (εξυπηρετητές) εφαρμογών και αρχείων, web servers για το internet,

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

υπερυπολογιστές χαμηλού κόστους και ευρείας κλίμακας δίκτυα υπολογιστών. Οι μικροεπεξεργαστές είναι επεξεργαστές συνόλου εντολών (instruction set processors, ISP). Ένας ISP εκτελεί εντολές ενός προκαθορισμένου συνόλου εντολών. Η λειτουργικότητα του εξαρτάται ανάλογα από το σύνολο εντολών που είναι ικανός να εκτελέσει ο μικροεπεξεργαστής. Σε αυτό το σύνολο εντολών κωδικοποιούνται όλα τα προγράμματα που τρέχουν σε έναν μικροεπεξεργαστή. Αυτό το προκαθορισμένο σύνολο εντολών ονομάζεται επίσης αρχιτεκτονική συνόλου εντολών (instruction set architecture, ISA). Το ISA χρησιμεύει ως μια διασύνδεση ανάμεσα στο λογισμικό (software) και το υλικό (hardware), δηλαδή ανάμεσα στα προγράμματα και τους επεξεργαστές.



Εικόνα 1.1: Μικροεπεξεργαστής

### 1.3 Μικροελεγκτής

#### 1.3.1 Ιστορική αναδρομή

Με τον όρο μικροελεγκτής αναφερόμαστε σε έναν τύπο επεξεργαστή ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Ένας μικροελεγκτής είναι ένα ενσωματωμένο τσιπ (ολοκληρωμένο κύκλωμα) που αποτελεί συχνά μέρος ενός συστήματος. Όπως ένας τυπικός Η/Υ έτσι και ο μικροελεγκτής περιλαμβάνει CPU, RAM, ROM για αποθήκευση δεδομένων και λογισμικού αντίστοιχα, μνήμη flash για μόνιμη αποθήκευση, θύρες εισόδου - εξόδου, μετατροπέα αναλογικού σε ψηφιακό σήμα και το αντίστροφο και τέλος timers. Καθώς είναι σχεδιασμένος να εκτελεί μόνο μια συγκεκριμένη εργασία για τον έλεγχο ενός απλού συστήματος, είναι πολύ μικρότερος και απλούστερα σχεδιασμένος ώστε να μπορεί να περιλαμβάνει όλες τις λειτουργίες που απαιτούνται σε ένα μόνο ολοκληρωμένο κύκλωμα. Χρησιμοποιείται Παγώνη Σ. Χριστίνα

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

ευρέως σε όλα τα ενσωματωμένα συστήματα έλεγχου χαμηλού και μεσαίου κόστους όπως για παράδειγμα αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά προϊόντα (ψηφιακές μηχανές ,παιχνίδια,...), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα. Στις μέρες μας η χρήση του μικροελεγκτή είναι καθολική για το λόγο ότι κάθε προϊόν το οποίο αλληλεπιδρά με έναν χρήστη περιλαμβάνει έναν μικροελεγκτή, ο οποίος παίζει τον ρόλο του «εγκεφάλου» των ηλεκτρονικών κυκλωμάτων.

## **1.4 Arduino**

### **1.4.1 Ιστορική αναδρομή**

Το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους. Το 2005 ξεκίνησε ένα σχέδιο ώστε να κατασκευαστεί μια συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο οικονομική από άλλα πρωτότυπα διαθέσιμα συστήματα εκείνη την περίοδο. Οι ιδρυτές αυτού του σχεδίου είναι οι Massimo Banzi και David Cueartielles και ονόμασαν το σχέδιο εμπνευσμένοι από Arduino της Ivrea. Ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα. Η Ιβρέα είναι κωμόπολη της επαρχίας του Τορίνο, στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας, όπου στην ίδια περιοχή στεγαζόταν η εταιρία υπολογιστών Olivetti. Το σχέδιο Arduino είναι μια διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες), παρόμοια με τη C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE). Το πρώτο Arduino που φτιάχτηκε ονομάστηκε «Serial Arduino» και περιλάμβανε μια ATmega8 με άμεση σύνδεση RS-232 με το μικροελεγκτή και όλα τα συστατικά του. Το πρόγραμμα Arduino έλαβε τιμητική μνεία στην κατηγορία Digital Communities στο Prix Ars Electronica το 2006.

Οι εκδόσεις που ακολούθησαν ήταν όλες FTDI USB μετατροπέα. Έπειτα, από τις εκδόσεις USBv1.0 και v2.0, κυκλοφόρησε το Arduino Extreme, όπου αυξήθηκε η ποσότητα των επιφανειακών εξαρτημάτων. Ακολούθησαν βελτιστοποιήσεις των εκδόσεων Arduino, όπως το Arduino Nuova Generazione με έναν απλούστερο μετατροπέα USB, όπου μετατρέπεται από ATmega8 σε Atmega168. Οι εκδόσεις Diecimila, Duemilanove και Uno αντικατέστησαν το μετατροπέα FTDI Atmega8U2, ο



Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

οποίος περιέχει έναν ελεγκτή USB και αναβαθμίζει τον μικροελεγκτή σε ATmega328.

Με την τελευταία έκδοση του Arduino Leonardo, εξαλείφθηκε ο μετρατοπέας και αναβαθμίστηκε σε ATmega32U4, το οποίο περιέχει ένα ελεγκτή USB απλουστεύοντας έτσι το σχεδιασμό σε μεγάλο βαθμό.

Οι μικρότερες εκδόσεις Arduino όπως Mini και Nano επιτρέπουν μικρότερες εγκαταστάσεις. Μερικές εκδόσεις είναι διαθέσιμες σε μορφή Arduino BT για συνδεσιμότητα με Bluetooth και σε μορφή Arduino Ethernet για σύνδεση σε LAN, το οποίο έχει επίσης υποστήριξη για Power over Ethernet, εάν μια add-on μονάδα έχει εγκατασταθεί στο πλακίδιο.



Εικόνα 1.2: Arduino Mini Atmega328

## ΚΕΦΑΛΑΙΟ 2

### ΠΛΑΤΦΟΡΜΑ ARDUINO

#### 2.1 Τι είναι ο Arduino

Ο Arduino είναι μια «ανοικτού κώδικα» πλατφόρμα «πρωτοτυποποίησης» ηλεκτρονικών βασισμένη σε ευέλικτο και εύκολο στη χρήση hardware και software που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα

Στην ουσία, πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα. Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου. Μάλιστα κάποιος θα μπορούσε να ισχυριστεί - και θα ήταν ένας αρκετά πετυχημένος παραλληλισμός - ότι λειτουργικά το Arduino μοιάζει πολύ με το NXT Brick των Lego Mindstorms NXT. Άλλωστε η ρομποτική είναι μια από τις πολλές εφαρμογές στις οποίες ο Arduino διαπρέπει.



Εικόνα 2.1:Arduino Uno

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

Ο Arduino βέβαια, δεν είναι ούτε ο μοναδικός, ούτε και ο καλύτερος δυνατός τρόπος για την δημιουργία μιας οποιασδήποτε διαδραστικής ηλεκτρονικής συσκευής. Όμως το κύριο πλεονέκτημά του είναι η τεράστια κοινότητα που το υποστηρίζει και η οποία έχει δημιουργήσει, συντηρεί και επεκτείνει μια ανάλογου μεγέθους online γνωσιακή βάση. Έτσι, παρότι ένας έμπειρος ηλεκτρονικός μπορεί να προτιμήσει διαφορετική πλατφόρμα ή εξαρτήματα ανάλογα με την εφαρμογή που έχει στον νου του, ο Arduino, με το εκτενές documentation, καταφέρνει να κερδίσει όλους αυτούς των οποίων οι γνώσεις στα ηλεκτρονικά περιορίζονται στα όσα λίγα έμαθαν στο σχολείο.

Ακριβώς επειδή απευθύνεται κυρίως σε αρχάριους των ηλεκτρονικών και επειδή, παρά τις αναλυτικότερες οδηγίες που υπάρχουν, δεν έχουν όλοι τις γνώσεις και τα μέσα να κατασκευάσουν μια ηλεκτρονική πλακέτα, κυκλοφορούν έτοιμες, προκατασκευασμένες πλακέτες Arduino που μπορείτε να προμηθευτείτε σε πολύ οικονομική τιμή. Με λίγα χρήματα παραπάνω μάλιστα, οι περισσότεροι προμηθευτές διαθέτουν Arduino Starter Kit, τα οποία, εκτός από το ίδιο ο Arduino, περιέχουν διάφορα άλλα εξαρτήματα και εργαλεία που μπορεί να χρειαστείτε για τις πρώτες σας εφαρμογές (όπως το απαραίτητο καλώδιο USB για την σύνδεση με τον υπολογιστή, ράστερ, καλώδια, LED, διακόπτες, ποτενσιόμετρα, αντιστάσεις, διόδους, τρανζίστορ κ.λπ.).



Εικόνα 2.2: Arduino Uno R3, Duemilanove, Diecimila, Leonardo

## 2.2 Μικροελεγκτής, η καρδιά του Arduino Uno

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

**α)** 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά σας για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στον Arduino σταματήσει ή αν γίνει reset.

**β)** 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.

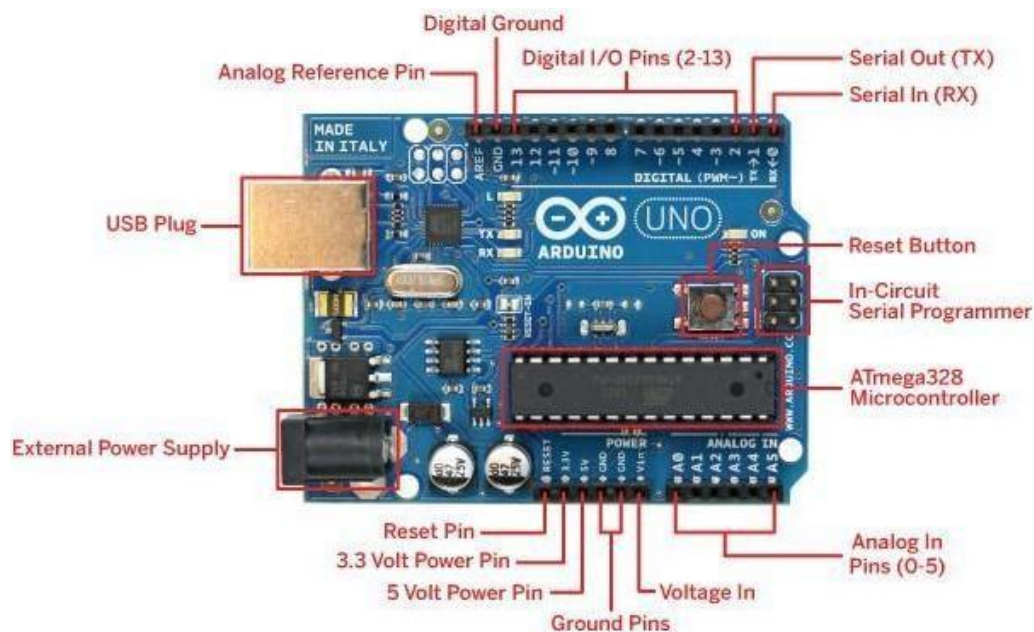
**γ)** 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για να εγκαταστήσετε τα δικά σας προγράμματα στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή σας. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για χρήση runtime μέσα από τα προγράμματά σας, λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2Kb SRAM + 1Kb EEPROM), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει την χρήση όσου χώρου περισσεύει (30Kb μείον το μέγεθος του προγράμματός σας σε μεταγλωτισμένη μορφή).

## 2.3 Χαρακτηριστικά του ArduinoUno

Καταρχήν ο Arduino διαθέτει σειριακό interface. Ο μικροελεγκτής ATmega υποστηρίζει σειριακή επικοινωνία, την οποία ο Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για τη μεταφορά των προγραμμάτων που

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

σχεδιάζονται από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται.



Εικόνα 2.3: Χαρακτηριστικά Arduino Uno

Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές εισοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA. Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμά σας σε κατάσταση HIGH ή LOW, οπότε ο Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Με αυτόν τον τρόπο μπορείτε λόγω χάρη να ανάψετε και να σβήσετε ένα LED που έχετε συνδέσει στο συγκεκριμένο pin. Αν πάλι ρυθμίσετε ένα από αυτά τα pin ως ψηφιακή είσοδο μέσα από το πρόγραμμά σας, μπορείτε με την κατάλληλη εντολή να διαβάσετε την κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχετε συνδέσει σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin (με αυτόν τον τρόπο λόγω χάρη μπορείτε να «διαβάζετε» την κατάσταση ενός διακόπτη).

Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές εισοδοι/έξοδοι έχουν και δεύτερη λειτουργία. Συγκεκριμένα:

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμά σας ενεργοποιεί τη σειριακή θύρα. Έτσι, όταν λόγω χάρη το πρόγραμμά σας στέλνει δεδομένα στη σειριακή, αυτά προωθούνται και στη θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. έναν δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμά σας ενεργοποιήσετε το σειριακό interface, χάνετε 2 ψηφιακές εισόδους/εξόδους.

Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορείτε να τα ρυθμίσετε μέσα από το πρόγραμμά σας ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει \*άμεσα\* και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι, μπορείτε να συνδέσετε λόγω χάρη ένα LED σε κάποιο από αυτά τα pin και να ελέγξετε πλήρως την φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να έχετε απλά τη δυνατότητα αναμμένο - σβηστό που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι. Είναι σημαντικό να καταλάβετε ότι το PWM (Pulse Width Modulation) δεν είναι πραγματικά αναλογικό σύστημα και ότι θέτοντας στην έξοδο τη τιμή 127, δεν σημαίνει ότι η έξοδος θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

Μέσα από τη βιβλιοθήκη του προγράμματος του Arduino χρησιμοποιούμε το analogWrite που έχει κλίμακα από 0-255, με άλλα λόγια μια συχνότητα PWM (Pulse Width Modulation) περίπου 500Hz. Έτσι αν analogWrite (255) είναι ένας κύκλος εργασίας 100% (πάντα on), αν analogWrite (127) είναι ένας κύκλος 50% (στο μισό χρόνο).

## 2.4 Τροφοδοσία

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω - αριστερή γωνία του Arduino.

Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC.

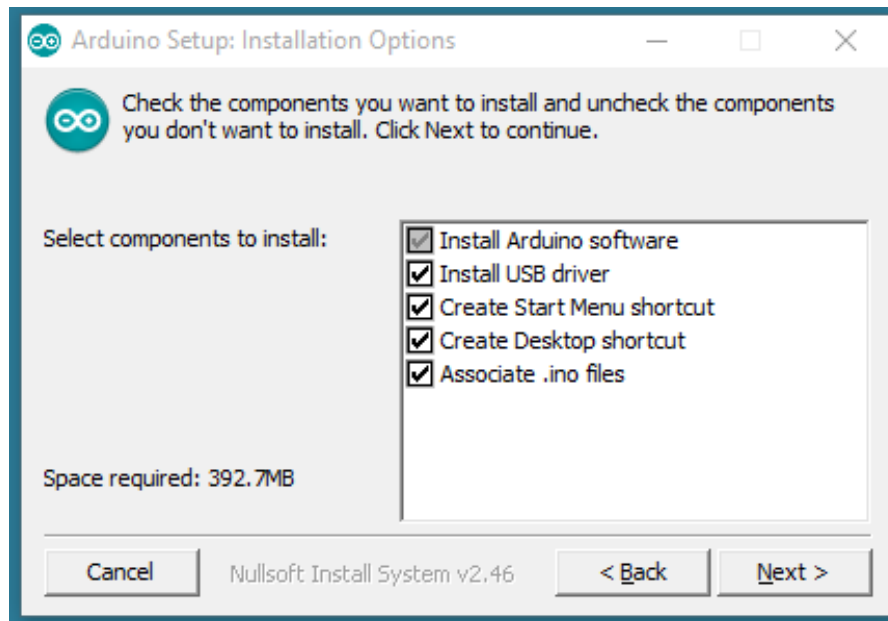
## **2.5 Εγκατάσταση περιβάλλοντος Arduino IDE**

Για να γίνει σωστή εγκατάσταση του προγράμματος, πρέπει να ακολουθηθεί μια σειρά από βήματα, ανάλογα με το λειτουργικό σύστημα που διαθέτει. Στη περίπτωση μας θα εγκατασταθεί σε λειτουργικό σύστημα Windows. Στη παρούσα εργασία επιλέχθηκε να χρησιμοποιηθεί η πλακέτα Arduino UNO R3. Θα χρησιμοποιήσουμε ένα καλώδιο USB για να συνδεθούν πλακέτα και υπολογιστής. Συνδέουμε τη πλακέτα Arduino UNO R3 στον υπολογιστή χρησιμοποιώντας το καλώδιο USB. Παρατηρούμε ότι το LED της πλακέτας ανάβει. Αρχίζουμε το κατέβασμα από τη σελίδα που προαναφέρθηκε και ακολουθούμε τα βήματα για την εγκατάσταση.

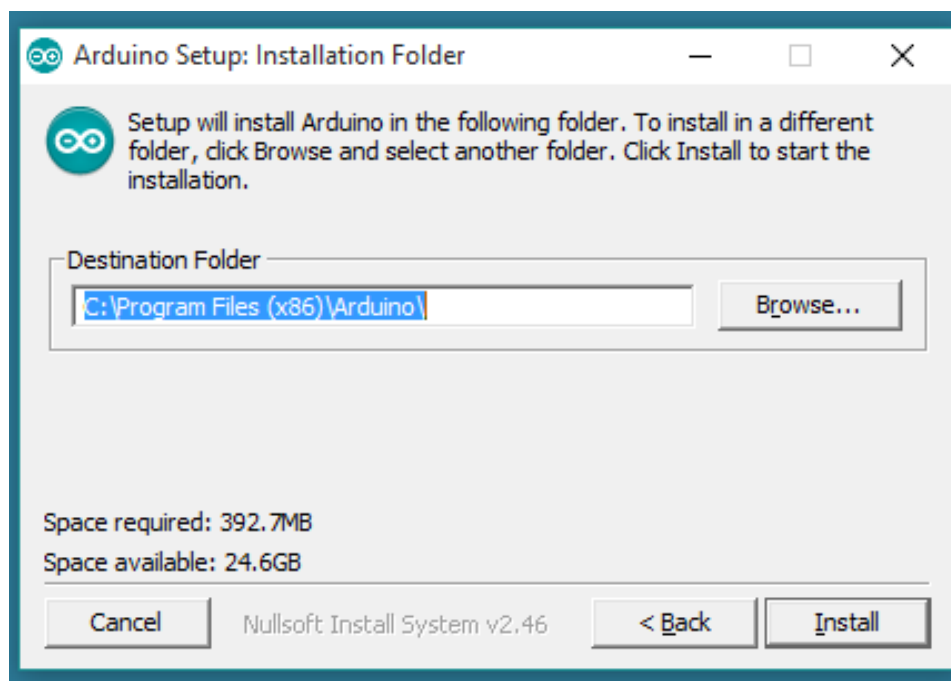
Μεταφορτώνουμε δωρεάν τη τελευταία έκδοση Arduino-1.8.0. από την ιστοσελίδα <http://arduino.cc/en/main/software>. Μπορείτε να επιλέξετε μεταξύ του πακέτου Installer (.exe) και του Zip. Προτείνουμε να χρησιμοποιήσετε τη πρώτη που εγκαθιστά άμεσα όλα όσα χρειάζεστε για να χρησιμοποιήσετε το λογισμικό Arduino (IDE), συμπεριλαμβανομένων των προγραμμάτων οδήγησης. Με το πακέτο Zip πρέπει να εγκαταστήσετε τα προγράμματα οδήγησης με μη αυτόματο τρόπο. Το αρχείο Zip είναι επίσης χρήσιμο εάν θέλετε να δημιουργήσετε μια φορητή εγκατάσταση.

Όταν ολοκληρωθεί η λήψη, συνεχίστε με την εγκατάσταση και παρακαλούμε επιτρέψτε τη διαδικασία εγκατάστασης του προγράμματος οδήγησης όταν λάβετε μια προειδοποίηση από το λειτουργικό σύστημα.

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.



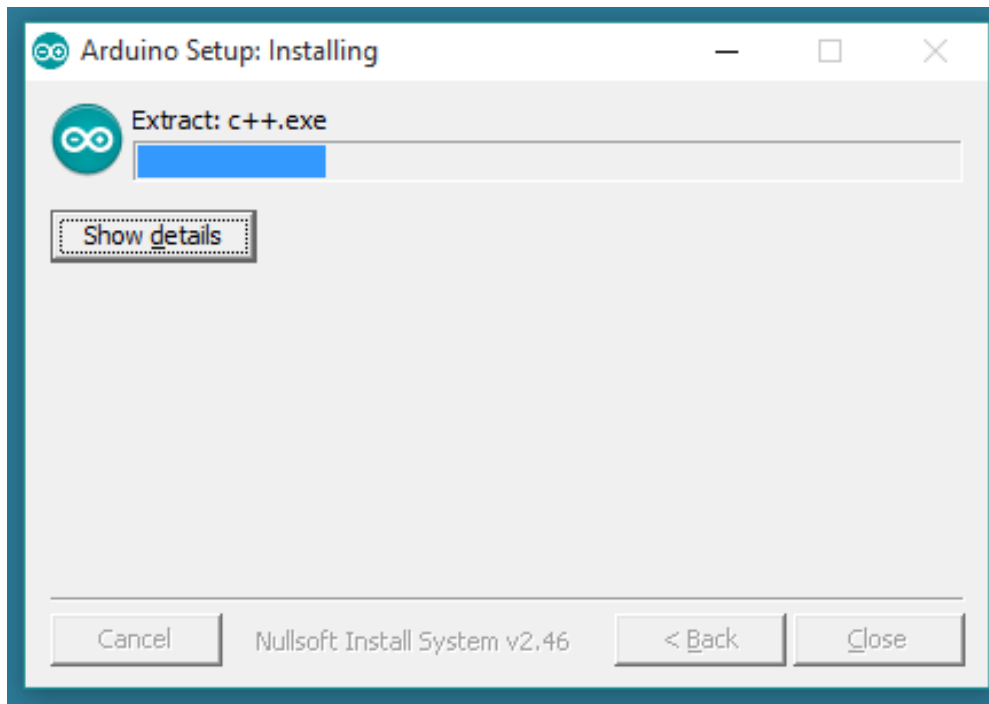
Εικόνα 2.4: Arduino Installation



Εικόνα 2.5: Arduino Installation



Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.



Εικόνα 2.6: Arduino Installation

## ΚΕΦΑΛΑΙΟ 3

### ΑΙΣΘΗΤΗΡΕΣ ΚΑΤΑΓΡΑΦΗΣ

#### 3.1 Γενικά οι αισθητήρες

Η μέτρηση των φυσικών μεγεθών είναι ένας από τους σπουδαιότερους τομείς στην επιστήμη. Στην σημερινή εποχή όπου επικρατούν οι αυτοματισμοί και οι υπολογιστές οι μετρήσεις των διαφόρων φυσικών μεγεθών έχουν αναχθεί σε μετρήσεις ηλεκτρικών μεγεθών και μετατροπή αυτών σε ψηφιακά σήματα για είσοδο σε ηλεκτρονικό υπολογιστή. Στις περισσότερες περιπτώσεις αδυνατούμε να μετρήσουμε απευθείας το φυσικό μέγεθος. Για να πετύχουμε τον σκοπό μας μετρούμε κάποιο άλλο μέγεθος και με βάση κάποιο φυσικό νόμο υπολογίζουμε το ζητούμενο. Για παράδειγμα ένας υπολογιστής αδυνατεί να πάρει ένα μέτρο και να μετρήσει μία διάσταση του δωματίου που βρίσκεται. Αν όμως βάζαμε ένα μεγάφωνο και ένα μικρόφωνο στον υπολογιστή τότε εκπέμποντας ένα ήχο και μετρώντας τον χρόνο που κάνει να έρθει η ανάκλαση θα μπορούσε να υπολογίσει την απόσταση ενός τοίχου. Με το παραπάνω παράδειγμα βλέπουμε ότι είναι δυνατόν να υπολογίσουμε διάφορα φυσικά μεγέθη μετρώντας κάποια άλλα που είναι περισσότερο «εύκολα» για τα μηχανήματα. Το φυσικό αίτιο επιδρά σε ένα σύστημα (αισθητήρα) το οποίο αλλάζει κάποια ηλεκτρικά χαρακτηριστικά (μεταβολή ηλεκτρικού σήματος). Το σήμα επεξεργάζεται αναλογικά (ενισχύεται, φιλτράρεται) και μετατρέπεται σε ψηφιακό με τη χρήση του καταλλήλου μετατροπέα (Analog to Digital Converter). Κατόπιν το υπολογιστικό σύστημα αναλαμβάνει την επεξεργασία, την αποθήκευση και την ένδειξη του μέτρου του φυσικού αίτιου. Ο αισθητήρα λοιπόν μπορεί να είναι ένας απλός μετατροπέας ή κάποια σύνθετη μορφή πολλών μετατροπέων που σκοπό έχουν να μετατρέψουν μια μορφή ενέργειας σε μία άλλη κατάλληλη προς μέτρηση (π.χ ηλεκτρικό σήμα). Οι αισθητήρες μπορούν να κατηγοριοποιηθούν σε δυο μεγάλες κατηγορίες:

- Παθητικοί αισθητήρες (Passive Sensors): Οι οποίοι μετατρέπουν ένα ελάχιστο ποσό ενεργείας σε παρατηρήσιμη μορφή. Π.χ. Το Θερμοζεύγος μετατρέπει τη θερμοκρασία σε σήμα τάσης.

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

- **Ενεργοποιητές (Active Sensors):** Οι οποίοι παρέχουν ενέργεια στο περιβάλλον σαν μέρος της διαδικασίας μέτρησης. Π.χ. Ένα σύστημα ραντάρ ή σόναρ στο οποίο η απόσταση κάποιου αντικειμένου μετράται με αποστολή ηλεκτρομαγνητικού ή ακουστικού κύματος με στόχο την ανάκλαση πάνω στο αντικείμενο και μέτρηση της απόστασης από τον αισθητήρα.

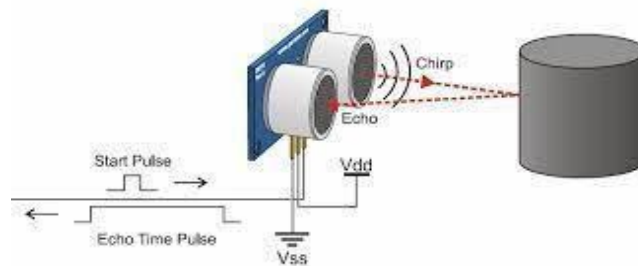
### **3.2 Αισθητήρας Υπερήχων HC-SR04**

Ο αισθητήρας που χρησιμοποιήθηκε στο σύστημα που περιγράφει η εργασία ανήκει στους πιεζοηλεκτρικούς μετατροπείς. Αρχή λειτουργίας των αισθητήρων αυτού του είδους είναι το πιεζοηλεκτρικό φαινόμενο το οποίο ανακαλύφθηκε από τους Pierre και Jacques Curie το 1880. Σύμφωνα με το φαινόμενο αυτό, η παραμόρφωση ενός κρυστάλλου λόγω μηχανικής πίεσης σε αυτόν προκαλεί επαναπροσανατολισμό του φορτίου και αντίστοιχα την μετακίνηση ίσων θετικών και αρνητικών φορτίων στις αντίθετες πλευρές του κρυστάλλου. Αποτέλεσμα των παραπάνω διεργασιών είναι η ανάπτυξη τάσης στα άκρα του κρυστάλλου ανάλογη της παραμόρφωσης που την προκάλεσε.

Το πιεζοηλεκτρικό φαινόμενο εμφανίζεται σε ορισμένους τύπους κρυστάλλων όπως οι κρύσταλλοι Rochelle και Quartz ή σε ορισμένα κεραμικά υλικά όπως το  $3 \text{ BaTiO}_3$ .

Μια σημαντική ιδιότητα του φαινομένου είναι η αντιστρεψιμότητα του, δηλαδή το γεγονός ότι η εφαρμογή τάσης στα άκρα του κρυστάλλου είναι ικανή να προκαλέσει την παραμόρφωση του. Το αντίστροφο αυτό φαινόμενο βρίσκει ευρύτατη εφαρμογή στη σχεδίαση ενεργοποιητών (actuators). Η μέτρηση της παραμόρφωσης του πιεζοηλεκτρικού κρυστάλλου μπορεί να πραγματοποιηθεί μετρώντας είτε το φορτίο που αναπτύσσεται είτε τη τάση που προκύπτει στα άκρα του φορτίου. Ωστόσο το φαινόμενο που παρατηρείται, λόγω διαρροής του φορτίου μέσω του κρυστάλλου (αυτοεκφόρτιση) ακόμα και στη περίπτωση που η παραμόρφωση παραμένει σταθερή, η τάση να ακολουθεί την καμπύλη εκ φόρτισης και να τείνει ασυμπτωτικά στο μηδέν, καθιστά τη χρήση των κρυστάλλων κυρίως για τη μέτρηση δυναμικών παραμορφώσεων, πιέσεων και δυνάμεων.

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.



Εικόνα 3.1: Αισθητήρας HC-SR04

Ο αισθητήρας που χρησιμοποιήσαμε εκμεταλλεύεται το φαινόμενο αυτό, μόνο που οι μηχανική παραμόρφωση του κρυστάλλου οδηγεί στην παραγωγή αρμονικών κυμάτων συχνότητας που βρίσκεται στη περιοχή των υπερήχων.

Για ένα σύστημα σαν και αυτό που αναπτύξαμε υπήρχε η ανάγκη για έναν πομπό και αντίστοιχα έναν δέκτη. Τις λειτουργίες αυτές τις αναλαμβάνει ο πιεζοηλεκτρικός κρύσταλλος που αναλύσαμε παραπάνω. Λειτουργεί ως πομπός καθώς εμείς του εφαρμόζουμε μια τάση και αυτός παράγει ένα ανάλογο υπερηχητικό κύμα στον χώρο. Ενώ στη συνέχεια ένας άλλος όμοιος κρύσταλλος δέχεται αυτό το μηχανικό κύμα και το μετατρέπει σε μια τάση. Για την πραγματοποίηση της μέτρησης βασιζόμαστε στη μέθοδο της ανάκλασης, το μηχανικό κύμα δηλαδή, που εκπέμπεται από τον αισθητήρα θα προσπέσει πάνω σε μία δονούμενη επιφάνεια και θα ανακλαστεί. Καθώς ο δέκτης ανιχνεύει το ανακλώμενο, το μετατρέπει σε ένα ηλεκτρικό σήμα (τάση) το οποίο στη συνέχεια οδηγείται για επεξεργασία. Στη περίπτωση που χρησιμοποιούνται δυο αισθητήρες, χρησιμοποιείται ένας που λειτουργεί μόνο ως πομπός (XT) και ένας που λειτουργεί ως δέκτης (XR), αυτοί οι αισθητήρες ονομάζονται μονοστατικοί. Ωστόσο υπάρχει και μια δεύτερη κατηγορία, χαρακτηριστικό της οποίας είναι το γεγονός ότι μπορούν να λειτουργούν ως πομποί και ως δέκτες, και ονομάζονται διστατικοί.

Ο αισθητήρας HC-SR04 Ultrasonic Distance sensor κυρίως χρησιμοποιείται στα ραντάρ. Ο HC-SR04 αισθητήρας ανιχνεύει αντικείμενα με παραγωγή κυμάτων συχνότητας και αντιλαμβάνεται την απόσταση.

### 3.3 Αισθητήρας Θερμοκρασίας DHT11

Το DHT11 είναι ένας βασικός, εξαιρετικά χαμηλού κόστους ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας. Χρησιμοποιεί ένα χωρητικό αισθητήρα υγρασίας και ένα θερμίστορ για τη μέτρηση του περιβάλλοντος αέρα και εκπέμπει ένα ψηφιακό σήμα

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

στον ακροδέκτη δεδομένων (δεν χρειάζονται αναλογικοί ακροδέκτες εισόδου). Είναι αρκετά απλό στη χρήση, αλλά απαιτεί προσεκτικό χρονοισμό για να αρπάξει τα δεδομένα. Το μόνο πραγματικό μειονέκτημα αυτού του αισθητήρα είναι ότι μπορείτε να λαμβάνετε μόνο νέα δεδομένα από αυτό μία φορά κάθε 2 δευτερόλεπτα, οπότε όταν χρησιμοποιείτε τη βιβλιοθήκη, οι μετρήσεις των αισθητήρων μπορούν να είναι έως και 2 δευτερόλεπτα.

Είναι ιδανικό για απομακρυσμένη μετεωρολογικούς σταθμούς, συστήματα home περιβαλλοντικού ελέγχου και συστήματα παρακολούθησης αγρόκτημα ή στον κήπο.

Γενικά ο αισθητήρας είναι μικρός σε μέγεθος, έχει αρκετά χαμηλή κατανάλωση ισχύος και είναι αρκετά ακριβής στις μετρήσεις. Έχει τρεις ακροδέκτες, ένα για την τροφοδοσία, για την γείωση και για τα δεδομένα. Λόγω αυτών των ιδιοτήτων του έχει χρησιμοποιηθεί σε πολλές εφαρμογές και υπάρχει πληθώρα βιβλιοθηκών στο διαδίκτυο. Τα τεχνικά χαρακτηριστικά του είναι:

Parameter	Min.	Typical	Max.	Unit
Working voltage	3	5	5.5	VDC
Working current (VCC=5V, T=25°C)	0.5	-	2.5	mA
Sampling interval	1	-	-	s
<b>Humidity</b>				
Accuracy	-	±4	-	%RH
	-	-	±5	%RH
Measurement range	20	-	95	%RH
Response time 25°C 20 - 95 %RH	6	10	15	s
<b>Temperature</b>				
Accuracy	±1	-	±2	°C
Measurement range	0	-	50	°C
Response time	6		60	s

Πίνακας 1: Τεχνικά Χαρακτηριστικά DHT11

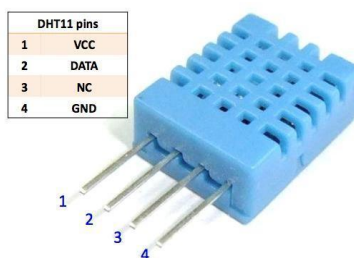
Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

Η διαδικασία της επικοινωνίας ξεκινάει όταν το Arduino στέλνει πρώτο σήμα, ο αισθητήρας μεταβαίνει από τη χαμηλή κατανάλωση λειτουργίας σε κανονική λειτουργία περιμένοντας το Arduino να τελειώσει με την αποστολή. Μόλις αυτό πραγματοποιηθεί ο αισθητήρας στέλνει πίσω στο Arduino ένα σήμα δεδομένων απάντησης των 40bit που περιλαμβάνουν τις ανάλογες πληροφορίες για την υγρασία και τη θερμοκρασία.

Συγκεκριμένα:

- 8 bit για την τιμή της υγρασίας
- 8 bit για την αέραια τιμή της υγρασίας
- 8 bit για την τιμή της θερμοκρασίας
- 8 bit για την αέραια τιμή της θερμοκρασίας
- 8 bit για το bit ελέγχου ισοτιμίας των δεδομένων.

Χωρίς το αρχικό σήμα από το Arduino ο αισθητήρας δεν στέλνει δεδομένα. Όταν τα δεδομένα σταλούν από τον αισθητήρας αυτός μεταβαίνει και πάλι στη χαμηλή κατανάλωση λειτουργία και περιμένει νέο σήμα.



Εικόνα 3.2: Αισθητήρας DHT11

### 3.3.1 Πώς μετρά το DHT11 την υγρασία και τη θερμοκρασία

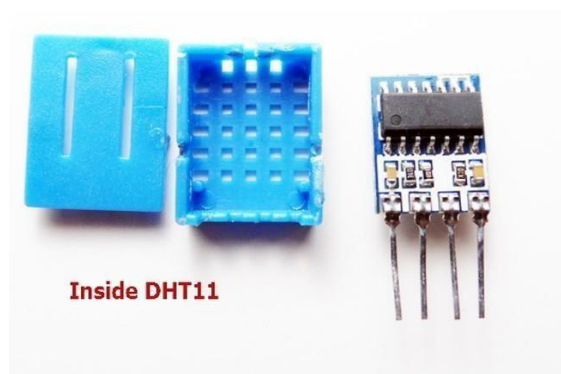
Το DHT11 ανιχνεύει υδρατμούς με μέτρηση της ηλεκτρικής αντίστασης ανάμεσα σε δύο ηλεκτρόδια. Το ανιχνευτικό συστατικό υγρασία είναι ένα υπόστρωμα συγκράτησης της υγρασίας με ηλεκτρόδια που εφαρμόζεται στην επιφάνεια. Όταν υδρατμός απορροφάται από το υπόστρωμα, τα ιόντα που απελευθερώνονται από το υπόστρωμα το οποίο αυξάνει την αγωγιμότητα μεταξύ των ηλεκτροδίων. Η αλλαγή στην αντίσταση μεταξύ των δύο ηλεκτροδίων είναι ανάλογη με τη σχετική υγρασία.

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

Υψηλότερη σχετική υγρασία μειώνει την αντίσταση μεταξύ των ηλεκτροδίων, ενώ οι χαμηλότερη σχετική υγρασία αυξάνει την αντίσταση μεταξύ των ηλεκτροδίων.

Το DHT11 μέτρα θερμοκρασία σε μια επιφάνεια με τον αισθητήρα θερμοκρασίας NTC (θερμίστορ) που είναι ενσωματωμένο στη μονάδα. Αν το πλαστικό περίβλημα αφαιρεθεί, μπορείτε να δείτε τα ηλεκτρόδια που εφαρμόζονται στο υπόστρωμα.

Μια IC τοποθετημένη στο πίσω μέρος της μονάδας μετατρέπει τη μέτρηση της σχετικής υγρασίας. Επίσης, αποθηκεύει τους συντελεστές βαθμονόμησης, και ελέγχει τη μετάδοση του σήματος δεδομένων μεταξύ του DHT11 και το Arduino.



Εικόνα:3.3 DHT11 Pins

Το DHT11 χρησιμοποιεί μόνο ένα καλώδιο σήματος για τη μετάδοση δεδομένων στο Arduino. Η ισχύς προέρχεται από το ξεχωριστό 5V και καλώδια γείωσης. Ένα 10K Ohm pull-up αντίσταση είναι απαραίτητη μεταξύ της γραμμής σήματος και 5V γραμμή για να βεβαιωθείτε ότι η στάθμη του σήματος παραμένει υψηλή από προεπιλογή.

Υπάρχουν δύο διαφορετικές εκδόσεις του DHT11 που μπορεί να συναντήσετε. Ένας τύπος έχει τέσσερις πείρους, και ο άλλος τύπος έχει τρεις ακίδες και είναι συναρμολογημένο σε ένα μικρό PCB. Η σύνδεση του DHT11 στο Arduino είναι πραγματικά εύκολη, αλλά οι συνδέσεις είναι διαφορετικές ανάλογα με τον τύπο που έχετε.

### 3.3.2 Απεικόνιση υγρασίας και θερμοκρασίας στο serial monitor

Για να μπορέσουμε να χρησιμοποιήσουμε το DHT11 για το Arduino, θα χρειαστεί να εγκαταστήσουμε την βιβλιοθήκη DHTLib. Διαθέτει όλες τις λειτουργίες που απαιτούνται για να πάρει τις μετρήσεις υγρασίας και θερμοκρασίας από τον αισθητήρα. Είναι εύκολο στην εγκατάσταση, απλά κατεβάζουμε το αρχείο DHTLib.zip Παγώνη Σ. Χριστίνα

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

και ανοίγουμε το Arduino IDE. Στη συνέχεια, πηγαίνουμε σε Sketch > Include Library > Add .ZIP Library και επιλέγουμε το αρχείο DHTLib.zip.

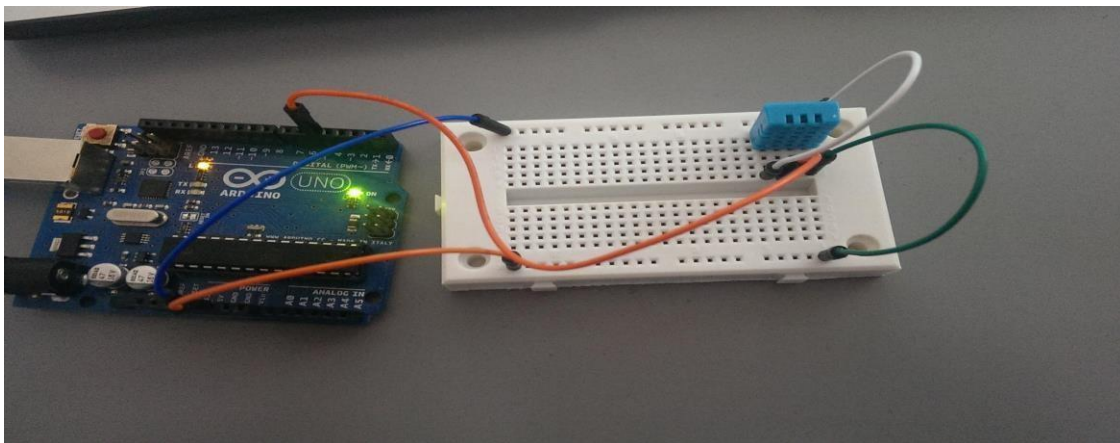
Μετά την προσθήκη της βιβλιοθήκης, φορτώνουμε το ακόλουθο παράδειγμα στο Arduino και ανοίγουμε τη σειριακό οθόνη:

```
#include <dht.h>
DHT DHT?
#define DHT11_PIN A0
void setup () {
  Serial.begin (9600)?
}
void loop (){}
```

Θα πρέπει να δείτε τις μετρήσεις της υγρασίας και της θερμοκρασίας να εμφανίζονται σε διαστήματα του ενός δευτερολέπτου

Εάν δεν θέλετε να χρησιμοποιήσετε το pin 7 για το σήμα των δεδομένων, μπορείτε να αλλάξετε τον αριθμό του pin στην γραμμή του κώδικα όπου λέει

```
#define DHT11_PIN A0
```



Εικόνα 3.4: Arduino with DHT11



## ΚΕΦΑΛΑΙΟ 4

### ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ARDUINO

#### 4.1 Η δομή του προγράμματος

Ένα τυπικό πρόγραμμα Arduino έχει την παρακάτω δομή:

```
//δήλωση μεταβλητών  
void setup ()  
{  
  //αρχικοποιήσει  
}  
void loop ()  
{  
  // Κώδικας  
}
```

Υπάρχουν δυο ειδικές συναρτήσεις που είναι μέρος του κάθε sketch του Arduino οι οποίες είναι η `setup()` και η `loop()`. Η `setup()` καλείται μια φορά, όταν το sketch ξεκινά ή όποτε κάνει επαναφορά (reset) η πλατφόρμα Arduino. Κυρίως, σε αυτήν γίνονται οι αρχικοποιήσεις των μεταβλητών, η ρύθμιση της κατάστασης των ακίδων (pins) και η προετοιμασία των βιβλιοθηκών.

Αντιθέτως, η συνάρτηση `loop()` καλείται ξανά και ξανά επιτρέποντας έτσι στο πρόγραμμα να ανταποκριθεί σε εξωτερικά ερεθίσματα.

Και οι δυο συναρτήσεις πρέπει να περιλαμβάνονται στο sketch, ακόμα και αν δεν περιέχουν κάτι και να είναι κενές.



Εικόνα:4.1 Δομή προγράμματος

## 4.2 Ψηφιακές ακίδες (Digital Pins)

Οι ακίδες αυτές στο Arduino μπορούν να ρυθμιστούν είτε ως είσοδοι είτε ως έξοδοι, όμως από προεπιλογή είναι ρυθμισμένες ως είσοδοι. Επίσης αξίζει να σημειωθεί, ότι η πλειοψηφία των αναλογικών ακίδων του Arduino (Atmega), μπορεί να ρυθμιστεί και να χρησιμοποιηθεί, με τον ίδιο ακριβώς τρόπο όπως οι ψηφιακές ακίδες. Οι συναρτήσεις ψηφιακής εισόδου και εξόδου είναι οι παρακάτω:

- `pinMode()`: Ρυθμίζει τη συγκεκριμένη ακίδα να συμπεριφέρεται ως είσοδος/έξοδος.

Σύνταξη: `pinMode(pin, mode)`

Παράμετροι `pin`: Ο αριθμός της ακίδας της οποίας η λειτουργία είναι επιθυμητό να αλλάξει. `mode`: INPUT/OUTPUT

- `digitalWrite()`: Γράφει μια υψηλή (HIGH) ή μια χαμηλή (LOW) τιμή σε μια

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

ψηφιακή ακίδα. Αν η ακίδα έχει ρυθμιστεί ως έξοδος με την συνάρτηση `pinMode()`, τότε η τάση της θα καθορίσει στην αντίστοιχη τιμή: 5V για HIGH και 0V για LOW. Αν η ακίδα έχει ρυθμιστεί ως είσοδος, γράφοντας HIGH στην συνάρτηση `digitalWrite()` θα ενεργοποιήσει μια εσωτερική pull up - αντίσταση των 20 K ενώ γράφοντας LOW θα την απενεργοποιήσει

Σύνταξη: `digitalWrite(pin,value)`

Παράμετροι: `pin`: Ο αριθμός της ακίδας της οποίας η λειτουργία είναι επιθυμητό να αλλάξει. `Value`: INPUT/OUTPUT

- `digitalRead()`: Διαβάζει την τιμή από μια συγκεκριμένη ψηφιακή ακίδα, που είναι είτε HIGH είτε LOW.

Σύνταξη: `digitalRead(pin)`

Παράμετροι: `pin`: Ο αριθμός της ακίδας της οποίας η λειτουργία είναι επιθυμητό να αλλάξει. Επιστρέφει: HIGH/LOW

### 4.3 Αναλογικές ακίδες (Analog Input Pins)

Οι ελεγκτές Atmega που χρησιμοποιούνται για τη πλατφόρμα Arduino περιέχουν έναν ενσωματωμένο αναλογικό σε ψηφιακό μετατροπέα 6 καναλιών. Ο μετατροπέας διαθέτει ανάλυση 10bit, επιστρέφοντας ακέραιους από 0 έως 1023. Ενώ η κύρια λειτουργία της αναλογικής ακίδας για τους περισσότερους χρήστες Arduino είναι να διαβάζει αναλογικούς αισθητήρες, οι αναλογικές ακίδες έχουν επίσης όλες τις λειτουργίες των γενικών ακίδων εισόδου/εξόδου. Οι συναρτήσεις αναλογικής εισόδου και εξόδου είναι οι παρακάτω:

- `analogWrite()`: Γράφει μια αναλογική τιμή (PWM κύμα) σε μια ακίδα. Μπορεί να χρησιμοποιηθεί για παράδειγμα να ανάψει ένα LED σε διάφορες φωτεινότητες ή να οδηγήσει ένα κινητήρα σε διάφορες ταχύτητες. Μετά από μια κλήση της `analogWrite()`, η ακίδα θα δημιουργήσει ένα σταθερό τετραγωνικό κύμα του καθορισμένου κύκλου λειτουργίας μέχρι την επόμενη κλήση της `analogWrite()` (ή μια κλήση της `digitalWrite()` ή `digitalRead()` για την ίδια ακίδα). Η συχνότητα του σήματος PWM είναι περίπου 490 Hz. Στις

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

περισσότερες πλατφόρμες Arduino η συνάρτηση αυτή λειτουργεί στις ακίδες 3, 5, 6, 9, 10, 11.

Σύνταξη: `analogWrite(pin, value)`

Παράμετροι: `pin`: Ο αριθμός της ακίδας της οποίας θα γράψει επάνω

`value`: ο κύκλος λειτουργίας μεταξύ 0 και 255

- `analogRead()`: Διαβάζει την τιμή από την καθορισμένη αναλογική ακίδα.

Σύνταξη: `analogRead(pin)`

Παράμετροι: `pin`: Ο αριθμός της αναλογικής ακίδας εισόδου από όπου θα

διαβάζει Επιστέφει: ακέραιο από 0 έως 1023

#### 4.4 Σειριακή Οθόνη (Serial Monitor)

Εμφανίζει τα σειριακά δεδομένα που αποστέλλονται από την πλακέτα Arduino. Πιο συγκεκριμένα, η αποστολή δεδομένων στην πλακέτα γίνεται, εισάγοντας κείμενο και πατώντας το κουμπί `send` ή πατώντας το `Enter`. Επίσης, στο κάτω μέρος της σειριακής οθόνης, μπορεί να γίνει η επιλογή της κατάλληλης ταχύτητας (`baud`) από τη λίστα που εμφανίζεται ανάλογα με τη τιμή που θα επιλεγεί στο προγραμματισμό του Arduino με το `Serial.begin()`.

## ΚΕΦΑΛΑΙΟ 5

### ΠΕΡΙΦΕΡΙΑΚΕΣ ΣΥΣΚΕΥΕΣ

Οι πρόσθετες περιφερειακές συσκευές που είναι απαραίτητες για τη επιτυχή λειτουργία και ολοκλήρωση του σταθμού μας είναι:

- Κάρτα Επέκτασης Δικτύου (Arduino Ethernet Shield) W5100
- MicroSd
- DHT11
- HC-SR04
- Breadboard
- Jumper Wire

#### 5.1 Κάρτα επέκτασης δικτύου (Arduino Ethernet Shield) W5100

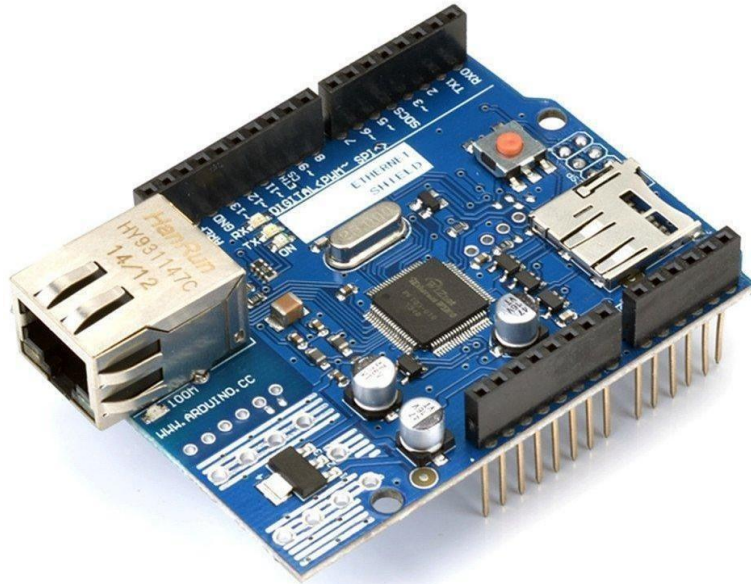
Το Arduino Ethernet Shield είναι μια πλακέτα που για να λειτουργήσει εφάπτεται πάνω στο μικροελεγκτή Arduino Uno καθώς και σε άλλες εκδόσεις του Arduino. Χρησιμοποιείται για συνδεσμολογία του Arduino με το διαδίκτυο, βασίζεται στο τσιπ Ethernet Wiznet W5100 που παρέχει μια στοίβα δικτύου (IP) ικανή τόσο για TCP όσο και για UDP. Το Arduino Ethernet Shield υποστηρίζει έως τέσσερις ταυτόχρονες συνδέσεις υποδοχών. Χρησιμοποιήστε τη βιβλιοθήκη Ethernet για να γράψετε σκίτσα που συνδέονται στο διαδίκτυο χρησιμοποιώντας την ασπίδα.

Το Arduino Ethernet Shield δίνει τη δυνατότητα για αποθήκευση δεδομένων στην SD κάρτα που διαθέτει για προβολή μέσω του δικτύου. Είναι συμβατό με το Arduino Uno και το Mega (χρησιμοποιώντας τη βιβλιοθήκη Ethernet). Μπορείτε να έχετε πρόσβαση στην ενσωματωμένη υποδοχή κάρτας SD χρησιμοποιώντας τη βιβλιοθήκη SD που περιλαμβάνεται στη τρέχουσα έκδοση του Arduino.

Ο Arduino επικοινωνεί τόσο με τη κάρτα W5100 όσο και με τη κάρτα SD χρησιμοποιώντας το δίαυλο SPI (μέσω της κεφαλίδας ICSP). Αυτό ισχύει για τις ψηφιακές ακίδες 11, 12 και 13 στο Uno και τις ακίδες 50, 51 και 52 στο Mega. Και στις δύο πλακέτες, ο ακροδέκτης 10 χρησιμοποιείται για την επιλογή του W5100 και του ακροδέκτη 4 για τη κάρτα SD. Αυτές οι καρφίτσες δεν μπορούν να χρησιμοποιηθούν για γενικές λειτουργίες i/o. Στο Mega, ο ακροδέκτης SS του υλικού,

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

53, δεν χρησιμοποιείται για να επιλέξει ούτε την κάρτα W5100 ούτε την κάρτα SD, αλλά πρέπει να διατηρηθεί ως έξοδος ή η διεπαφή SPI δεν θα λειτουργήσει.



Εικόνα: 5.1 Arduino Ethernet Shield W5100

Λάβετε υπόψη ότι επειδή η κάρτα W5100 και SD μοιράζονται το δίαυλο SPI, μόνο μία μπορεί να είναι ενεργή τη φορά. Εάν χρησιμοποιείτε και τα δύο περιφερειακά στο πρόγραμμά σας, αυτό θα πρέπει να ληφθεί μέριμνα από τις αντίστοιχες βιβλιοθήκες. Αν όμως δεν χρησιμοποιείτε κάποια από τις περιφερειακές συσκευές του προγράμματός σας, θα πρέπει να την καταρτίσετε. Για να το κάνετε αυτό με τη κάρτα SD, ορίστε τον ακροδέκτη 4 ως έξοδο και γράψτε ένα υψηλό σε αυτό. Για το W5100, ρυθμίστε τον ψηφιακό ακροδέκτη 10 ως υψηλή απόδοση.

Διαθέτει θύρα LAN και κουμπί Reset το οποίο βοηθά στην επανεκκίνηση του Arduino και του όλου συστήματος αφού καθώς εφάπτεται με το Arduino Uno ο χρήστης είναι αδύνατο να επανεκκινήσει το σύστημα χειροκίνητα αφού το κουμπί του Arduino βρίσκεται μεταξύ των δύο πλακετών.

## 5.2 Breadboard

Breadboard είναι μια μονάδα στην οποία μπορούν να κατασκευαστούν κυκλώματα χωρίς τη χρήση συγκόλλησης μεταξύ των εξαρτημάτων. Χρησιμοποιούνται τόσο για προσωρινά όσο και για μόνιμα κυκλώματα και μπορούν να κατασκευαστούν από τα πιο απλά έως τα πιο πολύπλοκα κυκλώματα. Λόγω του ότι τα εξαρτήματα δεν απαιτούν συγκόλληση, είναι επίσης ιδανικά για πειραματικά κυκλώματα ώστε να μη χρειάζεται να κολληθούν οι συνδέσεις κάθε φορά. Στις mini breadboard οι κάθετες νοητές γραμμές που σχηματίζουν οι συνδέσεις μεταξύ τους είναι συνδεδεμένες με μεταλλικές λωρίδες στις οποίες υπάρχουν κολλημένα κλιπ. Έτσι ο ακροδέκτης που συνδέεται σε μια σύνδεση της mini breadboard είναι συνδεδεμένος ηλεκτρικά με τις κάθετες σε αυτόν συνδέσεις λόγω των μεταλλικών λωρίδων.

Στα breadboard έχουμε επιπλέον 2 σειρές στις άκρες οι οποίες συνδέονται οριζόντια αντί για κάθετα και προορίζονται για παροχή ρεύματος στο κύκλωμα.



Εικόνα: 5.2 Breadboard

## 5.3 Jumper Wires

Jumper wires χρησιμοποιούνται για τη σύνδεση και την επικοινωνία μεταξύ των αισθητήρων και του Arduino αλλά και την παροχή ρεύματος.

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

Ένα καλώδιο άλματος (γνωστό και ως βραχυκύκλωμα, καλώδιο βραχυκυκλωτήρα, καλώδιο βραχυκυκλώματος, καλώδιο DuPont ή καλώδιο DuPont - το όνομα του κατασκευαστή του) είναι ένα ηλεκτρικό σύρμα ή μια ομάδα από αυτά σε καλώδιο με συνδετήρα ή πείρο σε κάθε άκρο.



Εικόνα: 5.3 Jumper Wire

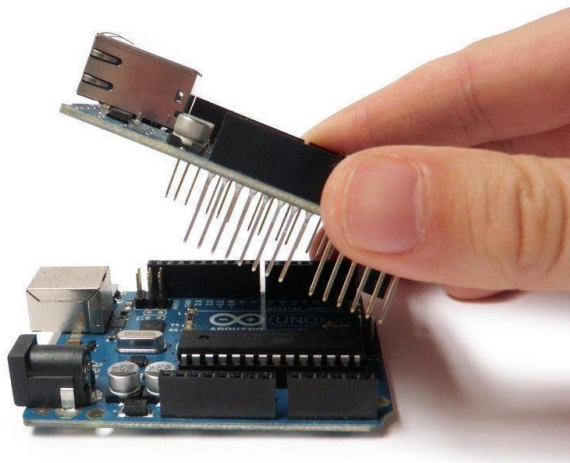


## ΚΕΦΑΛΑΙΟ 6

### ΣΥΝΔΕΣΗ ΠΕΡΙΦΕΡΙΑΚΩΝ – ΑΙΣΘΗΤΗΡΩΝ & ΥΛΟΠΟΙΗΣΗ

#### 6.1 Κάρτα επέκτασης δικτύου (Ethernet Shield) W5100

Αρχικά το Arduino Uno βρίσκεται κάτω από το Arduino Ethernet Shield W5100, το οποίο όμως έχει ακριβώς τους ίδιους ακροδέκτες που επικοινωνούν με το πρώτο. Η χρήση του Arduino Ethernet Shield W5100 έγινε συγκεκριμένα για την υποδοχή SD κάρτας μνήμης που προσφέρει καθώς το Uno δεν διαθέτει αυτή τη δυνατότητα. Για τη χρήση της υποδοχής της SD δεσμεύονται από το αναπτυξιακό σύστημα οι ψηφιακοί ακροδέκτες 10,11,12,13 καθώς και ο SPI δίαυλος. Σαφώς υπάρχει εξάρτημα που με συγκεκριμένη συνδεσμολογία μας προσφέρει αυτήν την υποδοχή.



Εικόνα 6.1: Ethernet Shield W5100 with Arduino Uno

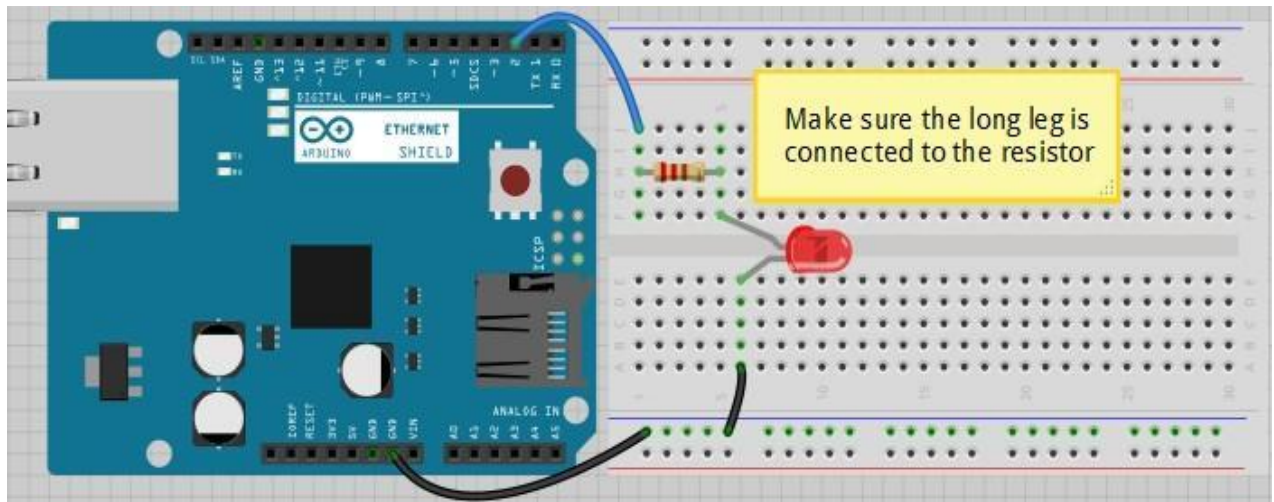
Όπως παρουσιάζεται στην ακόλουθη εικόνα, στη σειρά ένα (μαύρο καλώδιο) συνδέουμε ένα καλώδιο από τον ακροδέκτη της γείωσης GND του Arduino. Έτσι κάθε φορά που θέλουμε να συνδέσουμε κάποιο εξάρτημα με γείωση θα ενώνουμε τον ακροδέκτη του εξαρτήματος με τη σειρά ένα (μαύρο καλώδιο).

Ομοίως για τη τροφοδοσία συνδέουμε ένα καλώδιο από τον ακροδέκτη 5V του Arduino με την σειρά δύο (κόκκινο καλώδιο). Πάλι για τροφοδοσία των εξαρτημάτων θα ενώνουμε το εξάρτημα με την συγκεκριμένη σειρά. Αυτό γίνεται διότι υπάρχουν

Παγώνη Σ. Χριστίνα

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

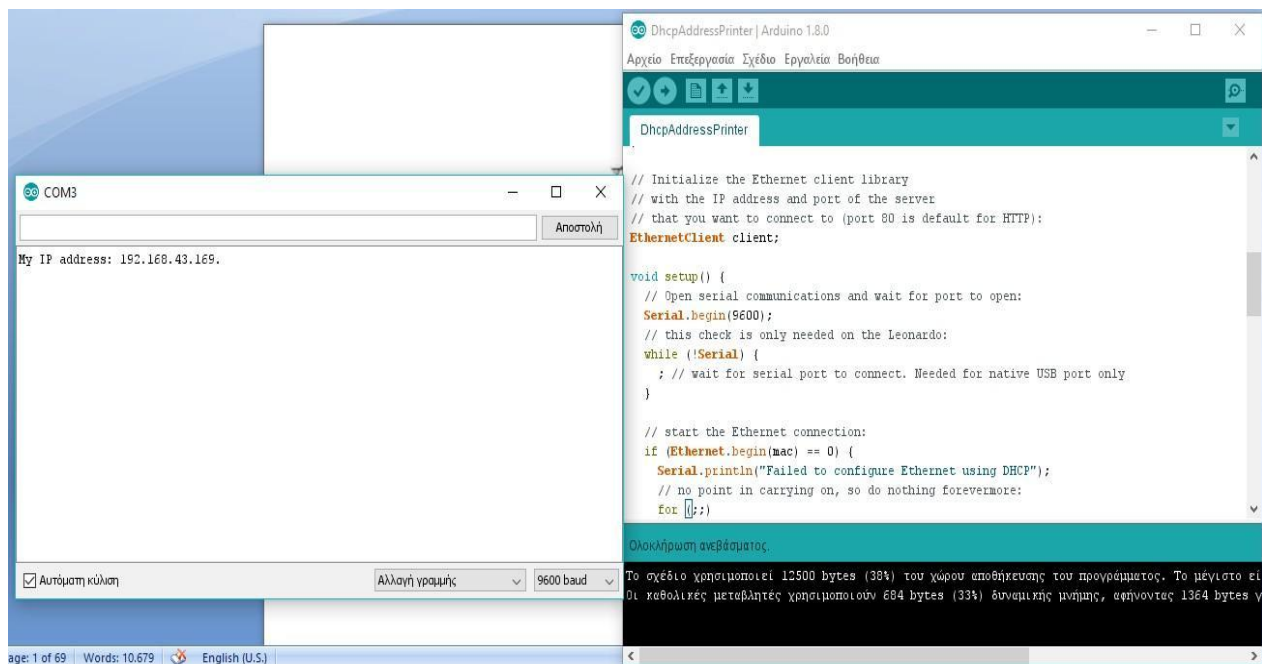
στο συγκεκριμένο κύκλωμα πολλοί αισθητήρες οι οποίοι χρειάζονται πάντα υποχρεωτικά τροφοδοσία και γείωση. Αφού η τροφοδοσία και η γείωση έχουν συνδεθεί για κάθε εξάρτημα, καθένα από αυτά έχει ακροδέκτες και επικοινωνία με το Arduino.



Εικόνα 6.2: Σύνδεση Arduino - W5100 - Breadbord

Εφόσον συνδέσουμε το arduino uno και το Ethernet shield w5100, θα πρέπει να συνδέουμε μέσω usb καλώδιο το arduino uno με τον ηλεκτρονικό υπολογιστή, ο οποίος θα αποτελεί τη πηγή τροφοδοσίας.

Έπειτα συνδέουμε και το καλώδιο Ethernet τόσο στον ηλεκτρονικό υπολογιστή όσο και στο Ethernet shields w5100. Ανοίγουμε την εφαρμογή Arduino 1.8.0 και ανεβάζουμε στο arduino το παράδειγμα Dhcp Address Printer (Ο κώδικας του παραδείγματος συμπεριλαμβάνεται στο παράρτημα Α). Το παράδειγμα αυτό θα μας τυπώσει στη σειριακή έξοδο (serial monitor) την Ip διεύθυνση, με την οποία βγαίνουμε στο internet.



Εικόνα 6.3: Εκτύπωση ip διεύθυνσης

Για τον προγραμματισμό του arduino θα χρειαστούμε και τη default gateway καθώς και τη μάσκα υποδικτύου. Τα δεδομένα αυτά μπορούμε να τα πάρουμε ανάγοντας το command prompt (cmd) και πληκτρολογώντας την εντολή ipconfig, θα μας εμφανιστούν όλα τα στοιχεία για το δίκτυο, καθώς και η ip διεύθυνση την οποία την βρήκαμε εκτελώντας το παράδειγμα Dhcp Address Printer. Ωστόσο μπορούμε για τον προγραμματισμό του arduino να ορίσουμε εμείς με ποια ip διεύθυνση (π.χ. IPAddress ip(192, 168, 43, 200);) θα βγαίνουμε στο internet.

## 6.2 Αισθητήρας Καταγραφής Θερμοκρασίας Υγρασίας DHT11

Οι αισθητήρες καταγραφής συνδέονται στις ψηφιακές και αναλογικές εισόδους του Arduino το οποίο κάνοντας χρήση των κατάλληλων βιβλιοθηκών, διαβάζει τις τιμές τάσης στη είσοδο του και στη συνέχεια μετατρέπει την εισερχόμενη τάση σε τιμές τις οποίες μπορούμε να αξιοποιήσουμε

Τα δεδομένα του αισθητήρα είναι ψηφιακά σαφώς λοιπόν ο τρίτος ακροδέκτης (out) θα συνδεθεί σε μία από τις ψηφιακές υποδοχές 0-13 του αναπτυξιακού(Εκτός των 10,11,12,13)

Τοποθετούμε στο breadboard τον αισθητήρα DHT11. Ο αισθητήρας DHT11

λειτουργεί με τάση 5V. Έτσι συνδέουμε ένα καλώδιο με τάση 5 Volt στο 1<sup>ο</sup> pin του

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

αισθητήρα, ένα καλώδιο που χρησιμοποιείται ως analog out δηλαδή το επιστρεφόμενο σήμα προς τη πλακέτα arduino στο 2<sup>ο</sup> pin του αισθητήρα και ένα καλώδιο γείωσης (ground) στο 4<sup>ο</sup> pin και συνδέονται τα analog out στις θέσεις A0 στο arduino.

## 6.3 Υλοποίηση

### 6.3.1 Πρόγραμμα και αξιοποιήσιμες βιβλιοθήκες

Όπως προαναφέρθηκε, για τον προγραμματισμό του Arduino χρησιμοποιείται η γλώσσα wiring που είναι παρόμοια με την C++. Ο συνολικός κώδικας, γράφτηκε στο IDE Arduino 1.8.0 για τον προγραμματισμό της πλακέτας Arduino. Αξιοποιήθηκαν βιβλιοθήκες που έχουν δημιουργηθεί από τους κατασκευαστές των αισθητήρων και υπάρχουν στην κατηγορία του “ανοιχτού κώδικα”. Για την αποθήκευση των καταγραφόντων δεδομένων χρειαζόμαστε μια κάρτα SD όπου θα αποθηκεύονται τα δεδομένα σε μορφή CSV.







Το περιβάλλον ανάπτυξης Arduino περιέχει μια περιοχή επεξεργασίας κειμένου για τη συγγραφή κώδικα, μια περιοχή μηνυμάτων, ένα μενού, μια γραμμή εργαλείων με κουμπιά για κοινές λειτουργίες, καθώς και μια σειρά από μενού. Συνδέεται με το υλικό Arduino για τη φόρτωση προγραμμάτων και για να επικοινωνούν μεταξύ τους.

### 6.3.2 Sketches

Ένα ολοκληρωμένο πρόγραμμα συνήθως ονομάζεται sketch. Αυτό το sketch είναι γραμμένο με το πρόγραμμα επεξεργασίας κειμένου. Έχει δυνατότητες για την αντιγραφή/επικόλληση και για την αναζήτηση/αντικατάσταση κειμένου. Η περιοχή μηνυμάτων μας δείχνει χρήσιμες πληροφορίες καθώς αποθηκεύουμε ή κάνουμε εξαγωγή το sketch μας καθώς και τυχών σφάλματα (errors) που θα προκύψουν.

Η κονσόλα απεικονίζει την έξοδο του κειμένου από το περιβάλλον Arduino συμπεριλαμβάνοντας πλήρη μηνύματα λάθους και άλλες πληροφορίες. Η κάτω δεξιά γωνία του παραθύρου απεικονίζει τον μικροελεγκτή Arduino που έχουμε επιλέξει καθώς και την σειριακή θύρα που βρίσκεται συνδεδεμένος (SerialPort). Τα κουμπιά της γραμμής εργαλείων επιτρέπουν τον έλεγχο και το ανέβασμα των προγραμμάτων, τη δημιουργία νέου sketch, το άνοιγμα και την αποθήκευση των sketch και άνοιγμα της σειριακής οθόνης.

Τα κουμπιά της γραμμής εργαλείων:

	Verify/Compile (Έλεγχος / Μεταγλώττιση): Έλεγχος για λάθη στον κώδικα
	Upload: Ανέβασμα του κώδικα στον μικροελεγκτή
	New(Νέο): Δημιουργεί ένα νέο sketch
	Open(Άνοιγμα): Παρουσιάζει ένα μενού με όλα τα sketch, κάνοντας κλικ σε ένα από αυτά θα ανοίξει μέσα στο τρέχον παράθυρο
	Save(Αποθήκευση): Αποθηκεύει το sketch
	Serial Monitor(Σειριακή οθόνη): Ανοίγει την σειριακή οθόνη ώστε να μπορούμε να δώσουμε δεδομένα από το πληκτρολόγιο

Πίνακας 2: Επεξήγηση κουμπιών της γραμμής εργαλείων

Στην παρούσα εργασία, για την λειτουργία του συστήματος, συγγράφηκε ένας κώδικας ο οποίος διαβάζει το analog out από κάθε αισθητήρα DHT11, το μετατρέπει σε τιμές θερμοκρασίας και υγρασίας και αποθηκεύει τις συγκεκριμένες τιμές ανά Χλεπτά στην κάρτα micro SD.

Όπως φαίνεται στην αρχή του κώδικα (ΠΑΡΑΡΤΗΜΑ Α) έχουν συμπεριληφθεί 3 libraries, έτοιμοι κώδικες δηλαδή που υπάρχουν στο Arduino και είναι η DHT.h , η SPI.h και η SD.h. Η πρώτη βοηθάει στη μετατροπή του analog out από τον αισθητήρα σε λογικές τιμές θερμοκρασίας και υγρασίας ενώ οι υπόλοιπες στην αποθήκευση των τιμών στη microSD card.

Επιπλέον εντολές βρίσκονται στα πέντε μενού: File, Edit, Sketch, Tools, Help. Τα μενού είναι context sensitive, το οποίο σημαίνει πως μόνο τα χαρακτηριστικά που σχετίζονται με το πρόγραμμα το οποίο φτιάχνουμε είναι διαθέσιμα.

### 6.3.3 Μενού File

- New

Δημιουργεί μια νέα φόρμα του Editor στην οποία βρίσκεται η βασική δομή ενός

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

Sketch.

- Open

Μας επιτρέπει να ανοίξουμε ένα παράθυρο πλοήγησης στον υπολογιστή για να βρούμε ένα υπάρχων αρχείο sketch.

- Open Recent

Μας παρέχει μία μικρή λίστα από τα πιο πρόσφατα Sketches, έτοιμα για άνοιγμα.

- Sketchbook

Μας ανοίγει ένα παράθυρο πλοήγησης μέσα στο φάκελο sketch book των υπάρχοντων sketches.

- Examples

Κάθε παράδειγμα που παρέχεται από το Arduino IDE ή κάθε βιβλιοθήκη εμφανίζεται σε αυτό το μενού. Όλα τα παραδείγματα είναι δομημένα σε μορφολογία δέντρου, κάτι το οποίο μας παρέχει εύκολη πρόσβαση ανά αντικείμενο ή βιβλιοθήκη.

- Close

Κλείνει τον editor του IDE τον οποίο επιλέξαμε.

- Save

Αποθηκεύει το Sketch με το ήδη υπάρχον όνομα. Εάν το αρχείο δεν έχει αποθηκευτεί προηγουμένως, ένα όνομα θα του δοθεί στο παράθυρο "Save as.."

- Save as...

Μας δίνει την δυνατότητα να αποθηκεύσουμε το ήδη υπάρχων sketch με ένα διαφορετικό όνομα.

- Page set up

Μας δείχνει το παράθυρο Page Setup για εκτύπωση του προγράμματος.

- Print

Εκτυπώνει το πρόγραμμα μας σύμφωνα με τις ρυθμίσεις που επιλέξαμε από το Παγώνη Σ. Χριστίνα

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

παράθυρο Page Set up.

- Preferences

Ανοίγει το παράθυρο προτιμήσεων του IDE όπου μπορούν κάποιες ρυθμίσεις του IDE να παραμετροποιηθούν , όπως πχ. Η γλώσσα του περιβάλλοντος.

- Quit

Κλείνει όλα τα παράθυρα του IDE. Τα ίδια sketch που ήταν ανοιχτά την στιγμή που πατήσαμε το κουμπί Quit , αυτόματα θα ξανανοίξουν της επόμενης φορά που θα ανοίξουμε το Arduino IDE.

### 6.3.4 Edit

- Undo/Redo

Πηγαίνει ένα η περισσότερα βήματα πίσω καθώς κάναμε edit το sketch μας.

- Cut /Copy /Paste

Αφαιρεί /Αντιγράφει /Τοποθετεί το επιλεγμένο κείμενο από τον editor και το τοποθετεί στο clipboard.

- Copy for Forum

Αντιγράφει τον κώδικα του Sketch στο clipboard σε μία φόρμα κατάλληλη για κοινοποίηση σε ένα forum, ολοκληρωμένο μαζί με το syntax coloring.

- Copy as HTML

Αντιγράφει τον κώδικα του Sketch στο clipboard σε μορφή HTML, κατάλληλο για εισαγωγή σε ιστοσελίδες.

- Select All

Επιλέγει όλο το περιεχόμενο κειμένου του editor.

- Comment/Uncomment

Τοποθετεί ή αφαιρεί τα // σημάδια σχολίων στην αρχή της κάθε επιλεγμένης γραμμής.

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

- Increase/Decrease Indent

Προσθέτει ή αφαιρεί ένα διάστημα στην αρχή της κάθε επιλεγμένης γραμμής, μετακινώντας το κείμενο ένα διάστημα προς τα δεξιά ή προς τα αριστερά.

- Find

Ανοίγει το παράθυρο Find and Replace στο οποίο μπορούμε να εισάγουμε το κείμενο που θέλουμε να βρούμε μέσα στον υπάρχων sketch μέσα από διάφορες επιλογές.

- Find Next

Υπογραμμίζει το επόμενο ίδιο κείμενο – εάν υπάρχει- μέσα στο ανοιχτό αρχείο στον text editor.

- Find Previous

Υπογραμμίζει το προηγούμενο ίδιο κείμενο – εάν υπάρχει- μέσα στο ανοιχτό αρχείο στον text editor.

### 6.3.5 Sketch

- Verify/Compile

Έλεγχος του sketch για σφάλματα κάνοντας το compile. Θα μας δώσει αναφορά για την μνήμη που χρησιμοποιεί ο κώδικας και οι μεταβλητές στην περιοχή της κονσόλας.

- Upload

Κάνει compile και φορτώνει το binary αρχείο στον προσαρμοσμένο μικροελεγκτή Arduino μας μέσω της αντίστοιχης σειριακής θύρας.

- Upload Using Programmer

Αυτή η επιλογή θα παρακάμψει τον boot loader που βρίσκεται στον μικροελεγκτή Arduino. Πρέπει να χρησιμοποιήσουμε την επιλογή Tools>BurnBootloader για να επαναφέρουμε τον boot loader και να μπορέσουμε να κάνουμε πάλι Upload στην USB σειριακή θύρα μας ξανά. Ωστόσο, η επιλογή αυτή μας δίνει την δυνατότητα να



Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

χρησιμοποιήσουμε όλη την χωρητικότητα της Flash μνήμης για το sketch μας. Αυτή η επιλογή δεν θα κάψει τις ασφάλειες.

- Export Compiled Binary

Αποθηκεύει ένα .hex αρχείο το οποίο μπορεί να αρχειοθετηθεί ή να σταλεί στο Arduino μας χρησιμοποιώντας άλλες μεθόδους.

- Show Sketch Folder

Ανοίγει τον υπάρχων φάκελο sketch.

- Include Library

Προσθέτει μια βιβλιοθήκη στο sketch μας εισάγοντας την με #include δήλωση στη αρχή του κώδικά μας. Επιπρόσθετα από το μενού αυτό έχουμε πρόσβαση στο Library Manager και να εισάγουμε νέες βιβλιοθήκες από .zip αρχεία.

- Add File...

Προσθέτει ένα source file στο sketch μας. (Αντιγράφεται από την υπάρχουσα θέση του). Το καινούριο αυτό αρχείο εμφανίζεται σε μία καινούργια καρτέλα στο sketch παράθυρο. Τα αρχεία μπορούν να αφαιρεθούν από το sketch χρησιμοποιώντας το tab menu κλικάροντας το μικρό τριγωνικό εικονίδιο κάτω από serial monitor στην δεξιά πλευρά του toolbar.

### 6.3.6 Tools

- Auto Format

Αυτόματη μορφοποίηση του κώδικα με σκοπό να είναι όμορφος οπτικά και ευανάγνωστος για τον debugging.

- Archive Sketch

Δημιουργεί ένα αντίγραφο του υπάρχοντος sketch σε .zip μορφή και αποθηκεύεται στο ίδιο directory που υπάρχει το sketch.

- Fix Encoding & Reload

Φτιάχνει τις τυχών διαφορές της κωδικοποίησης του editor charmap με των διαφορετικών λειτουργικών συστημάτων charmaps.

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

- Serial Monitor

Ανοίγει το serial monitor παράθυρο και ξεκινάει την ανταλλαγή δεδομένων με το οποιοδήποτε συνδεδεμένο Arduino μικροελεγκτή στην επιλεγμένη θύρα.

- Board

Επιλέγουμε το μοντέλο του Arduino μικροελεγκτή που χρησιμοποιούμε.

- Port

Αυτό το μενού περιέχει όλες στις σειριακές συσκευές στον υπολογιστή μας. Συνήθως ανανεώνεται αυτόματα κάθε φορά που ανοίγουμε τα επάνω παράθυρα από το tools menu.

- Programmer

Για την επιλογή νέου hardware programmer όταν προγραμματίζουμε ένα μικροελεγκτή ή chip χωρίς να χρησιμοποιούμε την ενσωματωμένη USB σειριακή σύνδεση. Συνήθως δεν μας χρειάζεται αυτή η λειτουργία παραμόνο ένα κάνουμε burn ένα νέο boot loader σε ένα καινούριο μικροελεγκτή.

- Burn Boot loader

Τα στοιχεία σε αυτό το μενού μας παρέχουν την δυνατότητα να γράψουμε έναν boot loader μέσα στον μικροελεγκτή ενός Arduino. Αυτό δεν χρειάζεται για την κανονική χρήση νέος Arduino ή Genuino αλλά είναι χρήσιμο ένα αγοράσουμε έναν καινούργιο ATmega μικροελεγκτή ( οι οποίοι συνήθως έρχονται χωρίς προ εγκατεστημένο bootloader). Σιγουρευτείτε ότι έχετε επιλέξει το σωστό board από το μενού Boards προτού γράψετε ένα boot loader στο αντίστοιχο board. Ακόμη, η επιλογή αυτή θα ρυθμίσει τις σωστές ασφάλειες.

### 6.3.7 Help

Σε αυτό το μενού μπορούμε να βρούμε εύκολα δεκάδες έγγραφα που παρέχονται με την εγκατάσταση του Arduino IDE. Έχουμε πρόσβαση στους οδηγούς Getting Started , Reference του IDE και σε άλλα έγγραφα χωρίς σύνδεση στο διαδίκτυο.

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

- Find In Reference

Η μόνη επιλογή στο μενού αυτό. Επιλέγει απευθείας την συσχετισμένη σελίδα από το τοπικό αντίγραφο της Reference για την λειτουργία ή εντολή κάτω από την θέση του κέρσορα.

### 6.3.8 Sketch book

Το Arduino IDE χρησιμοποιεί ένα πρότυπο μοντέλο sketch book: Ένα προκαθορισμένο φάκελο που μπορείς να αποθηκεύσεις τα προγράμματα ή sketches. Τα sketches που έχουμε δημιουργήσει μπορούν να ανοιχτού από το μενού File>Sketchbook ή από το κουμπί Open στην γραμμή εργαλείων. Την πρώτη φορά που θα χρησιμοποιήσουμε το λογισμικό αυτό, θα δημιουργήσει αυτόματα αυτό το φάκελο και θα τον καθορίσει ως το Sketchbook μας. Μπορούμε να βρούμε ή ακόμη και να αλλάξουμε την διαδρομή/τοποθεσία του sketchbook μας από την επιλογή Preferences στο IDE.

### 6.3.9 Uploading

Προτού φορτώσουμε (Upload) το sketch μας στο Arduino, πρέπει να έχουμε επιλέξει το σωστό Arduino και την σωστή σειριακή θύρα στα μενού Tools>Board και Tools>Port αντίστοιχα. Στα Mac , η σειριακήθύρα έχει συνήθως την μορφή /dev/tty.usbmodem241 (για το Uno ή Mega2560 ή το Leonardo) ή /dev/tty.usbserial-1B1 (για το Duemilanove ή κάποιο παλαιότερο USBboard), ή /dev/tty.USA19QW1b1P1.1 (για κάποια σειριακή πλακέτα (serialboard) συνδεδεμένη με έναν KeyspanUSB-to-Serial αντάπτορα). Στα Windows στα οποία θα υλοποιηθεί η εργασία μας συνήθως έχει την μορφή COM1 ή COM2 (για κάποια σειριακή πλακέτα) ή COM 4, COM5, COM7, κλπ (για πλακέτα με σύνδεση USB).Για να βρούμε την θύρα αυτή μπορούμε να ψάξουμε για μια USB serial device μέσα στο Device Manager των Windows. Στα Linux έχει την μορφή / dev/ ttyACMx , / dev/ ttyUSBx ή παρόμοια. Μόλις επιλέξουμε την σωστή σειριακή θύρα και πλακέτα Arduino, πατάμε το κουμπί upload στη γραμμή εργαλείων ή το κουμπί Upload από το Sketch μενού. Με την διαδικασία αυτή το Arduino μας θα κάνει reset και θα ξεκινήσει η διαδικασία φόρτωσης του κώδικα. Με Arduino παλαιότερα του Diecimila που δεν έχουν την δυνατότητα auto-reset (αυτόματης επαναφοράς) θα πρέπει να χρησιμοποιήσουμε το κουμπί reset στο Arduino αμέσως πριν ξεκινήσει η φόρτωση

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

του κώδικα. Στα περισσότερα Arduino θα δούμε τα LEDRX και TX να αναβοσβήνουν καθώς τους φορτώνουμε τον κώδικα του sketch.

Το Arduino Software (IDE) θα μας εμφανίσει ένα μήνυμα όταν η φόρτωση (upload) του sketch ολοκληρωθεί η αλλιώς ένα μήνυμα σφάλματος. Όταν φορτώνουμε ένα sketch, χρησιμοποιούμε τον Arduino bootloader , ένα μικρό προ εγκατεστημένο πρόγραμμα στον μικροελεγκτή του Arduino. Αυτό μας παρέχει την δυνατότητα να φορτώσουμε τον κώδικα μας χωρίς να χρειαστούν επιπλέον εργαλεία. Ο boot loader είναι ενεργός για μερικά δευτερόλεπτα καθώς το Arduino κάνει reset, μετά ξεκινάει με όποιο ήταν το τελευταίο Sketch το οποίο φορτώθηκε στον μικροελεγκτή. Ο boot loader θα αναβοσβήσει το pin 13 LED κάθε φορά που θα ξεκινήσει την λειτουργία του όπως πχ κατά την διαδικασία του reset.

### 6.3.10 Libraries (Βιβλιοθήκες)

Οι βιβλιοθήκες μάς παρέχουν επιπλέον λειτουργίες για τα sketch μας, όπως πχ. Να χρησιμοποιούμε διάφορα ηλεκτρονικά στοιχεία ή να χρησιμοποιούμε δεδομένα. Για να χρησιμοποιήσουμε μια βιβλιοθήκη σε ένα sketch , μπορούμε να την επιλέξουμε από το μενού Sketch>ImportLibrary . Με τον τρόπο αυτό μπορούμε να εισάγουμε μια ή περισσότερες δηλώσεις #include στο πάνω μέρος του sketch μας και να προσθέσουμε την βιβλιοθήκη αυτή στο sketch. Καθώς οι βιβλιοθήκες φορτώνονται στο Arduino μέσω του sketch, μεγαλώνουν τον χώρο που καταλαμβάνει το sketch στο Arduino. Ένα δεν μας χρειάζεται πια μία βιβλιοθήκη, απλά διαγράφουμε την #include δήλωσή της στο πάνω μέρος του κώδικα του Sketch μας. Υπάρχει μια μεγάλη λίστα βιβλιοθηκών. Κάποιες βιβλιοθήκες είναι προ εγκατεστημένες στο λογισμικό του Arduino. Άλλες μπορούν να κατεβαστούν από διάφορες πηγές μέσω του LibraryManager. Με βάση την έκδοση 1.0.5 του IDE μπορούμε να εισάγουμε μια βιβλιοθήκη ακόμη και από ένα zip αρχείο και να την χρησιμοποιήσουμε σε κάποιο sketch . Ακόμη μπορούμε να συγγράψουμε την δικιά μας βιβλιοθήκη.

### 6.3.11 Third-Party Hardware

Η υποστήριξη υλικών τρίτων κατασκευαστών μπορεί να προστεθεί στο Hardware directory του sketchbook. Εγκατεστημένες πλατφόρμες οι οποίες μπορεί να

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

περιέχουν ορισμούς συσκευών (οι οποίες εμφανίζονται στο μενού board), βιβλιοθήκες πυρήνα, boot loaders και προγραμματιστικούς ορισμούς. Για την εγκαταστάσεις τέτοιου είδους, δημιουργούμε ένα hardware directory, μετά κάνουμε εξαγωγή (unzip) την πλατφόρμα τρίτων κατασκευαστών στο δικό της υποκατάλογο (sub-directory).

Ποτέ δεν χρησιμοποιούμε το όνομα “arduino” ως το όνομα του υποκαταλόγου αλλιώς θα γράψουμε πάνω στην προ εγκατεστημένη πλατφόρμα του Arduino. Για να απεγκαταστήσουμε μια πλατφόρμα απλά διαγράψουμε τον υποκατάλογο της.

### **6.3.12 Serial Monitor**

Το Serial Monitor απεικονίζει την σειριακή αποστολή δεδομένων από το Arduino μέσω σειριακής επικοινωνίας ή USB. Για να στείλουμε δεδομένα στο Arduino, γράφουμε το κείμενο που θέλουμε και πατάμε το κουμπί “send” ή το enter του πληκτρολογίου. Επιλέγουμε τον ρυθμό baud από το drop-down μενού που αντιστοιχεί στο επιθυμητό ρυθμό που έχουμε επιλέξει στο Serial.begin του sketch μας. Είναι καλό να γνωρίζουμε πως σε όλα τα λειτουργικά συστήματα Windows, Mac ή Linux το Arduino θα κάνει reset καθώς συνδέεται με το serial monitor. Το Serial Monitor δεν επεξεργάζεται χαρακτήρες ελέγχου και εάν το sketch μας χρειάζεται πλήρες έλεγχο για σειριακή επικοινωνία με χαρακτήρες ελέγχου, μπορούμε να χρησιμοποιήσουμε ένα εξωτερικό τερματικό και να το συνδέσουμε με την COM θύρα που είναι συνδεδεμένο το Arduino μας.

### **6.3.13 Preferences (Προτιμήσεις)**

Κάποιες προτιμήσεις μπορούν να ρυθμιστούν από το μενού Preferences όπου βρίσκεται κάτω από το μενού Arduino στα Mac ή το μενού File στα Windows ή Linux. Οι υπόλοιπες μπορούν να βρεθούν στο αρχείο preferences που ή τοποθεσία του απεικονίζεται στο preference dialog.

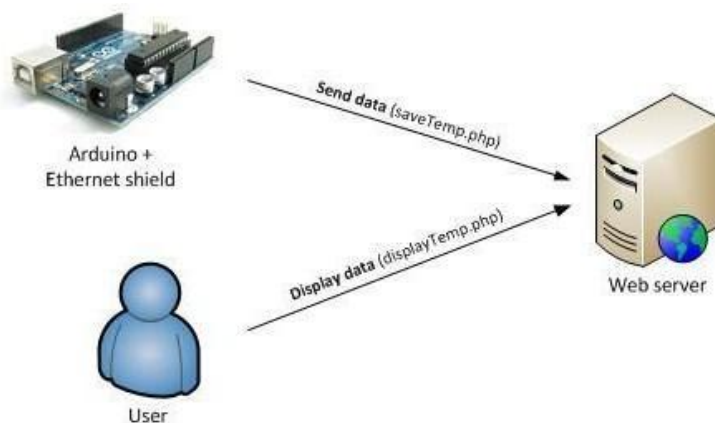
Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

## ΚΕΦΑΛΑΙΟ 7

### ΔΙΕΠΑΦΗ ΧΡΗΣΤΗ

Το περιβάλλον χρήστη βασίζεται στην τεχνολογία HTML με χρήση JavaScript. Η μεταφορά δεδομένων από το Arduino στην ιστοσελίδα πραγματοποιείται μέσω της τεχνολογίας XML, ενώ η προβολή των γραφημάτων γίνεται με τη χρήση έτοιμης βιβλιοθήκης JavaScript η οποία είναι ανοιχτού κώδικα και διατίθεται δωρεάν. Η διαδικασία λειτουργίας είναι η ακόλουθη:

Αρχικά αποστέλλεται από τον φυλλομετρητή, http αίτημα στο Arduino. Στην συνέχεια το αίτημα αναλύεται και παρουσιάζεται στον φυλλομετρητή. Οι ιστοσελίδες βρίσκονται αποθηκευμένες στη κάρτα μνήμης SD υπό τη μορφή αρχείων HTML. Για το λόγο αυτό θα πρέπει να ενσωματωθεί η βιβλιοθήκη SD για τη διαχείριση της κάρτας SD. Για οικονομία χώρου καθώς και της κίνησης μικρότερου όγκου δεδομένων, οι εικόνες και βιβλιοθήκες JavaScript που χρησιμοποιούνται δεν είναι αποθηκευμένες στον σταθμό, αλλά σε άλλους εξυπηρετητές.



Εικόνα 7.1: Διάγραμμα ροής διαδικασίας εξυπηρετητή διαδικτύου

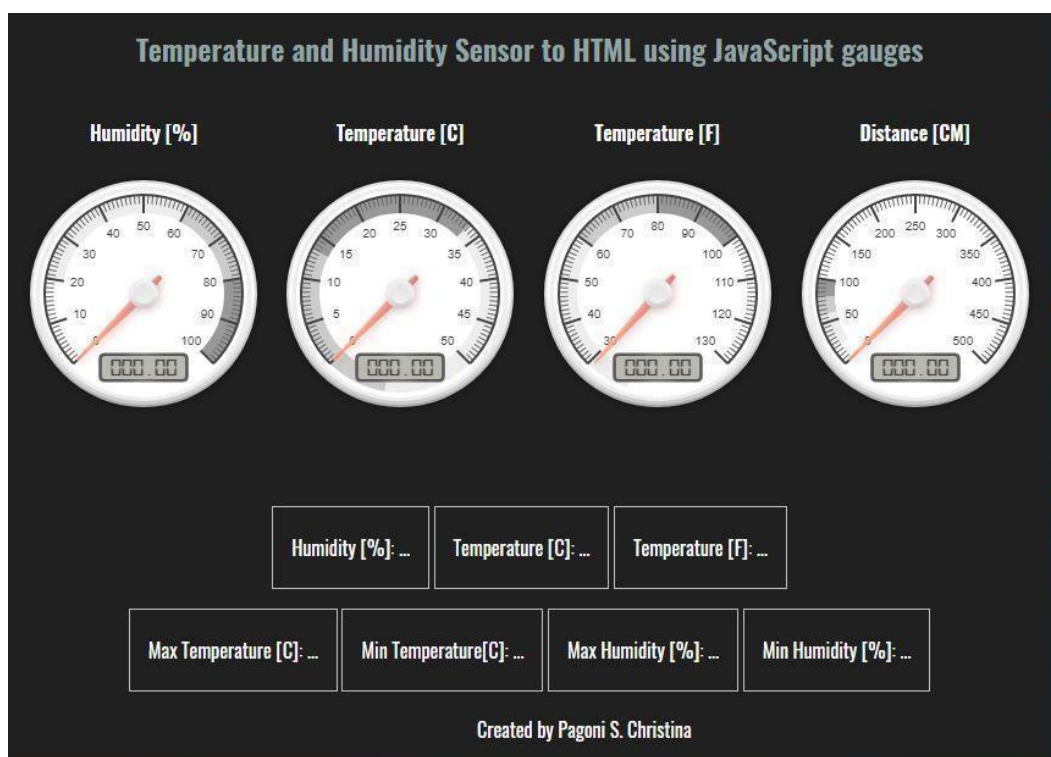
#### 7.1 Αρχική Σελίδα

Είναι η σελίδα στην οποία ο χρήστης παρακολουθεί όλες τις πληροφορίες για τις μετρήσεις που λαμβάνουμε. Αρχικά ο χρήστης εισάγει την διεύθυνση IP την οποία έχουμε ορίσει στο πρόγραμμα. Για τις ανάγκες της παρούσης εργασίας, η διεύθυνση IP δοκιμάστηκε σε τοπικό δίκτυο με διεύθυνση 192.168.43.200.

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

Μόλις ο χρήστης πληκτρολογήσει τη διεύθυνση στη γραμμή εντολών του φυλλομετρητή, αποστέλλεται το αίτημα http "HTTP/1.1 GET /" ή "HTTP/1.1 GET /index2.htm" το οποίο μεταφράζεται ως:

«πρωτόκολλο HTTP έκδοση 1.1 ζητώ την σελίδα index.html». Το Arduino εφόσον ελέγξει την ύπαρξη του αρχείου στη κάρτα SD, το αποστέλλει στον χρήστη. Η σελίδα που προβάλλεται είναι η κεντρική σελίδα και περιέχει τις μετρήσεις και τα αρχεία που υπάρχουν αποθηκευμένα μέσα στην microSD κάρτα, καθώς και το μέγεθός τους και την ημερομηνία τελευταίας τροποποίησής τους. Παρακάτω εμφανίζεται η κεντρική σελίδα του Arduino Ethernet.



Εικόνα 7.2: Αρχική Σελίδα

Αναλυτικά τα δεδομένα που προβάλλονται είναι:

- Τρέχουσα Υγρασία καθώς και η μέγιστη και ελάχιστη ημερήσια τιμή της.
- Τρέχουσα Θερμοκρασία σε Celsius, τη μέγιστη και ελάχιστη ημερήσια τιμή της.
- Τρέχουσα Θερμοκρασία σε Fahrenheit, τη μέγιστη και ελάχιστη ημερήσια τιμή της.
- Τρέχουσα Απόσταση καθώς και η μέγιστη και ελάχιστη ημερήσια τιμή της.

Όταν δεν αιτηθεί η προβολή κάποιου αρχείου, τότε το Arduino αποστέλλει ροή η οποία περιλαμβάνει όλες τις παραπάνω πληροφορίες. Στην συνέχεια ο



Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

φυλλομετρητής φορτώνει το αρχείο που αιτήθηκε ο εκάστοτε χρήστης και εμφανίζεται στο φυλλομετρητή το περιεχόμενο του αρχείου.

Το παραγόμενο XML έχει την παρακάτω δομή:

```
<?xmlversion = "1.0"?>
<inputs>
<tempC></tempC>//τρέχουσα θερμοκρασία σε *C
<tempF></tempF>//τρέχουσα θερμοκρασία σε F
<maxtempC></maxtempC>//μέγιστη θερμοκρασία σε *C
<mintempC></mintempC>//ελάχιστη θερμοκρασία σε *C
<maxtempF></maxtempF>//μέγιστη θερμοκρασία σε F
<mintempF></mintempF>//ελάχιστη θερμοκρασία σε F
<humidity></humidity>//τρέχουσα υγρασία %
<maxhum></maxhum>//μέγιστη υγρασία %
<minhum></minhum>//ελάχιστη υγρασία %
<distance></distance>//τρέχουσα απόσταση σε cm
</inputs>
```

Η χρήση των ελληνικών χαρακτήρων στο Arduino δεν είναι εύκολη υπόθεση καθώς λόγω της wiring, υποστηρίζεται ο πίνακας χαρακτήρων ASCII μόνο των 127 πρώτων χαρακτήρων. Βέβαια είναι δυνατή η προβολή των ελληνικών, αλλά απαιτείται προγραμματισμός και φυσικά δέσμευση της πολύτιμης μνήμης SRAM. Για τον λόγο αυτό ο Arduino αποστέλλει στην σελίδα έναν κωδικό. Στην σελίδα υπάρχει ένας πίνακας forecast σε γλώσσα JavaScript ο οποίος περιέχει την μετάφραση του αντίστοιχου λεκτικού στα ελληνικά. Για την άμεση κατανόηση των μετρήσεων εκτός του λεκτικού κειμένου, χρησιμοποιείται οπτικό βοήθημα που αναπαριστά την αντίστοιχη πρόγνωση με χρήση εικονιδίων. Εκτός του forecast υπάρχει και ο πίνακας forecast Icons ο οποίος περιέχει την πλήρη διαδρομή των εικόνων της κάθε κατάστασης των μετρήσεων στα ελληνικά.

Τέλος υπάρχει δυνατότητα προβολής των καταγεγραμμένων μεγίστων και ελαχίστων τιμών της θερμοκρασίας και υγρασίας υπό την μορφή γραφήματος.

## 7.2 Προβολή Στατιστικών

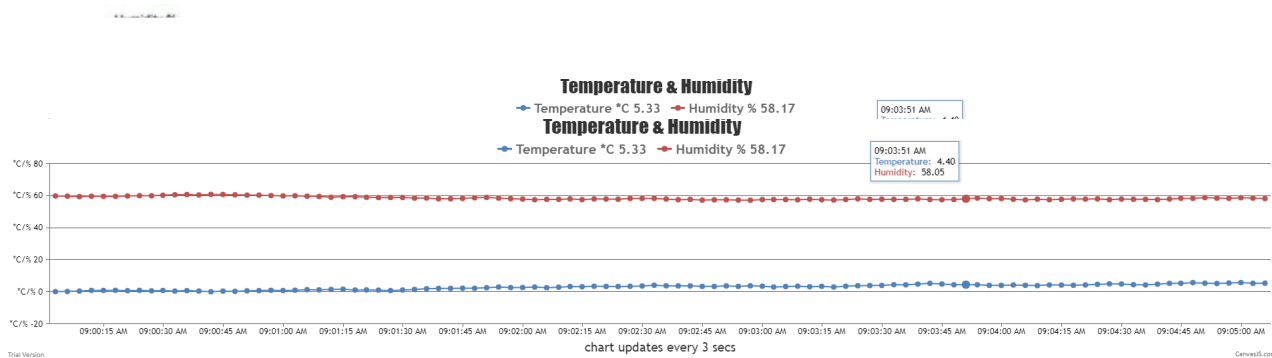
Ο Arduino αποθηκεύει τις τιμές της θερμοκρασίας και της υγρασίας. Η αποθήκευση

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

γίνεται σε μορφή csv στο αρχείο data.csv. Το αρχείο αποθηκεύεται σύμφωνα με το πρότυπο μορφοποίησης ημερομηνίας και αριθμών ISO 8601. Τα περιεχόμενα του αρχείου είναι:

Humidity, Temperature \*C, Temperature F, Distance CM

Στην αρχική σελίδα ο χρήστης έχει τη δυνατότητα να επιλέξει τον υπερσύνδεσμο “για να δείτε τα στατιστικά πατήστε εδώ” και να εμφανίσει τη σελίδα stats.html. Σε αυτή την σελίδα ο χρήστης μπορεί να δει τις τιμές των μετρήσεων που έχει καταγράψει ο σύστημα.



Εικόνα 7.3: Σελίδα προβολής Στατιστικών stats.htm

Στην παραπάνω Εικόνα εμφανίζεται η προβολή της σελίδας στατιστικών. Για την παραγωγή του γραφήματος χρησιμοποιήθηκε η βιβλιοθήκη digraph σε γλώσσα JavaScript. Η βιβλιοθήκη είναι ανοιχτού λογισμικού, και διατίθεται δωρεάν. Η βιβλιοθήκη δεν έχει αποθηκευτεί στο Arduino αλλά γίνεται η σύνδεση της απ’ ευθείας από την σελίδα του δημιουργού της, για εξοικονόμηση χώρου και όγκου των δεδομένων. Η κατασκευή άλλωστε, έχει σχεδιαστεί για διασύνδεση στο διαδίκτυο, οπότε η αναφορά σε άλλους ιστότοπους θεωρείται δεδομένη.

## ΚΕΦΑΛΑΙΟ 8

### ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ

#### 8.1 Σύνοψη της πτυχιακής εργασίας

Ο στόχος της παρούσης Πτυχιακής Εργασίας ήταν η υλοποίηση ενός Web Server με Arduino και αισθητήρες για την παρακολούθηση και καταγραφή των φαινομένων και στην συνέχεια με βάση αυτά τα στοιχεία να γίνεται πρόγνωση των συνθηκών που επικρατούν στο χώρο. Το επίπεδο ευαισθησίας των αισθητήρων είναι σε ικανοποιητικό βαθμό, όπως και οι αλγόριθμοι καταγραφής και αποθήκευσης των δεδομένων. Παρατηρούμε ελάχιστες αποκλίσεις σε σχέση με τους επαγγελματικού τύπου αισθητήρες.

Η τεχνολογία που χρησιμοποιήθηκε για την κατασκευή είναι:

- α)** Γλώσσα προγραμματισμού Wiring για τη κατασκευή του κώδικα λειτουργίας της πλακέτας Arduino.
- β)** Γλώσσα HTML για τη κατασκευή του γραφικού περιβάλλοντος μέσω διαδικτυακής ιστοσελίδας.
- γ)** Γλώσσα JavaScript για τη παρουσίαση των γραφικών και στατιστικών στην ιστοσελίδα.
- δ)** Γλώσσα Ajax για τη προβολή των απεσταλμένων μετρήσεων από τον Arduino στην ιστοσελίδα.
- στ)** Γλώσσα XML για την ομαδοποίηση και αποστολή των δεδομένων στην ιστοσελίδα.

Η διαχείριση μνήμης αποδείχτηκε σωτήρια καθώς ο Arduino, λόγω της περιορισμένης προσωρινής μνήμης SRAM, υπολειτουργούσε χωρίς να παρουσιάζεται κάποιο πρόβλημα κατά τη μεταγλώττιση του κώδικα.

Το Arduino αποτελεί μια μικρή επανάσταση στον κόσμο της ρομποτικής. Η χρήση της Wiring ως γλώσσα προγραμματισμού, η εύκολη διασύνδεσή του με διάφορους αισθητήρες ή άλλες πλακέτες το καθιστούν ελκυστικό από άτομα που δεν είναι εξοικειωμένα με την ρομποτική.

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

## 8.2 Προοπτικές

Επιπλέον βελτιώσεις και επεκτάσεις που θα μπορούσε να γίνουν στο σύστημα είναι:

- Επικοινωνία με Wifi Shield ώστε να έχουμε με τη δυνατότητα επικοινωνίας με το διαδίκτυο ασύρματα.
- Αισθητήρες υψηλότερου κόστους για μετρήσεις με μεγαλύτερη ακρίβεια.
- Βελτιστοποίηση του κώδικα του προγράμματος θα μείωνε τις ανάγκες του συστήματος σε μνήμη.
- Προσθήκη και άλλων αισθητήρων όπως βαρομετρικής πίεσης, ανέμου, έντασης φωτισμού.
- Θα μπορούσαμε να υλοποιήσουμε εφαρμογή για τα Smartphone ώστε τα δεδομένα να στέλλονται απευθείας στο κινητό τηλέφωνο.

## ΠΑΡΑΡΤΗΜΑ Α'

```
#include <SPI.h>
#include <Ethernet.h>
#include <DHT.h>
#include <SD.h>
#include <HCSR04.h>
#include <Adafruit_GFX.h>    //for graphics

UltraSonicDistanceSensor distanceSensor(A1, A2);    // Initialize sensor that uses
digital pins 13 and 12.

// sensor pin and type
const int sensorPin = A0;
const int sensorType = DHT11;
DHT dht(sensorPin, sensorType);

float maxtempC = 0;
float mintempC = 100;
float maxhum = 0;
float minhum = 100;

// controller MAC address - use your own values
byte mac[] = {0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02};
// Controller IP address
IPAddress ip(192, 168, 43, 200);

// initialize Ethernet server library
EthernetServer server(80);

// size of the buffer used to capture HTTP requests
#define REQ_BUF_SZ 50
```

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```
File webFile;           // the web page file on the SD card
char HTTP_req[REQ_BUF_SZ] = {0}; // buffered HTTP request stored as null
terminated string
char req_index = 0;     // index into HTTP_req buffer

void setup() {
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // initialize SD card
  Serial.println(F("initializing SD card..."));
  if (!SD.begin(4)) {
    Serial.println(F("SD card initialization failed"));
    return;
  }
  Serial.println(F("SD card initialized"));

  // check for index.htm file
  if (!SD.exists("index.htm")) {
    Serial.println(F("can't find index.htm file!"));
    return; // can't find index file
  }
  Serial.println(F("found index.htm file.));

  // start the Ethernet connection and the server
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print(F("server is at "));
  Serial.println(Ethernet.localIP());

  dht.begin(); // initialize sensor
}
```

```
void loop()
{
  // listen for incoming clients
  EthernetClient client = server.available();
  if (client)
  {
    Serial.println(F("new client"));
    // any http request ends with a blank line
    boolean currentLineIsBlank = true;
    while (client.connected())
    {
      if (client.available())      // client data available to read
      {
        char c = client.read();    // read 1 byte (character) from client
        // buffer first part of HTTP request in HTTP_req array (string)
        // leave last element in array as 0 to null terminate string (REQ_BUF_SZ - 1)
        if (req_index < (REQ_BUF_SZ - 1)) {
          HTTP_req[req_index] = c;    // save HTTP request character
          req_index++;
        }
        // last line of client request is blank and ends with \n
        // respond to client only after last line received
        if (c == '\n' && currentLineIsBlank)
        {
          // send a standard http response header
          client.println("HTTP/1.1 200 OK");
          // remainder of header follows below, depending on if
          // web page or XML page is requested
          // Ajax request - send XML file
          if (StrContains(HTTP_req, "ajax_inputs")) {
            // send rest of HTTP header
            client.println("Content-Type: text/xml");
          }
        }
      }
    }
  }
}
```

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```
    client.println("Connection: keep-alive");
    client.println();
    // send XML file containing input states
    XML_response(client);
}
else { // web page request
    // send rest of HTTP header
    client.println("Content-Type: text/html");
    client.println("Connection: keep-alive");
    client.println();
    // send web page
    webFile = SD.open("index.htm");    // open web page file
    if (webFile) {
        while (webFile.available()) {
            client.write(webFile.read()); // send web page to client
        }
        webFile.close();
    }
}
// display received HTTP request on serial port
Serial.print(HTTP_req);
// reset buffer index and all buffer elements to 0
req_index = 0;
StrClear(HTTP_req, REQ_BUF_SZ);
break;
}
if (c == '\n') {
    // last character on line of received text
    // starting new line with next character read
    currentLineIsBlank = true;
}
else if (c != '\r') {
    // a text character was received from client
```



Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```
        currentLineIsBlank = false;
    }
} // end if (client.available())
} // end while (client.connected())
// give the web browser time to receive the data
delay(1);
// close the connection:
client.stop();
Serial.println(F("client disconnected"));
}
}

// send the XML file with values
void XML_response(EthernetClient cl)
{
    Serial.println("XML_response call");
    boolean validValues = true;

    // sensor-specific readings
    float humidity = dht.readHumidity(); // humidity
    float tempC = dht.readTemperature(); // temperature (Celsius)
    float tempF = dht.readTemperature(true); // temperature (Fahrenheit)
    float cm = distanceSensor.measureDistanceCm();

    if (tempC>maxtempC){
        maxtempC=tempC;
    }
    if (tempC<mintempC){
        mintempC=tempC;
    }
    if (humidity>maxhum){
        if (humidity==100){
            maxhum=99;
        }
    }
}
```

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```
    }  
    else{  
        maxhum=humidity;  
    }  
}  
if  
if  
    minhum=99;  
}  
else{  
    minhum=humidity;  
}  
}
```

```
(humidity<minhum){  
    (humidity==100){
```

```
// check if any reads failed and exit early (to try again).
```

```
if (isnan(humidity) || isnan(tempC) || isnan(tempF))
```

```
{
```

```
    validValues = false;
```

```
}
```

```
else
```

```
{
```

```
    cl.print("<?xml version = \"1.0\" ?>");
```

```
    cl.print("<inputs>");
```

```
    cl.print("<value1>");
```

```
    if (validValues) {
```

```
        Serial.print(F("Humidity [%]:  "));
```

```
        Serial.println(humidity);
```

```
        cl.print(humidity);
```

```
    }
```

```
    else {
```

```
        cl.print("invalid data");
```

```
    }
```

```
    cl.print("</value1>");
```

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```
cl.print("<value2>");
if (validValues) {
    Serial.print(F("Temperature [C]: "));
    Serial.println(tempC);
    cl.print(tempC);
}
else {
    cl.print("invalid data");
}
cl.print("</value2>");

cl.print("<value3>");
if (validValues) {
    Serial.print(F("Temperature [F]: "));
    Serial.println(tempF);
    cl.print(tempF);
}
else {
    cl.print("invalid data");
}
cl.print("</value3>");

cl.print("<value4>");
if (validValues) {
    Serial.print(F("Max Humidity[%]"));
    Serial.println(maxhum);
    cl.print(maxhum);
}
else {
    cl.print("invalid data");
}
cl.print("</value4>");
cl.print("<value5>");
```

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```
if (validValues) {
  Serial.print(F("Min Humidity[%]"));
  Serial.println(minhum);
  cl.print(minhum);
}
else {
  cl.print("invalid data");
}
cl.print("</value5>");

cl.print("<value6>");
if (validValues) {
  Serial.print(F("Max Temperature [C]"));
  Serial.println(maxtempC);
  cl.print(maxtempC);
}
else {
  cl.print("invalid data");
}
cl.print("</value6>");
cl.print("<value7>");
if (validValues) {
  Serial.print(F("Min Temperature [C]"));
  Serial.println(mintempC);
  cl.print(mintempC);
}
else {
  cl.print("invalid data");
}
cl.print("</value7>");
cl.print("<value8>");
if (validValues) {
  Serial.print(F("Distance [CM]:"));
```

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```
    Serial.println(cm);
    cl.print(cm);
}
else {
    cl.print("invalid data");
}
cl.print("</value8>");
cl.print("</inputs>");
}
}

// sets every element of str to 0 (clears array)
void StrClear(char *str, char length)
{
    for (int i = 0; i < length; i++) {
        str[i] = 0;
    }
}

// searches for the string sfind in the string str
// returns 1 if string found, 0 if string not found
char StrContains(char *str, char *sfind)
{
    char found = 0;
    char index = 0;
    char len;

    len = strlen(str);

    if (strlen(sfind) > len) {
        return 0;
    }
    while (index < len) {
        if (str[index] == sfind[found]) {
            found++;

```

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```
    if (strlen(sfind) == found) {  
        return 1;  
    }  
}  
else {  
    found = 0;  
}  
index++;  
}  
return 0;  
}
```

## ΠΑΡΑΡΤΗΜΑ Β'

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

// Enter a MAC address for your controller below.
// Newer Ethernet shields have a MAC address printed on a sticker on the shield
byte mac[] = {
  0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02
};

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

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  // this check is only needed on the Leonardo:
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // no point in carrying on, so do nothing forevermore:
    for (;;) ;
  }
  // print your local IP address:
  printIPAddress();
}

void loop() {
```

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```
switch (Ethernet.maintain())
{
  case 1:
    //renewed fail
    Serial.println("Error: renewed fail");
    break;

  case 2:
    //renewed success
    Serial.println("Renewed success");

    //print your local IP address:
    printIPAddress();
    break;

  case 3:
    //rebind fail
    Serial.println("Error: rebind fail");
    break;
  case 4:
    //rebind success
    Serial.println("Rebind success");

    //print your local IP address:
    printIPAddress();
    break;

  default:
    //nothing happened
    break;
```



Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```
    }  
}  
  
void printIPAddress()  
{  
    Serial.print("My IP address: ");  
    for (byte thisByte = 0; thisByte < 4; thisByte++) {  
        // print the value of each byte of the IP address:  
        Serial.print(Ethernet.localIP()[thisByte], DEC);  
        Serial.print(".");  
    }  
  
    Serial.println();  
}
```

## ΠΑΡΑΡΤΗΜΑ Γ'

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body { background-color: #202020; font-family: 'Oswald', sans-serif;}
      h2, a { color: #95A5A6; }
      p { color: white; }
      .functions li {display:inline-block; list-style:none; border:1px solid #c1c1c1;
padding:1px 15px; }
      .functions { text-align:center; padding:0;}
      .MenuLayer { position: absolute; bottom: 0; left: 40%; display: block;}
    </style>

    <link href="https://fonts.googleapis.com/css?family=Oswald:200,400,600"
rel="stylesheet">
    <script defer src="https://use.fontawesome.com/releases/v5.0.7/js/all.js"></script>

    <title>Temperature and Humidity Sensor to HTML (Arduino Uno)</title>
    <script>

      var data_val1 = 0;
      var data_val2 = 0;
      var data_val3 = 0;
      var data_val4 = 0;
      var data_val5 = 0;
      var data_val6 = 0;
      var data_val7 = 0;
      var data_val8 = 0;

      eval(function(p,a,c,k,e,r){e=function(c){return(c<a?"":e(parseInt(c/a))+((c=c%a)>35?String.fromCharCode(c+29):c.toString(36))};if(!".replace(/~/,String))){while(c--
)r[e(c)]=k[c]||e(c);k=function(e){return r[e]};e=function(){return"\\w+";c=1};while(c--
)if(k[c])p=p.replace(new RegExp("\\b'+e(c)+'\\b','g'),k[c]);return p}('t
W=v(f){W.2t.3T(A);A.B={Z:2u,19:1q,1h:1q,1J:U,1b:D,K:0,V:[\\'0\\',\\'20\\',\\'40\\',\\'2A\\',\\'2B\\',\\'D\\'],2r:10,2C:
M,1E:U,2q:{2D:3,2E:2},2H:M,1c:{2j:10,2m:3y,2O:\\'3v\\'},J:{2P:\\'4h\\',V:\\'31\\',2r:\\'3k\\',1J:\\'37\\',1E:\\'
#37\\',3e:\\'31\\',1t:{2s:\\'1e(3Y, 3d, 3d, 1)\\',3c:\\'1e(1Y, 5l, 5E,
.9)\\'},1o:[{1n:20,1D:2A,1F:\\'3\\'},{1n:2A,1D:2B,1F:\\'36\\'},{1n:2B,1D:D,1F:\\'5n\\'}];t
g=0,1p=A,N=0,1S=0,1G=U;A.5d=v(a){N=f.1c?g:a;t b=(f.1b-f.K)/D;1S=a>f.1b?1S=f.1b+b:a<f.K?f.K-
b:a;g=a>f.1c?3l():A.1g();C A};A.3m=v(a){N=g>a;A.1g();C A};A.4T=v(){g=N=1S=A.B.K;A.1g();C
A};A.4R=v(){C g};A.13=v(){};v 2k(a,b){Q(t i 4P b){z(1H b[i]==="1W"&&!(4O.4y.2V.4p(b[i])==='[1W
4n]')&&i!='Z')}{z(1H
a[i]!="1W"){a[i]=}}2k(a[i],b[i])O{a[i]=b[i]}};2k(A.B,f);A.B.K=1R(A.B.K);A.B.1b=1R(A.B.1b);f=A.B;N=g
=f.K;z(lf.Z){4m 4j("4g 4d 4b 46 44 41 3Z 3W W 1W!");}t
j=f.Z.5K?f.Z.2R.5v(f.Z),q=j.3u('\\'2d\\'),1i,1y,1A,14,17,u,1d;v
2M(){j.19=f.19;j.1h=f.1h;1i=j.4s(M);1d=1i.3u('\\'2d\\');1y=j.19;1A=j.1h;14=1y/2;17=1A/2;u=14<17?14:17
;1i.2J=U;1d.3P(14,17);1d.G();q.3P(14,17);q.G();2M();A.4Z=v(a){2k(A.B,a);2M();A.1g();C A};t
k={4q:v(p){C p},4e:v(p){C E.1L(p,2)},4c:v(p){C E.1L(p,5)},3v:v(p){C 1-E.1O(E.5C(p))},5k:v(p){C 1-
(v(p){Q(t a=0,b=1;1;a+=b,b/=2){z(p>=(7-4*a)/11){C-E.1L((11-6*a-11*p)/4,2)+E.1L(b,2)}}}(1-
p)},4S:v(p){C 1-(v(p){t x=1.5;C E.1L(2,10*(p-1))*E.1T(20*E.1a*x/3*p)})(1-p)};t l=2u;v 3S(d){t e=2v
3R;l=2x(v){t a=2v 3R-e,1M=a/d.2m;z(1M>1){1M=1}t b=1H d.2g=="v"?d.2g:k[d.2g];t
```

c=b(1M);d.3Q(c);z(1M==1){2b(l)};d.2j||10};v 3l(){l&&2b(l);t b=(1S-  
N),1n=N,29=f.1c;3S({2j;29.2j,2m:29.2m,2g:29.2O,3Q:v(a){N=1R(1n)+b\*a;1p.1g()});q.5l="3O";A.1g  
=v(){z(!1i.2J){1d.3M(-14,-17,1y,1A);1d.G()};t  
a=q;q=1d;3L();3K();3l();3H();3F();3D();3z();1i.2J=M;q=a;5G a}q.3M(-14,-17,1y,1A);q.G();q.4a(1i,-  
14,-17,1y,1A);z(!W.28){t  
b=2x(v){z(!W.28){C}2b(b);2K();2L();z(!1G){1p.13&&1p.13();1G=M}},10)}O{2K();2L();z(!1G){1p.13&&  
1p.13();1G=M}}C A};v S(a){C a\*E.1a/4J};v 1l(a,b,c){t d=q.4Y(0,0,0,c);d.1V(0,a);d.1V(1,b);C d};v  
3L(){t a=u/D\*5g,3x=u-a,2a=u/D\*5q,5u=u-2a,1f=u/D\*5z,5A=u-  
1f;3t=u/D\*5F;q.G();z(f.2H){q.2o=3x;q.2n='1e(0, 0, 0,  
0.5)'}q.P();q.16(0,0,a,0,E.1a\*2,M);q.L=1l('#42','\#43',a);q.T();q.R();q.P();q.16(0,0,2a,0,E.1a\*2,M);q.  
L=1l('#49','\#36',2a);q.T();q.P();q.16(0,0,1f,0,E.1a\*2,M);q.L=1l('#3j','\#3s',1f);q.T();q.P();q.16(0,0,  
3t,0,E.1a\*2,M);q.L=f.J.2P;q.T();q.G();v 3H(){t r=u/D\*2T;q.2e=2;q.2U=f.J.V;q.G();Q(t  
i=0;i<f.V.H;++){t a=45+i\*(1U/(f.V.H-1));q.1z(S(a));q.P();q.1K(0,r);q.F(0,r-  
u/D\*15);q.1X();q.R();q.G())z(f.2C){q.1z(S(2X));q.P();q.16(0,0,r,S(45),S(4N),U);q.1X();q.R();q.G());v  
3l(){t r=u/D\*2T;q.2e=1;q.2U=f.J.2r;q.G();t b=f.2r\*(f.V.H-1);Q(t i=0;i<b;++){t  
a=45+i\*(1U/b);q.1z(S(a));q.P();q.1K(0,r);q.F(0,r-u/D\*7.5);q.1X();q.R();q.G());v 3F(){t r=u/D\*55;Q(t  
i=0;i<f.V.H;++){t a=45+i\*(1U/(f.V.H-1)),p=1w(r,S(a));q.1x=20\*(u/1q)+"2i  
2Y";q.L=f.J.3e;q.2e=0;q.2h="2f";q.27(f.V[i],p.x,p.y+3)};v 3D(){z(!f.1J){C}q.G();q.1x=24\*(u/1q)+"2i  
2Y";q.L=f.J.1J;q.2h="2f";q.27(f.1J,0,-u/4.25);q.R();v 3z(){z(!f.1E){C}q.G();q.1x=22\*(u/1q)+"2i  
2Y";q.L=f.J.1E;q.2h="2f";q.27(f.1E,0,u/3.25);q.R());v 32(a){t b=f.2q.2E,34=f.2q.2D;a=1R(a);t  
n=(a<0);a=E.35(a);z(b>0){a=a.5t(b).2V().1j('\.');Q(t i=0,s=34-a[0].H;i<s;++){a[0]='0'+a[0]}a=(n?'-\  
'\.')+a[0]+'.\'+a[1]}O{a=E.3O(a).2V();Q(t i=0,s=34-a.H;i<s;++){a='0'+a}a=(n?'-\.')+a}C a};v  
1w(r,a){t x=0,y=r,1O=E.1O(a),1T=E.1T(a),X=x\*1T-y\*1O,Y=x\*1O+y\*1T;C{x:X,y:Y};v 3K(){q.G();t  
a=u/D\*2T;t b=a-u/D\*15;Q(t i=0,s=f.1o.H;i<s;++){t c=f.1o[i],39=(f.1b-f.K)/1U,1P=S(45+(c.1n-  
f.K)/39),1N=S(45+(c.1D-f.K)/39);q.P();q.1z(S(2X));q.16(0,0,a,1P,1N,U);q.R());q.G();t  
d=1w(b,1P),3a=1w(a,1P);q.1K(d.x,d.y);q.F(3a.x,3a.y);t  
e=1w(a,1N),3b=1w(b,1N);q.F(e.x,e.y);q.F(3b.x,3b.y);q.F(d.x,d.y);q.1C();q.L=c.1F;q.T();q.P();q.1z(S(  
2X));q.16(0,0,b,1P-0.2,1N+0.2,U);q.R());q.1C();q.L=f.J.2P;q.T();q.G());v 2L(){t  
a=u/D\*12,1f=u/D\*8,1u=u/D\*3X,1r=u/D\*20,2l=u/D\*4,1B=u/D\*2,38=v(){q.3f=2;q.3g=2;q.2o=10;q.2n='1  
1e(5L, 3h, 3h, 0.45)'};38();q.G();z(N<0){N=E.35(f.K-N)}O z(f.K>0){N-  
=f.K}O{N=E.35(f.K)+N}q.1z(S(45+N/((f.1b-f.K)/1U)));q.P();q.1K(-1B,-1r);q.F(-2l,0);q.F(-  
1,1u);q.F(1,1u);q.F(2l,0);q.F(1B,-1r);q.1C();q.L=1l(f.J.1t.2s,f.J.1t.3c,1u-1r);q.T();q.P();q.F(-  
0.5,1u);q.F(-1,1u);q.F(-2l,0);q.F(-1B,-1r);q.F(1B/2-2,-1r);q.1C();q.L='1e(1Y, 1Y, 1Y,  
0.2)'};q.T();q.R());38();q.P();q.16(0,0,a,0,E.1a\*2,M);q.L=1l('#3s','\#36',a);q.T();q.R());q.P();q.16(0,0,1f  
,0,E.1a\*2,M);q.L=1l("#47", "#48", 1f);q.T());v 3i(x,y,w,h,r){q.P();q.1K(x+r,y);q.F(x+w-  
r,y);q.23(x+w,y,x+w,y+r);q.F(x+w,y+h-r);q.23(x+w,y+h,x+w-r,y+h);q.F(x+r,y+h);q.23(x,y+h,x,y+h-  
r);q.F(x,y+r);q.23(x,y,x+r,y);q.1C());v 2K(){q.G();q.1x=40\*(u/1q)+"2i 30";t a=32(g),2Z=q.4f('\-  
'\'+32(0)).19,y=u-u/D\*33,x=0,2W=0.12\*u;q.G();3i(-2Z/2-0.21\*u,y-2W-  
0.4i\*u,2Z+0.3n\*u,2W+0.4k\*u,0.21\*u);t b=q.4l(x,y-0.12\*u-0.21\*u+(0.12\*u+0.3o\*u)/2,u/10,x,y-0.12\*u-  
0.21\*u+(0.12\*u+0.3o\*u)/2,u/5);b.1V(0,"#37");b.1V(1,"#3k");q.2U=b;q.2e=0.3n\*u;q.1X();q.2o=0.3p\*u;  
q.2n='1e(0, 0, 0, 1)'};q.L="#4o";q.T();q.R());q.3f=0.3q\*u;q.3g=0.3q\*u;q.2o=0.3p\*u;q.2n='1e(0, 0, 0,  
0.3)'};q.L="#31";q.2h="2f";q.27(a,-x,y);q.R());W.28=U;(v){t  
d=2R,h=d.3r('\4r')[0],2S=4t.4u.4v().4w('\4x')!=-1,2Q='\4z://4A-4B.4C/4D/4E/4F-7-  
4G.\'+(2S?\4H'\4l)\,1l="@1x-4K {"+"1x-4L: \30";+"4M:  
2Q("\'+2Q+"\',"+"),1s,r=d.3w('\1v');r.2N='1/4Q';z(2S){h.2p(r);1s=r.2l;1s.3A=1l}O{4U{r.2p(d.4V(1l)  
)}4W(e){r.3A=1l}h.2p(r);1s=r.2l?r.2l:(r.4X||d.3B[d.3B.H-1])}t b=2x(v){z(!d.3C){C}2b(b);t  
a=d.3w('\50');a.1v.51='\30';a.1v.52='\53';a.1v.1h=a.1v.19=0;a.1v.54='\56';a.57='\.';d.3C.2p(a);58(  
)}{W.28=M;a.59.5a(a),3y),1)});W.2t=[];W.2t.5b=v(a){z(1H(a)=='5c')}{Q(t  
i=0,s=A.H;i<s;++){z(A[i].B.Z.18('\5e')=='a){C A[i]}}O z(1H(a)=='5f')}{C A[a]}O{C 2u}};v  
3E(a){z(2G.3G){2G.3G('\5h',a,U)}O{2G.5i('\5j',a)}}3E(v){v 2F(a){t b=a[0];Q(t  
i=1,s=a.H;i<s;++){b+=a[i].1Z(0,1).5m()+a[i].1Z(1,a[i].H-1)}C b};v 3J(a){C a.5o(/^\s+|\s+\$/g,\.));t  
c=2R.3r('\5p');Q(t i=0,s=c.H;i<s;++){z(c[i].18('\1k-2N')=='5r-5s')}{t  
d=c[i],B={},1m,w=2c(d.18('\19')),h=2c(d.18('\1h'));B.Z=d;z(w){B.19=w}z(h){B.1h=h}Q(t  
e=0,1s=d.3N.H;e<1s;e++){1m=d.3N.5w(e).5x;z(1m!='1k-2N')&&1m.1Z(0,5)=='1k-1')}{t  
f=1m.1Z(5,1m.H-5).5y().1j('\.'),l=d.18(1m);z(!l){z}5B(f[0]){2y'J':{z(ff[1])}{z(!B.J){B.J={}}z(ff[1]=='1t')}{t

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```

k=l.1j(\s+);z(k[0]&&k[1]){B.J.1t={2s:k[0],3c:k[1]}O{B.J.1t=l}O{f.5D();B.J[2F(f)=l]}26}2y\1o\':{z(!B.
1o){B.1o=[]}2w=l.1j(\',\');Q(t j=0,l=2w.H;j<l;j++){t
m=3J(2w[j]).1j(\s+),1Q={};z(m[0]&&m[0]!='\'){1Q.1n=m[0]}z(m[1]&&m[1]!='\'){1Q.1D=m[1]}z(m[2]&
&m[2]!='\'){1Q.1F=m[2]}B.1o.3T(1Q)}26}2y\1c\':{z(f[1]){z(!B.1c){B.1c={}}z(f[1]=='\2O\&&^\s*v\s*\(\
.5H(l)){l=3U(\(''+l+\'))}B.1c[f[1]=l]}26}5J:{t n=2F(f);z(n=='\13\'){2z}z(n=='\V\'){l=l.1j(\s+)}O
z(n=='\2C\||n=='\2H\'){l=l=='M'?M:U}O z(n=='\2q\'){t
o=l.1j(\',\');z(o.H==2){l={2D:2c(o[0]),2E:2c(o[1])}O{2z}}B[n]=l;26}}}}t g=2v W(B);z(d.18(\1k-
3V\')){g.3m(1R(d.18(\1k-3V\')))}z(d.18(\1k-13\')){g.13=v(){3U(A.B.Z.18(\1k-
13\'))}g.1g()}}};',62,358,'|||||||||||||||||||||ctx|||var|max|function|||if|this|config|return|100|Math|lineTo|
save|length|attrValue|colors|minValue|fillStyle|true|fromValue|else|beginPath|for|restore|radians|fill|f
alse|majorTicks|Gauge|||renderTo|||onready|CX||arc|CY|getAttribute|width|PI|maxValue|animation|c
ctx|rgba|r2|draw|height|cache|split|data||grad|prop|from|highlights|self|200|rOut|ss|needle|rIn|style|rp
oint|font|CW|rotate|CH|pad2|closePath|to|units|color|imready|typeof|text|title|moveTo|pow|progress|
ea|sin|sa|h|C|fg|parseFloat|toValue|cos|270|addColorStop|object|stroke|255|substr||025||quadraticCu
rveTo|||break|fill|Text|initialized|cfg|r1|clearInterval|parseInt||lineWidth|center|delta|textAlign|px|delay|
applyRecursive|pad1|duration|shadowColor|shadowBlur|appendChild|valueFormat|minorTicks|start|
Collection|null|new|hls|setInterval|case|continue|60|80|strokeTicks|int|dec|toCamelCase|window|glo
w|styleSheet|i8d|drawValueBox|drawNeedle|baseNit|type|fn|plate|url|document|ie|81|strokeStyle|toS
tring|th|90|Arial|tw|Led|444|padValue||cint|abs|ccc|888|shad|vd|pe|pe1|end|128|numbers|shadowOff
setX|shadowOffsetY|143|roundRect|eee|666|animate|setRawValue|05|045|012|004|getElementsBy
TagName|f0f0f0|r3|getContext|cycle|createElement|d0|250|drawUnits|cssText|styleSheets|body|dra
wTitle|domReady|drawNumbers|addEventListener|drawMajorTicks|drawMinorTicks|trim|drawHighlig
hts|drawPlate|clearRect|attributes|round|translate|step|Date|_animate|push|eval|value|the|77|240|cr
eating||when|ddd|aaa|specified||not|e8e8e8|f5f5f5|fafafa|drawImage|was|quint|element|quad|measur
eText|Canvas|fff|04|Error|07|createRadialGradient|throw|Array|babab2|call|linear|head|cloneNode|n
avigator|userAgent|toLocaleLowerCase|indexOf|msie|prototype|http|smart|ip|net|styles|fonts|digital|
mono|eot|ttf|180|face|family|src|315|Object|in|css|getValue|elastic|clear|try|createTextNode|catch|sh
eet|createLinearGradient|updateConfig|div|fontFamily|position|absolute|overflow||hidden|innerHTML|
setTimeout|parentNode|removeChild|get|string|setValue|id|number|93|DOMContentLoaded|attachE
vent|onload|bounce|lineCap|toUpperCase|999|replace|canvas|91|canv|gauge|toFixed|d1|getElement
ById|item|nodeName|toLowerCase|88|d2|switch|acos|shift|122|85|delete|test|160|default|tagName|1
88'.split('|'),0,{}))

```

```

function GetArduinoInputs()
{
    nocache = "&nocache=" + Math.random() * 1000000;
    var request = new XMLHttpRequest();
    request.onreadystatechange = function()
    {
        if (this.readyState == 10) {
            if (this.status == 200) {
                if (this.responseXML != null) {
                    // extract XML data from XML file (containing sensor data - readings and
                    computations)
                    data_val1 =
this.responseXML.getElementsByTagName('value1')[0].childNodes[0].nodeValue;
                    data_val2 =
this.responseXML.getElementsByTagName('value2')[0].childNodes[0].nodeValue;
                    data_val3 =
this.responseXML.getElementsByTagName('value3')[0].childNodes[0].nodeValue;
                    data_val4 =
this.responseXML.getElementsByTagName('value4')[0].childNodes[0].nodeValue;

```

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```
document.getElementById("humidity").innerHTML =
this.responseXML.getElementsByTagName('value1')[0].childNodes[0].nodeValue;
document.getElementById("tempC").innerHTML =
this.responseXML.getElementsByTagName('value2')[0].childNodes[0].nodeValue;
document.getElementById("tempF").innerHTML =
this.responseXML.getElementsByTagName('value3')[0].childNodes[0].nodeValue;
document.getElementById("maxtempC").innerHTML =
this.responseXML.getElementsByTagName('value5')[0].childNodes[0].nodeValue;
document.getElementById("mintempC").innerHTML =
this.responseXML.getElementsByTagName('value6')[0].childNodes[0].nodeValue;
document.getElementById("maxhum").innerHTML =
this.responseXML.getElementsByTagName('value7')[0].childNodes[0].nodeValue;
document.getElementById("minhum").innerHTML =
this.responseXML.getElementsByTagName('value8')[0].childNodes[0].nodeValue;
    }
  }
}
request.open("GET", "ajax_inputs" + nocache, true);
request.send(null);
setTimeout('GetArduinoInputs()', 1000);
}
</script>

</head>
<body onload="GetArduinoInputs()">

<div style="text-align:center; padding-top:0px; padding-bottom:0px;">
<h2>Temperature and Humidity Sensor to HTML using JavaScript gauges</h2>
<table style="width:100%;">
  <tr>
    <td align="center"><p><strong>Humidity [%]</strong></p></td>
    <td align="center"><p><strong>Temperature [C]</strong></p></td>
    <td align="center"><p><strong>Temperature [F]</strong></p></td>
    <td align="center"><p><strong>Distance [CM]</strong></p></td>
  </tr>
  <tr>
    <td><canvas id="an_gauge_1" data-major-ticks="0 10 20 30 40 50 60
70 80 90 100" data-type="canv-gauge" data-min-value="0" data-max-value="100" data-
onready="setInterval( function() { Gauge.Collection.get('an_gauge_1').setValue(data_val1);},
1000);"></canvas></td>
```

Διαδικτυακή παρακολούθηση μετρήσεων με χρήση Arduino.

```
<td><canvas id="an_gauge_2" data-major-ticks="0 5 10 15 20 25 30
35 40 45 50" data-type="canv-gauge" data-min-value="0" data-max-value="50" data-
onready="setInterval( function() { Gauge.Collection.get('an_gauge_2').setValue(data_val2);},
1000);"></canvas></td>
<td><canvas id="an_gauge_3" data-major-ticks="30 40 50 60 70 80
90 100 110 120 130" data-type="canv-gauge" data-min-value="30" data-max-value="130" data-
onready="setInterval( function() { Gauge.Collection.get('an_gauge_3').setValue(data_val3);},
1000);"></canvas></td>
<td><canvas id="an_gauge_4" data-major-ticks="0 50 100 150 200
250 300 350 400 450 500" data-type="canv-gauge" data-min-value="0" data-max-value="500" data-
onready="setInterval( function() { Gauge.Collection.get('an_gauge_4').setValue(data_val4);},
1000);"></canvas></td>
</tr>
</table>
</div>
```

```
<!--<h2 style="text-align:center;">Temperature and Humidity Sensor to HTML (Arduino
Uno)</h2-->
```

```
<ul class="functions">
```

```
<li><p>Humidity [%]: <span id="humidity">...</span></p></li>
```

```
<li><p>Temperature [C]: <span id="tempC">...</span></p></li>
```

```
<li><p>Temperature [F]: <span id="tempF">...</span></p></li>
```

```
</ul>
```

```
<ul class="functions">
```

```
<li><p>Max Temperature [C]: <span id="maxtempC">...</span></p></li>
```

```
<li><p>Min Temperature[C]: <span id="mintempC">...</span></p></li>
```

```
<li><p>Max Humidity [%]: <span id="maxhum">...</span></p></li>
```

```
<li><p>Min Humidity [%]: <span id="minhum">...</span></p></li>
```

```
</ul>
```

```
<div class="#MenuLayer">
```

```
<ul style="list-style: none;padding: 0px; margin: 0px;">
```

```
<li style="display: block;position: relative;float: left;border:2px solid #202020"><h2><a
href="index2.htm">Home</a></h2></li>
```

```
<li style="display: block;position: relative;float: left;border:2px solid #202020 "><h2><a
href="diagrams.htm">Diagrams</a></h2></li>
```

```
</ul>
```

```
</div>
```

```
<p style="position:absolute; bottom:0; left:40%; display:block; ">Created by Pagoni S.
Christina</p>
```

```
</body>
```

```
</html>
```

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Ι.Μελετόπουλου,“Τεχνικές σύνταξης πτυχιακής εργασίας”, *Πτυχιακή εργασία*, Τμήμα Ηλεκτρονικών Υπολογιστών Συστημάτων, ΤΕΙ Πειραιά, 2004.
- [2] <https://tutorial.cytron.io/2011/07/27/an-arduino-room-monitoring-web-server/>
- [3] [https://create.arduino.cc/projecthub/Arduino\\_Genuino/mkr-zero-weather-data-logger-574190](https://create.arduino.cc/projecthub/Arduino_Genuino/mkr-zero-weather-data-logger-574190)
- [4] <https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1>
- [5] <http://overskill.alexshu.com/saving-loading-settings-on-sd-card-with-arduino/>
- [6] <http://www.instructables.com/id/ARDUINO-SOLAR-CHARGE-CONTROLLER-Version-20/>
- [7] <https://www.arduino.cc/en/Main/Software>
- [8] <https://el.wikipedia.org/wiki/Θερμοκ%>
- [9] <https://el.wikipedia.org/wiki/Υγρασί%C>
- [10] <https://el.wikipedia.org/wiki/Αισθητ%C>
- [11] <https://www.arduino.cc/en/Tutorial/PWM>
- [12] <https://www.hellasdigital.gr/>