



**ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ**

ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

HOU-CS-UGP-2013-08

**«ΑΝΑΠΤΥΞΗ ΑΥΤΟΝΟΜΟΥ ΣΤΑΘΜΟΥ ΚΑΤΑΓΡΑΦΗΣ
& ΠΡΟΒΛΕΨΗΣ ΜΕΤΕΩΡΟΛΟΓΙΚΩΝ ΦΑΙΝΟΜΕΝΩΝ»**



**HELLENIC OPEN UNIVERSITY
SCHOOL OF SCIENCES AND TECHNOLOGY**





Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

Πτυχιακή Εργασία *HOU-CS-UGP-2013-08*

Ανάπτυξη Αυτόνομου Σταθμού

Καταγραφής και Πρόβλεψης Μετεωρολογικών Φαινομένων

Τριανταφύλλου Γεώργιος



© ΕΑΠ, 2013

Η παρούσα διατριβή, η οποία εκπονήθηκε στα πλαίσια της ΘΕ ΠΛΗ40, και τα λοιπά αποτελέσματα της αντίστοιχης Πτυχιακής Εργασίας (ΠΕ) αποτελούν συνιδιοκτησία του ΕΑΠ και του φοιτητή, ο καθένας από τους οποίους έχει το δικαίωμα ανεξάρτητης χρήσης και αναπαραγωγής τους (στο σύνολο ή τμηματικά) για διδακτικούς και ερευνητικούς σκοπούς, σε κάθε περίπτωση αναφέροντας τον τίτλο και το συγγραφέα και το ΕΑΠ όπου εκπονήθηκε η ΠΕ καθώς και τον επιβλέποντα και την επιτροπή κρίσης.



Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής και Πρόβλεψης Μετεωρολογικών Δεδομένων

Τριανταφύλλου Γεώργιος

Όνοματεπώνυμο Επιβλέποντα

Φλώρος Ανδρέας
Επίκουρος Καθηγητής
Ιονίου Πανεπιστημίου

Όνοματεπώνυμο Μέλους 1

Βερύκιος Σ. Βασίλειος
Αναπληρωτής Καθηγητής
Ελληνικού Ανοικτού
Πανεπιστημίου

Όνοματεπώνυμο Μέλους 2

Αναγνωστόπουλος
Χρήστος – Νικόλαος
Επίκουρος Καθηγητής
Πανεπιστημίου Αιγαίου

Περίληψη

Στην παρούσα εργασία παρουσιάζεται η διαδικασία προγραμματισμού της υπολογιστικής πλατφόρμας Arduino με σκοπό την αξιοποίηση της ως αυτόνομο μετεωρολογικό σταθμό, με δυνατότητες καταγραφής, επεξεργασίας και αποθήκευσης των μετεωρολογικών δεδομένων της περιοχής, με σκοπό την βραχυπρόθεσμη πρόβλεψη του καιρού σύμφωνα με τα δεδομένα που έχουν συλλεχθεί, χρησιμοποιώντας ένα απλοποιημένο μετεωρολογικό μοντέλο πρόγνωσης καιρού.

Αρχικά γίνεται μια ανάλυση των μετεωρολογικών φαινομένων και παρουσιάζεται η σημασία τους για την πρόγνωση καιρού καθώς και η μεθοδολογία καταγραφής τους, σύμφωνα με τους αλγόριθμους που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Μετεωρολογίας.

Παρουσιάζονται οι αισθητήρες που χρησιμοποιήθηκαν, τα κριτήρια επιλογής τους, ο τρόπος συνδεσμολογίας και οι βιβλιοθήκες που αξιοποιήθηκαν για την διεπαφή τους με την πλακέτα Arduino. Στην συνέχεια γίνεται παρουσίαση του μετεωρολογικού μοντέλου Ζαμπρέτι (Zambretti) βάση του οποίου πραγματοποιούνται οι προγνώσεις.

Αφού ολοκληρωθούν όλοι οι απαραίτητοι έλεγχοι για την διαπίστωση σφαλμάτων στον αλγόριθμο συλλογής και καταγραφής των μετεωρολογικών δεδομένων, σχεδιάζεται η κύρια οθόνη διεπαφής με την οποία αλληλεπιδρά ο χρήστης με τον σταθμό (Graphic User Interface).



Γίνεται επεξήγηση της διαδικασίας καταγραφής των μετεωρολογικών δεδομένων του σταθμού και παρουσιάζονται τα προβλήματα που εντοπίστηκαν κατά την διάρκεια των δοκιμών και αναλύονται οι τρόποι αντιμετώπισης τους.

Για να διαπιστωθεί η αξιοπιστία των αισθητήρων καθώς και των αλγορίθμων καταγραφής, γίνεται σύγκριση των δεδομένων που συλλέχθηκαν με τα αντίστοιχα από υπάρχον πιστοποιημένο μετεωρολογικό σταθμό της περιοχής, το αποτέλεσμα της οποίας είναι ικανοποιητικό.

Στην τελική φάση γίνεται έλεγχος της ποιότητας του μετεωρολογικού μοντέλου όπου για το χρονικό διάστημα των δεκαεννέα (19) ημερών, στις 9:00 καταγράφονταν η πρόγνωση που εμφάνιζε ο σταθμός με βάση τα δεδομένα που είχαν συλλεχθεί. Στην συνέχεια μετά από παρατήρηση των καιρικών συνθηκών της περιοχής για τις επόμενες 12 ώρες γίνονταν σύγκριση αυτών στις 21:00 με την πρόβλεψη του σταθμού. Ο αλγόριθμος πρόγνωσης υπήρξε ακριβής σε ποσοστό 90% αν και τα δείγματα καταγραφής ήταν λίγα ώστε να μπορέσουμε να έχουμε μια ολοκληρωμένη εικόνα αναφορικά με την απόδοση του.

Ο Αυτόνομος Μετεωρολογικός Σταθμός με χρήση της υπολογιστικής πλατφόρμας Arduino, αποτελεί ιδανική λύση σε όσους επιθυμούν να γνωρίζουν τα καιρικά φαινόμενα που θα επικρατήσουν τις επόμενες δώδεκα ώρες στην περιοχή εμβέλειάς του, χωρίς η δαπάνη κατασκευής του σταθμού να απαιτεί υψηλό κόστος.

Λέξεις-κλειδιά: Arduino, Μετεωρολογικός Σταθμός, Πρόγνωση Καιρού, Μετεωρολογικό Μοντέλο, Wiring

Περιεχόμενο: κείμενο, φωτογραφίες, πρόγραμμα σε γλώσσα Wiring, πρωτότυπο εφαρμογής, HTML κώδικας



**Development of Autonomous Station to
Record and Forecast Meteorological Phenomenon**

Triantafillou Georgios

Supervisor	Member 1	Member 2
Floros Andreas Assistant Professor Ionian University	Vassilios S. Verykios Associate Professor Hellenic Open University	Anagnostopoulos Christos-Nikolaos Assistant Professor Aegean University

Abstract:

In the current dissertation, the process of programming the prototype platform of Arduino is being described in order to evaluate and leverage it as an autonomous meteorological station, with better features than the ability to load and save data. This platform will be able to make short term weather predictions based on the collected data.

Initially all the meteorological phenomena in use will be presented according to the World Meteorological Organization.

The sensors that will be in use and the criteria of choosing them will be presented along with the way of accembrance those sensors with the wiring board. Next, the meteorological model known as “Zambretti”, used for prognosis, will be presented.

After all tests have been completed, the main interface screen will be planned in order the user to be able to interact with the station even from a long distance. The way of recording data and how this data will be used and utilized by the final user will be explained.

The problems encountered with the memory management and the ways to address them will be presented. Finally, a check control on the accuracy of the meteorological



data will be completed by logging data for a period of 19 days and observing for the next 12 hours in its operating state. In this final stage, the accuracy of prediction will be verified.



Πίνακας Περιεχομένων

1. Εισαγωγή	10
2. Αντικείμενο της Εργασίας	12
2.1 Αυτόνομος Μετεωρολογικός Σταθμός (ΑΜΣ)	13
2.2 Καιρός – Επιστήμη της μετεωρολογίας– Μετεωρολογικά φαινόμενα.....	14
2.2.1 Θερμοκρασία	14
2.2.2 Ατμοσφαιρική Πίεση	15
2.2.3 Βαρομετρική τάση	16
2.2.4 Υγρασία	17
2.2.5 Σημείο δρόσου ή Σημείο υγροποίησης ή σημείο κόρου ατμόσφαιρας.....	17
2.2.6 Δείκτης Δυσφορίας ή Δείκτης Θερμότητας.....	18
2.2.7 Άνεμος	19
2.3 Η υπολογιστική πλατφόρμα Arduino	20
2.3.1 Το Υλικό	20
2.3.2 Το Λογισμικό	21
2.4 Περιγραφή μεθοδολογίας της λύσης του προβλήματος	22
2.5 Περιγραφή των αισθητήρων καταγραφής.	24
2.5.1 Αισθητήρας Θερμοκρασίας / Υγρασίας DHT-22	25
2.5.2 Αισθητήρας Ατμοσφαιρικής Πίεσης BMP-085	27
2.5.3 Αισθητήρας Κατεύθυνσης Ανέμου 80422.....	29
2.6 Απλοποιημένα Μετεωρολογικά μοντέλα	30
2.6.1 Αλγόριθμος Ζαμπρέτι (Zambretti Algorithm)	31
3. Καταγραφή και ανάλυση απαιτήσεων	37
3.1 Πλάνο έργου (Project Plan)	39
3.2 Χρονοδιάγραμμα εργασιών	40
4. Ανάλυση και σχεδίαση του σταθμού	42
4.1 Περιγραφή Αρχιτεκτονικής Μετεωρολογικού Σταθμού	42
4.2 Σχεδιασμός Λογισμικού Σταθμού.....	43
4.3 Διάγραμμα περίπτωσης χρήσης (Use Case Diagram)	45
4.4 Δομή WeatherData.....	47
4.5 Βιβλιοθήκη WeatherStation.....	49
5. Σύνδεση Περιφερειακών Συσκευών – Αισθητήρων.....	52
5.1 Κάρτα επέκτασης δικτύου (EthernetShield).....	52
5.2 Αισθητήρας καταγραφής θερμοκρασίας – υγρασίας DHT-22.	54
5.3 Σύνδεση αισθητήρα ατμοσφαιρικής πίεσης BMP-085	57
5.4 Αισθητήρας Διεύθυνσης Ανέμου.....	59
6. Υλοποίηση	65
6.1 Πλατφόρμες και προγραμματιστικά εργαλεία.....	65
6.1.1 Πρόγραμμα και αξιοποιήσιμες βιβλιοθήκες.....	65
6.1.2 Ιστοσελίδες και Javascript	65
6.1.3 Επιπρόσθετες βιβλιοθήκες υποστήριξης έργου	65
6.2 Ανάλυση αλγορίθμων	67
6.2.1 Υπολογισμός μέσης τιμής μετεωρολογικών δεδομένων (ανά λεπτό)	67
6.2.2 Υπολογισμός μέσης τιμής μετεωρολογικών δεδομένων (ωριαία).....	67



6.2.3 Καταγραφή Βαρομετρικής τάσης	67
6.2.4 Υπολογισμός Βαρομετρικής Τάσης.....	68
6.2.5 Αλγόριθμος Πρόγνωσης Καιρού Zambretti	69
6.2.6 Συνάρτηση Καταγραφής Δεδομένων στην κάρτα SD	69
6.2.7 Συνάρτηση Αναγωγής στην επιφάνεια της θάλασσας.....	69
7. Διαχείριση Μνήμης.....	70
7.1 Διαχείριση Μνήμης Προγράμματος	70
7.2 Διαχείριση Μνήμης SRAM	73
7.2.1 Χρήση της συνάρτησης F().....	74
7.2.2 Χρήση της PROGREM.....	74
8. Διεπαφή Χρήστη	76
8.1 Αρχική Σελίδα.....	77
8.2 Προβολή Στατιστικών.....	80
9. Έλεγχος Συστήματος	82
9.1 Έλεγχος Μετεωρολογικών Δεδομένων	83
9.1.1 Θερμοκρασία	83
9.1.2 Υγρασία	86
9.1.3 Ατμοσφαιρική πίεση	87
9.2 Απόδοση Μετεωρολογικού Μοντέλου	89
10. Αδυναμίες και βελτιώσεις.....	94
11. Συμπεράσματα.	96
Βιβλιογραφία – Αναφορές	98
ΠΑΡΑΡΤΗΜΑΤΑ	101
Παράρτημα Α – Κώδικας διασύνδεσης κάρτας Ethernet Shield.....	101
Παράρτημα Β – Διαδικασία διασύνδεσης αισθητήρα DHT-22.....	102
Παράρτημα Γ – Αλγόριθμος Διασύνδεσης αισθητήρα BMP-085	103
Παράρτημα Δ – Διαδικασία διασύνδεσης αισθητήρα διεύθυνσης ανέμου.	104
Παράρτημα Ε – Συνάρτηση καταγραφής δειγμάτων Ατμ. πίεσης	105
Παράρτημα ΣΤ - Συνάρτηση Υπολογισμού Βαρομετρικής τάσης.....	106
Παράρτημα Ζ – Συνάρτηση αποθήκευσης βαρομετρικής τάσης	106
Παράρτημα Η – Συνάρτηση ανάγνωσης βαρομετρικής τάσης.....	107
Παράρτημα Θ – Συνάρτηση αλγορίθμου Ζαμπρέτι	107
Παράρτημα Ι – Διαδικασία καταχώρησης στην κάρτα SD	109
Παράρτημα ΙΑ – Συνάρτηση λειτουργίας εξυπηρετητή διαδικτύου.	110



1. Εισαγωγή

Για την παρατήρηση των καιρικών φαινομένων έχουν κατασκευαστεί ποικίλα μετεωρολογικά συστήματα τα οποία κυμαίνονται σε ένα ευρύ φάσμα τιμών και δυνατοτήτων, ώστε να απευθύνονται σε κοινό το οποίο αποτελείται από ερασιτέχνες καθώς και επαγγελματίες όπως αγρότες, μετεωρολόγους, αλιείς κτλ. Αυτές οι λύσεις μπορεί να είναι από μια απλή απεικόνιση των δεδομένων, έως πολύπλοκα συστήματα τα οποία περιλαμβάνουν την καταγραφή και τη βραχυπρόθεσμη πρόβλεψη του καιρού. Όμως σε τι θα χρησίμευε ένα σύστημα καταγραφής και πρόβλεψης μετεωρολογικών φαινομένων στον απλό χρήστη; Η αλήθεια είναι ότι στις καθημερινές του ασχολίες, ο σύγχρονος άνθρωπος της πόλης δεν επηρεάζεται ιδιαίτερα από τα φαινόμενα του καιρού. Όμως υπάρχουν άνθρωποι οι οποίοι επηρεάζονται άμεσα και σημαντικά από τις μεταβολές των καιρικών συνθηκών. Για παράδειγμα ο αγρότης, ο οποίος θέλει να γνωρίζει με ακρίβεια τις μεταβολές του καιρού ώστε να προγραμματίζει τις εργασίες του και να προετοιμάζεται κατάλληλα για κάθε αλλαγή των καιρικών φαινομένων. Τα περισσότερα όμως δελτία καιρού, τα οποία μεταδίδονται από τα τηλεοπτικά μέσα, αναφέρονται συνήθως στις αστικές περιοχές όπου και διαμένει το μεγαλύτερο ποσοστό του πληθυσμού της χώρας. Στις περιπτώσεις αυτές λοιπόν, όπου οι πληροφορίες οι οποίες δίδονται στους επαγγελματίες είναι λιγοστές και ελλιπείς, την λύση έρχονται να δώσουν οι φορητοί μετεωρολογικοί σταθμοί, οι οποίοι συλλέγουν δεδομένα για την περιοχή που είναι εγκατεστημένοι και εξάγουν μια σχετικά ασφαλή πρόγνωση. Συνήθως, τα μηχανήματα αυτά έχουν υψηλό κόστος, με αποτέλεσμα για ορισμένους χρήστες και σε συνάρτηση πάντα με τα οφέλη τα οποία προκύπτουν από την χρήση τέτοιων σταθμών, να είναι απαγορευτικά.

Στον αντίποδα αυτών των συστημάτων, το Arduino αποτελεί ένα χαμηλού κόστους υπολογιστικό σύστημα, με μεγάλο εύρος περιφερειακών συσκευών, εύκολα προγραμματίσιμο και με μεγάλη κοινότητα μελών, οι οποίοι είναι έτοιμοι να παράσχουν κάθε βοήθεια μεταξύ τους.

Στόχος της παρούσης εργασίας είναι η κατασκευή ενός μετεωρολογικού σταθμού ο οποίος θα συλλέγει τα απαραίτητα δεδομένα μέσω των αισθητήρων του και στην συνέχεια με βάση τα δεδομένα αυτά θα αποδίδει μια αξιόπιστη πρόβλεψη για τις επόμενες 12 έως 24 ώρες.

Ένα βασικό μειονέκτημα των περισσότερων μετεωρολογικών σταθμών είναι ότι απαιτείται η φυσική παρουσία του χρήστη στην τοποθεσία που βρίσκεται εγκατεστημένος ο σταθμός. Το



πρόβλημα αυτό αντιμετωπίζεται εν μέρη στα ποιο ακριβά μοντέλα τα οποία διαθέτουν ασύρματη μετάδοση των δεδομένων τους. Ο υπό κατασκευή σταθμός θα πρέπει να έχει όσο το δυνατόν μεγαλύτερο εύρος δράσης και η πρόσβαση σε αυτόν θα επιτρέπεται με μέσα που διαθέτει ήδη ο χρήστης ώστε να μην απαιτηθεί η κατασκευή ή αγορά του δέκτη, μειώνοντάς με αυτόν τον τρόπο το τελικό κόστος κατασκευής του.

Σε αυτήν την περίπτωση η ιδανική λύση που προτείνεται είναι η σύνδεση του σταθμού στο διαδίκτυο, καθώς οι τεχνολογίες του μας επιτρέπουν να εκτελούμε εργασίες οι οποίες μας διευκολύνουν είτε σε επαγγελματικό είτε σε προσωπικό επίπεδο. Με αυτόν τον τρόπο, η προβολή των δεδομένων θα γίνεται μέσω ιστοσελίδας έτσι ώστε με την χρήση ενός προσωπικού υπολογιστή ή ακόμα ενός «έξυπνου» κινητού τηλεφώνου θα επιτρέπεται η πρόσβαση στα δεδομένα του σταθμού από οποιοδήποτε σημείο και να βρίσκεται ο χρήστης, χωρίς επιπλέον κόστος.



2. Αντικείμενο της Εργασίας



Σχήμα 1 Αυτόματος Μετεωρολογικός Σταθμός σε λειτουργία

Αντικείμενο της παρούσης εργασίας είναι η κατασκευή ενός αυτόνομου μετεωρολογικού σταθμού (ΑΜΣ) με δυνατότητα πρόγνωσης καιρού, κάνοντας χρήση απλοποιημένων μοντέλων μετεωρολογίας. Ο ΑΜΣ θα βασίζεται στην λειτουργία του στην υπολογιστική πλατφόρμα Arduino, μία πρότυπη τεχνολογία η οποία διαθέτει διαδραστικότητα με το περιβάλλον μέσω των υφιστάμενων αισθητήρων και αποτελεί μια ιδανική λύση για εφαρμογές αυτοματισμού και απλοποιημένης ρομποτικής. Ο μικροελεγκτής θα δέχεται στις εισόδους του, αισθητήρες καταγραφής μετεωρολογικών φαινομένων οι οποίοι είναι:

- αισθητήρας θερμότητας
- αισθητήρας υγρασίας
- αισθητήρας ατμοσφαιρικής πίεσης
- αισθητήρας διεύθυνσης ανέμου

Στην συνέχεια οι τιμές των αισθητήρων θα καταγράφονται σύμφωνα με τα διεθνή πρότυπα που έχουν καθοριστεί από τον Παγκόσμιο Οργανισμό Μετεωρολογίας (World Meteorological Organization - WMO), και θα αποθηκεύονται στην κάρτα SD για ιστορικούς και στατιστικούς λόγους. Το Arduino θα προγραμματισθεί έτσι ώστε να λειτουργεί και ως εξυπηρετητής διαδικτύου (web server), με σκοπό την εμφάνιση



των δεδομένων των αισθητήρων στο διαδικτυακό τόπο του ΑΜΣ με την χρήση οποιουδήποτε φυλλομετρητή. Τέλος, με χρήση απλοποιημένων μετεωρολογικών μοντέλων στην ιστοσελίδα του σταθμού, θα εμφανίζεται και η πρόγνωση του καιρού της περιοχής της οποίας είναι εγκατεστημένος ο σταθμός.

Αρχικά θα πρέπει να συνδεθούν οι αισθητήρες υγρασίας, ατμοσφαιρικής πίεσης και θερμοκρασίας. Στην συνέχεια, αφού γίνει ο έλεγχος της λειτουργίας τους θα υλοποιηθεί το λογισμικό διασύνδεσης τους. Απαιτείται η μελέτη και κατασκευή του γραφικού περιβάλλοντος διεπαφής (Graphic User Interface – GUI) υπό την μορφή ιστοσελίδας, στο οποίο θα εισέρχεται ο χρήστης από φυλλομετρητή της επιλογής του και θα προβάλλονται τα δεδομένα και η πρόγνωση του καιρού.

Τέλος κρίνεται σκόπιμο, χωρίς να είναι απαραίτητο, η τοποθέτηση μιας οθόνης υγρών κρυστάλλων (LCD) η οποία θα έχει ως σκοπό την εμφάνιση των ενδείξεων του σταθμού (θερμοκρασία, υγρασία, ατμοσφαιρική πίεση, ταχύτητα ανέμου), και αν απαιτηθεί, πληροφορίες για την λειτουργία του (διεύθυνση ip, κατάσταση αισθητήρων κλπ).

2.1 Αυτόνομος Μετεωρολογικός Σταθμός (ΑΜΣ)

Αυτόνομος μετεωρολογικός σταθμός (ΑΜΣ) [1] ορίζεται ένας μετεωρολογικός σταθμός στον οποίο γίνονται λήψεις μετεωρολογικών δεδομένων με αυτοματοποιημένη διαδικασία και στην συνέχεια τοπικά ή σε κεντρικές μονάδες όπου αποστέλλονται, γίνεται η επεξεργασία των δεδομένων τους. Στους ΑΜΣ, οι μετρήσεις των οργάνων λαμβάνονται από μια κεντρική μονάδα δεδομένων, στην περίπτωση μας από το Arduino. Ο αυτόνομος σταθμός καιρού μπορεί να αποτελούν μέρος ενός δικτύου συνολικής καταγραφής, ονομάζονται δε συνήθως «Αυτοματοποιημένο Σύστημα Παρατήρησης Καιρικών Συνθηκών» (Automated Weather Observing System - AWOS) ή «Αυτοματοποιημένο Σύστημα Παρατήρησης Επιφάνειας» (Automatic System Observation Surface - ASOS). Όμως η ευρεία χρήση αναφοράς σε ένα τέτοιο σύστημα είναι ΑΜΣ, αν και ο όρος "σταθμός" δεν ανταποκρίνεται πλήρως σε αυτόν τον ορισμό. Παρ'όλα αυτά, στην παρούσα εργασία η αναφορά στον όρο ΑΜΣ θα εννοεί ένα τέτοιο σύστημα.



2.2 Καιρός – Επιστήμη της μετεωρολογίας– Μετεωρολογικά φαινόμενα

Καιρός ονομάζεται το σύνολο των μετεωρολογικών φαινομένων που παρατηρούνται στην ατμόσφαιρα της Γης σε καθορισμένο τόπο και χρόνο. Τέτοια φαινόμενα είναι η θερμοκρασία, η ατμοσφαιρική πίεση, οι κινήσεις των ανέμων, η παρουσία νεφών κλπ. Ο καιρός προσδιορίζεται για μια συγκεκριμένη χρονική στιγμή και η πρόγνωση του βασίζεται στα αποτελέσματα των παρατηρήσεων του. Στην πραγματικότητα είναι μια συνάρτηση στην οποία εισάγονται μετεωρολογικά δεδομένα, ο τόπος, ο χρόνος κλπ. και με βάση τις καταγραφές και παρατηρήσεις που έχουν γίνει, εξάγεται ένα αποτέλεσμα. Το αποτέλεσμα αυτό αποτελεί την πρόγνωση του καιρού και η ακρίβεια του εξαρτάται από το μετεωρολογικό μοντέλο που θα χρησιμοποιηθεί. Με την εξέλιξη των υπολογιστικών συστημάτων τα οποία πλέον εκτελούν τρισεκατομμύρια πράξεις ανά δευτερόλεπτο και τους δορυφόρους από τους οποίους μπορούμε να παρατηρούμε τα μετεωρολογικά φαινόμενα με μεγαλύτερη ακρίβεια και για μεγαλύτερο γεωγραφικό εύρος, τα μοντέλα αυτά έχουν εξελιχθεί και εξάγουν ασφαλέστερα και πιο μακροπρόθεσμα αποτελέσματα [2].

2.2.1 Θερμοκρασία

Θερμοκρασία ατμόσφαιρας ονομάζεται η θερμοκρασία την οποία έχει ο ατμοσφαιρικός αέρας πάνω από μια περιοχή. Η πρόγνωση του καιρού σε μια περιοχή βασίζεται κυρίως στη γνώση της εκάστοτε ατμοσφαιρικής πίεσης και της θερμοκρασίας της ατμόσφαιρας της υπ’ όψιν περιοχής και των γύρω αυτής εκτάσεων.

Συνεπώς η αναφορά της θερμοκρασίας σχετίζεται πάντα με κάποια περιοχή, είτε μικρή, είτε μεγάλη, στην περίπτωση μας δε στην περιοχή εμβέλειας του σταθμού. Η αναφορά σε παγκόσμια κλίμακα, ανάγεται σε αντικείμενο άλλης επιστήμης [3].

Η θερμοκρασία της ατμόσφαιρας μετριέται με τα θερμόμετρα και υπάρχουν διάφορες κλίμακες μέτρησης, με συνηθισμένες κλίμακες τις Κελσίου (Celsius, σύμβολο C°), Κέλβιν (Kelvin, σύμβολο K°) και Φαρενάιτ (Fahrenheit, σύμβολο F°). Στην Ελλάδα χρησιμοποιείται η κλίμακα Κελσίου και ορίζεται ως « Το σημείο βρασμού του νερού είναι στους 100 C° και το σημείο παγιοποίησης του, στους 0 C°».



2.2.2 Ατμοσφαιρική Πίεση

Η **ατμοσφαιρική πίεση** είναι ένα από τα πιο σημαντικά μετεωρολογικά στοιχεία, γιατί οι καιρικές καταστάσεις και οι μεταβολές τους συνδέονται άμεσα μαζί της [4]. Για τον λόγο αυτό, το μοντέλο που θα χρησιμοποιηθεί για την πρόγνωση του ΑΜΣ βασίζεται κυρίως στις μεταβολές της ατμοσφαιρικής πίεσης. Ατμοσφαιρική πίεση ή «Βαρομετρική πίεση» ονομάζεται η πίεση που ασκεί η ατμόσφαιρα, με το βάρος της, στην επιφάνεια της Γης.

Στην επιφάνεια της Γης η ατμοσφαιρική πίεση ισούται κατά μέσον όρο με το βάρος στήλης ύδατος ύψους 11 μ.(m) περίπου, ή 1 εκατομμύριοδύνες ανά cm^2 . Στην μετεωρολογία 1000 δύνες/ cm^2 αντιστοιχούν σε ένα χιλιοστόμετρο (milibar). Κατά μέσο όρο στην επιφάνεια της γης η ατμοσφαιρική πίεση είναι 1000 milibar. Με απόφαση του Διεθνούς Μετεωρολογικού Οργανισμού (WMO) καθιερώθηκε ως μονάδα μέτρησης της ατμοσφαιρικής πίεσης το Εκτοπασκάλ (1 hPa) το οποίο η αντιστοιχία είναι 1 hPa=1 milibar [5][6].

Η πίεση μειώνεται ανάλογα με υψόμετρο. Για τον λόγο αυτό οι μετρήσεις που γίνονται σε ύψος μεγαλύτερο από αυτό της επιφάνειας της θάλασσας, πρέπει να τροποποιηθούν ώστε να αντιστοιχούν σε μηδενικό ύψος. Η ενέργεια αυτή ονομάζεται ως **αναγωγή στην επιφάνεια της θάλασσας**.

Η σχέση με την οποία εκφράζεται η αναγωγή στην επιφάνεια της θάλασσας ορίζεται ως

$$P = P_0 \left(1 - \frac{0,0065h}{\text{°C} + 0,0065h + 213,75} \right)^{-5,257} \quad (1)$$

Όπου:

P_0 = Η τρέχουσα τιμή της ατμοσφαιρικής πίεσης σε hPa.

h = Το υψόμετρο του σταθμού από την επιφάνεια της θάλασσας σε μέτρα (m).

°C = η τρέχουσα θερμοκρασία σε βαθμούς Κελσίου.



2.2.3 Βαρομετρική τάση

Βαρομετρική τάση ονομάζεται η μεταβολή της τιμής της ατμοσφαιρικής πίεσης μέσα στην χρονική περίοδο παρατήρησης των 3 ωρών . Στον πίνακα 1 [4] παρατηρούμε τον τρόπο υπολογισμού της βαρομετρικής τάσης με συμβολισμούς:

- $p_0 = t - 3$ ώρες (όπου t η τρέχουσα ώρα),
- $p_1 = t - 2$ ώρες,
- $p_2 = t - 1$ ώρα και
- $p_3 = t$.

Πίνακας 1 Τρόπος Υπολογισμού Βαρομετρικής Τάσης

Code digit	Descriptions		Graphical representation	p_0-p_3	$p_0+p_3-p_1-p_2$	p_0-p_1
0	air pressure is the same as or higher than three hours earlier	rising, then falling		+ 0	- -	- +, 0, -
1	air pressure is higher than three hours earlier	rising, then steady		+	-	+, 0
2	air pressure is higher than three hours earlier	rising (steadily or irregularly)		+	0	+, 0, -
3	air pressure is higher than three hours earlier	falling or steady, then rising; or rising, then rising more rapidly		+	+	+, 0, -
4	air pressure is the same as three hours earlier	steady		0	0	+, 0, -
5	air pressure is the same as or lower than three hours earlier	falling, then rising		0 -	+ +	+, 0, - +
6	air pressure is lower than three hours earlier	falling, then steady; or falling, then falling more slowly		-	+	0, -
7	air pressure is lower than three hours earlier	falling (steadily or irregularly)		-	0	+, 0, -
8	air pressure is lower than three hours earlier	rising or steady, then falling		-	-	+, 0, -

+: result > 0, -: result < 0.



2.2.4 Υγρασία

Ο ατμοσφαιρικός αέρας περιέχει υδρατμούς με διαφορετική ποσότητα από τόπο σε τόπο και από ώρα σε ώρα. Ο αέρας όμως δεν είναι δυνατόν να περιέχει απεριόριστη ποσότητα υδρατμών, αλλά για κάθε θερμοκρασία υπάρχει μια μέγιστη δυνατή περιεκτικότητα υδρατμών. Όταν ο αέρας περιέχει τη μέγιστη τέτοια ποσότητα ονομάζεται κορεσμένος.

Όσο ψυχρότερος είναι ο αέρας τόσο μικρότερη ποσότητα υδρατμών μπορεί να συγκρατήσει. Αν λοιπόν μια μάζα υγρού και θερμού αέρα ψυχθεί θα φτάσει σε μια θερμοκρασία όπου δεν είναι δυνατόν πλέον να συγκρατήσει άλλους τους υδρατμούς από τους οποίους περιέχει. Οι υδρατμοί που περισσεύουν θα συμπυκνωθούν ως σταγονίδια πάνω στα αιωρούμενα μικροσωματίδια και θα δημιουργήσουν το νέφος. Αν δε, συμπυκνωθούν πάνω σε ψύχρα αντικείμενα θα δημιουργήσουν τη δρόσο. Η θερμοκρασία στην οποία ο ακόρεστος αέρας καθώς ψύχεται φτάνει στο κορεσμό, ονομάζεται σημείο δρόσου.

Η υγρασία παίζει σημαντικό ρόλο στην μορφή του κλίματος, τη βλάστηση και τη ζωή ενός τόπου. Μετράμε την υγρασία με τα υγρόμετρα, τα οποία δείχνουν πόσους υδρατμούς περιέχει η ατμόσφαιρα επί της εκατό (%) (όπου 100 θεωρούνται οι υδρατμοί οι οποίοι θα περιέχονταν για την ίδια θερμοκρασία αν είχαμε κορεσμό).

2.2.5 Σημείο δρόσου ή Σημείο υγροποίησης ή σημείο κόρου ατμόσφαιρας

Σημείο δρόσου χαρακτηρίζεται το σημείο εκείνο της θερμοκρασίας που όταν οι υδρατμοί ψυχθούν δημιουργούν το φαινόμενο της δρόσου, δηλαδή τις σταγόνες δρόσου. Στη θερμοκρασία αυτή εξυπακούεται πως όταν ο αέρας είναι κορεσμένος και δεν μπορεί να συγκρατήσει άλλους υδρατμούς η σχετική υγρασία να είναι 100%.

Σημειώνεται όμως ότι η θερμοκρασία κορεσμού της ατμόσφαιρας ή του "σημείου δρόσου" μπορεί να είναι οποιαδήποτε θερμοκρασία, πάνω από τους 0°C.

Η θερμοκρασία αυτή εξαρτάται μόνο από την ποσότητα των υδρατμών που περιέχει 1 κυβικό μέτρο αέρος, συνεπώς εξαρτάται από την απόλυτη υγρασία.

Η θερμοκρασία του σημείου δρόσου αποτελεί σπουδαίο μετεωρολογικό στοιχείο για



ένα τόπο και γι’ αυτό πάντοτε αναφέρεται στους μετεωρολογικούς χάρτες με τα σύμβολα D.P. από τα αρχικά του αγγλικού όρου Dew Point (Δρόσου Σημείο).

2.2.6 Δείκτης Δυσφορίας ή Δείκτης Θερμότητας

Ο δείκτης δυσφορίας [8] δεν αφορά ένα αμιγές μετεωρολογικό φαινόμενο, αλλά παρουσιάζει «πόσο ζέστη αισθανόμαστε πραγματικά» με το συνδυασμό της τρέχουσας θερμοκρασίας και της σχετικής υγρασίας. Εκφράζει δηλαδή το βαθμό δυσφορίας του ανθρώπου λόγω των επιδράσεων των θερμοϋγρομετρικών συνθηκών.

Σε συνθήκες αυξημένης θερμοκρασίας το ανθρώπινο σώμα ρυθμίζει τη θερμοκρασία του μέσω της εφίδρωσης του σώματος. Αποβάλλει θερμότητα με την εξάτμιση του ιδρώτα. Σε συνθήκες αυξημένης σχετικής υγρασίας, η εξάτμιση του ιδρώτα περιορίζεται, με αποτέλεσμα να αισθανόμαστε τη θερμοκρασία υψηλότερη από ό,τι πραγματικά είναι.

Ο υπολογισμός των τιμών του δείκτη δυσφορίας μετριέται σε βαθμούς Κελσίου (°C) και προκύπτει από την παρακάτω σχέση:

$$HI = c_1 + c_2T + c_3R + c_4TR + c_5T^2 + c_6R^2 + c_7T^2R + c_8TR^2 + c_9T^2R^2 \quad (2)$$

Όπου:

- HI= δείκτης δυσφορίας
- T= θερμοκρασία σε βαθμούς Farenait
- R= Η σχετική υγρασία σε ποσοστό %

και οι σταθερές

- $c_1 = -42,379$
- $c_2 = 2,04901523$
- $c_3 = 10,14333127$
- $c_4 = -0,22475541$
- $c_5 = -6,83783 \times 10^{-3}$
- $c_6 = -5,481717 \times 10^{-2}$
- $c_7 = 1,22847 \times 10^{-3}$
- $c_8 = 8,5282 \times 10^{-4}$
- $c_9 = -1,99 \times 10^{-6}$



Ο δείκτης δυσφορίας χωρίζεται σε πέντε ζώνες [7]:

• έως 29 C°	Ιδανικές Συνθήκες.
• από 30 έως 34 C°	Αίσθηση ελαφριάς δυσφορίας.
• από 35 έως 39 C°	Αίσθηση έντονης δυσφορία.
• από 40 έως 45 C°	Αίσθηση υψηλής δυσφορίας.
• από 46 έως 53 C°	Σοβαρός κίνδυνος: σταματήστε κάθε δραστηριότητα.
• από 54 C° και άνω	Κίνδυνος θανάτου: πιθανή ανακοπή καρδίας.

Πίνακας 2 Τιμές Θερμοκρασίας / Υγρασίας για τον υπολογισμό του δείκτη δυσφορίας

	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
42°	48	50	52	55	57	59	62	64	66	68	71	73	75	77	80	82
41°	46	48	51	53	55	57	59	61	64	66	68	70	72	74	76	79
40°	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73	75
39°	43	45	47	49	51	53	55	57	59	61	63	65	66	68	70	72
38°	42	44	45	47	49	51	53	55	56	58	60	62	64	66	67	69
37°	40	42	44	45	47	49	51	52	54	56	58	59	61	63	65	66
36°	39	40	42	44	45	47	49	50	52	54	55	57	59	60	62	63
35°	37	39	40	42	44	45	47	48	50	51	53	54	56	58	59	61
34°	36	37	39	40	42	43	45	46	48	49	51	52	54	55	57	58
33°	34	36	37	39	40	41	43	44	46	47	48	50	51	53	54	55
32°	33	34	36	37	38	40	41	42	44	45	46	48	49	50	52	53
31°	32	33	34	35	37	38	39	40	42	43	44	45	47	48	49	50
30°	30	32	33	34	35	36	37	39	40	41	42	43	45	46	47	48
29°	29	30	31	32	33	35	36	37	38	39	40	41	42	43	45	46
28°	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
27°	27	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
26°	26	26	27	28	29	30	31	32	33	34	35	36	37	38	39	39
25°	25	25	26	27	27	28	29	30	31	32	33	34	34	35	36	37
24°	24	24	24	25	26	27	28	28	29	30	31	32	33	33	34	35
23°	23	23	23	24	25	25	26	27	28	28	29	30	31	32	32	33
22°	22	22	22	22	23	24	25	25	26	27	27	28	29	30	30	31

2.2.7 Άνεμος

Άνεμος ονομάζεται η ροή του αέρα πάνω από την επιφάνεια της Γης και οφείλεται στη μετακίνηση του, ανάμεσα σε δυο περιοχές διαφορετικής πίεσης. Η ροή του αέρα γίνεται από τις περιοχές με υψηλότερη πίεση σε αυτή με την χαμηλότερη. Ο όρος άνεμος αναφέρεται πάντοτε σε οριζόντιες μετακινήσεις του ατμοσφαιρικού αέρα.

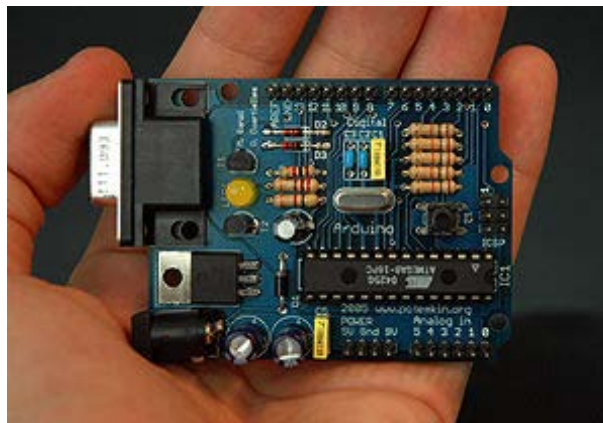
Η διεύθυνση ανέμου είναι το σημείο του ορίζοντα από όπου πνέει ο άνεμος. Η ένταση του ανέμου εκφράζεται από την ταχύτητα του. Η διεύθυνση μετριέται με τον ανεμοδείκτη. Η διαίρεση του ορίζοντα σε 16 διευθύνσεις (ανά 22,5 μοίρες) λέγεται ανεμολόγιο. Οι κύριοι άνεμοι είναι οι Βορράς, Ανατολικός, Νότιος, Δυτικός στις 00, 90, 180, 270 μοίρες αντίστοιχα. Δευτερεύοντες άνεμοι είναι ο Βορειοανατολικός, Νοτιοανατολικός, Νοτιοδυτικός, Βορειοδυτικός με μοίρες 45, 135, 225, 315 αντίστοιχα.



2.3 Η υπολογιστική πλατφόρμα Arduino

Το Arduino είναι μια πλακέτα που βασίζεται σε ανοιχτού υλικού μικροεπεξεργαστή, ο οποίος προγραμματίζεται σε γλώσσα Wiring, μια παραλλαγή της γνωστής γλώσσας C++ [9]. Το Arduino χαρακτηρίζεται για το ελάχιστο κόστος κατασκευής του, την χαμηλή κατανάλωση σε ισχύ, κάτι που αξιοποιείται κυρίως σε φορητές ή και αυτόνομες λύσεις, το μεγάλο πλήθος από περιφερειακές κάρτες για πιο πολύπλοκες και εξειδικευμένες λύσεις, όπως η κάρτα Ardupilot η οποία περιέχει συναρτήσεις και αισθητήρες για την δημιουργία συστήματος αυτόματης πλοήγησης και χρησιμοποιείται για την κατασκευή τηλεκατευθυνόμενων αεροσκαφών [10].

2.3.1 Το Υλικό



Σχήμα 2 Η πλακέτα Arduino με διασύνδεση RS-232

Αποτελείται από μια πλακέτα με πυρήνα, ένα μικροελεγκτή των 8-bit (Atmel AVR – ATmega328 ή ATmega168) και διαθέτει από οχτώ έως δεκαέξι σειριακές θύρες (RS-232) και οχτώ ψηφιακές, αναλόγως την έκδοση της πλακέτας, με τις οποίες αλληλεπιδρά με διάφορες συσκευές. Επίσης, υπάρχουν κάποιες επιπλέον κάρτες γνωστές ως “shields”, οι οποίες προσφέρουν περισσότερες δυνατότητες στον μικροελεγκτή. Αυτές οι κάρτες κατασκευάστηκαν έτσι ώστε να παρέχεται η δυνατότητα πρόσθετης διασύνδεσης του μικροελεγκτή, όπως με δίκτυο τύπου Ethernet (Ethernet shield), ασύρματο δίκτυο wifi, ασύρματη σειριακή επικοινωνία με τεχνολογία Bluetooth (BT Shield) κ.α. Λόγω της ευρείας χρήσης του Arduino, όλο



και περισσότερες shields κατασκευάζονται από τρίτους κατασκευαστές, όπως δέκτης GPS για λήψη της τρέχουσας τοποθεσίας, εμφάνιση πληροφοριών σε οθόνη OLED κλπ. Υπάρχουν πολλών ειδών πλακέτες Arduino οι οποίες έχουν σχεδιαστεί έτσι ώστε να καλύψουν ένα μεγάλο εύρος απαιτήσεων. Για την εργασία χρησιμοποιήθηκε η βασική πλακέτα Arduino με τα εξής τεχνικά χαρακτηριστικά:

Πίνακας 3 Τεχνικά χαρακτηριστικά Arduino Uno

Μικροεπεξεργαστής	ATmega328
Τάση λειτουργίας	5V
Προτεινόμενη τάση εισόδου	7-12V
Τάση εισόδου (οριακή)	6-20V
Ψηφιακές εισοδοί /έξοδοι	14 (6 εκ των οποίων παρέχουν PWM έξοδος)
Αναλογικές εισοδοί	6
Τιμή DC τάσης ανά I/O	40 mA
Τιμή DC τάσης για την 3.3V	50 mA
Μέγεθος μνήμης Flash	32 KB (ATmega328)
Μνήμη SRAM	2 KB (ATmega328)
Μνήμη EEPROM	1 KB (ATmega328)
Ταχύτητα ρολογιού	16 MHz

2.3.2 Το Λογισμικό

Το Ολοκληρωμένο Περιβάλλον Ανάπτυξης (Integrated Development Environment – IDE) με το οποίο θα προγραμματίσουμε το λογισμικό του μετεωρολογικού σταθμού, έχει υλοποιηθεί με την γλώσσα Java η οποία εκτελείται ανεξάρτητα από το λειτουργικό ή υλικό του υπολογιστή, είναι ευρέως διαδεδομένη, και έχει πολλές ομοιότητες με την γλώσσα C. Κατά συνέπεια, το συγκεκριμένο IDE είναι πολλαπλού λειτουργικού (multi-platform) και μπορεί να δουλεύει άψογα σε κάθε λειτουργικό σύστημα (windows, Linux, OSX). Η γλώσσα wiring είναι μια παραλλαγή της C++ και όπως και το IDE, είναι μια πλατφόρμα ανοιχτού λογισμικού που χρησιμοποιείται για προγραμματισμό μικροελεγκτών. Έχει επίσης σχεδιαστεί λαμβάνοντας υπόψη μια ευρεία γκάμα χρηστών, από αρχάριους μέχρι επαγγελματίες, με σκοπό την δημιουργία μιας κοινότητας η οποία θα ανταλλάσσει ιδέες, τεχνογνωσία και εμπειρία,



με τελικό στόχο την κατασκευή “έξυπνων συσκευών” οι οποίες θα περιορίζονται μόνο από την φαντασία του εκάστοτε δημιουργού τους.

```
Blink | Arduino 1.0
File Edit Sketch Tools Help

Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repe
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}
```

Σχήμα 3: Το περιβάλλον ανάπτυξης του Arduino

Κύριο χαρακτηριστικό της wiring είναι ότι απαιτεί την ύπαρξη δυο τουλάχιστον συναρτήσεων. Της setup() και της loop(). Η setup() περιέχει όλες εκείνες τις μεταβλητές οι οποίες απαιτούνται για την αρχικοποίηση του προγράμματος. Η loop(), αφού έχει εκτελεστεί η setup(), περιέχει την δομή του υπόλοιπου προγράμματος και εκτελείται συνεχόμενα πλην της περίπτωσης να έχει δοθεί εντολή τερματισμού μετά από συγκεκριμένες συνθήκες.



Σχήμα 4 Διάγραμμα ροής λειτουργίας της γλώσσας wiring

2.4 Περιγραφή μεθοδολογίας της λύσης του προβλήματος

Πριν την οποιαδήποτε σχεδίαση του σταθμού θα πρέπει να επιλέξουμε ένα μετεωρολογικό μοντέλο ικανό, έτσι ώστε με τα δεδομένα που εισάγονται από τους αισθητήρες του να είναι δυνατή η πρόγνωση του καιρού. Σύμφωνα με τον Διεθνή



Οργανισμό Μετεωρολογίας (World Meteorological Organization – WMO), και το εγχειρίδιο που έχει εκδώσει [1], τα μετεωρολογικά δεδομένα θα πρέπει να συλλέγονται με συγκεκριμένη δειγματοληψία. Στην συνέχεια τα δεδομένα αυτά θα χρησιμοποιηθούν για την πρόγνωση του καιρού η οποία συνίσταται να προβλέπεται στις 06:00, 12:00, 18:00 και 00:00 τοπικής ώρας για τα χρονικά διαστήματα των 6, 12, 18 και 24 ωρών, αναλόγως του μοντέλου [11].

Η πρόγνωση καιρού είναι μια επίμονη διαδικασία για την οποία, πέραν των τοπικών δεδομένων που έχουν συλλεχθεί από κάποιον ΑΜΣ, απαιτείται η κατασκευή και μελέτη ενός μετεωρολογικού χάρτη ο οποίος θα εμφανίζει την κίνηση διαφόρων συστημάτων (υφέσεων, αντικυκλώνων, μετώπων κλπ) για την χρονική περίοδο στην οποία αναφέρεται η πρόγνωση, βασιζόμενη όμως σε ευρύτερο γεωγραφικό πεδίο. Οι μέθοδοι που ακολουθούνται για την πρόγνωση του καιρού αξιοποιούν κυρίως εμπειρικές και στατιστικές παρατηρήσεις από τις οποίες μπορούν να εξαχθούν ασφαλείς αλλά όχι απόλυτες προβλέψεις. Επιγραμματικά τέτοιες μέθοδοι είναι [12]:

Ο κανόνας της Εμμονής: “Οι τιμές των κινήσεων και των μεταβολών των συστημάτων που επικρατούσαν στο πρόσφατο παρελθόν, τείνουν να διατηρηθούν και στο μέλλον.” Υπολογίζουμε την ταχύτητα κίνησης των βαρομετρικών συστημάτων και μετώπων. Βάσει αυτού, εκτιμούμε ότι ο καιρός στην περιοχή μας, θα είναι ο χαρακτηριστικός καιρός καθενός από αυτά όταν φτάσουν σε εμάς.

Ο κανόνας του Γεωστροφικού Ανέμου. “Οι αέριες μάζες μετακινούνται με την ταχύτητα περίπου του Γεωστροφικού Ανέμου”. Τα θερμά μέτωπα κινούνται με ταχύτητες 60%-80% του Γεωστροφικού Ανέμου. Τα ψυχρά μέτωπα κινούνται με ταχύτητες 80%-90% του ανέμου.

Ο κανόνας της βαρομετρικής τάσης: “Υπολογίζουμε τις τιμές της βαρομετρικής πίεσης στο μέλλον ως αποτέλεσμα του γινομένου της βαρομετρικής τάσης επί τον χρόνο”. Υπολογίζουμε την μεταβολή της ατμοσφαιρικής πίεσης εντός τριώρου και εάν υπάρχει αισθητή διαφορά, τότε με βάση εμπειρικές παρατηρήσεις συμπεραίνουμε ότι στην περιοχή θα επικρατήσει καλοκαιρία ή κακοκαιρία.

Η ακρίβεια των εν λόγω κανόνων εξαρτάται κυρίως από τις τοπικές περιβαλλοντολογικές συνθήκες όπως το υψόμετρο, η γεωγραφική θέση ακόμα και το ημισφαίριο στο οποίο ανήκει η περιοχή. Κατά συνέπεια η χρήση τέτοιων κανόνων



θα πρέπει να γίνεται εφόσον διερευνηθεί το μετεωρολογικό ιστορικό της περιοχής και εντοπιστούν οι παράμετροι εκείνοι οι οποίες καθορίζουν την λειτουργία του κάθε μοντέλου. Τα μοντέλα που θα χρησιμοποιηθούν και περιγράφονται αναλυτικά παρακάτω βασίζονται σε παρατηρήσεις που έχουν γίνει μόνο για το βόρειο ημισφαίριο.

Τα τελευταία χρόνια έχουν γίνει μεγάλα άλματα στα μαθηματικά μοντέλα προγνώσεως καιρού λόγω της αύξησης της υπολογιστικής ισχύος, τα οποία μάλιστα έχουν την δυνατότητα πρόβλεψης καιρού για χρονικά διαστήματα μεγαλύτερα των 24 ωρών. Τέτοια μοντέλα όμως είναι απαγορευτικά για την υπολογιστική ισχύ του μικροεπεξεργαστή του Arduino καθώς πλέον της ισχύος απαιτείται και μεγάλη μνήμη λόγω της εισόδου μεγάλου όγκου μετεωρολογικών δεδομένων. Κατά συνέπεια δεν είναι δυνατή η εφαρμογή και αξιοποίηση τους στην παρούσα εργασία [2].

Λόγω των ανωτέρω, το μετεωρολογικό μοντέλο που θα χρησιμοποιηθεί στην παρούσα εργασία θα βασίζεται στις παρατηρήσεις του κανόνα βαρομετρικής πίεσης καθώς θα απαιτηθεί μόνο η καταγραφή της τρέχουσας ατμοσφαιρικής πίεσης και της κατεύθυνσης του ανέμου. Στην συνέχεια οι τιμές αυτές θα συγκριθούν με βάση τους γενικούς κανόνες που ακολουθεί το μοντέλο και θα επιστρέφεται η πρόγνωση του καιρού.

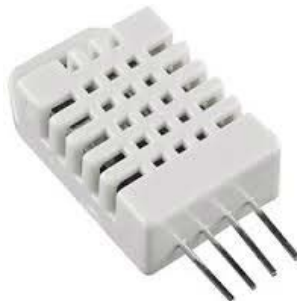
2.5 Περιγραφή των αισθητήρων καταγραφής.

Οι αισθητήρες καταγραφής συνδέονται στις ψηφιακές και αναλογικές εισόδους του Arduino το οποίο κάνοντας χρήση των κατάλληλων βιβλιοθηκών, διαβάζει τις τιμές τάσης στην είσοδο του και στην συνέχεια μετατρέπει την εισερχόμενη τάση σε τιμές τις οποίες μπορούμε να αξιοποιήσουμε. Αναλυτικά οι αισθητήρες που θα χρειαστούμε για την κατασκευή του σταθμού είναι:

- α) Αισθητήρας θερμοκρασίας / υγρασίας DHT-22
- β) Αισθητήρας Ατμοσφαιρικής Πίεσης BMP-085
- γ) Αισθητήρας Κατεύθυνσης Ανέμου



2.5.1 Αισθητήρας Θερμοκρασίας / Υγρασίας DHT-22



Σχήμα 5 ο Αισθητήρας DHT-22

Ο αισθητήρας DHT 22 [13] είναι ένας φθηνός και αξιόπιστος αισθητήρας. Έχει μικρό μέγεθος, μικρή κατανάλωση ρεύματος και μπορεί να μεταδίδει δεδομένα στην πλακέτα κάθε 2 sec. Λόγω αυτών των ιδιοτήτων του έχει χρησιμοποιηθεί σε πολλές εφαρμογές και υπάρχει πληθώρα βιβλιοθηκών στο διαδίκτυο. Τα τεχνικά χαρακτηριστικά του είναι:

Πίνακας 4 Τεχνικά Χαρακτηριστικά DHT-22

	Υγρασία	Θερμοκρασία
Τάση	3.3-5V	
Έξοδος	Ψηφιακή	
Αισθητήρας	Πολυμερής Πυκνωτής	DS18B20
Εύρος	0-100% Rh	-40 – 125 C°
Ακρίβεια	2-5%	±0,5 C°
Δειγματοληψία	0,5 Hz	

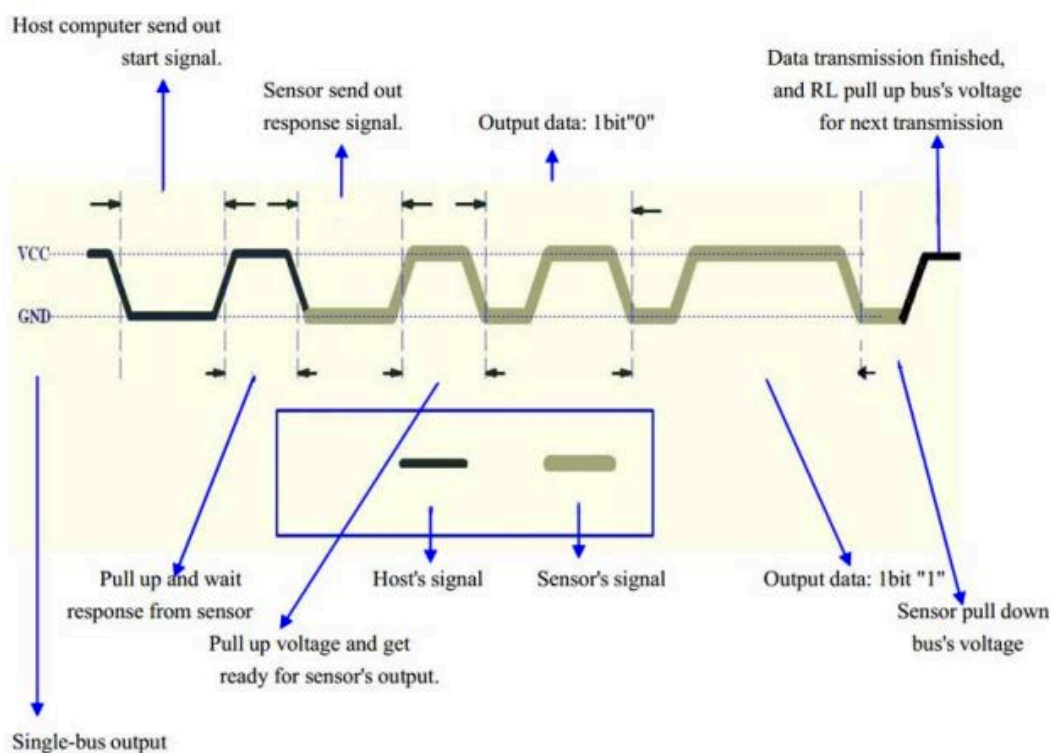
Σύμφωνα με τον WMO η ευαισθησία του αισθητήρα θερμοκρασίας για καταγραφή περιβαλλοντολογικής θερμοκρασίας έχει οριστεί από -30 C° έως 50 C° [14] και για την υγρασία από 5% έως 100% [15], άρα η χρήση του εν λόγω αισθητήρα είναι η ιδανική δεδομένου του κόστους του.

Όπως παρατηρούμε στο σχήμα 5 το Arduino στέλνει ένα σήμα έναρξης της τάξης των 500us περίπου και αναμένει απάντηση από τον αισθητήρα. Ο αισθητήρας με την μείωση του σήμα στα 80us στέλνει εκ νέου ένα όμοιο σήμα και τέλος στέλνει ένα ακόμα σήμα της τάξης των 50us (διαδικασία χειραψίας). Τα επόμενα σήματα που



ακολουθούν είναι ψηφιακά με αντιστοιχία στα 28 us για το 0 και 70 για το λογικό 1. Κάθε δύο δευτερόλεπτα, ο αισθητήρας εκπέμπει 40 bit πληροφορίας τα οποία αντιστοιχούν σε

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



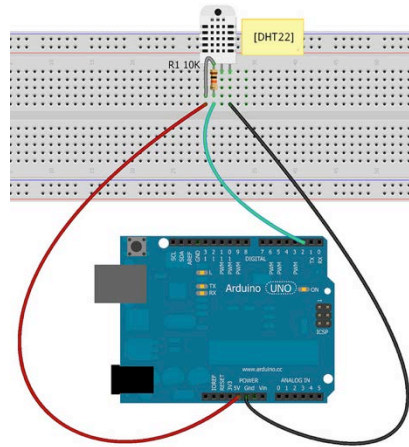
Σχήμα 6 Διάγραμμα μεταφοράς δεδομένων DHT-22

Συνδεσμολογία με την πλακέτα Arduino

Η συνδεσμολογία μεταξύ του αισθητήρα και της πλακέτας Arduino είναι αρκετά απλή. Στον ακροδέκτη #1 του αισθητήρα συνδέεται η γείωση (ground), στον ακροδέκτη #4 η τάση (+5v ή 3.3v) και στον ακροδέκτη #2 μία από τις ψηφιακές

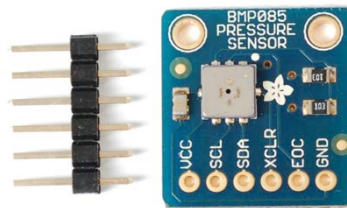


εισόδους του Arduino, στην εργασία αυτή έχει συνδεθεί στην θύρα #5. Στο σχήμα 7 βλέπουμε τον τρόπο συνδεσμολογίας του αισθητήρα με την πλακέτα.



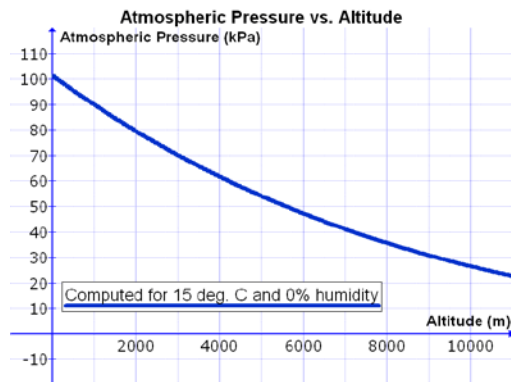
Σχήμα 7 Υπόδειγμα συνδεσμολογίας του αισθητήρα με το Arduino

2.5.2 Αισθητήρας Ατμοσφαιρικής Πίεσης BMP-085



Σχήμα 8 ο αισθητήρας BMP-085

Ο αισθητήρας BMP-085 [16] έχει σχεδιαστεί για την μέτρηση ειδικά της ατμοσφαιρικής πίεσης αλλά και της θερμοκρασίας της ατμόσφαιρας. Επίσης λόγω της συσχέτισης της ατμοσφαιρικής πίεσης και του ύψους στην οποία γίνεται η μέτρηση, η βιβλιοθήκη παρέχει και την επιπλέον λειτουργία, της ένδειξης του τρέχοντος υψομέτρου της περιοχής του αισθητήρα. Η γραφική παράσταση της συνάρτησης μεταξύ του ύψους και της ατμοσφαιρικής πίεσης εμφανίζεται στο σχήμα 9.



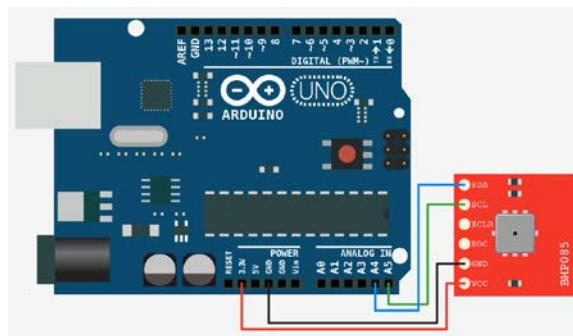
Σχήμα 9 Γραφική παράσταση σχέσης ύψους και ατμοσφαιρικής πίεσης (πηγή: wikipedia)

Οι τεχνικές προδιαγραφές του αισθητήρα είναι :

- Εύρος ευαισθησίας αισθητήρα: 300-1100 hPa (-500m έως 9000m υψόμετρο)
- Ευαισθησία: 0.03 hPa / 0.25m
- Θερμοκρασία περιβάλλοντος: -40 C έως +85 C
- Τροφοδοσία: 3,3 V – 5V

Σύμφωνα με τον WMO οι απαιτήσεις μέτρησης της ατμοσφαιρικής πίεσης είναι από 500 έως 1080 hPa και με ευαισθησία 0.1 hPa [17], το οποίο καθιστά τον αισθητήρα ιδανικό για ακριβείς μετρήσεις, σε σχέση με το κόστος του.

Συνδεσμολογία αισθητήρα με την πλακέτα Arduino



Σχήμα 10 Συνδεσμολογία του αισθητήρα με την πλακέτα Arduino

Στο σχήμα 10 βλέπουμε την συνδεσμολογία του αισθητήρα με την πλακέτα. στον ακροδέκτη VCC συνδέεται η τάση 3.3V, στο GRD η γείωση. Ο ακροδέκτης SCL συνδέεται με το ρολόι του arduino για τον συγχρονισμό του αισθητήρα. Για το



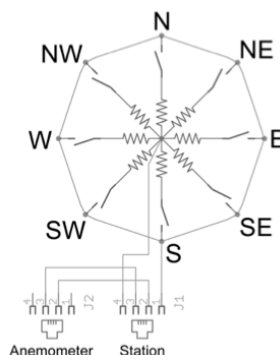
Arduino UNO αυτό βρίσκεται στην αναλογική είσοδο #5. Ο ακροδέκτης SDA συνδέεται με το πρωτόκολλο επικοινωνίας i2c και είναι υπεύθυνο για την ανάγνωση των δεδομένων του αισθητήρα.

2.5.3 Αισθητήρας Κατεύθυνσης Ανέμου 80422



Σχήμα 11 ο Αισθητήρας κατεύθυνσης ανέμου 80422

Πρόκειται για έναν σχετικά οικονομικό αισθητήρα, ο οποίος βασίζεται στην εξής λειτουργία: υπάρχει ένα δίκτυο από 8 διαφορετικές αντιστάσεις, μια για κάθε κατεύθυνση του ανέμου [21]. Όπως φαίνεται στο σχήμα 12 όταν ο δείκτης στραφεί προς μια κατεύθυνση τότε γίνεται η σύνδεση μεταξύ των 2 αντιστάσεων. Στην έξοδο (ακροδέκτης 4) του αισθητήρα διέρχεται ένα μοναδικό αναλογικό σήμα, το οποίο ανταποκρίνεται σε μια κατεύθυνση (Σχήμα 12)



Σχήμα 12 Συνδεσμολογία αντιστάσεων για τον εντοπισμό της κατεύθυνσης του ανέμου



Πίνακας 5 Αντιστοιχία μεταξύ μοιρών και τάσης εξόδου του αισθητήρα

Direction (Degrees)	Resistance (Ohms)	Voltage (V=5v, R=10k)
0	33k	3.84v
22.5	6.57k	1.98v
45	8.2k	2.25v
67.5	891	0.41v
90	1k	0.45v
112.5	688	0.32v
135	2.2k	0.90v
157.5	1.41k	0.62v
180	3.9k	1.40v
202.5	3.14k	1.19v
225	16k	3.08v
247.5	14.12k	2.93v
270	120k	4.62v
292.5	42.12k	4.04v
315	64.9k	4.78v
337.5	21.88k	3.43v

2.6 Απλοποιημένα Μετεωρολογικά μοντέλα

Στις αρχές του 20^{ου} αιώνα όπου δεν υπήρχαν ηλεκτρονικοί υπολογιστές οι οποίοι θα μπορούσαν να πραγματοποιούν επίλυση διαφορικών εξισώσεων με μεγάλη ταχύτητα, η μετεωρολογία βασιζόταν κυρίως στις παρατηρήσεις της μεταβολής της βαρομετρικής τάσης. Έτσι παρατηρήθηκε π.χ. πως όταν η πίεση πέφτει γρήγορα σε διάστημα 3 ωρών τότε θα ακολουθήσει συννεφιά [18]. Με βάση αυτές τις παρατηρήσεις κατασκευάστηκαν μηχανές στις οποίες ο χειριστής εισήγαγε μετεωρολογικά δεδομένα και εξήγαγε μια υποτυπώδη πρόγνωση. Βέβαια τα αποτελέσματα δεν είχαν την ακρίβεια που απαιτούνται για να χρησιμοποιηθούν πχ σε πολεμικές επιχειρήσεις, αλλά σε ερασιτεχνικές ή ημιεπαγγελματικές χρήσεις όπως πχ στην γεωργία, στην αλιεία κλπ. Δύο από τις πιο διαδεδομένες συσκευές πρόγνωσης καιρού, μέχρι την δεκαετία του 60, είναι ο προγνώστης «Σάγκερ» (Sager) [19] και ο προγνώστης «Ζαμπρέτι» (Zambretti) [20]. Το μοντέλο που θα χρησιμοποιηθεί, προκύπτει από την μελέτη του τρόπου λειτουργίας του προγνώστη Ζαμπρέτι, με μεταβλητές:

- την κατεύθυνση του ανέμου,
- την τρέχουσα τιμή της ατμοσφαιρικής πίεσης,
- την βαρομετρική τάση,



- την μέγιστη και ελάχιστη καταγραφείσα τιμή της ατμοσφαιρικής πίεσης
- η εποχή (καλοκαίρι ή χειμώνας),
- η κατεύθυνση του ανέμου,

και επιστρέφει την πρόγνωση του καιρού για τις επόμενες δώδεκα ώρες.

2.6.1 Αλγόριθμος Ζαμπρέττι (Zambretti Algorithm)



Σχήμα 13 Μετεωρολογικό όργανο πρόγνωσης καιρού Ζαμπρέττι

Ο αλγόριθμος σχεδιάστηκε από τους Νεργέτι (Nergetii) και Ζάμπρα (Zambra) δύο Άγγλους κατασκευαστές οργάνων το 1915. Το όργανο αυτό έχει ακρίβεια σε ποσοστό 91% και χρονική διάρκεια έως 12 ώρες [20] και ισχύει μόνο για το βόρειο ημισφαίριο. Η χρήση του θεωρείται ιδανική για αγρότες, περιβαλλοντολόγους, αθλητές κλπ. Το όργανο αυτό αποτελείται από τρεις δίσκους. Ο πρώτος δίσκος (εξωτερικός) περιέχει πληροφορίες σχετικά με την κατεύθυνση του ανέμου. Ο δεύτερος έχει ένα δείκτη για να ευθυγραμμίζεται σωστά με την κατεύθυνση του ανέμου στον δίσκο Α και η ίδια φέρει κλίμακα βαθμονομημένη σε μονάδες της βαρομετρικής πίεσης (ίντσες του υδραργύρου ή millibars.) Ο εσωτερικός δίσκος C έχει ένα ευρετήριο ρυθμιζόμενο με την ατμοσφαιρική πίεση στην Β και τρεις οπές οι οποίες επιτρέπουν την επιλογή της ανάγνωσης η οποία βασίζεται από την άνοδο ή την πτώση του βαρόμετρου το χειμώνα ή το καλοκαίρι. Μέσα από την οπή προβάλλεται ένα γράμμα της αγγλικής αλφαβήτου, το οποίο αποτελεί το κλειδί για την πρόβλεψη που μπορεί να διαβαστεί στην εμπρόσθια πλευρά της συσκευής.

Υπάρχουν τρεις διαφορετικές συναρτήσεις, μία για κάθε κατάσταση της

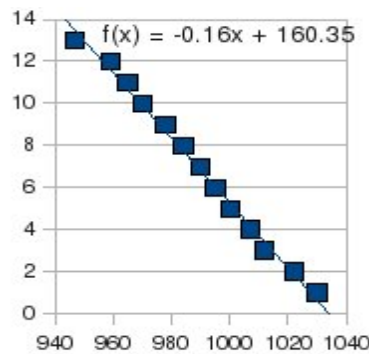


βαρομετρικής τάσης, από όπου εξάγεται το αποτέλεσμα πρόγνωσης:

α) Όταν η βαρομετρική τάση ανεβαίνει τότε η συνάρτηση που αντιστοιχεί είναι

$$Z_{rizing} = 130 - \left(\frac{p}{81}\right) \quad (3)$$

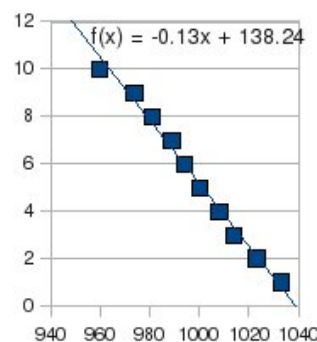
όπου p η τρέχουσα ατμοσφαιρική πίεση σε hPa, η γραφική παράσταση της οποίας διαγράφεται στο σχήμα 14.



Σχήμα 14 γραφική παράσταση συνάρτησης αυξανόμενης βαρ. τάσης
β) όταν η βαρομετρική τάση παραμένει σταθερή, τότε ισχύει:

$$Z_{steady} = 147 - \left(\frac{5 \cdot p}{376}\right) \quad (4)$$

η γραφική παράσταση της οποίας παρουσιάζεται στο σχήμα 15



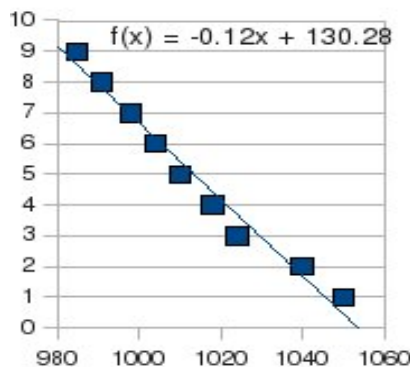
Σχήμα 15 γραφική παράσταση συνάρτησης σταθερής βαρ. τάσης



γ) όταν η βαρομετρική τάση πέφτει τότε ισχύει:

$$Z_{falling} = 179 - \left(\frac{2 \cdot p}{129}\right) \quad (5)$$

η γραφική παράσταση της οποίας είναι



Σχήμα 16 Γραφική παράσταση φθίνουσας βαρομετρικής τάσης (πηγή:
<http://www.meteormetrics.com/zambretti.htm>)

Το Z που προκύπτει είναι ένας ακέραιος αριθμός στον οποίο αντιστοιχεί η παρακάτω πρόβλεψη:

Πίνακας 6 Αποτελέσματα εξίσωσης και αντιστοιχία τους με τον δίσκο Zambretti

Αποτέλεσμα εξίσωσης Z	Γράμμα δίσκου Ζαμπρετι	Πρόγνωση	Ατμ. Πίεση
1;	A;	Αίθριος αμετάβλητος	1050
2;	B;	Αίθριος	1040
3;	D;	Αίθριος μεταβαλλόμενος (Αίθριος, λιγότερο αμετάβλητος)	1024
4;	H;	Σχεδόν αίθριος, αργότερα πιθανώς ελαφρές βροχοπτώσεις.	1018
5;	O;	Ελαφρές βροχοπτώσεις εν συνεχεία άστατος.	1010
6;	R;	Άστατος, αργότερα βροχοπτώσεις.	1004
7;	U;	Βροχοπτώσεις κατά διαστήματα, αργότερα επιδείνωση.	998
8;	V;	Βροχοπτώσεις κατά διαστήματα, εν συνέχεια έντονα άστατος.	991
9;	X;	Έντονα άστατος, βροχοπτώσεις.	985
10;	A;	Αίθριος αμετάβλητος	1033
11;	B;	Αίθριος	1023



12;	E;	Αίθριος, πιθανές ελαφρές βροχοπτώσεις	1014
13;	K;	Σχεδόν αίθριος, πολύ πιθανόν ελαφρές βροχοπτώσεις	1008
14;	N;	Ελαφρές βροχοπτώσεις, με διαστήματα ηλιοφάνειας	1000
15;	P;	Μεταβλητός με μερικές βροχοπτώσεις.	994
16;	S;	Άστατος, βροχοπτώσεις κατά διαστήματα.	989
17;	W;	Συχνές βροχοπτώσεις.	981
18;	X;	Έντονα άστατος, βροχοπτώσεις.	974
19;	Z;	Θυελλώδης, έντονες βροχοπτώσεις.	960
20;	A;	Αίθριος αμετάβλητος	1030
21;	B;	Αίθριος	1022
22;	C;	Μεταβολή σε αίθριο	1012
23;	F;	Σχεδόν αίθριος, σε βελτίωση	1007
24;	G;	Σχεδόν αίθριος, γρήγορα πιθανώς ελαφρές βροχοπτώσεις.	1000
25;	I;	Αρχικά ελαφρές βροχοπτώσεις, βελτίωση.	995
26;	J;	Μεταβλητός σε βελτίωση.	990
27;	L;	Σχεδόν άστατος, βελτίωση αργά στην ημέρα	984
28;	M;	Άστατος, πιθανή βελτίωση	978
29;	Q;	Άστατος, αίθριος κατά διαστήματα.	970
30;	T;	Έντονα άστατος, αίθριος κατά διαστήματα.	965
31;	Y;	Θυελλώδης, πιθανή βελτίωση.	959
32;	Z;	Θυελλώδης, έντονες βροχοπτώσεις.	947

Υπολογισμός Ανέμου / εποχής

Μέχρι τώρα δεν έχει ληφθεί υπόψη η κατεύθυνση του ανέμου, ούτε η εποχή στην οποία γίνεται η πρόγνωση. Ο βοριάς υπολογίζεται γενικά ότι βελτιώνει τον καιρό από 1 μονάδα πρόβλεψης και μια αφαίρεση από την τιμή του Z του πίνακα 6.

Ομοίως, ο νότιος εκτιμάται ότι θα επιδεινώσει τον καιρό μέχρι και 2 μονάδες.

Ο αλγόριθμος υπολογίζει με βάση τις εποχές (χειμώνας ή καλοκαίρι). Ο χειμώνας γενικά οδηγεί σε χαμηλότερη τιμή του Z κατά 1 μονάδα σε σύγκριση με το καλοκαίρι



που. Παρομοίως το καλοκαίρι, βελτιώνει την πρόγνωση του Z κατά 1 μονάδα επάνω.

Συμπέρασμα

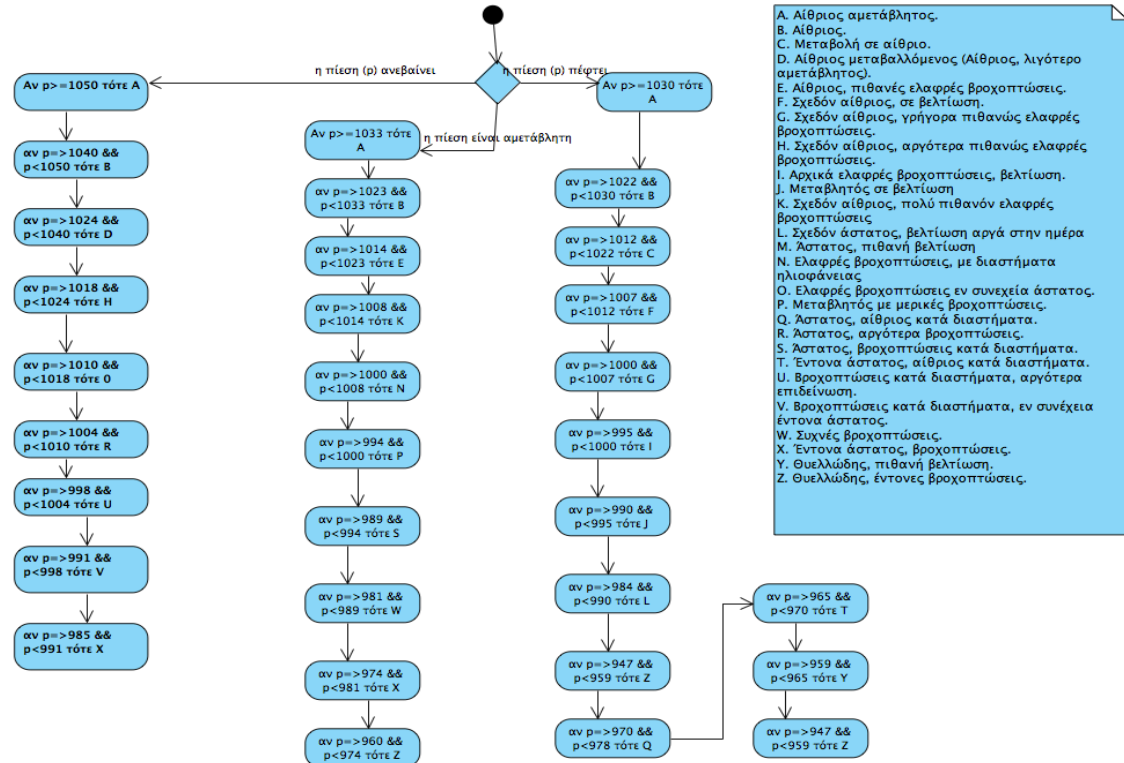
Για να κάνουμε χρήση του αλγορίθμου θα πρέπει να ακολουθήσουμε τα ακόλουθα βήματα:

- 1) Υπολογισμός της βαρομετρικής τάσης
- 2) Με βάση το 1), χρήση της ανάλογης συνάρτησης Z_t , Z_s , Z_f
- 3) Μέτρηση της τρέχουσας ατμοσφαιρικής πίεσης
- 4) Υπολογισμός του Z
- 5) Τροποποίησε το Z σύμφωνα με την κίνηση του ανέμου
- 6) Τροποποίησε Z σύμφωνα με την εποχή,
- 7) Το αποτέλεσμα Z χρησιμοποίησε το και βρες την αντιστοιχία στον πίνακα 6.

Στο σχήμα 17 παρατηρούμε το διάγραμμα ροής του αλγόριθμου Ζαμπρέττι.



Τριανταφύλλου Γεώργιος,
 'Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
 Πρόβλεψης Μετεωρολογικών Φαινομένων



Σχήμα 17 Διάγραμμα Ροής Αλγόριθμου Ζαμπρέττι



3. Καταγραφή και ανάλυση απαιτήσεων

Ο μετεωρολογικός σταθμός θα πρέπει να είναι σε θέση να καταγράφει τα δεδομένα με ακρίβεια και σε προκαθορισμένο σε μερικές περιπτώσεις χρονικό περιθώριο. Για τον λόγο αυτό το ρολόι του συστήματος θα πρέπει να είναι συγχρονισμένο με την τοπική ώρα και να είναι τακτικός ο έλεγχος για την ακρίβειά της. Κατά την εκκίνηση του σταθμού θα πρέπει να συνδέεται με κάποιον εξυπηρετητή πρωτοκόλλου ώρας (NTP Server), και σε περίπτωση απόκλισης να ρυθμίζεται αυτόματα η ημερομηνία / ώρα. Στην συνέχεια θα πρέπει να γίνεται έλεγχος για την σωστή λειτουργία των αισθητήρων και σε περίπτωση σφάλματος, αυτό να εμφανίζεται στην LCD οθόνη καθώς και στην ιστοσελίδα του ΑΜΣ. Σύμφωνα με τις οδηγίες του WMO [25] η διαδικασία καταγραφής για την θερμοκρασία, υγρασία, ατμοσφαιρική πίεση και κατεύθυνση του ανέμου είναι κοινή και περιγράφεται ως εξής:

α) Κάθε δώδεκα (12) δευτερόλεπτα συλλέγεται ένα δείγμα για κάθε μετεωρολογικό φαινόμενο. Αυτά έχουν καθοριστεί στα δευτερόλεπτα 12'', 24'', 36'', 48'' και 00'' του επόμενου λεπτού. Στην συνέχεια ο μέσος όρος των 5 αυτών δειγμάτων αποτελεί την μέτρηση για το λεπτό της ώρας. Αυτή η μέτρηση θα αποθηκεύεται σε κάρτα SD για στατιστικούς λόγους, καθώς και στην προσωρινή μνήμη για τον υπολογισμό των μέσων ωριαίων τιμών.

β) Κάθε μια ώρα προκύπτει η μέση τιμή των τελευταίων 60 μετρήσεων (κάθε λεπτού δηλαδή). Ο χρόνος που μπορεί να υπολογιστεί ο μέσος όρος είναι ανάμεσα στο 50' και στο 58' λεπτό της ώρας. Η διαδικασία αυτή είναι απαραίτητη για τον υπολογισμό της βαρομετρικής τάσης.



γ) κάθε τρεις ώρες υπολογίζεται η βαρομετρική τάση η οποία βασίζεται στον εξής αλγόριθμο. Αν P_n η τρέχουσα μέση τιμή της πίεσης και P_{n-3} η τιμή της πριν από 3 ώρες τότε αν:

- i) $P_n - P_{n-3} > 0$ η βαρομετρική τάση ανεβαίνει,
- ii) $P_n - P_{n-3} < 0$ η βαρομετρική τάση πέφτει,
- iii) $P_n - P_{n-3} = 0$ η βαρομετρική τάση παραμένει σταθερή.

εκτός των παραπάνω θα πρέπει να λάβουμε υπόψη τα ακόλουθα:

α) για τον υπολογισμό της μέγιστης και ελάχιστης τιμής της θερμοκρασίας, θα πρέπει η ώρα καταγραφής να είναι από την 06:00 ώρα UTC έως την 18:00 UTC [26].

β) για τον υπολογισμό του σημείου δρόσου θα χρειαστούμε τους εξής τύπους:

$$k = \frac{\log_{10}(h) - 2}{0,4343} + \frac{17,62 * t}{243,12 + t} \quad (6)$$

$$dewpoint = \frac{243.12 * k}{17.62 - k} \quad (7)$$

όπου h η τρέχουσα υγρασία και όπου t η τρέχουσα θερμοκρασία.

γ) Η πρόγνωση του καιρού γίνεται κάθε 6 ώρες και συγκεκριμένα στις 06:00, 12:00, 18:00, 00:00 τοπική ώρα. Ο αλγόριθμος που θα χρησιμοποιήσουμε προβλέπει την κατάσταση του καιρού για την χρονική διάρκεια των 12 ωρών.

δ) Τα μετεωρολογικά δεδομένα θα αποθηκεύονται στην κάρτα SD με μορφή αρχείου κειμένου CSV. Θα αποθηκεύονται πέντε αρχεία, από ένα για κάθε μετεωρολογικό δεδομένο με το σύνολο των μετρήσεων της ημέρας και επιπλέον ένα αρχείο με την καταγραφή όλων των μέσων τιμών της ημέρας.



Θα πρέπει να ληφθεί υπόψη το υψόμετρο του σημείου όπου βρίσκεται ο ΑΜΣ για την ορθή ρύθμιση του αισθητήρα ατμοσφαιρικής πίεσης, ο οποίος σε κάθε μετακίνηση του θα πρέπει να υπολογίζεται εκ νέου σύμφωνα με την διαδικασία αναγωγής στην επιφάνεια της θάλασσας. Ο σταθμός έχει εγκατασταθεί στην περιοχή του Νέου Φαλήρου Αττικής στο γεωγραφικό σημείο φ:37° 56' 49'' Β και λ:23° 39' 59'' Α. Το υψόμετρο του σημείου που έχει εγκατασταθεί ο ΑΜΣ είναι 71m από την επιφάνεια της θάλασσας και έχει υπολογιστεί με την χρήση δέκτη GPS. Για την ορθή μέτρηση της ατμοσφαιρικής πίεσης θα πρέπει γίνει η απαραίτητη αναγωγή έτσι ώστε να μπορεί να υπολογιστεί σωστά η ατμοσφαιρική πίεση [18].

3.1 Πλάνο έργου (Project Plan)

Με βάση τις παραπάνω απαιτήσεις τα υποέργα τα οποία απαρτίζουν την παρούσα εργασία είναι:

- α) Καταγραφή και ανάλυση απαιτήσεων.
- β) Αγορά και συναρμολόγηση αισθητήρων / breadboard / Ethernet shield
- γ) Σχεδίαση βιβλιοθηκών και προσαρμογή των υπαρχουσών στα δεδομένα της εργασίας.
- δ) Σχεδίαση και κατασκευή βασικής βιβλιοθήκης η οποία θα έχει συγκεντρωμένες όλες τις συναρτήσεις / διαδικασίες που απαιτούνται για την λειτουργία του σταθμού.
- ε) Σχεδίαση και κατασκευή του αλγορίθμου Zambretti.
- στ) Σχεδίαση του γραφικού προγράμματος (GUI) της δικτυακής ιστοσελίδας του ΑΜΣ καθώς και μεθόδους αλληλεπίδρασης του με τον χρήστη.
- ζ) Σύνδεση του σταθμού στο δίκτυο και έλεγχος λειτουργίας
- η) Ολοκλήρωση συστήματος.



θ) Δοκιμές συστήματος.

ι) Διορθώσεις σφαλμάτων και βελτιώσεις

3.2 Χρονοδιάγραμμα εργασιών

Οι φάσεις με την οποία θα χωριστεί η κατασκευή του ΑΜΣ είναι:

1η Φάση (1/12/12-7/12/12).

1.1 Ανάλυση απαιτήσεων

1.2 Αγορά και συναρμολόγηση εξοπλισμού και αισθητήρων.

2^η Φάση (8/12/12-24/12/12)

2.1 Σχεδίαση Διαγραμμάτων ροής / περιπτώσεων χρήσεων/ διάγραμμα δραστηριοτήτων

3η Φάση (24/12/12 – 6/1/13).

Διακοπές Χριστουγέννων

4η Φάση (7/1/13 – 22/1/13).

3.1 Σύνδεση αισθητήρων με την πλακέτα Arduino

3.2 Έλεγχος δεδομένων – Σύγκριση με τα δεδομένα της ΕΜΥ

5η Φάση (23/1/13 – 10/2/13).

4.1 Σχεδίαση βασικής βιβλιοθήκης σταθμού

4.2 Σχεδίαση απαιτούμενων συναρτήσεων.

6^η Φάση (11/2/13 – 20/2/13)

Διακοπή εργασιών

7^η Φάση (21/2/13 – 5/3/13)

7.1 Σχεδίαση Αλγορίθμου Ζαμπρέττι

7.2 Υλοποίηση και έλεγχος ορθής του λειτουργίας.

8^η Φάση (6/3/13 – 17/3/13)

8.1 Σχεδίαση Γραφικού Περιβάλλοντος Διαδικτύου.

8.2 Υλοποίηση γραφικού περιβάλλοντος.

8.3 Έλεγχος σφαλμάτων



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

9η Φάση (18/03/13 – 30/03/13)

9.1 Σύνδεση στο διαδίκτυο

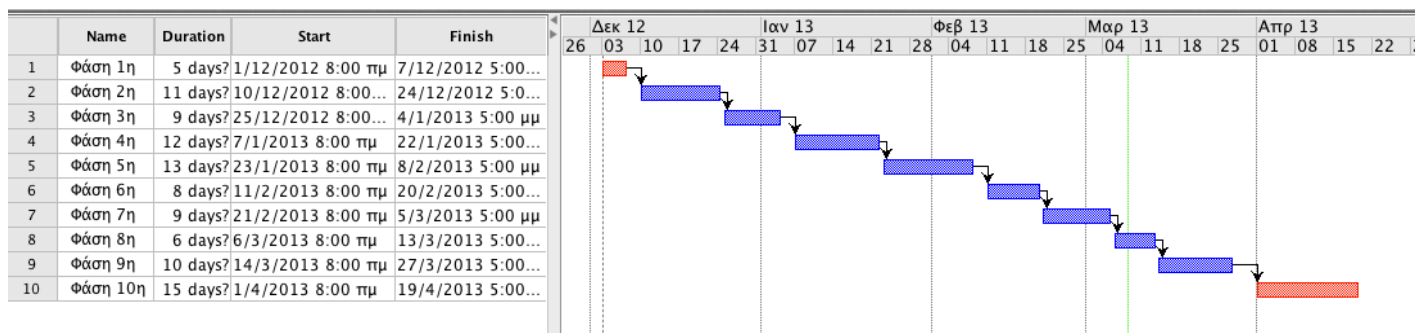
9.2 Έλεγχος και δοκιμές σε πραγματικό περιβάλλον

10^η Φάση(1/4/13-19/5/13)

10.1 Διορθώσεις σφαλμάτων

10.2 Βελτιώσεις

10.3 Σύγκριση δεδομένων και αποτελεσμάτων της πρόγνωσης καιρού.



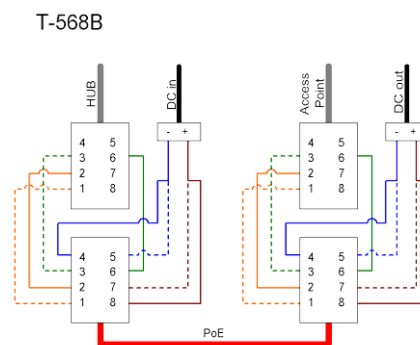
Σχήμα 18 Χρονοδιάγραμμα εργασίας



4. Ανάλυση και σχεδίαση του σταθμού

4.1 Περιγραφή Αρχιτεκτονικής Μετεωρολογικού Σταθμού

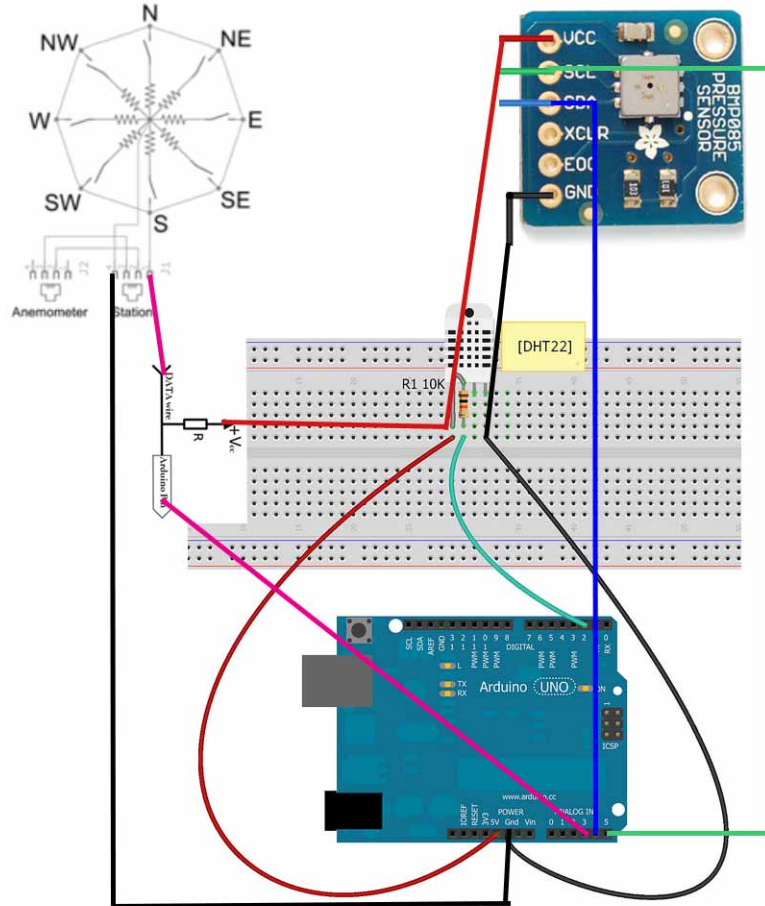
Ο σταθμός αποτελείται από τρεις αισθητήρες (θερμοκρασίας, βαρομετρικής πίεσης, κατεύθυνσης ανέμου). Ο αισθητήρας κατεύθυνσης ανέμου πρέπει να τοποθετηθεί σε θέση που να μην υπάρχει ορατό εμπόδιο σε απόσταση 100m. Ο αισθητήρας θερμοκρασίας πρέπει να βρίσκεται σε ύψος 1,5 μέτρο από το έδαφος. Σε αντίθετη περίπτωση πρέπει να υπολογιστεί η διαφορά. Επίσης θα πρέπει να μην υπάρχει σε θέση που είναι εκτεθειμένος απευθείας στις ακτίνες του ηλίου. Στον ίδιο κλωβό θα βρίσκεται και ο αισθητήρας βαρομετρικής πίεσης. Επειδή ο σταθμός βρίσκεται σε αρκετή απόσταση από την πλησιέστερη παροχή ρεύματος, για την αποφυγή της εγκατάστασης παροχής ηλεκτρικού ρεύματος, η τροφοδοσία του γίνεται εκμεταλλευόμενοι την τεχνολογία PoE (Power over Ethernet) [28] η οποία επιτρέπει την μεταφορά ρεύματος για μικρές τάσεις όπως αυτή του Arduino (σχήμα 19). Τέλος η συνδεσμολογία του σταθμού με τους αισθητήρες εμφανίζεται στο σχήμα 20.



Σχήμα 19 Συνδεσμολογία καλωδίου UTP για μεταφορά ηλεκτρικού φορτίου (PoE)



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων



Σχήμα 20 Συνδεσμολογία μεταξύ πλακέτας Arduino και αισθητήρων

4.2 Σχεδιασμός Λογισμικού Σταθμού.

Ο σταθμός κατά την εκκίνηση του θα ελέγχει ότι οι αισθητήρες είναι συνδεδεμένοι και ότι η επικοινωνία μεταξύ τους θα είναι ορθή. Στην συνέχεια θα γίνεται σύνδεση με το δίκτυο, φορτώνοντας τις αντίστοιχες βιβλιοθήκες .Ο σταθμός θα πρέπει να είναι όσο το δυνατόν ακριβής με τις μετρήσεις του, κατά συνέπεια κρίνεται σκόπιμο στην επόμενη φάση να γίνεται σύνδεση με εξυπηρετητή NTP από τον οποίο θα συγχρονίζεται και θα λαμβάνει την τρέχουσα ώρα [29]. Σε κάθε περίπτωση



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

σφάλματος θα πρέπει να ενημερώνεται ο χρήστης στην ιστοσελίδα του σταθμού καθώς και στην οθόνη LCD.

Ο σταθμός πλέον βρίσκεται σε πλήρη λειτουργία και εκτελεί τις ακόλουθες ενέργειες οι οποίες είναι κοινές για όλους τους αισθητήρες:

α) Κάθε 12 δευτερόλεπτα και συγκεκριμένα στα 12'', 24'', 36'', 48'' και 00'' του επομένου λεπτού γίνεται σύνδεση με τους αισθητήρες από όπου αντλεί τα δεδομένα και τα αποθηκεύει στην προσωρινή μνήμη.

β) Μόλις φτάσει στο δευτερόλεπτο 00'' (του επόμενου λεπτού) έχουν ήδη συλλεχθεί 5 δείγματα, οπότε υπολογίζουμε τον μέσο όρο αυτών, το αποτέλεσμα του οποίου αποτελεί την τιμή του μετεωρολογικού φαινομένου για το προηγούμενο λεπτό. Στην συνέχεια αποθηκεύεται η τιμή αυτή στην προσωρινή μνήμη του σταθμού καθώς και στην κάρτα SD.

γ) Κάθε ώρα και συγκεκριμένα από το 50' έως 58' λεπτό της ώρας, καθορίζεται ο χρόνος στον οποίο θα γίνει ο υπολογισμός της μέσης ωριαίας τιμής με βάση τις τελευταίες 60 μετρήσεις (μία για κάθε λεπτό).

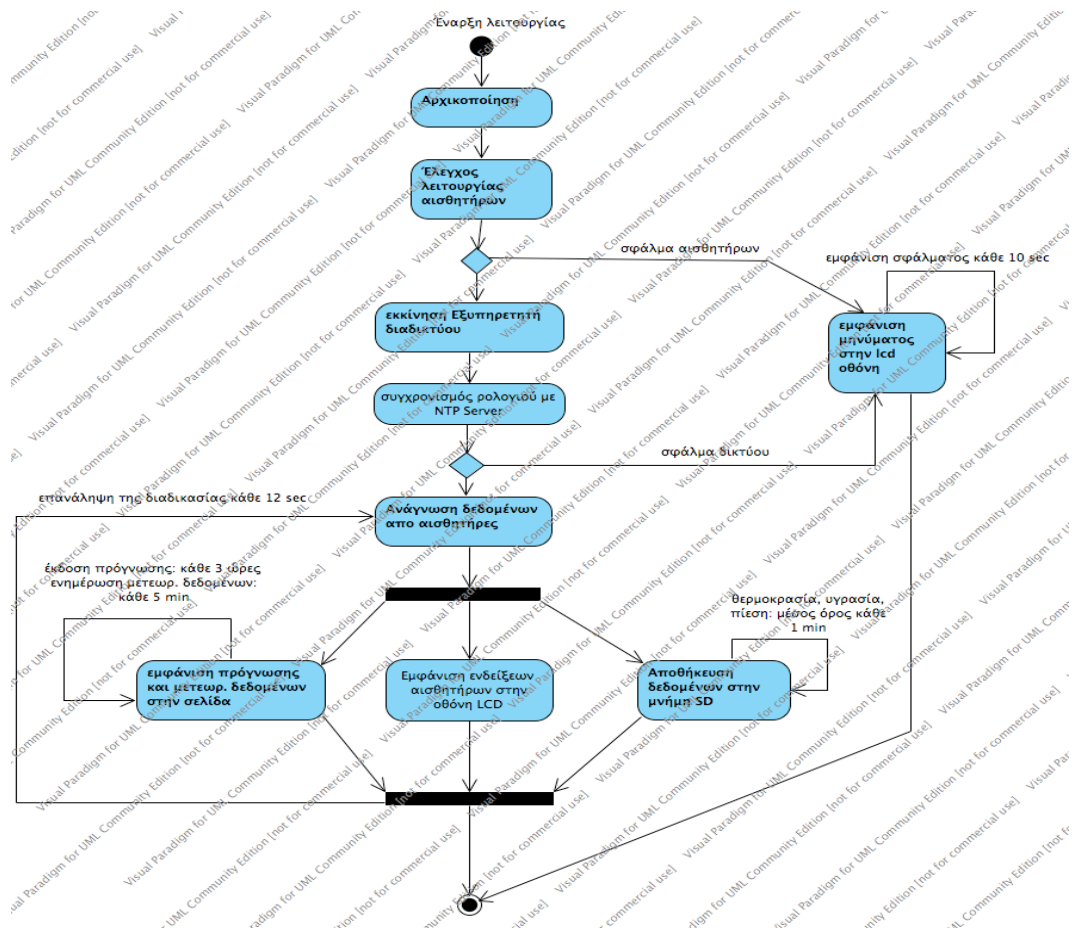
δ) Με βάση το γ) γίνεται ο υπολογισμός της βαρομετρικής τάσης της οποίας η διαδικασία έχει αναφερθεί στο Κεφάλαιο 2.

ε) Κάθε πέντε δευτερόλεπτα η ιστοσελίδα του AMΣ ανανεώνεται, ενημερώνοντας έτσι τον χρήστη για τις τιμές των μετεωρολογικών φαινομένων που έχουν καταγραφεί.

Από τα παραπάνω προκύπτει το διάγραμμα δραστηριότητας του σταθμού το οποίο θα είναι και ο οδηγός μας για την κατασκευή της συνάρτησης **loop()**.



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων



Σχήμα 21 Διάγραμμα δραστηριότητας του Αυτόνομου Μετεωρολογικού Σταθμού
4.3 Διάγραμμα περίπτωσης χρήσης (Use Case Diagram)

Τίτλος Περίπτωσης χρήσης: loop()

Κύριος Χειριστής: αισθητήρας (όλοι οι αισθητήρες)

Περιγραφή: Οι αισθητήρες εισάγουν τα δεδομένα που απαιτούνται για την καταγραφή

Προϋποθέσεις:

Οι αισθητήρες να είναι συνδεδεμένοι και να λειτουργούν



Μετασυνθήκες:

Δεν υπάρχουν

Βασική ροή:

1. Με την έναρξη γίνεται συγχρονισμός με τον NTP server.
2. Οι αισθητήρες αποστέλλουν δεδομένα στο Arduino
3. Κάθε λεπτό αποθηκεύεται στην κάρτα SD οι μετρήσεις.
4. Σε προκαθορισμένα χρονικά διαστήματα επαναλαμβάνεται η διαδικασία (βήμα 2-4).

Εναλλακτική ροή 1

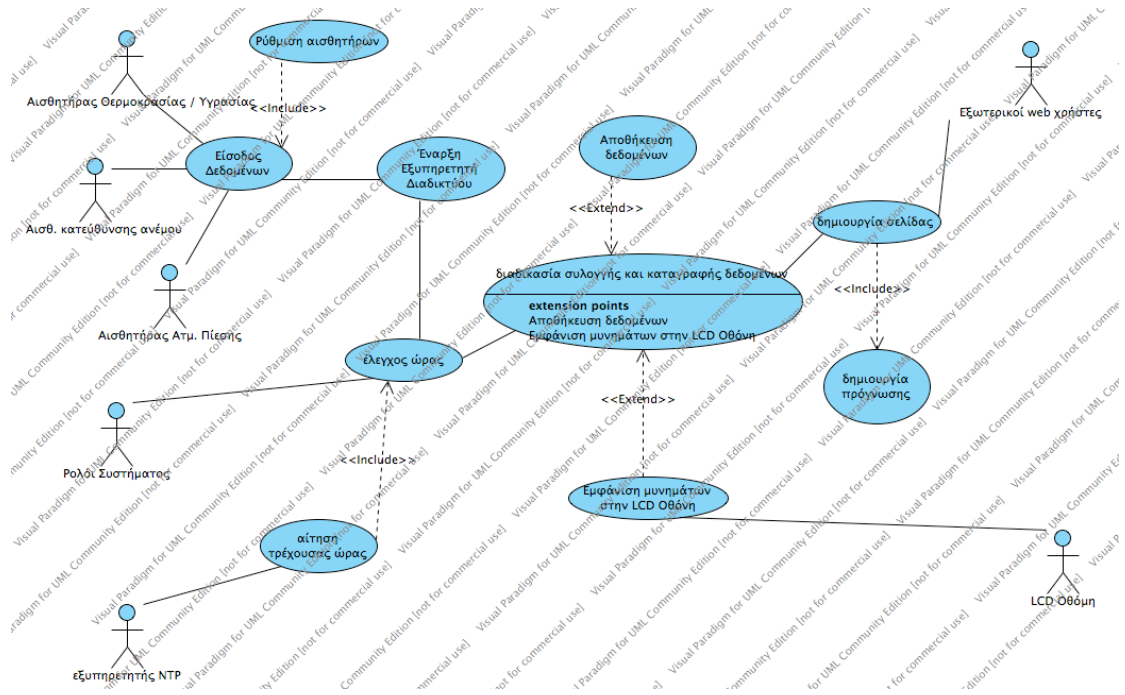
- 2.α.1 κάποιιοι αισθητήρες δεν αποστέλλουν δεδομένα
- 2.α.2 Στην οθόνη LCD και στην ιστοσελίδα εμφανίζεται μήνυμα λάθους.
- 2.α.3 Επανάληψη των βημάτων 1 και 2

Εναλλακτική ροή 2

- 2.β.1 Ο σταθμός δεν μπορεί να συνδεθεί στο δίκτυο.
- 2.β.2 Εμφάνιση σχετικού μηνύματος στην οθόνη LCD
- 2.β.3 Επανάληψη των βημάτων 1 και 2 έως την επιδιόρθωση του σφάλματος



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων



Σχήμα 22 περίπτωσης χρήσης βασικής συνάρτησης loop()

4.4 Δομή WeatherData.

Η δομή WeatherData αποτελεί τον ενδιάμεσο κρίκο μεταξύ της βιβλιοθήκης WeatherStation και της συνάρτησης **loop()**. Η δομή περιέχει συγκεντρωτικά όλες τις πληροφορίες που απαιτούνται για την διαχείριση των μετεωρολογικών δεδομένων από την εφαρμογή.

Το διάγραμμα της δομής παρουσιάζεται στο σχήμα 23



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

weatherData
-prognosis : char
-temperature : float
-humidity : float
-pressure : int
-windDirectionDegrees : int
-heatIndex : float
-dewPoint : float
-maxTemp : float
-minTemp : float
-maxHumi : float
-minHumi : float
-maxPres : int
-minPres : int
-trend : int
-prognosisCode : int

Σχήμα 23 Διάγραμμα δομής weatherData

Αναλυτικά τα στοιχεία της δομής είναι:

Πίνακας 7 Αναλυτικά στοιχεία δομής weatherData

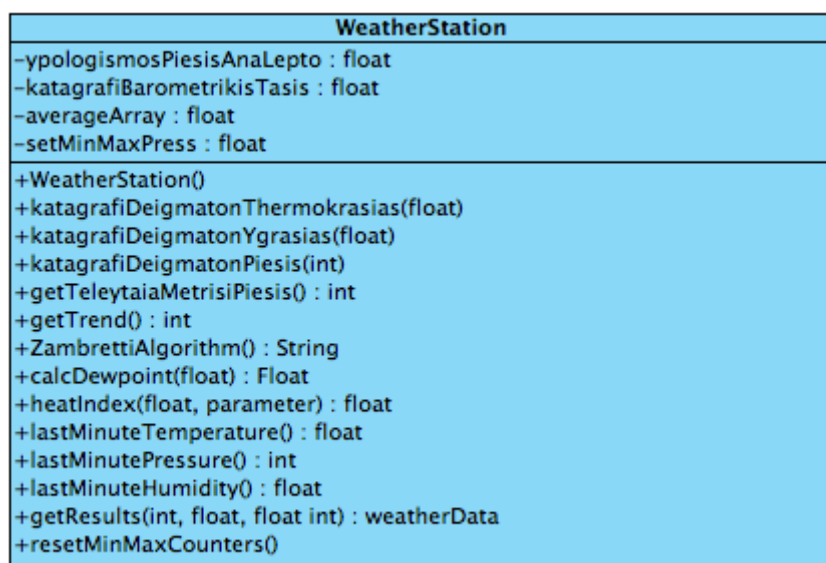
όνομα	τύπος	περιγραφή
prognosis	char	Αποθηκεύεται το κείμενο της πρόγνωσης καιρού
temperature	Float	Αποθηκεύεται η τιμή της θερμοκρασίας
himidity	Float	Αποθηκεύεται η τιμή της υγρασίας
pressure	Int	Αποθηκεύεται η τιμή της ατμοσφαιρικής πίεσης
windDirectionDegrees	Int	Αποθηκεύεται η τιμή της διεύθυνσης ανέμου σε μοίρες
heatIndex	Float	Αποθηκεύεται η τιμή του δείκτη δυσφορίας
dewPoint	Float	Αποθηκεύεται η τιμή του σημείου δρόσου
maxTemp	Float	Αποθηκεύεται η τιμή της μέγιστης θερμοκρασίας
minTemp	Float	Αποθηκεύεται η τιμή της ελάχιστης θερμοκρασίας
maxHumi	Float	Αποθηκεύεται η τιμή της μέγιστης υγρασίας
minHumi	Float	Αποθηκεύεται η τιμή της ελάχιστης υγρασίας
maxPres	Float	Αποθηκεύεται η ημερήσια μέγιστη τιμή της ατμ. πίεσης
minPres	Float	Αποθηκεύεται η ημερήσια ελάχιστη τιμή της ατμ. πίεσης
trend	int	Αποθηκεύεται η βαρομετρική τάση με την εξής κωδικοποίηση. <ul style="list-style-type: none">• 0 είναι σταθερή



		<ul style="list-style-type: none">• 1 είναι αυξητική• 2 είναι μειούμενη• -1 Όταν δεν μπορεί να υπολογιστεί η βαρομετρική τάση
prognosisCode	int	Αποθηκεύεται ο κωδικός της πρόγνωσης καιρού.

4.5 Βιβλιοθήκη WeatherStation.

Η βιβλιοθήκη WeatherStation περιέχει όλες εκείνες τις συναρτήσεις, τις οποίες χρειάζεται ο σταθμός για τον υπολογισμό της μέσης τιμής υγρασίας, θερμοκρασίας, ατμοσφαιρικής πίεσης, κατεύθυνσης ανέμου (σε μοίρες). Επίσης περιέχει τον αλγόριθμο Ζαμπρέτι καθώς και την συνάρτηση υπολογισμού της βαρομετρικής τάσης. Το διάγραμμα κλάσεως της εν λόγω βιβλιοθήκης παρουσιάζεται στο σχήμα 24.



Σχήμα 24 Διάγραμμα κλάσης WeatherStation



Η κλάση περιέχει τις ακόλουθες μεθόδους / ιδιότητες:

Πίνακας 8 αναλυτική περιγραφή της κλάσης WeatherStation

Όνομα	Ιδιότητα	Περιγραφή
WeatherStation	Μέθοδος	Είναι ο κατασκευαστής της κλάσης. Δεν απαιτεί μεταβλητές.
katagrafiDeigmatonThermokrasias	Μέθοδος	Η μέθοδος αυτή δέχεται ως είσοδο την τιμή της τρέχουσας θερμοκρασίας και την τοποθετεί στην προσωρινή μνήμη για τον υπολογισμό της μέσης τιμής.
katagrafiDeigmatonYgrasias	Μέθοδος	Η μέθοδος αυτή δέχεται ως είσοδο την τιμή της τρέχουσας υγρασίας και την τοποθετεί στην προσωρινή μνήμη για τον υπολογισμό της μέσης τιμής.
katagrafiDeigmatonPiesis	Μέθοδος	Η μέθοδος αυτή δέχεται ως είσοδο την τιμή της τρέχουσας ατμοσφαιρικής πίεσης και την τοποθετεί στην προσωρινή μνήμη για τον υπολογισμό της μέσης τιμής.
getTrend	Μέθοδος	Όταν καλείται, επιστρέφει σε αριθμητική απεικόνιση την βαρομετρική τάση του τελευταίου τριώρου. Όταν επιστρέφει 0 τότε δεν υπάρχει μεταβολή, όταν επιστρέφει 1 τότε υπάρχει αύξηση, ενώ όταν επιστρέφει 2 υπάρχει πτώση.
katagrafiBarometrikisTasis	Μέθοδος	Δέχεται ως είσοδο τον μέσο όρο των 60 τελευταίων μετρήσεων και τον αποθηκεύει για τον υπολογισμό της βαρομετρικής τάσης
ZambrettiAlgorithm	Μέθοδος	Δέχεται ως ορίσματα την ελάχιστη και μέγιστη τιμή της ατμοσφαιρικής πίεσης για την περιοχή που θα εγκατασταθεί ο σταθμός, την βαρομετρική τάση (με την κωδικοποιημένη μορφή που



Τριανταφύλλου Γεώργιος,
'Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

calcDewpoint	Μέθοδος	επιστρέφει η συνάρτηση getTrend), η κατεύθυνση του ανέμου σε μοίρες, και η τρέχουσα ατμοσφαιρική πίεση. Δέχεται ως όρισμα την τρέχουσα θερμοκρασία και υγρασία και επιστρέφει την θερμοκρασία του σημείου δρόσου.
heatIndex	Μέθοδος	Δέχεται ως όρισμα την τρέχουσα θερμοκρασία και υγρασία και επιστρέφει τον δείκτη δυσφορίας
lastMiniuteTemperature	Μέθοδος	Επιστρέφει τον μέσο όρο των τελευταίων 5 μετρήσεων της θερμοκρασίας.
lastMiniutePressure	Μέθοδος	Επιστρέφει τον μέσο όρο των τελευταίων 5 μετρήσεων της ατμοσφαιρικής πίεσης.
lastMiniuteHumidity	Μέθοδος	Επιστρέφει τον μέσο όρο των τελευταίων 5 μετρήσεων της υγρασίας.
getResult	Μέθοδος	Δέχεται ως ορίσματα την τρέχουσα θερμοκρασία, υγρασία, ατμ. πίεση, διεύθυνση ανέμου και επιστρέφει τη δομή weatherData συμπληρωμένη.
resetMinMaxCounters	Μέθοδος	Καθημερινά στις 00:00 διαγράφει τα δεδομένα των προσωρινών μνημών, για τον υπολογισμό νέων.



5. Σύνδεση Περιφερειακών Συσκευών – Αισθητήρων.

5.1 Κάρτα επέκτασης δικτύου (EthernetShield)

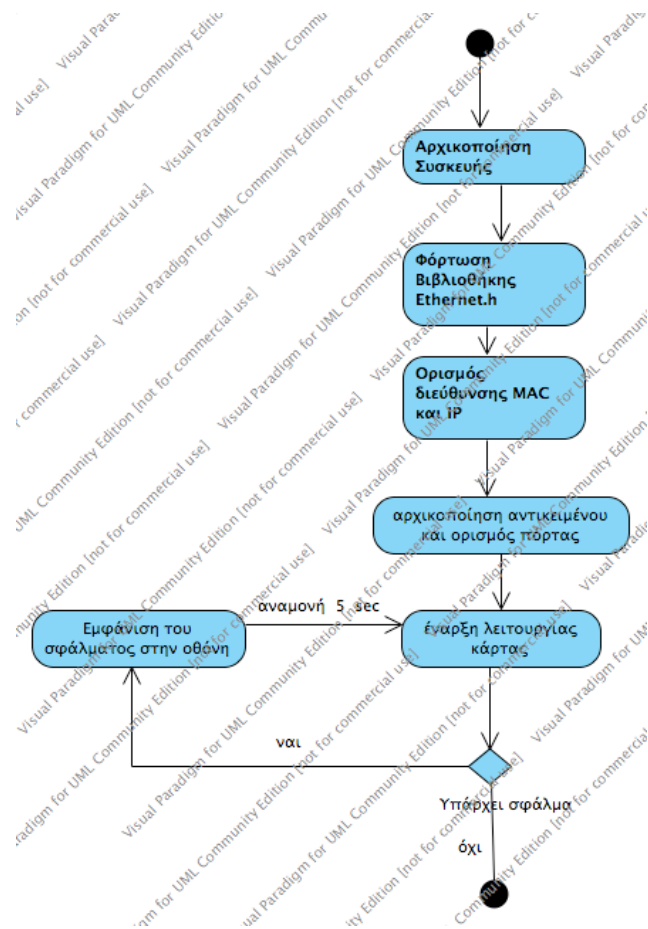
Κύρια λειτουργία της κάρτας Ethernet είναι η διασύνδεση της πλακέτας Arduino με δίκτυο με σκοπό την μετάδοση πληροφοριών. Στην παρούσα εργασία ο σταθμός έχει τοποθετηθεί σε ενσύρματο δίκτυο τύπου Ethernet (οικιακό δίκτυο), από όπου γίνεται η πρόσβαση από όλους τους σταθμούς εργασίας του τοπικού δικτύου. Χρησιμοποιείται το πρωτόκολλο TCP/IP καθώς στις απαιτήσεις της εργασίας προαπαιτούμενος τρόπος διασύνδεσης με τον σταθμό είναι μέσω φυλλομετρητή ιστοσελίδων.

Κατά την εκκίνηση του σταθμού γίνεται αρχικοποίηση των ρυθμίσεων της κάρτας δικτύου. Αρχικά εισάγουμε την βιβλιοθήκη **Ethernet.h** [30] η οποία βρίσκεται προεγκατεστημένη στο IDE του Arduino και περιέχει όλες τις συναρτήσεις για την διασύνδεση της κάρτας με το Arduino. Στην συνέχεια ορίζουμε την διεύθυνση IP του δικτύου, το υποδίκτυο στο οποίο ανήκει ο σταθμός, την διεύθυνση MAC της κάρτας, αρχικοποιούμε (instantiate) το αντικείμενο και ορίζουμε την πόρτα στην οποία θα ανταποκρίνεται ο εξυπηρετητής. Εξ’ ορισμού η πόρτα στην οποία ανταποκρίνονται όλες οι αιτήσεις του HTTP πρωτοκόλλου είναι η 80. Στη συνάρτηση **setup()** γίνεται η αρχικοποίηση και έναρξη λειτουργίας της κάρτας. Σε περίπτωση που υπάρχει σφάλμα κατά την διαδικασία σύνδεσης της κάρτας, τότε ενημερώνεται ο χρήστης με σχετικό μήνυμα στην οθόνη LCD. Η ανωτέρω διαδικασία επαναλαμβάνεται έως ότου η επικοινωνία αποκατασταθεί.

Στο σχήμα 25 παρατίθεται το διάγραμμα ροής της λειτουργίας της κάρτας δικτύου Ethernetshield.



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων



Σχήμα 25 Διάγραμμα ροής λειτουργίας Ethernet Shield



Ο κώδικας με τον οποίο υλοποιείται η σύνδεση περιλαμβάνεται στο παράρτημα Α. Για να ελέγξουμε την ορθή λειτουργία του σταθμού εκτελούμε την εντολή ping από την γραμμή εντολών (ισχύει σε όλα τα λειτουργικά συστήματα, η δοκιμή έγινε σε mac OSX) και παρατηρούμε ότι ο σταθμός απαντά στο ερώτημα μας (Σχήμα 26) με επιτυχία.

```
MacBookPro:~ gtriandafillou$ ping -c 5 192.168.2.100
PING 192.168.2.100 (192.168.2.100): 56 data bytes
64 bytes from 192.168.2.100: icmp_seq=0 ttl=128 time=2.700 ms
64 bytes from 192.168.2.100: icmp_seq=1 ttl=128 time=4.256 ms
64 bytes from 192.168.2.100: icmp_seq=2 ttl=128 time=2.652 ms
64 bytes from 192.168.2.100: icmp_seq=3 ttl=128 time=4.108 ms
64 bytes from 192.168.2.100: icmp_seq=4 ttl=128 time=3.550 ms

--- 192.168.2.100 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 2.652/3.453/4.256/0.677 ms
MacBookPro:~ gtriandafillou$
```

Σχήμα 26 αποτελέσματα δοκιμών διασύνδεσης κάρτας Ethernet Shield

5.2 Αισθητήρας καταγραφής θερμοκρασίας – υγρασίας DHT-22.

Ο αισθητήρας DHT-22 είναι υπεύθυνος για την ανάγνωση της θερμοκρασίας και υγρασίας. Για την διασύνδεση του αισθητήρα με την πλακέτα Arduino γίνεται μέσω της βιβλιοθήκης DHTlib [31]. Η σύνδεση με το Arduino γίνεται μέσω ψηφιακής εισόδου στην οποία εισέρχεται τάση η οποία έχει σταθμιστεί ανάλογα με την τιμή θερμοκρασίας / υγρασίας του περιβάλλοντος. Η τάση αυτή μετατρέπεται σε τιμή η οποία αντιστοιχεί σε αυτή της θερμοκρασίας και της υγρασίας. Η βιβλιοθήκη αναλαμβάνει να μετατρέψει την εισερχόμενη τάση στις τιμές που αντιστοιχούν. Η κλάση όταν αρχικοποιείται θα πρέπει να δηλωθεί η αναλογική είσοδος στην οποία είναι συνδεδεμένη. Η κλάση περιέχει τρεις μεθόδους:



readData(): επιστρέφει την κατάσταση του αισθητήρα. Επιστρέφονται οι τιμές:

- DHT_ERROR_NONE: δεν υπάρχουν σφάλματα
- DHT_ERROR_CHECKSUM: σφάλμα κατά τον έλεγχο με το bit ισοτιμίας
- DHT_BUS_HUNG: σφάλμα διαύλου
- DHT_ERROR_NOT_PRESENT: δεν εντοπίστηκε ο αισθητήρας
- DHT_ERROR_ACK_TOO_LONG: το σήμα ACK έληξε
- DHT_ERROR_SYNC_TIMEOUT: σφάλμα κατά τον συγχρονισμό
- DHT_ERROR_DATA_TIMEOUT: σφάλμα δεδομένων
- DHT_ERROR_TOOQUICK: ζητήθηκε γρηγορότερα από τον αναμενόμενο χρόνο (<2sec) αίτημα για νέα δεδομένα

getHumidity(): επιστρέφει την τρέχουσα τιμή του αισθητήρα για την υγρασία σε

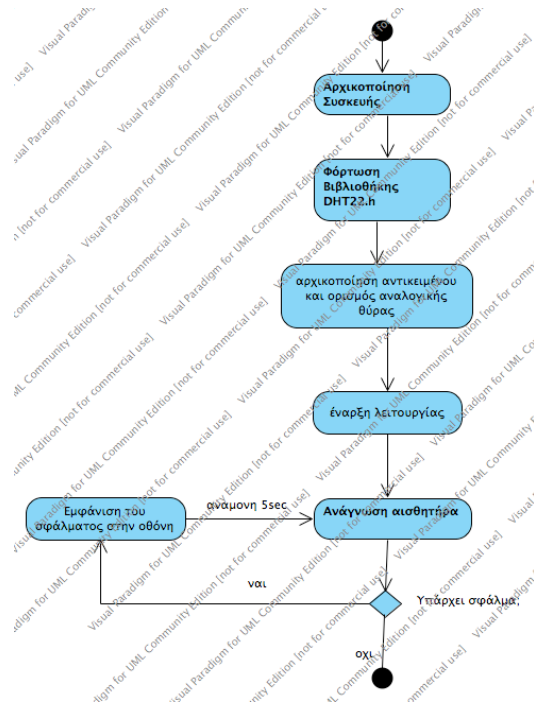
getTemperatureC() επιστρέφει την τρέχουσα τιμή της θερμοκρασίας σε βαθμούς κελσίου.

Αρχικά εισάγουμε την βιβλιοθήκη <DHT22.h> στο πρότυπο. Στην συνέχεια αρχικοποιούμε (instantiate) το αντικείμενο ορίζοντας την αναλογική είσοδο στην οποία θα συνδεθεί ο αισθητήρας. Τέλος, ελέγχουμε την κατάσταση του αισθητήρα έτσι ώστε σε περίπτωση λάθους να ενημερώνει τον χρήστη και γίνεται έλεγχος λειτουργίας του αισθητήρα κάθε 5 δευτερόλεπτα.

Στο ακόλουθο διάγραμμα ροής βλέπουμε την διαδικασία που θα ακολουθήσουμε για την σύνδεση και τον έλεγχο του αισθητήρα.



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων



Σχήμα 27 Διάγραμμα ροής διασύνδεσης με τον αισθητήρα DHT-22

Ο κώδικας με τον οποίο γίνεται η σύνδεση με την πλακέτα Arduino, παρουσιάζεται στο παράρτημα Β.

Για να ελέγξουμε την ορθή λειτουργία του αισθητήρα ανοίγουμε το παράθυρο σειριακής θύρας και παρατηρούμε ότι ο σταθμός αποστέλλει την τρέχουσα θερμοκρασία και υγρασία (Σχήμα 28).



Τριανταφύλλου Γεώργιος,
'Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

```
Integer-only reading: Temperature 23.7 C, Humidity 53.3 % RH  
Requesting data...Got Data 23.70C 53.30%  
Integer-only reading: Temperature 23.7 C, Humidity 53.3 % RH  
Requesting data...Got Data 23.70C 53.40%  
Integer-only reading: Temperature 23.7 C, Humidity 53.4 % RH  
Requesting data...Got Data 23.70C 53.40%  
Integer-only reading: Temperature 23.7 C, Humidity 53.4 % RH  
Requesting data...Got Data 23.60C 53.30%  
Integer-only reading: Temperature 23.6 C, Humidity 53.3 % RH  
Requesting data...Got Data 23.60C 53.40%  
Integer-only reading: Temperature 23.6 C, Humidity 53.4 % RH  
Requesting data...Got Data 23.60C 53.40%  
Integer-only reading: Temperature 23.6 C, Humidity 53.4 % RH  
Requesting data...Got Data 23.60C 53.40%  
Integer-only reading: Temperature 23.6 C, Humidity 53.4 % RH  
Requesting data...Got Data 23.60C 53.40%
```

Σχήμα 28 Αποτελέσματα ελέγχου αισθητήρα θερμοκρασίας / υγρασίας

5.3 Σύνδεση αισθητήρα ατμοσφαιρικής πίεσης BMP-085

Ο αισθητήρας BMP-085 είναι υπεύθυνος για την λήψη της ατμοσφαιρικής πίεσης και της θερμοκρασίας. Για την διασύνδεση με το Arduino θα γίνει χρήση της βιβλιοθήκης Adafruit_BMP085.h [32] την οποία έχει αναπτύξει η εταιρία Adafruit και είναι ανοιχτού κώδικα. Η κλάση περιέχει τις μεθόδους:

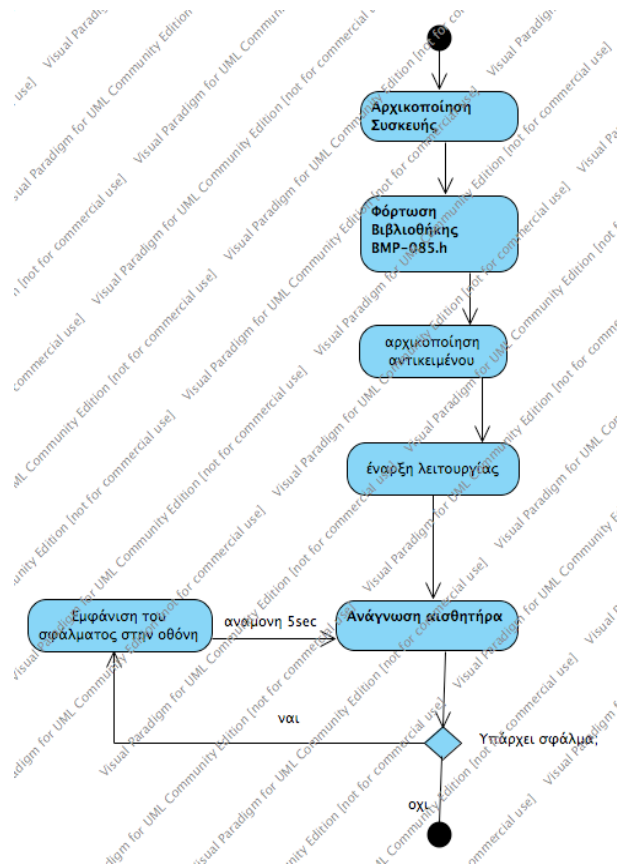
- **begin()**: γίνεται η έναρξη της ανάγνωσης από τον αισθητήρα.
- **readTemperature()**: επιστρέφει σε float την τιμή της τρέχουσας θερμοκρασίας.
- **readPressure()**: επιστρέφει σε float την τιμή της τρέχουσας ατμοσφαιρικής πίεσης.
- **readAltitude(int)**: δέχεται ως όρισμα την τιμή της ατμοσφαιρικής πίεσης και επιστρέφει το υψόμετρο.
- **seaLevelPressure(υψος, πίεση, ατμόσφαιρα)**: δέχεται ως ορίσματα την τιμή της ατμοσφαιρικής πίεσης σε (hPa), του ύψους (σε μέτρα) και της



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

θερμοκρασίας (σε βαθμούς Celsius) και επιστρέφει την τιμή της ατμοσφαιρικής πίεσης της επιφάνειας της θάλασσας.

Για να χρησιμοποιήσουμε την βιβλιοθήκη αρκεί η αναφορά της στην αρχικοποίηση του προγράμματος. Στην συνάρτηση **setup()** γίνεται η σύνδεση του αισθητήρα με το Arduino. Στην περίπτωση σφάλματος ενημερώνεται ο χρήστης με σχετικό μήνυμα και κάθε 5 δευτερόλεπτα γίνεται νέα προσπάθεια σύνδεσης με τον αισθητήρα έως ότου η σύνδεση ολοκληρωθεί. Όπως και στους προηγούμενους αισθητήρες έτσι και εδώ τονίζεται πως ο έλεγχος γίνεται κατά την εκκίνηση και όχι κατά την λειτουργία των αισθητήρων όπου εκεί υπάρχει άλλη διαδικασία.



Σχήμα 28 Διάγραμμα ροής αισθητήρα ατμοσφαιρικής πίεσης



Ο κώδικας με τον οποίο γίνεται η σύνδεση παρουσιάζεται στο παράρτημα Γ.

Στο Σχήμα 29 βλέπουμε την έξοδο του αισθητήρα κατά την διάρκεια των δοκιμών. Θα πρέπει να λάβουμε υπόψη ότι η έξοδος της βιβλιοθήκης επιστρέφει την τιμή της ατμοσφαιρικής πίεσης σε Pa οπότε θα πρέπει να γίνει μετατροπή της σε hPa. Η μετατροπή εκφράζεται ως:

$$hPa = Pa \times 0,01 \quad (8)$$

Temperature = 22.70 *C
Pressure = 101365 Pa

Temperature = 22.71 *C
Pressure = 101368 Pa

Temperature = 22.73 *C
Pressure = 101369 Pa

Temperature = 22.71 *C
Pressure = 101364 Pa

Temperature = 22.69 *C
Pressure = 101371 Pa

Σχήμα 29 Έλεγχος λειτουργίας του αισθητήρα ατμοσφαιρικής πίεσης

5.4 Αισθητήρας Διεύθυνσης Ανέμου

Για την διασύνδεση του αισθητήρα διεύθυνσης ανέμου χρειάζεται να αναπτυχθεί η συνάρτηση ανάγνωσης των σημάτων του αισθητήρα καθώς δεν έχει υπάρξει σχετική βιβλιοθήκη. Σύμφωνα με το εγχειρίδιο του αισθητήρα [21] στην έξοδο παρουσιάζεται μια μοναδική μεταβαλλόμενη τάση η τιμή της οποίας εξαρτάται από την κατεύθυνση



του ανέμου. Έτσι για παράδειγμα όταν ο δείκτης είναι προσανατολισμένος στον Βορρά, η έξοδος είναι 3,84 V.

Για την διαχείριση των αναλογικών σημάτων, το Arduino διαθέτει ενσωματωμένο έναν μετατροπέα σήματος από αναλογικό σε ψηφιακό (Analog to Digital Converter) με ακρίβεια δέκα ψηφίων (10 bit) [22] άρα το αναλογικό σήμα δειγματίζεται σε $2^{10}=1024$ βαθμίδες.

Η συνάρτηση που θα κατασκευαστεί, θα διαβάζει το εισερχόμενο σήμα από την αναλογική είσοδο και στην συνέχεια ανάλογα με την τιμή που θα παρουσιάζει, θα επιστρέφει το ανάλογο μήνυμα. Κατά συνέπεια θα πρέπει να εντοπίσουμε τις διακριτές τιμές στις οποίες μετατρέπεται το εισερχόμενο αναλογικό σήμα.

Στον Πίνακα 9 φαίνεται η αντιστοιχία του αναλογικού σήματος συναρτήσει της κατεύθυνσης του αισθητήρα.

Πίνακας 9 Αντιστοιχίες μεταξύ της διεύθυνσης ανέμου και του αναλογικού σήματος.

Direction (Degrees)	Resistance (Ohms)	Voltage (V=5v, R=10k)
0	33k	3.84v
22.5	6.57k	1.98v
45	8.2k	2.25v
67.5	891	0.41v
90	1k	0.45v
112.5	688	0.32v
135	2.2k	0.90v
157.5	1.41k	0.62v
180	3.9k	1.40v
202.5	3.14k	1.19v
225	16k	3.08v
247.5	14.12k	2.93v
270	120k	4.62v
292.5	42.12k	4.04v
315	64.9k	4.78v
337.5	21.88k	3.43v

Στην συνέχεια από την σχέση (9) υπολογίζουμε την τιμή την οποία επιστρέφει η συνάρτηση **analogRead()** αναλόγως του εισερχόμενου αναλογικού σήματος [23].



$$adc = \left\lfloor \left(\frac{VIN}{VREF} \right) * 1023 \right\rfloor \quad (9)$$

όπου:

- adc η διακριτή τιμή του εισερχόμενου σήματος.
- VIN η τιμή του εισερχόμενου σήματος.
- $VREF$ η μέγιστη (θεωρητική) τιμή του μετατροπέα. Στο Arduino είναι 5 (Volt).

Για παράδειγμα η διακριτή τιμή του εισερχόμενου σήματος που αντιστοιχεί στον Βορρά είναι:

$$adc = \left\lfloor \left(\frac{VIN}{VREF} \right) * 1023 \right\rfloor \Leftrightarrow adc = \left\lfloor \frac{3,84}{5} * 1023 \right\rfloor \Leftrightarrow adc = [785,664]$$

$$\Leftrightarrow adc = 786$$

Άρα η τιμή της **readAnalog()** θα πρέπει να ισούται με 786.

Το παραπάνω όμως δεν ανταποκρίνεται πλήρως στην πραγματικότητα γιατί η τιμή $VREF$ δεν είναι |5 βολτ| , αλλά **τείνει** προς τα 5 βολτ, η τιμή της οποίας εξαρτάται από εξωγενής κυρίως παράγοντες όπως είναι η τάση του τροφοδοτικού.

Για την επαλήθευση της ανωτέρω διαπίστωσης, προστέθηκε η συνάρτηση **readVcc()** [24] η οποία επιστρέφει την τιμή της τάση τροφοδοσίας του Arduino. Η συνάρτηση επέστρεφε τις τιμές 4,935 ή 4,954 εναλλάξ κάθε 2 δευτερόλεπτα. Αυτό συνέβαινε διότι το τροφοδοτικό δεν απέδιδε απόλυτα σταθερή τάση, αλλά παρουσίαζε αυτή την μικρή απόκλιση και μάλιστα εναλλασσόμενα. Κατά συνέπεια κατά τον υπολογισμό της διακριτής τιμής στην 1η περίπτωση θα έχουμε διακριτή τιμή $adc = 796$ και στην 2η περίπτωση $adc = 793$. Για τον λόγο αυτό, η διακριτή τιμή adc δεν μπορεί είναι απόλυτη ως προς την αντιστοιχία της αλλά θα πρέπει να καλύπτει ένα εύρος που να ανταποκρίνεται στις παραπάνω παρατηρήσεις. Έτσι υπολογίστηκαν αρχικά οι διακριτές τιμές που θα απέδιδε ο μετατροπέας ADC σύμφωνα με την τιμή αναφοράς,



Τριανταφύλλου Γεώργιος,
'Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

δηλαδή τα 5 volt και στην συνέχεια ορίστηκε ένα εύρος στην διακριτή τιμή ± 12 . Με αυτό τον τρόπο αν η τιμή adc βρίσκεται στο εύρος $774 > adc > 798$ τότε αυτή η τιμή αντιστοιχεί στον βορρά. Στον πίνακα που ακολουθεί αναγράφεται το εύρος των διακριτών τιμών σε σχέση με την θέση διεύθυνσης ανέμου.

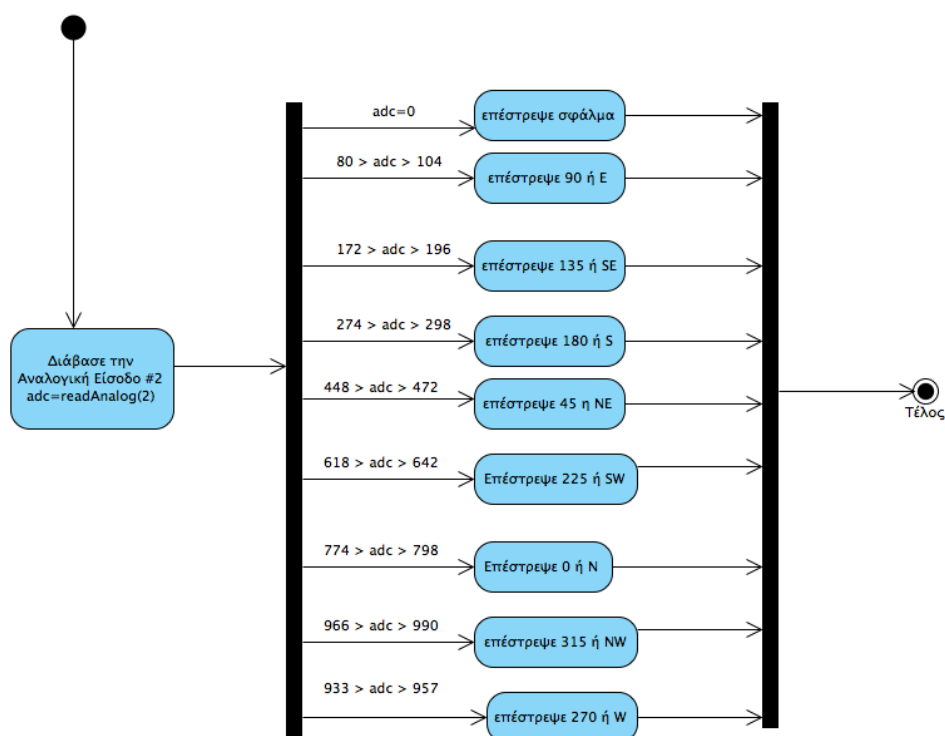
Πίνακας 10 εύρος διακριτών τιμών σε σχέση με την κατεύθυνση του δείκτη

Θέση διεύθυνσης ανέμου	Εύρος ADC
0	$774 > adc > 798$
45	$448 > adc > 472$
90	$80 > adc > 104$
135	$172 > adc > 196$
180	$274 > adc > 298$
225	$618 > adc > 642$
270	$933 > adc > 957$
315	$966 > adc > 990$

Σε περίπτωση που η τάση τροφοδοσίας του Arduino είναι μικρότερη των 4,932 volt και μεγαλύτερη των 5,074 volt, τότε ο ανωτέρω πίνακας παύει να ισχύει και θα πρέπει να υπολογιστεί εκ νέου.

Με βάση τα παραπάνω θα σχεδιαστεί η συνάρτηση διασύνδεσης του Arduino με τον αισθητήρα. Αρχικά θα πρέπει να οριστεί στις προκαθορισμένες τιμές (define) η αναλογική θύρα που θα χρησιμοποιηθεί για την επικοινωνία με τον σταθμό.

Στην περίπτωση που γίνει διακοπή της επικοινωνίας η τιμή $adc = 0$, άρα όταν ισχύει τότε θα επιστρέφεται η τιμή error disconnected. Στο Σχήμα 30 βλέπουμε το διάγραμμα ροής του αλγορίθμου ανάγνωσης της κατεύθυνσης του ανέμου.



Σχήμα 30 Διάγραμμα ροής συνάρτησης ανάγνωσης κατεύθυνσης ανέμου

Στο παράρτημα Δ παρουσιάζεται η διαδικασία διασύνδεσης του αισθητήρα με την πλακέτα Arduino.

Στο Σχήμα 31 βλέπουμε την έξοδο του Arduino κατά την διάρκεια των δοκιμών για τον έλεγχο ανάγνωσης του αισθητήρα. Επισημαίνεται ότι οι 2 τελευταίες ενδείξεις αναφέρουν σφάλμα ανάγνωσης σκοπίμως, καθώς αποσυνδέθηκε ο αισθητήρας για να ελεγχθεί την αντίδραση του συστήματος.



Τριανταφύλλου Γεώργιος,
'Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

NW
W
S
SW
SE
E
NE
NE
N
NW
NW
NW
NW
sensor disconnected
sensor disconnected

Σχήμα 31 Έξοδος αισθητήρα κατεύθυνσης ανέμου



6. Υλοποίηση

6.1 Πλατφόρμες και προγραμματιστικά εργαλεία

6.1.1 Πρόγραμμα και αξιοποιήσιμες βιβλιοθήκες

Ο συνολικός κώδικας, γράφτηκε στο IDE Arduino 1.5.2 για τον προγραμματισμό της πλακέτας Arduino. Αξιοποιήθηκαν βιβλιοθήκες που έχουν δημιουργηθεί από τους κατασκευαστές των αισθητήρων και υπάγονται στην κατηγορία του “ανοιχτού κώδικα”. Για την αποθήκευση των καταγραφέντων δεδομένων χρειαζόμαστε μια κάρτα SD όπου θα αποθηκεύονται τα δεδομένα σε μορφή CSV.

6.1.2 Ιστοσελίδες και Javascript

Οι ιστοσελίδες που δημιουργήθηκαν, υλοποιήθηκαν με το εργαλείο Netbeans 7.2. Η παρουσίαση των γραφημάτων γίνεται με την χρήση javascript μέσω της δωρεάν βιβλιοθήκης digraph [33].

6.1.3 Επιπρόσθετες βιβλιοθήκες υποστήριξης έργου

Εκτός των βιβλιοθηκών που χρειάστηκαν για την υποστήριξη των αισθητήρων αξιοποιήθηκαν και άλλες οι οποίες είχαν μεν υποστηρικτικό έργο αλλά όμως κρίθηκε απαραίτητη η εκμετάλλευσή τους για την σωστή και εύρυθμη λειτουργία του σταθμού. Αυτές επιγραμματικά είναι:

ClockLibrary [34]: Πρόκειται για μια βιβλιοθήκη η οποία είναι υπεύθυνη για τον συγχρονισμό του σταθμού με εξυπηρετητή συγχρονισμού ρολογιού (NTP Server). Η ενέργεια αυτή είναι απαραίτητη καθώς το Arduino δεν μπορεί να διατηρήσει την ημερομηνία / ώρα, με αποτέλεσμα σε κάθε επανεκκίνηση του, το ρολόι να χάνει τις ρυθμίσεις του γεγονός το οποίο επηρεάζει την ποιότητα των δεδομένων που συλλέγονται. Στο σχήμα 32 βλέπουμε την έξοδο κατά τον συγχρονισμό του Arduino με τον εξυπηρετητή πρωτοκόλλου NTP.



```
-----  
Refreshing NTPTime  
NTPTime response received  
2013/04/15 09:34:06  
2013/04/15 09:34:07  
2013/04/15 09:34:08  
2013/04/15 09:34:09  
2013/04/15 09:34:10  
2013/04/15 09:34:11  
2013/04/15 09:34:12  
2013/04/15 09:34:13  
2013/04/15 09:34:14  
2013/04/15 09:34:15
```

Σχήμα 32 Έλεγχος διασύνδεσης με τον διακομιστή πρωτοκόλλου NTP

LiquidDisplay [35]: πρόκειται για την βιβλιοθήκη η οποία είναι υπεύθυνη για την προβολή δεδομένων στην οθόνη υγρών κρυστάλλων (Liquid Crystal Display – LCD) του σταθμού. Αν και τα δεδομένα παρουσιάζονται αναλυτικά στον ιστότοπο του σταθμού, κρίθηκε σκόπιμη η ενσωμάτωση της οθόνης για την άμεση προβολή πληροφοριών που αφορούν κυρίως την συντήρηση του. Τέτοιες πληροφορίες είναι η μέτρηση της τάσης του σταθμού, η IP διεύθυνση του, σε περιπτώσεις λάθους να εμφανίζει το σχετικό σφάλμα πχ: σε περίπτωση σφάλματος ενός αισθητήρα τότε εμφανίζεται, σχετικό μήνυμα στην οθόνη κλπ.

SD (Secure Disk) [36]: τα δεδομένα του σταθμού θα καταγράφονται σε αρχείο τύπου CSV έτσι ώστε να είναι αξιοποιήσιμα από εφαρμογές (όπως πχ το Microsoft Excel). Τα δεδομένα αυτά θα αποθηκεύονται σε κάρτα SD καθώς η κάρτα επέκτασης Ethernet Shield παρέχει αυτή την δυνατότητα. Αυτή η δυνατότητα κρίνεται σκόπιμο να εγκατασταθεί καθώς με αυτόν τον τρόπο αποφεύγεται η συχνή σύνδεση του σταθμού με τον εξυπηρετητή, με αποτέλεσμα τον περιορισμό του όγκου δεδομένων που διακινούνται από και προς τον σταθμό.



6.2 Ανάλυση αλγορίθμων

6.2.1 Υπολογισμός μέσης τιμής μετεωρολογικών δεδομένων (ανά λεπτό)

Γνωρίζουμε ότι απαιτούνται 5 δείγματα ανά λεπτό για την συλλογή των δειγμάτων. Ο μέσος όρος των δειγμάτων αυτών αποτελεί την μέτρηση για το συγκεκριμένο λεπτό που γίνεται η καταγραφή. Το αποτέλεσμα αποθηκεύεται στην κάρτα SD καθώς και στον πίνακα για τον υπολογισμό της ωριαίας τιμής για περαιτέρω επεξεργασία. Ο κώδικας που εκτελεί την διαδικασία είναι ίδιος για όλα τα μετεωρολογικά δεδομένα (θερμοκρασία, υγρασία, ατμοσφαιρική πίεση). Στο παράρτημα Ε περιλαμβάνεται ο κώδικας καταγραφής της ατμοσφαιρικής πίεσης.

6.2.2 Υπολογισμός μέσης τιμής μετεωρολογικών δεδομένων (ωριαία)

Για την εκτέλεση αυτής της συνάρτησης απαιτείται η πλήρη συμπλήρωση του πίνακα **mesiTimiPiesis** της προηγούμενης συνάρτησης. Για τον λόγο αυτό η συνάρτηση καλείται μόλις συμπληρωθεί και η πέμπτη θέση του πίνακα. Σύμφωνα με τον WMO ο υπολογισμός της ωριαίας μέσης τιμής θα πρέπει να γίνεται μεταξύ του 52⁰⁰ έως 58⁰⁰ λεπτού της ώρας. Κατά σύμβαση έχουμε ορίσει, όλες τις μετρήσεις θα γίνονται στο 55⁰ λεπτό της ώρας. Η μέση τιμή της θερμοκρασίας/υγρασίας και διεύθυνσης του ανέμου, δεν έχουν καμία αξία στην παρούσα εργασία, οπότε δεν κρίνεται σκόπιμη η υλοποίησή τους. Για τον λόγο αυτό αναπτύχθηκε μόνο η μέτρηση της ατμοσφαιρικής πίεσης καθώς είναι απαραίτητη για τον υπολογισμό της βαρομετρικής τάσης. Στο παράρτημα ΣΤ παρατίθεται ο κώδικας με τον οποίο υπολογίζεται η βαρομετρική τάση.

6.2.3 Καταγραφή Βαρομετρικής τάσης

Η βαρομετρική τάση είναι απαραίτητη μεταβλητή για τον υπολογισμό της πρόγνωσης του καιρού από τον αλγόριθμο Zambretti. Γνωρίζουμε ότι η βαρομετρική τάση υπολογίζεται αφαιρώντας την μέση ωριαία τιμή της ατμοσφαιρικής πίεσης που



μετρήθηκε πριν 3 ώρες με την τιμή της τρέχουσας ώρας. Άρα χρειαζόμαστε έναν πίνακα που να αποθηκεύονται αυτές οι μετρήσεις. Ήδη από την προηγούμενη συνάρτηση παρατηρούμε ότι όταν το ρολόι φτάσει στο 55^ο λεπτό της ώρας καλείται η συνάρτηση `katagrafiBarometrikisTasis(int pressure)`. Η συνάρτηση αυτή αποθηκεύει την μέση τιμή της ωριαίας ατμοσφαιρικής πίεσης στον πίνακα `hourPressureTrend`.

6.2.4 Υπολογισμός Βαρομετρικής Τάσης

Για τον υπολογισμό της βαρομετρικής τάσης έχουμε ότι :

$$Trend = P_t - P_{t-3} \quad (10)$$

με P_t η τρέχουσα μέση τιμή της ατμοσφαιρικής πίεσης και

P_{t-3} την προ τριώρου μέση τιμή της ατμοσφαιρικής πίεσης.

Η συνάρτηση επιστρέφει 4 διακριτές τιμές οι οποίες είναι:

- 0: όταν το αποτέλεσμα της αφαίρεσης είναι 0 δηλαδή δεν υπάρχει μεταβολή (steady).
- 1: όταν το αποτέλεσμα είναι θετικός αριθμός σημαίνει ότι η τάση είναι αυξανόμενη (rising)
- 2: όταν το αποτέλεσμα είναι αρνητικός αριθμός τότε η τάση είναι μειούμενη (falling)
- -1: όταν δεν έχουν συμπληρωθεί 3 ώρες μετρήσεων, οπότε δεν γίνεται να υπολογιστεί η βαρομετρική τάση.

Η συνάρτηση που επιστρέφει την κατάσταση της βαρομετρικής τάσης παρουσιάζεται στο παράρτημα Η.



6.2.5 Αλγόριθμος Πρόγνωσης Καιρού *Zambretti*

Ο αλγόριθμος *Zambretti* δέχεται ως είσοδο την μέγιστη και ελάχιστη καταγραφείσα τιμή της ατμοσφαιρικής πίεσης της περιοχής [20][37], την βαρομετρική τάση (σε κωδικοποίηση όπως περιγράφηκε στην προηγούμενη συνάρτηση) , την διεύθυνση του ανέμου (σε μοίρες) και την τρέχουσα τιμή της ατμοσφαιρικής πίεσης. Όταν καλείται ο αλγόριθμος δέχεται ως είσοδο μόνο την βαρομετρική τάση, καθώς όλα τα υπόλοιπα δεδομένα εισάγονται άμεσα από τους αισθητήρες. Τονίζεται ότι ο αλγόριθμος δεν λειτουργεί αν δεν έχουν παρέλθει τουλάχιστον τρεις ώρες λειτουργίας του σταθμού, έτσι ώστε να υπάρχει δυνατότητα υπολογισμού της βαρομετρικής τάσης της περιοχής. Στο παράρτημα Θ βρίσκεται ο κώδικας του αλγορίθμου Ζαμπρέττι.

6.2.6 Συνάρτηση Καταγραφής Δεδομένων στην κάρτα *SD*

Τα δεδομένα αφού ληφθούν μπορούν να αποθηκευθούν στην κάρτα *SD* για μελλοντική χρήση. Η συνάρτηση αφού λάβει τα τέσσερα δεδομένα (ένα για κάθε μετεωρολογικό δεδομένο) προς καταγραφή προσθέτει την τρέχουσα ημερομηνία και ώρα και αποθηκεύει την εγγραφή σε προκαθορισμένη *csv* μορφή. Ο κώδικας της διαδικασίας που αποθηκεύει τα δεδομένα κάθε λεπτό, βρίσκεται στο παράρτημα Ι

6.2.7 Συνάρτηση Αναγωγής στην επιφάνεια της θάλασσας

Οι μετρήσεις της ατμοσφαιρικής πίεσης πρέπει να είναι όσο το δυνατόν ακριβέστερες καθώς η πρόγνωση του καιρού είναι άρρηκτα συνδεδεμένη μαζί τους. Για τον λόγο αυτό θα πρέπει να γίνει η απαραίτητη αναγωγή στην επιφάνεια της θάλασσας. Ο συνάρτηση είναι απλή εφαρμογή της σχέσης (1) και περιγράφεται στο παράρτημα Κ.



7. Διαχείριση Μνήμης

Το Arduino UNO το οποίο χρησιμοποιήθηκε για την ανάπτυξη του σταθμού, διαθέτει περιορισμένες δυνατότητες λόγω του μικρού μεγέθους της μνήμης του. Οι περιορισμοί αυτοί, μπορεί να αναγκάσουν τον προγραμματιστή να περικόψει, ακόμα και να επανασχεδιάσει τον εκτελέσιμο κώδικα ώστε να μην υπερβαίνει το μέγιστο όριο μνήμης. Για παράδειγμα, ένα από τα κυριότερα προβλήματα που παρουσιάστηκαν ήταν η υπερχειλίση της προσωρινής μνήμη, κάτι το οποίο δεν ήταν δυνατόν να εντοπισθεί κατά την ανάπτυξη του κώδικα παρά μόνο στην εκτέλεση του προγράμματος. Το Arduino UNO διαθέτει 3 ειδών μνήμες, η κάθε μια με τα δικά της μοναδικά χαρακτηριστικά. Συγκεκριμένα αποτελείται από τις:

1. Μνήμη προγράμματος ή flash με μέγεθος 32KB
2. Μνήμη τυχαίας προσπέλασης ή Προσωρινή Μνήμη ή SRAM με μέγεθος 2KB.
3. Επαναπρογραμματιζόμενη μνήμη ή EEPROM με μέγεθος 1KB

7.1 Διαχείριση Μνήμης Προγράμματος

Σε αυτή την μνήμη αποθηκεύεται ο εκτελέσιμος κώδικας του Arduino, γνωστός και ως «σχέδιο» (sketch). Πρόκειται για μια αναπρογραμματιζόμενη μνήμη στην οποία τα δεδομένα παραμένουν αναλλοίωτα κατά την λειτουργία της πλακέτας, ενώ δεν διαγράφονται ακόμα και κατά την απώλεια της ηλεκτρικής τάσης. Ο μόνος τρόπος επεξεργασίας της είναι κατά την φόρτωση νέου εκτελέσιμου κώδικα μέσω της εφαρμογής του Arduino IDE ή άλλου παρόμοιου λογισμικού. Στην μνήμη προγράμματος είναι προτιμότερο να αποθηκεύονται σταθερές μεταβλητές (constant, define) καθώς αυτή διαθέτει μεγαλύτερη χωρητικότητα από την προσωρινή μνήμη SRAM, γεγονός το οποίο δίνει την δυνατότητα να αποθηκευτούν μη τροποποιήσιμες τιμές κατά την διάρκεια εκτέλεσης του προγράμματος.



Κατά την ανάπτυξη του κώδικα λειτουργίας του σταθμού παρατηρήθηκε το φαινόμενο της αύξησης του μεγέθους του «σχεδίου» πάνω από το όριο των 32KB που είναι το μέγεθος της μνήμης προγράμματος που διαθέτει το Arduino UNO. Για να αποφευχθεί το ενδεχόμενο της αγοράς άλλης μεγαλύτερης πλακέτας (πχ Arduino MEGA) έγινε προσπάθεια επανασχεδίασης του κώδικα αφαιρώντας δευτερευούσης σημασίας τμήματά του. Επίσης βιβλιοθήκες οι οποίες καταλάμβαναν όγκο είτε αφαιρέθηκαν, είτε τροποποιήθηκαν με αποτέλεσμα να υπάρχουν εμφανή αποτελέσματα στην εξοικονόμηση μνήμης. Αναλυτικά διενεργήθηκαν οι εξής τροποποιήσεις:

1. Αφαίρεση βιβλιοθήκης **Average.h** η οποία προορίζονταν για τον υπολογισμό του μέσου όρου των δειγμάτων μέτρησης, υγρασίας, θερμοκρασίας και ατμοσφαιρικής πίεσης. Στην θέση της γράφτηκε η συνάρτηση **averageArray()**, η οποία δέχεται ως όρισμα τον πίνακα και το πλήθος των γραμμών του, και επιστρέφει τον μέσο όρο των τιμών του. Κέρδος σε μνήμη 623 bytes.

2. Αφαιρέθηκαν όλες οι επαναλαμβανόμενες κλήσεις συναρτήσεων κυρίως όσον αφορά στην ανάγνωση των αισθητήρων. Για τον λόγο αυτό σχεδιάστηκε η συνάρτηση **getResults()** της κλάσης **weatherStation()** η οποία δέχεται ως όρισμα τα 4 μετεωρολογικά δεδομένα (θερμοκρασία, υγρασία, πίεση, διεύθυνση ανέμου σε μοίρες) και επιστέφει τα δεδομένα αυτά συν την μέγιστη και ελάχιστη καταγραφή τιμών αυτών, το σημείο δρόσου, την αισθητή θερμοκρασία, την διεύθυνση του ανέμου ονομαστικά και την πρόγνωση του καιρού σύμφωνα με τον αλγόριθμο Zambretti. Η **getResults()** επιστρέφει τα ανωτέρω δεδομένα μέσω της δομής **weatherData**, μια δομή η οποία σχεδιάστηκε ειδικά για την διαχείριση των εν λόγω δεδομένων. Κάθε δύο δευτερόλεπτα γίνεται λήψη των μετεωρολογικών δεδομένων από τους αισθητήρες και στην συνέχεια η **getResults()** εκτελώντας τις εσωτερικές μεθόδους της **weatherStation** επιστέφει τα αποτελέσματα των μετρήσεων. Πλέον η δομή **weatherData**, περιέχει όλες τις πληροφορίες που χρειάζονται ώστε



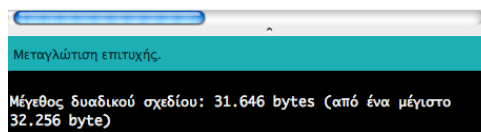
Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

ανατρέχοντας σε αυτήν, αποφεύγεται η άσκοπη χρήση του επεξεργαστή και ελαττώνεται το μέγεθος του σχεδίου. Κέρδος σε μνήμη: 456bytes.

3. Περιορισμός των κειμένων και σύμπτυξη αυτών στα άκρως απαραίτητα, για παράδειγμα το κείμενο “Station Starting...” περιορίστηκε στην λέξη “starting”. Αυτή η ενέργεια εξυπηρετεί και στην διαχείριση της μνήμης SRAM που εξηγείται παρακάτω. Κέρδος σε μνήμη: 195 bytes

4. Το μεγάλο κέρδος όμως στην εξοικονόμηση της μνήμης προγράμματος είναι η κατάργηση του αλγόριθμου Zambretti από το σχέδιο του σταθμού. Στα σχήματα 33 και 34 παρατηρείται ότι το μέγεθος σχεδίου ελαττώνεται κατά 1476 bytes ενώ επιπλέον έχουμε και κέρδος στην εξοικονόμηση της προσωρινής μνήμης κατά 712 bytes (όσα είναι τα bytes που καταλαμβάνουν τα λεκτικά στην προσωρινή μνήμη). Αυτό οφείλεται διότι υπάρχουν μεγάλα κείμενα διότι χρησιμοποιούνται για την προβολή της πρόγνωσης καιρού. Για τον λόγο αυτό κρίθηκε απαραίτητο η τροποποίηση του αλγορίθμου Zambretti, στον οποίο αντί να επιστρέφεται το λεκτικό της πρόγνωσης, θα επιστρέφεται ένας κωδικός στον οποίο αντιστοιχεί η κάθε πρόγνωση. Με αυτόν τον τρόπο δεν αποθηκεύεται το λεκτικό και δεν θα επιβαρύνεται ο σταθμός με επιπλέον εργασία.

```
data.pressure=pressure;  
data.humidity=humidity;  
data.temperature=temperature;  
data.windDirectionDegrees=windDirection;  
data.windDirection=convertDegreesToString(windDirecti  
data.heatIndex=heatIndex(humidity,temperature);  
data.dewPoint=calcDewpoint(humidity,temperature);  
data.trend=getTrend();  
data.prognosis=zambrettiAlgorithm(pressure,windDirec  
return data;  
}
```

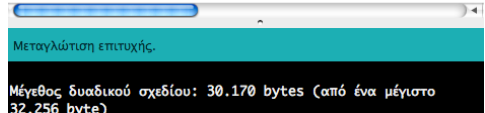


Σχήμα 33 Μέγεθος σχεδίου με τον αλγόριθμο Zambretti



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

```
data.maxTemp=maxTmp;  
data.minHumi=minHumi;  
data.maxHumi=maxHumi;  
data.pressure=pressure;  
data.humidity=humidity;  
data.temperature=temperature;  
data.windDirectionDegrees=windDirection;  
data.windDirection=convertDegreesToString(windDirectio  
data.heatIndex=heatIndex(humidity, temperature);  
data.dewPoint=calcDewpoint(humidity, temperature);  
data.trend=getTrend();  
//data.prognosis=zambrettiAlgorithm(pressure, windDire  
return data;  
}
```



Σχήμα 34 Μέγεθος σχεδίου χωρίς τον αλγόριθμο Zambretti

7.2 Διαχείριση Μνήμης SRAM

Κατά την διάρκεια των δοκιμών για την αποσφαλμάτωση του κώδικα, διαπιστώθηκε μια απροσδιόριστη συμπεριφορά ως προς την λειτουργία του σταθμού. Ο σταθμός δεν εμφάνιζε σωστά τα μηνύματα της κατάστασης του ή την προβολή της δικτυακής σελίδας. Συγκεκριμένα ενώ ο κώδικας δεν παρουσίαζε λάθος κατά την μεταγλώττιση του, ο σταθμός δεν εκκινούσε την λειτουργία του ή η λειτουργία του ήταν μη ελεγχόμενη. Αυτό οφείλονταν κυρίως στο γεγονός ότι κατά την εκκίνηση του σταθμού όλες οι μεταβλητές, κείμενα, συναρτήσεις κλπ φορτώνονται αρχικά στην προσωρινή μνήμη. Η υπέρβαση του μεγίστου ορίου χωρητικότητας της, οδηγούσε σε υπερχείλιση με αποτέλεσμα το πρόγραμμα να τερματίζεται απροσδόκητα (crash).

Για να διαπιστωθεί η κατάσταση της μνήμης SRAM γράφτηκε η συνάρτηση **FreeRam()** [38] η οποία επιστρέφει την τιμή της διαθέσιμης ελεύθερης μνήμης. Κατά την εκτέλεση της διαπιστώθηκε ότι η προσωρινή μνήμη είχε πράγματι εξαντληθεί. Ο κυριότερος λόγος είναι η χρήση κειμένων τα οποία αθροιστικά μπορεί να ξεπεράσουν το μέγεθος μνήμης. Για παράδειγμα στο κείμενο «Meteo Station Start!» καταλαμβάνονται 20 bytes, τα οποία στο σύνολο των 2KB είναι μεν λίγα, όμως η συνεχής ροή δεδομένων έχει ως αποτέλεσμα στην αύξηση χρήσης της προσωρινής μνήμης. Ειδικά δε κατά την δημιουργία και προβολή της σελίδας HTML όπου πρέπει



να ακολουθηθεί συγκεκριμένη δομή με χρήση ετικετών (tags) ο όγκος της αιτούμενης ιστοσελίδας μεγαλώνει υπερβολικά. Λαμβάνοντας υπόψη ότι μόνο τα λεκτικά για την πρόγνωση καιρού καταλαμβάνουν 719 bytes, αντιλαμβανόμαστε ότι ήδη έχει καταληφθεί το 72% της διαθέσιμης προσωρινής μνήμης.

7.2.1 Χρήση της συνάρτησης $F()$

Για την αντιμετώπιση αυτού του προβλήματος η λύση που προτείνεται είναι η αξιοποίηση της διαθέσιμης μνήμης προγράμματος PROGRAM MEMORY. Για να εκμεταλλευτούμε αυτή την δυνατότητα, θα πρέπει όλα τα σταθερά λεκτικά (πχ μηνύματα της LCD οθόνης) να τα ορίσουμε ως σταθερές μεταβλητές (const). Θέτοντας μια μεταβλητή σε σταθερά, παραμένει αναλλοίωτη καθ' ολη την διάρκεια λειτουργίας του προγράμματος. Άρα δεν θα αποθηκευτεί στην προσωρινή μνήμη αλλά θα βρίσκεται μόνιμα στην μνήμη προγράμματος. Ένας άλλος έξυπνος τρόπος είναι η χρήση της συνάρτησης $F()$ σε κάθε λεκτικό ή μήνυμα που υπάρχει. Για παράδειγμα στην εντολή `lcd.print("μήνυμα»)` μπορεί να προστεθεί η συνάρτηση $F()$ και να γραφτεί ως `lcd.print(F(«μήνυμα»))`. Η συνάρτηση $F()$ αναλαμβάνει να ορίσει ως σταθερή μεταβλητή το κείμενο που δέχεται ως όρισμα, με αποτέλεσμα την απελευθέρωση της προσωρινής μνήμης.

7.2.2 Χρήση της **PROGMEM**

Η χρήση της $F()$ είναι ιδανική για χρήση κειμένων, αλλά όμως τι μπορούμε να κάνουμε για άλλου είδους μεταβλητές ή πίνακες; Η απάντηση είναι και εδώ πολύ απλή: θα γίνει χρήση της PROGMEM. Για να γίνει χρήση της PROGMEM θα πρέπει αρχικά να δηλώσουμε την βιβλιοθήκη **avr/pgmspace.h**. Η PROGMEM αναλαμβάνει να αποθηκεύσει τα δεδομένα των μεταβλητών που έχει τεθεί, αντί για την SRAM στην μνήμη προγράμματος. Προτιμήθηκε να χρησιμοποιηθεί η PROGMEM σε



*Τριανταφύλλου Γεώργιος,
'Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων*

μεταβλητές με σταθερά δεδομένα καθώς και σε πίνακες. Παρακάτω παρατίθεται ο τρόπος χρήσης της PROGRAM:

```
const char weather0[] PROGRAM="Settled fine";
const char weather1[] PROGRAM="Fine weather";
const char weather2[] PROGRAM="Becoming fine";
const char weather3[] PROGRAM="Fine, becoming less settled";
const char weather4[] PROGRAM="Fine, possible showers";
const char weather5[] PROGRAM="Fairly fine, improving";
const char weather6[] PROGRAM="Fairly fine, possible showers early";
const char weather7[] PROGRAM="Fairly fine, showery later";
const char weather8[] PROGRAM="Showery early, improving";
const char weather9[] PROGRAM="Changeable, mending";
const char weather10[] PROGRAM="Fairly fine, showers likely";
const char weather11[] PROGRAM="Rather unsettled clearing later";
const char weather12[] PROGRAM="Unsettled, probably improving";
const char weather13[] PROGRAM="Showery, bright intervals";
const char weather14[] PROGRAM="Showery, becoming less settled";
const char weather15[] PROGRAM="Changeable, some rain";
const char weather16[] PROGRAM="Unsettled, short fine intervals";
const char weather17[] PROGRAM="Unsettled, rain later";
const char weather18[] PROGRAM="Unsettled, some rain";
const char weather19[] PROGRAM="Mostly very unsettled";
const char weather20[] PROGRAM="Occasional rain, worsening";
const char weather21[] PROGRAM="Rain at times, very unsettled";
const char weather22[] PROGRAM="Rain at frequent intervals";
const char weather23[] PROGRAM="Rain, very unsettled";
const char weather24[] PROGRAM="Stormy, may improve";
const char weather25[] PROGRAM="Stormy, much rain";

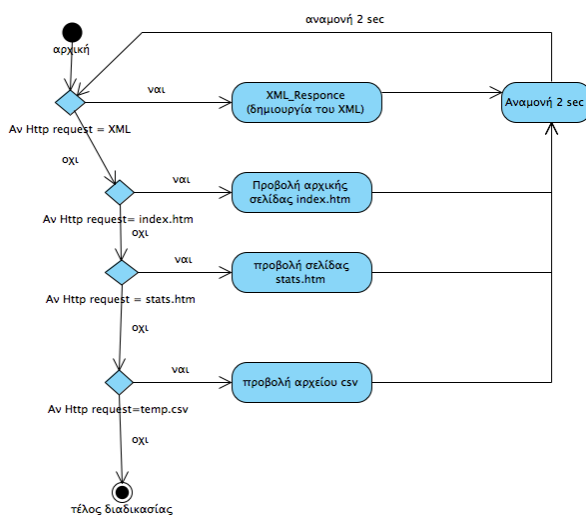
const char* forecast[] PROGRAM = {weather0, weather1,weather2,
weather3,weather4,
weather5,weather6,weather7,weather8,weather9,weather10,weather11,weat
her12,
weather13,weather14,weather15,weather16,weather17,weather18,weather19
,weather20, weather21,weather22, weather23,weather24, weather25};
// sisxetismos me ton A disko tou Zambretti broadcast. analoga me
tin barometriki tasi epilegetai kai to analogo apotelesma
const prog_uint8_t rise_options[] PROGRAM = {
25,25,25,24,24,19,16,12,11,9,8,6,5,2,1,1,0,0,0,0,0,0 };
const prog_uint8_t steady_options[] PROGRAM = {
25,25,25,25,25,25,23,23,22,18,15,13,10,4,1,1,0,0,0,0,0,0 };
const prog_uint8_t fall_options[] PROGRAM = {
25,25,25,25,25,25,25,25,23,23,21,20,17,14,7,3,1,1,1,0,0,0 };
```



8. Διεπαφή Χρήστη

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 την χρήση έτοιμης βιβλιοθήκης javascript η οποία είναι ανοιχτού κώδικα και διατίθεται δωρεάν. Η διαδικασία λειτουργίας είναι η ακόλουθη:

Αρχικά αποστέλλεται απο τον φυλλομετρητή, http αίτημα στο Arduino. Στην συνέχεια το αίτημα αναλύεται και παρουσιάζεται στον φυλλομετρητή. Οι ιστοσελίδες βρίσκονται αποθηκευμένες στην κάρτα μνήμης SD υπό την μορφή αρχείων HTML. Για τον λόγο αυτό θα πρέπει να ενσωματωθεί η βιβλιοθήκη SD για την διαχείριση της κάρτας SD. Η διαδικασία λειτουργίας του Arduino ως εξυπηρετητή απεικονίζεται στο Σχήμα 35. 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.



Σχήμα 35 Διάγραμμα ροής διαδικασίας εξυπηρετητή διαδικτύου



Ο κώδικας της λειτουργίας ως εξυπηρετητή διαδικτύου παρατίθεται στο παράρτημα ΙΑ.

8.1 Αρχική Σελίδα

Είναι η σελίδα την οποία ο χρήστης παρακολουθεί όλες τις πληροφορίες του καιρού. Αρχικά ο χρήστης εισάγει την διεύθυνση IP ή το πλήρως αναγνωρισμένο όνομα τομέα (Fully Qualified Domain Name) στην οποία ανήκει ο σταθμός. Για τις ανάγκες της παρούσης εργασίας, η διεύθυνση IP δοκιμάστηκε σε τοπικό δίκτυο με διεύθυνση 192.168.2.20.

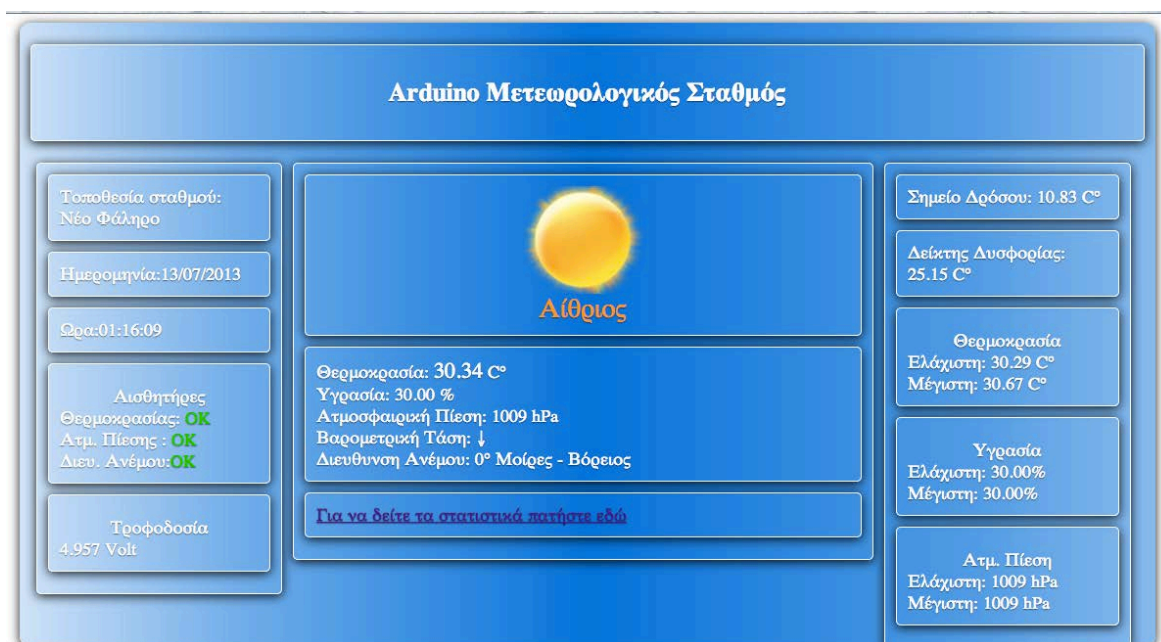
Μόλις ο χρήστης πληκτρολογήσει την διεύθυνση στην γραμμή εντολών του φυλλομετρητή, αποστέλλεται το αίτημα http

```
"HTTP/1.1 GET /" ή "HTTP/1.1 GET /index.htm"
```

το οποίο μεταφράζεται ως: «πρωτόκολλο HTTP έκδοση 1.1 ζητώ την σελίδα index.htm» Το Arduino εφόσον ελέγξει την ύπαρξη του αρχείου στην κάρτα SD, το αποστέλλει στον χρήστη. Η σελίδα που προβάλλεται είναι η κεντρική σελίδα του σταθμού και περιέχει τα μετεωρολογικά και άλλα δεδομένα που περιέχουν πληροφορίες σχετικά με την κατάσταση του σταθμού. Στο Σχήμα 36 εμφανίζεται η κεντρική σελίδα του σταθμού.



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων



Σχήμα 36 Το γραφικό περιβάλλον του μετεωρολογικού σταθμού

Αναλυτικά τα δεδομένα που προβάλλονται είναι:

- Οπτικοποιημένη ένδειξη της πρόγνωσης του καιρού με χρήση εικόνων,
- Την πρόγνωση του καιρού
- Τρέχουσα θερμοκρασία καθώς και η μέγιστη και ελάχιστη ημερήσια τιμή της
- Τρέχουσα Υγρασία καθώς και η μέγιστη και ελάχιστη ημερήσια τιμή της
- Τρέχουσα ατμ. πίεση καθώς και η μέγιστη και ελάχιστη ημερήσια τιμή της
- Το σημείο δρόσου
- Τον δείκτη δυσφορίας
- Την διεύθυνση ανέμου σε μοίρες
- Ημερομηνία ώρα σταθμού
- Κατάσταση αισθητήρων
- Τιμή της ηλεκτρικής τάσης σε βολτ

Όταν δεν αιτηθεί η προβολή κάποιου αρχείου, τότε το Arduino αποστέλλει ροή υπό την μορφή XML η οποία περιλαμβάνει όλες τις παραπάνω πληροφορίες. Στην συνέχεια ο φυλλομετρητής, κάνοντας χρήση της τεχνολογίας javascript και AJAX



Τριανταφύλλου Γεώργιος,
'Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

αναλύει το εισερχόμενο XML και δυναμικά ενημερώνει την ιστοσελίδα με τα τρέχοντα μετεωρολογικά δεδομένα. Με τον τρόπο αυτό αποφεύγεται η συνεχής ανανέωση της ιστοσελίδας του σταθμού, περιορίζοντας τον μεταφερόμενο όγκο δεδομένων, στον ελάχιστο δυνατό. Το παραγόμενο XML έχει την παρακάτω δομή:

```
<?xmlversion = "1.0" ?>
<inputs>
  <trend></trend> // βαρομετρική τάση
  <prognosis></prognosis> // πρόγνωση καιρού
  <temperature></temperature> // τρέχουσα θερμοκρασία
  <humidity></humidity> // τρέχουσα υγρασία
  <pressure></pressure> // τρέχουσα πίεση
  <windDirection></windDirection> // κατεύθυνση ανέμου (λεκτικό)
  <windDirecionDegrees></windDirecionDegrees> // κατεύθυνση ανέμου
  (μοίρες)
  <maxTemp></maxTemp> // μέγιστη θερμοκρασία
  <minTemp></minTemp> // ελάχιστη θερμοκρασία
  <maxPress></maxPress> // μέγιστη πίεση
  <minPress></minPress> // ελάχιστη πίεση
  <maxHumi></maxHumi> // μέγιστη υγρασία
  <minHumi></minHumi> // ελάχιστη υγρασία
  <heatIndex></heatIndex> // δείκτης δυσφορίας
  <dewPoint></dewPoint> // σημείο δρόσου
  <prognosisCode></prognosisCode> // κωδικός πρόγνωσης
  <stationDate></stationDate> // ημερομηνία σταθμού
  <stationDate></stationDate> // ώρα σταθμού
  <voltage></voltage> // ηλεκτρική τάση σταθμού
</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 όπως έχει περιγραφεί στο κεφάλαιο 7. Για τον λόγο αυτό ο σταθμός αποστέλλει στην σελίδα έναν κωδικό, ο οποίος αντιστοιχεί στο γράμμα του αλγορίθμου *Zambretti*. Στην σελίδα υπάρχει ένας πίνακας **forecast** σε γλώσσα *javascript* ο οποίος περιέχει την μετάφραση του αντίστοιχου λεκτικού στα ελληνικά. Για την άμεση κατανόηση της πρόγνωσης του καιρού εκτός του λεκτικού κειμένου, χρησιμοποιείται οπτικό βοήθημα που αναπαριστά την αντίστοιχη πρόγνωση με χρήση εικονιδίων. Εκτός του



forecast υπάρχει και ο πίνακας **forecastIcons** ο οποίος περιέχει την πλήρη διαδρομή των εικόνων της κάθε κατάστασης του καιρού στα ελληνικά.

Τέλος υπάρχει δυνατότητα προβολής των καταγραμμένων μεγίστων και ελαχίστων τιμών της θερμοκρασίας υπό την μορφή γραφήματος. Για την προβολή των γραφημάτων, ο χρήστης επιλέγει τον υπερσύνδεσμο “για να δείτε τα στατιστικά πατήστε εδώ” που βρίσκεται στο τέλος της σελίδας.

8.2 Προβολή Στατιστικών

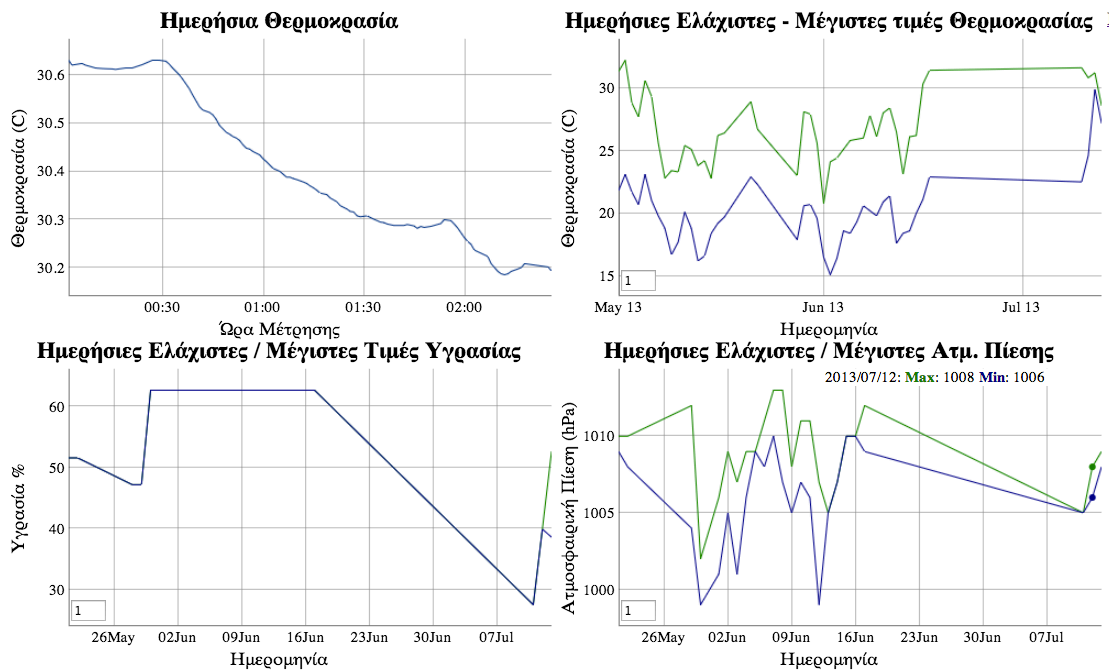
Ο σταθμός κατά την ώρα 23:59 κάθε ημέρας αποθηκεύει την μέγιστη και ελάχιστη στις τιμές της θερμοκρασίας, υγρασίας και ατμοσφαιρικής πίεσης. Η αποθήκευση γίνεται σε μορφή csv στα αρχεία *temp.csv*, *humid.csv*, *pres.csv* αντίστοιχα . Το αρχείο αποθηκεύεται σύμφωνα με το πρότυπο μορφοποίησης ημερομηνίας και αριθμών ISO 8601 [39] . Τα περιεχόμενα του αρχείου είναι:

```
date, max, min  
ΕΕΕΕΜΜΗΗ, 0.00, 0.00
```

Στην αρχική σελίδα ο χρήστης έχει την δυνατότητα να επιλέξει τον υπερσύνδεσμο “για να δείτε τα στατιστικά πατήστε εδώ” και να εμφανίσει την σελίδα *stats.htm*. Σε αυτή την σελίδα ο χρήστης μπορεί να δει υπό την μορφή γραφημάτων, τις μέγιστες και ελάχιστες τιμές των μετεωρολογικών δεδομένων (ανά ημέρα) καθώς και τις μετρήσεις της τρέχουσας ημέρας.



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων



Σχήμα 37 Σελίδα προβολής στατιστικών stats.htm

Στο Σχήμα 37 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 [33]. 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 από τον σταθμό. Ο σταθμός άλλωστε, έχει σχεδιαστεί για διασύνδεση στο διαδίκτυο, οπότε η αναφορά σε άλλους ιστότοπους θεωρείται δεδομένη.



9. Έλεγχος Συστήματος

Στο κεφάλαιο αυτό περιγράφεται ο τρόπος με τον οποίο έγινε η καταγραφή και ο έλεγχος απόδοσης του συστήματος. Από την 1^η έως την 19^η Μαΐου έγινε καταγραφή δεδομένων από τον σταθμό, τα αποτελέσματα των οποίων αποθηκεύτηκαν στην κάρτα SD. Τα δεδομένα αυτά αποθηκεύονται σε τέσσερα διαφορετικά αρχεία ως ακολούθως:

α) αρχείο *temp.csv* για την καταγραφή της μέγιστης και ελάχιστης τιμής θερμοκρασίας

β) αρχείο *humi.csv* για την καταγραφή της μέγιστης και ελάχιστης τιμής υγρασίας

γ) αρχείο *press.csv* για την καταγραφή της μέγιστης και ελάχιστης τιμής της ατμοσφαιρικής πίεσης.

δ) αρχεία της μορφής *εεεεμμηη.csv*. Καθημερινά δημιουργείται ένα αρχείο υπό την μορφή *εεεεμμηη.csv* με τυποποίηση εεεε τα 4 ψηφία του έτους, μμ τα 2 ψηφία του μήνα και ηη τα 2 ψηφία της ημέρας. Στο αρχείο αυτό αποθηκεύεται ο μέσος όρος των 5 μετρήσεων ανά λεπτό σύμφωνα με την διαδικασία που έχει αναλυθεί στο 2^ο κεφάλαιο. Εφόσον η καταγραφή γίνεται σε κάθε λεπτό της ώρας, το αρχείο θα περιέχει 3600 μετρήσεις. Για παράδειγμα στο αρχείο *20130515.csv* περιέχονται οι μετρήσεις που πραγματοποιήθηκαν την 15/05/2013. Η δομή των αρχείων CSV είναι η ακόλουθη:

```
Date,Pressure,Temperature,Humidity,WindDirection
.....
2013/05/15 12:44,1010,23.02,56.30,N
2013/05/15 12:45,1010,23.02,56.30,N
2013/05/15 12:46,1010,23.02,56.30,N
2013/05/15 12:47,1010,23.01,56.30,N
.....
```



Στην συνέχεια αντλήθηκαν δεδομένα από την ιστοσελίδα www.meteo.gr που τελεί υπό την εποπτεία του Εθνικού Αστεροσκοπείου Αθηνών και συγκεκριμένα στον υποσταθμό Παλαιού Φαλήρου (<http://penteli.meteo.gr/stations/faliro/>) ο οποίος βρίσκεται σε απόσταση 4,2 χμ. σε ευθεία νοητή γραμμή, από το σημείο που έχει εγκατασταθεί ο μετεωρολογικός σταθμός.

Θα πρέπει να ληφθεί υπόψη ότι ο σταθμός δεν έχει τοποθετηθεί σε ειδικό μετεωρολογικό κλωβό αλλά σε απλό κουτί για την προστασία του από τα καιρικά φαινόμενα.

9.1 Έλεγχος Μετεωρολογικών Δεδομένων

Σε αυτό το στάδιο συγκρίθηκαν τα δεδομένα που συνέλεξε ο σταθμός και βρίσκονται αποθηκευμένα στα αρχεία *temp.csv*, *humid.csv*, *pres.csv* με τα αντίστοιχα μετεωρολογικά δεδομένα που βρίσκονται στην διεύθυνση <http://penteli.meteo.gr/stations/faliro/>. Με βάση αυτά τα δεδομένα δημιουργήθηκαν γραφικές παραστάσεις για να αξιολογηθεί η απόδοση των αισθητήρων και του αλγορίθμου καταγραφής.

9.1.1 Θερμοκρασία

Τα δεδομένα αποθηκεύτηκαν στο αρχείο *temp.csv* και στο οποίο συμπεριλαμβάνεται η μέγιστη και ελάχιστη θερμοκρασία την συγκεκριμένη χρονική περίοδο. Σε αυτό το σημείο θα πρέπει να υπενθυμιστεί, σύμφωνα με τον Παγκόσμιο Οργανισμό Μετεωρολογίας, η μέγιστη και ελάχιστη τιμή θερμοκρασίας καταγράφεται μεταξύ 06:00-18:00 τοπική ώρα.



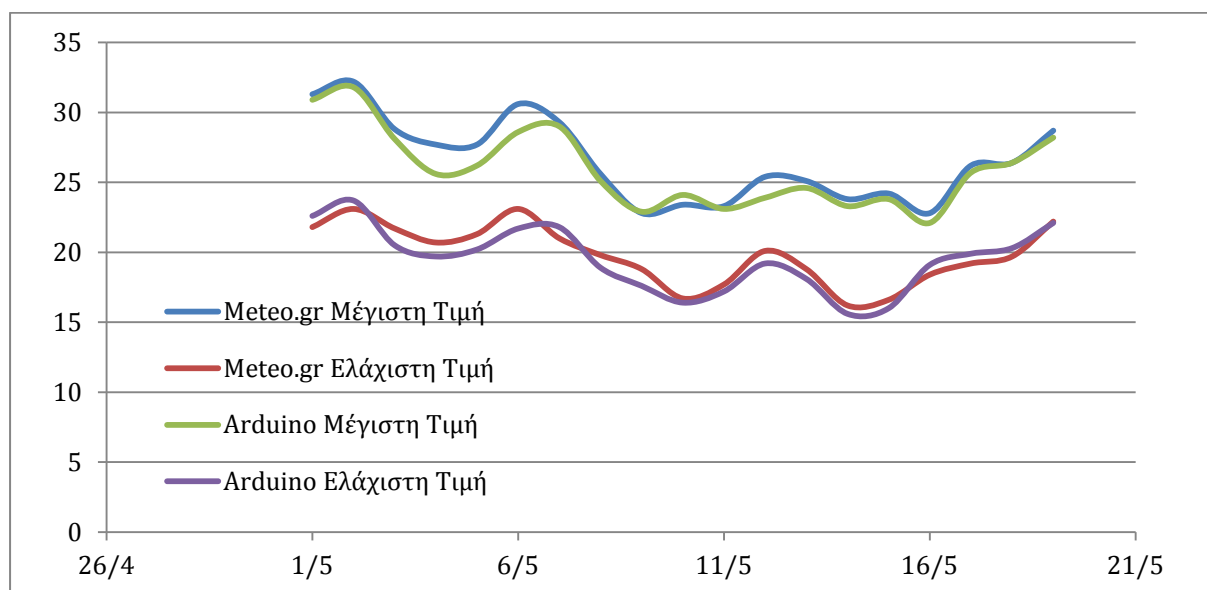
Τριανταφύλλου Γεώργιος,
'Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

Πίνακας 11 Καταγεγραμμένα δεδομένα Arduino και Meteo.gr

Ημερομηνία	Meteo.gr Μέγιστη Τιμή	Meteo.gr Ελάχιστη Τιμή	Arduino Μέγιστη Τιμή	Arduino Ελάχιστη Τιμή
1/5/13	31,3	21,8	30,9	22,6
2/5/13	32,2	23,1	31,8	23,7
3/5/13	28,8	21,7	28,1	20,5
4/5/13	27,7	20,7	25,6	19,7
5/5/13	27,7	21,3	26,2	20,2
6/5/13	30,6	23,1	28,6	21,7
7/5/13	29,3	21,0	29,0	21,8
8/5/13	25,6	19,8	25,1	18,9
9/5/13	22,8	18,8	22,9	17,6
10/5/13	23,4	16,7	24,1	16,4
11/5/13	23,3	17,7	23,1	17,2
12/5/13	25,4	20,1	23,9	19,2
13/5/13	25,1	18,8	24,6	18,1
14/5/13	23,8	16,2	23,3	15,6
15/5/13	24,2	16,6	23,8	16,0
16/5/13	22,8	18,4	22,1	19,1
17/5/13	26,2	19,2	25,7	19,9
18/5/13	26,4	19,7	26,4	20,3
19/5/13	28,7	22,2	28,2	22,1



Τριανταφύλλου Γεώργιος,
'Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων



Σχήμα 38 Συγκριτική απεικόνιση δεδομένων Meteo.gr / Arduino

Στο Σχήμα 38 βλέπουμε την γραφική παράσταση των δεδομένων μεταξύ του σταθμού και σελίδας Meteo.gr για τον εντοπισμό τυχόν αποκλίσεων στις μετρήσεις του σταθμού καθώς και της ποιότητας του αλγορίθμου καταγραφής δειγμάτων. Λαμβανομένου υπόψη ότι οι αισθητήρες του σταθμού βρίσκονται σε περιβάλλον που δεν πληροί τις προϋποθέσεις που έχει θεσπίσει ο παγκόσμιος οργανισμός μετεωρολογίας, τα ανωτέρω δεδομένα κρίνονται ικανοποιητικά και φέρουν σε μεγάλο βαθμό ομοιογένεια μεταξύ τους. Κατά συνέπεια, αν και το δείγμα είναι σχετικά μικρό για να μπορέσει να εξαχθεί ένα ασφαλές συμπέρασμα, δεν παρουσιάζονται μεγάλες ή σοβαρές αποκλίσεις σε σχέση με τα αντίστοιχες μετρήσεις της περιοχής, άρα η λειτουργία του σταθμού αποδίδει ασφαλείς μετρήσεις ως προς την καταγραφή της θερμοκρασίας.



9.1.2 Υγρασία

Τα δεδομένα αποθηκεύτηκαν στο αρχείο *hum1.csv* και στο οποίο συμπεριλαμβάνεται η μέγιστη και ελάχιστη θερμοκρασία την συγκεκριμένη χρονική περίοδο. Σε αυτό το σημείο πρέπει να τονιστεί ότι τα δεδομένα της υγρασίας από την ιστοσελίδα *Meteo.gr* είναι αποθηκευμένα σε ακέραιο αριθμό.

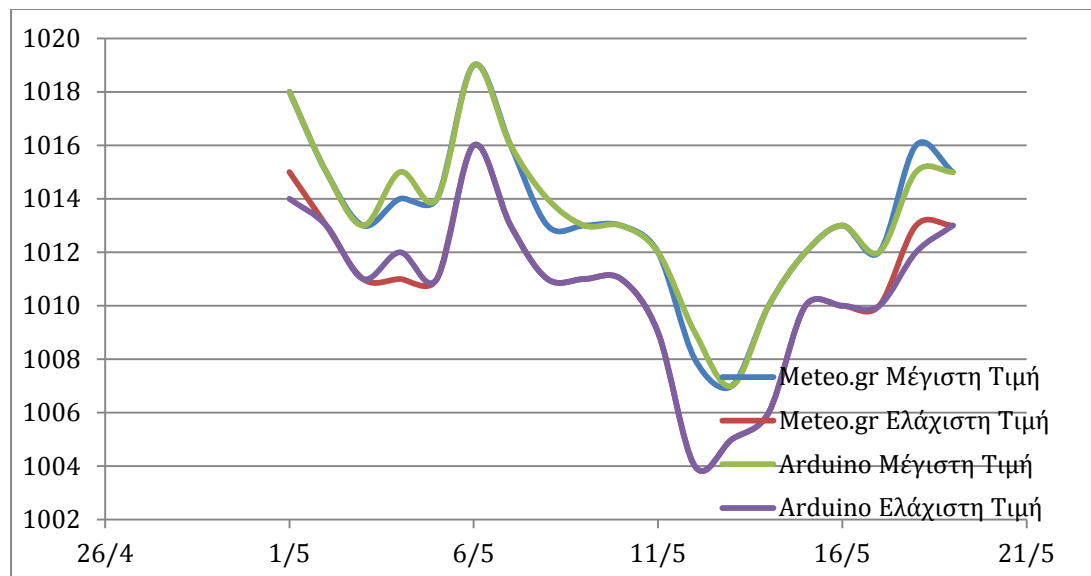
Παρατίθεται ο πίνακας με τις καταγεγραμμένες τιμές της υγρασίας μεταξύ του σταθμού και του *Meteo.gr*

Πίνακας 12 Καταγραφείσες τιμές υγρασίας της ιστοσελίδας *meteo.gr* και του σταθμού *Arduino*

Ημερομηνία	Meteo.gr Μέγιστη Τιμή	Meteo.gr Ελάχιστη Τιμή	Arduino Μέγιστη Τιμή	Arduino Ελάχιστη Τιμή
1/5/13	38,0	23,0	38,1	22,4
2/5/13	34,0	15,0	33,4	14,2
3/5/13	62,0	26,0	60,9	23,9
4/5/13	60,0	33,0	57	34,7
5/5/13	57,0	34,0	54,6	35,9
6/5/13	55,0	27,0	53,8	28,5
7/5/13	61,0	27,0	60,1	28,2
8/5/13	76,0	35,0	75,5	34,1
9/5/13	85,0	51,0	83	51,2
10/5/13	81,0	45,0	79,9	46,8
11/5/13	80,0	50,0	72,1	52,2
12/5/13	73,0	44,0	61,4	42,9
13/5/13	72,0	40,0	71,9	41,7
14/5/13	83,0	45,0	72,7	43,6
15/5/13	74,0	45,0	80,5	53,2
16/5/13	80,0	52,0	79,6	49,4
17/5/13	83,0	48,0	82,2	45,9
18/5/13	74,0	44,0	76,7	33,1
19/5/13	69,0	31,0	69,9	36,9



Με βάση τον ανωτέρω πίνακα προκύπτει το παρακάτω διάγραμμα.



Σχήμα 39 Συγκριτική απεικόνιση δεδομένων Meteo.gr / Arduino

Σύμφωνα με τα παραπάνω παρατηρείται μεγάλη ακρίβεια των αισθητήρων του σταθμού, με τα αντίστοιχα δεδομένα της ιστοσελίδας Meteo.gr. Όπως και στην περίπτωση της θερμοκρασίας, η καταγραφή των δεδομένων υγρασίας κυμαίνεται στα ίδια επίπεδα ακρίβειας, δίνοντας μας ικανοποιητικά αποτελέσματα.

9.1.3 Ατμοσφαιρική πίεση

Τα στατιστικά δεδομένα της ατμοσφαιρικής πίεσης αποθηκεύονται στο αρχείο *pres.csv*. Η ατμοσφαιρική πίεση μετριέται σε ακεραίους αριθμούς και σε κλίμακα hPa. Αξίζει να σημειωθεί ότι ο αισθητήρας BMP-085 συλλέγει δεδομένα σε Pa, οπότε για να προβληθούν τα δεδομένα μετατρέπονται σε hPa μέσω του τύπου

$$hPa = \frac{Pa}{100}$$



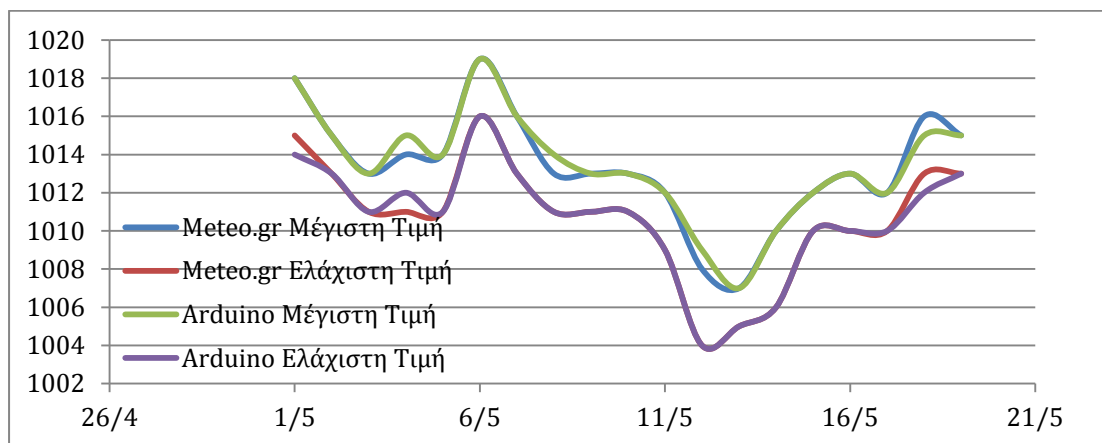
Τριανταφύλλου Γεώργιος,
'Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

Παρατίθεται ο πίνακας με τις καταγεγραμμένες τιμές ατμοσφαιρικής πίεσης μεταξύ του σταθμού και του Meteo.gr

Πίνακας 13 Καταγεγραμμένα δεδομένα ατμοσφαιρικής πίεσης απο meteo.gr και arduino

Ημερομηνια	Meteo.gr Μέγιστη Τιμή	Meteo.gr Ελάχιστη Τιμή	Arduino Μέγιστη Τιμή	Arduino Ελάχιστη Τιμή
1/5/13	1018	1015	1018	1014
2/5/13	1015	1013	1015	1013
3/5/13	1013	1011	1013	1011
4/5/13	1014	1011	1015	1012
5/5/13	1014	1011	1014	1011
6/5/13	1019	1016	1019	1016
7/5/13	1016	1013	1016	1013
8/5/13	1013	1011	1014	1011
9/5/13	1013	1011	1013	1011
10/5/13	1013	1011	1013	1011
11/5/13	1012	1009	1012	1009
12/5/13	1008	1004	1009	1004
13/5/13	1007	1005	1007	1005
14/5/13	1010	1006	1010	1006
15/5/13	1012	1010	1012	1010
16/5/13	1013	1010	1013	1010
17/5/13	1012	1010	1012	1010
18/5/13	1016	1013	1015	1012
19/5/13	1015	1013	1015	1013

Με βάση τον ανωτέρω πίνακα προκύπτει το ακόλουθο γράφημα:



Σχήμα 40 Γραφική παράσταση καταγραφείσας ατμοσφαιρικής πίεσης meteo.gr και Arduino

Σύμφωνα με τα παραπάνω παρατηρείται μεγάλη ακρίβεια των αισθητήρων του σταθμού, με τα αντίστοιχα δεδομένα της ιστοσελίδας Meteo.gr. Όπως και στις προηγούμενες δύο περιπτώσεις, η καταγραφή των δεδομένων ατμοσφαιρικής πίεσης κυμαίνεται σε ικανοποιητικά επίπεδα, ώστε να συμπεράνουμε ότι η λειτουργία του σταθμού αποδίδει ασφαλείς μετρήσεις και στην καταγραφή της ατμοσφαιρικής πίεσης.

9.2 Απόδοση Μετεωρολογικού Μοντέλου

Για να μελετηθεί η απόδοση του μετεωρολογικού μοντέλου, τηρήθηκε η ακόλουθη διαδικασία:

Κατασκευάστηκε πίνακας ο οποίος περιείχε τις προς μελέτη ημερομηνίες. Οι σχεδιαστές του αλγορίθμου προτείνουν ότι η ιδανική ώρα κατά την οποία μπορούμε να λάβουμε όσο το δυνατόν ακριβέστερη πρόγνωση είναι η 09:00. Έχοντας αυτόν τον κανόνα υπόψη, καθημερινά από 09:00 έως 10:00 καταγράφονταν η πρόγνωση του σταθμού στο αντίστοιχο πεδίο. Ταυτόχρονα καταγράφονταν η ατμοσφαιρική πίεση, η διεύθυνση του ανέμου καθώς και η βαρομετρική τάση, μεταβλητές απαραίτητες για την πρόγνωση καιρού με το μοντέλο που χρησιμοποιούμε. Στην συνέχεια για το



*Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων*

υπόλοιπο διάστημα των 12 ωρών και συγκεκριμένα στις ώρες 12:00, 15:00, 18:00 και 21:00 ο καιρός παρατηρούνταν οπτικά και σε περίπτωση τροποποίησης του, ενημερώνονταν το σχετικό πεδίο. Ανάμεσα στο χρονικό διάστημα 21:00 και 22:00, καταγράφονταν τα καιρικά φαινόμενα που τελικά επικρατούσαν στην περιοχή. Ο πίνακας 10 είναι αυτός που τελικά προέκυψε από την παραπάνω διαδικασία.



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

Ημερ	Πραγματική Κατάσταση Καιρού	Πρόγνωση Σταθμού με χρήση αλγορίθμου Zambretti	Άνεμος	ατμ. πιεση	βαρ. τάση
1/5/13	Αίθριος χωρίς σύννεφα	Αίθριος.	BΔ	1018	Σταθ
2/5/13	Αίθριος χωρίς σύννεφα	σχεδόν αίθριος, πολύ πιθανόν ελαφρές βροχοπτώσεις	BΔ	1015	Ανοδ
3/5/13	Αίθριος με λίγα σύννεφα	Ελαφρές βροχοπτώσεις, με διαστήματα ηλιοφάνειας	BΔ	1013	Σταθ
4/5/13	Αίθριος χωρίς σύννεφα	Αίθριος αμετάβλητος.	B	1013	Ανοδ
5/5/13	Αίθριος χωρίς σύννεφα	Αίθριος αμετάβλητος.	B	1016	Ανοδ
6/5/13	Αίθριος χωρίς σύννεφα	Αίθριος αμετάβλητος.	B	1018	Ανοδ
7/5/13	Αίθριος χωρίς σύννεφα	Αίθριος.	B	1016	Σταθ
8/5/13	βροχή με διαστήματα ηλιοφάνειας	Άστατος, αργότερα βροχοπτώσεις.	BΔ	1012	Πτωτ
9/5/13	Συννεφιά - βροχή μετά της 13:00 – 19:00 αίθριος	Σχεδόν άστατος, βελτίωση αργά στην ημέρα	B	1012	Πτωτ
10/5/13	Βροχή - Σταδιακή βελτίωση στις 15:00 -17:00 Αίθριος	Αίθριος.	BΔ	1012	Ανοδ
11/5/13	Αίθριος χωρίς σύννεφα	Αίθριος.	BΔ	1012	Σταθ
12/5/13	Αίθριος χωρίς σύννεφα - 11:00 συννεφιά - 13:00 εκτεταμένες βροχές	Βροχοπτώσεις κατά διαστήματα, αργότερα επιδείνωση.	BΔ	1006	Πτωτ
13/5/13	Αίθριος με λίγα σύννεφα - 11:00 συννεφιά - 17:00 εκτεταμένες βροχές	Αίθριος.	BΔ	1006	Ανοδ



Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων

14/5/13	Αίθριος με λίγα σύννεφα - 11:00 συνεφιά - 14:00 λίγες βροχές	Αίθριος.	ΒΔ	1007	Ανοδ
15/5/13	Αίθριος χωρίς σύννεφα	Αίθριος.	ΒΔ	1011	Ανοδ
16/5/13	Αίθριος χωρίς σύννεφα - 12:00 συνεφιά - 19:00 αίθριος	Σχεδόν αίθριος, σε βελτίωση.	N	1012	Ανοδ
17/5/13	Αίθριος χωρίς σύννεφα - 12:00 βροχή κατά διαστήματα - 17:00 αίθριος	Βροχοπτώσεις κατά διαστήματα, εν συνέχεια έντονα άστατος.	NA	1010	Πτωτ
18/5/13	Αίθριος χωρίς σύννεφα	Αίθριος.	NA	1015	Ανοδ
19/5/13	Αίθριος χωρίς σύννεφα	Αίθριος.	NA	1016	Σταθ

Πίνακας 14 Απόδοση Αλγορίθμου Zambretti σε σχέση με την πραγματική κατάσταση του καιρού



Με βάση τον ανωτέρω πίνακα προκύπτει ένα μεγάλο ποσοστό επιτυχίας του αλγορίθμου Zambretti μεταξύ του προβλεπόμενου και του πραγματικού καιρού. Αν και τα δείγματα είναι ελάχιστα για να ελέγξουμε την ποιότητα του μοντέλου, από τις ανωτέρω καταγραφές μπορούμε να εξάγουμε τις εξής παρατηρήσεις:

- α) 12 φορές εντοπίζεται η πρόγνωση «αίθριος ή αίθριος αμετάβλητος» εκ των οποίων οι 10 από αυτές υπήρξαν ορθές ως προς τα καιρικά φαινόμενα τελικά επικράτησαν.
- β) 5 φορές εντοπίζεται πρόγνωση σχετικά με βροχοπτώσεις, εκ των οποίων και οι 5 υπήρξαν ορθές.
- γ) υπάρχει 1 πρόγνωση «Σχεδόν αίθριος, σε βελτίωση» η οποία και επαληθεύτηκε.
- δ) εντοπίζεται 1 πρόγνωση “σχεδόν αίθριος, πολύ πιθανόν ελαφρές βροχοπτώσεις” η οποία αν και δεν επαληθεύτηκε κατά γράμμα μπορεί να θεωρηθεί σωστό καθώς ο αλγόριθμος θεωρεί ως «πιθανόν ελαφρές βροχοπτώσεις».

Στατιστικά έχουμε $\frac{2}{19} \approx 11\%$ πιθανότητα η πρόγνωση να είναι εσφαλμένη και 89% η πρόγνωση να είναι ορθή. Λαμβάνοντας υπόψη ότι ο αλγόριθμος έχει ποσοστό επιτυχίας 91% [20] μπορούμε να συμπεράνουμε ότι το μοντέλο εξάγει σχετικά ασφαλή προγνώσεις των καιρικών φαινομένων που θα επικρατήσουν στην περιοχή.



10. Αδυναμίες και βελτιώσεις

Ο σταθμός όσον αφορά τον τρόπο καταγραφής των δεδομένων, έχει κατασκευαστεί σύμφωνα με τα πρότυπα του Παγκόσμιου Οργανισμού Μετεωρολογίας και οι αισθητήρες που επιλέχθηκαν καλύπτουν τις ελάχιστες απαιτήσεις που έχει θεσπίσει ο εν λόγω οργανισμός. Παρόλα αυτά όμως, για την επέκταση της εφαρμογής στον εν λόγω σταθμό κρίνεται σκόπιμο να προστεθούν ορισμένες δυνατότητες. Μερικές από αυτές είναι:

α) Διασύνδεση με σύστημα RDBMS για την αποθήκευση των μετεωρολογικών δεδομένων όχι στην μνήμη του σταθμού, αλλά σε κεντρικό εξυπηρετητή. Με αυτό τον τρόπο απελευθερώνονται πολύτιμοι πόροι και μετακυλύεται η εργασία από το περιορισμένων δυνατοτήτων Arduino, στα μεγάλα συστήματα διαχείρισης βάσεων δεδομένων, από όπου ο χρήστης θα μπορεί να διαχειριστεί τα δεδομένα του με περισσότερη ευελιξία και ποιο αποδοτικά.

β) Εφόσον γίνεται αναφορά σε αυτόνομο μετεωρολογικό σταθμό, θα πρέπει αυτός να λειτουργεί, χωρίς να απαιτείται η ύπαρξη του ανθρώπου για την αντιμετώπιση προβλημάτων που τυχόν θα εντοπιστούν. Έτσι θα πρέπει να ληφθεί μέριμνα έτσι ώστε ο σταθμός να λειτουργεί για μεγάλο χρονικό διάστημα σε περίπτωση λόγου χάρη πτώσης του ηλεκτρικού ρεύματος ή αντικατάσταση της κάρτας Ethernet με αντίστοιχη κάρτα δικτύωσης μέσω δικτύου κινητής τηλεφωνίας.

γ) θα πρέπει να τροποποιηθεί η παρούσα αρχιτεκτονική σε πιο σύγχρονη και εύχρηστη στον τελικό χρήστη. Για παράδειγμα ο σταθμός αυτή την στιγμή εξυπηρετεί μόνο έναν πελάτη ανά αίτηση. Έτσι αν κάποιος άλλος πελάτης ζητήσει εκείνη την στιγμή να έχει πρόσβαση στα μετεωρολογικά δεδομένα, τότε η επιβάρυνση θα είναι υπερβολική για τον μικρό επεξεργαστή, με αποτέλεσμα την πιθανή κατάρρευση του. Αυτό το πρόβλημα μπορεί να λυθεί αντικαθιστώντας το Arduino από σύστημα με κεντρικό ρόλο, σε σύστημα με περιφερειακό ρόλο. Έτσι αντί να επεξεργάζεται / αποθηκεύει τα ανωτέρω δεδομένα, θα τα συλλέγει από τους αισθητήρες και θα τα αποστέλλει στο RDBMS. Στην συνέχεια, τα δεδομένα αυτά θα προβάλλονται μέσω ειδικής web εφαρμογής από εξειδικευμένο εξυπηρετητή το



οποίο θεωρητικά σημαίνει ότι ο ανωτέρω περιορισμός πλέον θα αντιμετωπιστεί επιτυχώς.

δ) Λόγω των περιορισμών του Arduino δεν είναι δυνατή η υλοποίηση πιο σύνθετων και ακριβέστερων αλγορίθμων πρόγνωσης καιρού όπως ο αλγόριθμος Sager. Η υλοποίηση του 3^{ου} βήματος θα εξυπηρετούσε σημαντικά στην κατασκευή πιο πολύπλοκων αλγορίθμων, μιας και η επεξεργασία, όπως έχει ήδη ειπωθεί, θα γίνεται στους εξυπηρετητές και όχι στο Arduino.

ε) Δεν έχει γίνει κάποια έρευνα ως προς την ακεραιότητα και εμπιστευτικότητα των δεδομένων που αποστέλλονται ή αποθηκεύονται σε αυτό. Έτσι υπάρχει η πιθανότητα κάποιος κακόβουλος χρήστης, να εντοπίσει κάποια ευπάθεια και να εισβάλει στο σύστημα με σκοπό την παραποίηση ή την μη εξουσιοδοτημένη διαγραφή των δεδομένων, οπότε είναι πολύ πιθανό να υπάρχουν κενά ασφαλείας ως προς την χρήση του, τα οποία θα πρέπει να εντοπιστούν και να αντιμετωπιστούν.

στ) Τέλος θα πρέπει οι αισθητήρες του σταθμού να εγκατασταθούν σε ειδικό μετεωρολογικό κλωβό, έτσι ώστε οι μετρήσεις να είναι όσο το δυνατόν ακριβέστερες.



11. Συμπεράσματα.

Ο στόχος της παρούσης Πτυχιακής Εργασίας ήταν η υλοποίηση ενός αυτόνομου μετεωρολογικού σταθμού για την παρακολούθηση και καταγραφή μετεωρολογικών φαινομένων και στην συνέχεια με βάση αυτά τα στοιχεία να γίνεται πρόγνωση καιρού. Μετά από αρκετή αναζήτηση στο διαδίκτυο καταλήξαμε στη χρήση του αλγορίθμου Ζαμπρέτι, για την πρόγνωση του καιρού. Το επίπεδο ευαισθησίας των αισθητήρων είναι σε ικανοποιητικό βαθμό, όπως και οι αλγόριθμοι καταγραφής και αποθήκευσης των δεδομένων. Αυτό αποδεικνύεται στα συγκριτικά του κεφαλαίου 9 όπου παρατηρούμε ελάχιστες αποκλίσεις σε σχέση με τους επαγγελματικού τύπου αισθητήρες. Επίσης παρατηρείται ότι και ο αλγόριθμος Zambretti παρά το σύντομο χρονικό διάστημα που δοκιμάστηκε, σε ποσοστό 89% παρείχε επιτυχή πρόγνωση.

Η τεχνολογία που χρησιμοποιήθηκε για την κατασκευή του εν λόγω σταθμού είναι:

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

Η διαχείριση μνήμης αποδείχτηκε σωτήρια καθώς ο μετεωρολογικός σταθμός, λόγω της περιορισμένης προσωρινής μνήμης SRAM, υπολειτουργούσε χωρίς να παρουσιάζεται κάποιο πρόβλημα κατά την 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



*Τριανταφύλλου Γεώργιος,
‘Ανάπτυξη Αυτόνομου Σταθμού Καταγραφής &
Πρόβλεψης Μετεωρολογικών Φαινομένων*

εξοικειωμένα με την ρομποτική. Η δημιουργία ενός αυτόνομου μετεωρολογικού σταθμού είναι μόνο η κορυφή του παγόβουνου καθώς οι δυνατότητες του εξελίσσονται με τέτοιο τρόπο, που το μόνο πλέον που μπορεί να το περιορίσει είναι η ανθρώπινη φαντασία.



Βιβλιογραφία – Αναφορές

- [1]. World Meteorological Organization, “Guide to Meteorological Instruments and Methods of Observation”, 7th edition, p.385, 2008
- [2]. Wikipedia, "Weather Forecasting",
http://en.wikipedia.org/wiki/Weather_forecasting , (Ημ. Προσπέλασης 17/6/2013)
- [3]. Wikipedia Θερμοκρασία Ατμόσφαιρας,
http://el.wikipedia.org/wiki/%CE%98%CE%B5%CF%81%CE%BC%CE%BF%CE%BA%CF%81%CE%B1%CF%83%CE%AF%CE%B1_%CE%B1%CF%84%CE%BC%CF%8C%CF%83%CF%86%CE%B1%CE%B9%CF%81%CE%B1%CF%82 , ,
(Ημ. Προσπέλασης 17/6/2013)
- [4]. Koninklijk Nederlands Meteorologisch Instituut, "Handbook for the Meteorological Observation", p.44, September 2000
- [5]. World Meteorological Organization, "Procedures And Formats For Exchange Of MONITORING RESULTS",
<https://www.wmo.int/pages/prog/www/DPS/Monitoring-home/mon-procedures.htm> ,
(ημ προσπέλασης 12/6/2013)
- [6]. World Meteorological Organization, “Guide to Meteorological Instruments and Methods of Observation”, 7th edition, p.65, 2008
- [7]. Εθνικό Αστεροσκοπείο Αθηνών,
http://www.meteo.gr/pdf/deiktis_thermotitas.pdf , (Ημ Προσπέλασης 22/6/2013)
- [8]. Wikipedia, http://en.wikipedia.org/wiki/Heat_index , (Ημ Προσπέλασης 22/6/2013)
- [9]. Arduino Επίσημη Ιστοσελίδα, www.arduino.cc (Ημ προσπέλασης 12/6/2012)
- [10]. Ardupilot, <http://www.diydrones.com/notes/ArduPilot> , (Ημ προσπέλασης 15/6/2012)
- [11]. Καραπιέρης Λ., «Ναυτική Μετεωρολογία», Ευγενίδιο Ίδρυμα, σ. 102, Αθήνα, 1997
- [12]. Καραπιέρης Λ., «Ναυτική Μετεωρολογία», Ευγενίδιο Ίδρυμα, σ. 108, Αθήνα, 1997
- [13]. Datasheet DHT-22,
<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Weather/RHT03.pdf> , (Ημ προσπέλασης 5/6/2013)
- [14]. World Meteorological Organization, “Guide to Meteorological Instruments and Methods of Observation”, 7th edition, p.44, 2008



- [15]. World Meteorological Organization, “Guide to Meteorological Instruments and Methods of Observation”, 7th edition, p.92, 2008
- [16]. Datasheet BMP-085,
http://www.adafruit.com/datasheets/BMP085_DataSheet_Rev.1.0_01July2008.pdf ,
(Ημ προσπέλασης 7/6/2013)
- [17]. World Meteorological Organization, “Guide to Meteorological Instruments and Methods of Observation”, 7th edition, pp.65-66, 2008
- [18]. "Atmospheric Pressure and Altimeters",
<http://aviationweather.gov/general/pubs/front/docs/oct-02.pdf> (Ημ προσπέλασης 12/6/2013)
- [19]. Raymond M Sager, "The Sager weathercaster;: A scientific instrument for accurate prediction of the weather", October House, 1969
- [20]. Meteormetrics, "Local Forecasting based on the Zambretti Forecaster",
<http://www.meteormetrics.com/zambretti.htm> , (Ημ. προσπέλασης 2/6/2013)
- [21]. Wind Sensor Datasheet,
<http://www.sparkfun.com/datasheets/Sensors/Weather/Weather%20Sensor%20Assembled.pdf> , (Ημ. προσπέλασης: 12/6/2013)
- [22]. Arduino AnalogRead, <http://arduino.cc/en/Tutorial/AnalogInputPins> , (Ημ προσπέλασης 13/6/2013)
- [23]. <http://www.marulaberry.co.za/index.php/tutorials/code/arduino-adc/> , (Ημ προσπέλασης 14/6/2013)
- [24]. "Making accurate ADC readings on the Arduino",
<http://hacking.majenko.co.uk/making-accurate-adc-readings-on-arduino> , (Ημ προσπέλασης 9/7/2013)
- [25]. Koninklijk Nederlands Meteorologisch Instituut, "Handbook for the Meteorological Observation", pp.(30,47,62), September 2000
- [26]. Koninklijk Nederlands Meteorologisch Instituut, "Handbook for the Meteorological Observation", pp.44-45, September 2000
- [27]. Koninklijk Nederlands Meteorologisch Instituut, "Handbook for the Meteorological Observation", pp.(31), September 2000
- [28]. Wikipedia, http://en.wikipedia.org/wiki/Power_over_Ethernet , (Ημ Προσπέλασης 2/6/2013)
- [29]. Wikipedia, http://en.wikipedia.org/wiki/Network_Time_Protocol , (Ημ Προσπέλασης 11/6/2013)
- [30]. Πληροφορίες για το EthernetShield,
<http://arduino.cc/en/Main/ArduinoEthernetShield> , (Ημ. προσπέλασης 1/7/2013)
- [31]. Πληροφορίες για την βιβλιοθήκη DHTlib,
<https://github.com/ringerc/Arduino-DHT22> , (Ημ. Προσπέλασης 2/6/2013)



- [32]. Πληροφορίες για την βιβλιοθήκη BMP-085, <http://learn.adafruit.com/bmp085/overview> , (Ημ. Προσπέλασης 2/6/2013)
- [33]. Βιβλιοθήκη dygraph, <http://www.chartjs.org> , (Ημ προσπέλασης 2/7/2013)
- [34]. Clock Library, http://www.blueleafsoftware.com/Resources/EmbeddedSand/Clock_library , (Ημ προσπέλασης 3/7/2013)
- [35]. Arduino, "LiquidCrystal Library", <http://arduino.cc/en/Tutorial/LiquidCrystal>, (Ημ προσπέλασης 3/7/2013)
- [36]. Arduino, "SD Library", <http://arduino.cc/de/Reference/SD>, (Ημ προσπέλασης 3/7/2013)
- [37]. Μετεωρολογικός Σταθμός NTUA, <http://openmeteo.org/db/timeseries/d/3813/>, (ημ προσπέλασης: 10/7/13)
- [38]. Margolis Michael, “Arduino Cookbook”, O’Reilly, Sebastopol, p.310, 2011
- [39]. <http://startingelectronics.com/tutorials/arduino/ethernet-shield-web-server-tutorial/SD-card-AJAX-web-server/> , (ημ. Προσπέλασης: 01/05/2013)
- [40]. ISO 8601, https://en.wikipedia.org/wiki/ISO_8601 , (Ημ προσπ. 3/7/2013)

Βιβλιογραφία

- [1]. Margolis Michael, “Arduino Cookbook”,(O’Reilly,Sebastopol,2011)
- [2]. R.E. Munn, Compendium of Meteorology, (WMO, Volume II, 1979)
- [3]. Riley Mike, «Programming your home with Arduino, Android, and your Computer» (The Pragmatic Programmers ,2012)
- [4]. Massimo Banzi, «Getting Start with Arduino»,(O’Reilly, Sebastopol, 2009)
- [5]. World Meteorological Organization, “Guide to Meteorological Instruments and Methods of Observation”, (7th edition, 2008)
- [6]. Vasquez Tim, “International Weather Watchers Observer Handbook”, (1998)
- [7]. Wilcher Don, “Learns Electronics with Arduino”, (Apress, New York, 2012)
- [8]. Warren John-David, “Arduino Robotics”, (Apress, New York, 2012)
- [9]. Καλαμαράς Νικόλαος, “Μετεωρολογικά Όργανα”, (ΕΜΥ, Αθήνα,2010)
- [10]. Καραπιπέρης Λ., «Μετεωρολογία», (Ευγενίδιο Ίδρυμα, Αθήνα, 1997)



ΠΑΡΑΡΤΗΜΑΤΑ

Παράρτημα Α – συνάρτηση διασύνδεσης κάρτας Ethernet Shield

```
#include <Ethernet.h> //δήλωση βιβλιοθήκης
//δήλωση υπολοίπων βιβλιοθηκών
...
//δήλωση διευθύνσεων MAC και IP
byte mac_addr[]={0xDE,0xAD,0xBE,0xEF,0xFE,0xED};
byte ip_addr[]={192,168,2,100};
test_host[]={192,168,2,1};//
//υπόλοιπες δηλώσεις
...
EthernetClient client;
//συνάρτηση setup()
void setup()
{
    Ethernet.begin(mac_addr, ip_addr);//έναρξη λειτουργίας
    client.connect(test_host, 80);//σύνδεση στο router για έλεγχο
    λειτουργίας δικτύου
    while (!client.connected()){ //όσο δεν έχει συνδεθεί στο δίκτυο
        Serial.println('error connect to client');
        LCD.print('sfalmadiktyog');//εκτύπωση στην οθόνη LCD
        delay(5000); αναμονή 5 sec
    }
    ... υπόλοιπος κώδικας.
}
}
```



Παράρτημα Β – συνάρτηση διασύνδεσης αισθητήρα DHT-22

```
#include<DHT22.h> //εισαγωγή βιβλιοθήκης
// υπόλοιπες βιβλιοθήκες
...
// αρχικοποίηση αισθητήρα
DHT22 senson(7)// ορίζουμε την αν. είσοδο 7.

//συνάρτηση ανάγνωσης αισθητήρα
voidreadHumAndTemp(void){
  DHT22_ERROR_t errorCode; //ανάγνωση σφαλμάτων αισθητήρα

  // ανάννωση κάθε 2 δευτερόλεπτα
  delay(2000);

  errorCode = myDHT22.readData(); //ανάγνωση σφαλμάτων
  //για κάθε περίπτωση σφάλματος κάνε τα ακόλουθα
  switch(errorCode)
  {
    //αν δεν εχει σφάλματα εκτύπωσε τα δεδομένα
    case DHT_ERROR_NONE:
      LCD.print("Got Data ");
      LCD.print(myDHT22.getTemperatureC());
      LCD.print("C ");
      LCD.print(myDHT22.getHumidity());
      LCD.println("%");

    break;
    //περίπτωσηλάθουςbit ισοτιμίας
    case DHT_ERROR_CHECKSUM:
      LCD.print("check sum error ");
      break;
    case DHT_BUS_HUNG:
      Serial.println("BUS Hung ");
      break;
    case DHT_ERROR_NOT_PRESENT:
      Serial.println("Not Present ");
      break;
    case DHT_ERROR_ACK_TOO_LONG:
      Serial.println("ACK time out ");
      break;
    case DHT_ERROR_SYNC_TIMEOUT:
      Serial.println("Sync Timeout ");
      break;
    case DHT_ERROR_DATA_TIMEOUT:
      Serial.println("Data Timeout ");
      break;
    case DHT_ERROR_TOOQUICK:
      Serial.println("Polled to quick ");
      break;
  }
}
```



Παράρτημα Γ – συνάρτηση Διασύνδεσης αισθητήρα BMP-085

```
#include "Adafruit_BMP085.h"//εισαγωγή βιβλιοθήκης
....
Adafruit_BMP085 pSensor; //αρχικοποίηση αισθητήρα
....
void setup(){
....
while(!pSensor.begin()){//αν δεν ξεκινήσει η ανάγνωση
  LCD.print('sfalma aisthitira piesis');//τύπωσε σφάλμα
  Delay(5000)//αναμονή 5 sec
}
pSensor.begin();//έναρξη ανάγνωσης αισθητήρα

....
}
//συνάρτηση ανάγνωσης και εκτύπωσης δεδομένων
readPSensor(){
LCD.print("Temperature = ");
LCD.print(bmp.readTemperature());
LCD.println(" *C");

LCD.print("Pressure = ");
LCD.print(bmp.readPressure());
LCD.println(" hPa");

Serial.println();
}
```



Παράρτημα Δ – συνάρτηση διασύνδεσης αισθητήρα διεύθυνσης ανέμου.

```
int getWindDirectionDegrees() {  
    unsigned int adc;  
    adc = analogRead(WIND_DIRECTION_PIN); // get the current reading from the  
    sensor  
  
    if (adc > 80 && adc < 104) return (90); //E  
    if (adc > 172 && adc < 196) return (135); //SE  
    if (adc > 274 && adc < 298) return (180); //S  
    if (adc > 488 && adc < 472) return (45); //NE  
    if (adc > 618 && adc < 642) return (225); //SW  
    if (adc > 774 && adc < 798) return (0); //N  
    if (adc > 966 && adc < 990) return (315); //NW  
    if (adc > 933 && adc < 957) return (270); //W  
    return (-1); // error, disconnected?  
}
```




Παράρτημα Ε – Συνάρτηση καταγραφής δειγμάτων Ατμ. πίεσης

```
int mesiTimiPiesis[5]; //πίνακας στον οποίο αποθηκεύεται η μέση τιμή. // Σε
αυτόν τον πίνακα αποθηκεύεται η ατμοσφαιρική πίεση.

loop(){
    ...
    Int hPa=bmp.readPressure()/100;//λήψη τιμής ατμ. Πίεσης και
μετατροπής αποPa σε hPa
    katagrafiDeigmatonPiesis(hPa); //κλήση συνάρτησης καταγραφής
    ...υπόλοιπος κώδικας...
}
void katagrafiDeigmatonPiesis(intpressu){
int seconds = second();
//αν τα δευτερόλεπτα του ρολογιού είναι 12, 24, 36, 48 , 00 τότε
καταγράφεται η τρέχουσα τιμή στον πίνακα
//επειδή θέλουμε να υπολογίσει τα 00 του επόμενου λεπτού. Για παράδειγμα η
καταγραφή είναι μεταξύ των
//12:12, 12:24, 12:36, 12:48, 13:00. Άρα όταν το ρολόι δείξει 0 θα πρέπει
να το προσπεράσει ενώ όταν θα δείξει 59 θα περιμένει 1 δευτερόλεπτο ( θα
φτάσει δηλαδή 0) και θα γίνει η καταγραφή.
if (seconds==0) {
    delay(1000);
    seconds=1;
} else if (seconds==59) {
    delay(1000);
    seconds=0;
}
switch (seconds){
case 12:
    mesiTimiPiesis[0] = pressu;
    Serial.print("katagrafi lou Deigmatos:");
    Serial.println(pressu);
    break;
case 24:
    mesiTimiPiesis[1] = pressu;
    Serial.print("katagrafi 2ou Deigmatos:");
    Serial.println(pressu);
    break;
case 36:
    mesiTimiPiesis[2] = pressu;
    Serial.print("katagrafi 3ou Deigmatos:");
    Serial.println(pressu);
    break;
case 48:
    mesiTimiPiesis[3] = pressu;
    Serial.print("katagrafi 4ou Deigmatos:");
    Serial.println(pressu);
    break;
case 0:
    mesiTimiPiesis[4] = pressu;
    Serial.print("katagrafi 5ou Deigmatos:");
    Serial.println(pressu);
    kataxwrisiDeigmatosAnaLepto(mode(mesiTimiPiesis,5)); //όταν φτάσει στο
τελευταίο δείγμα ξεκινάει η διαδικασία καταχώρησης της μέτρησης του
λεπτού.
    break;
}
}
```



Παράρτημα ΣΤ - Συνάρτηση Υπολογισμού Βαρομετρικής τάσης

```
MesiTimiOrieasPiesis[60];

void kataxwrisiDeigmatosAnaLepto(inttimiPiesis){
    int minutes = minute();
    // symfwna me ton algoritmo katagrafis gia ton ypologismo tou mesou
    orou tis oriaias barometrikis piesis
    // h katagrafi ginete anamesa sta 50 kai oxi parapanw apo to 58
    leptou tis wras. edw orizw oti
    // h katagrafi tha oloklironetai sto 55 leptou tis wras
    // an den einaisto 55 leptou tote kategrapse tin mesitimi ton
    proigoumenwn 5 metrisewn
    // etsi apothikeuetai ston pinaka stin uesi px 8 i mesi timi sta 8
    leptou tis wras klp.
    if (minutes !=55){
        MesiTimiOrieasPiesis[minutes]=timiPiesis;
    }
    // an einai tote kategrapse tin timikai bale kataxorise ton meso
    oro tis wras ston pinaka gia ton
    // ypologismo tis barometrikis tasis
    else if (minutes==55){
        MesiTimiOrieasPiesis[55]=timiPiesis;
        float pressure =mode(MesiTimiOrieasPiesis,60);
        katagrafiBarometrikisTasis(pressure); //καταγραφή ωριαίας μέσης τιμής
        ατμοσφαιρικής πίεσης για τον υπολογισμό της βαρομετρικής τάσης.
    }
    Serial.print("katagrafiDeigmatos ");
    printLCDStatus("katagrafiDeigmatos:");

    Serial.print(minutes);
    Serial.print(" leptou: ");

    Serial.println(mode(mesiTimiPiesis,5));
    printLCDStatus((char*)mode(mesiTimiPiesis,5));
}
}
```

Παράρτημα Ζ – Συνάρτηση αποθήκευσης βαρομετρικής τάσης

```
void katagrafiBarometrikisTasis(int pressure){
    //Μετακίνηση των προηγούμενων τιμών στις n+1 θέσεις
    for(int i=3;i=1;i++){
        hourPressureTrend[i]=hourPressureTrend[i-1];
    }
    hourPressureTrend[0]=pressure; // καταχώρηση της τρέχουσας
    τιμής τ0
}
}
```



Παράρτημα Η – Συνάρτηση ανάγνωσης βαρομετρικής τάσης

```
Int getTrend(){
    if ((hourPressureTrend[3]-hourPressureTrend[0])==0){
        return 0;
    } else if((hourPressureTrend[3] - hourPressureTrend[0]) > 0){
        return 1;
    } else { //
        return 2;
    }
}
```

Παράρτημα Θ – Συνάρτηση αλγορίθμου Ζαμπρέτι

```
char* zambrettiAlgorithm(int trend){
    float minPressure=922.9; //ελάχιστη τιμή περιοχής
    float maxPressure=1037.5; //μεγιστη τιμή περιοχής
    intwindDirection = getWindDirectionDegrees(1);//διευθ. Ανέμου
    float hPaz = bmp.readPressure()/100; //μετατροπή σεhPa
    char* forecast[] = {"Settled fine", "Fine weather", "Becoming
fine", "Fine, becoming less settled", "Fine, possible showers",
"Fairly fine, improving", "Fairly fine, possible showers early",
"Fairly fine, showery later", "Showery early, improving",
"Changeable, mending", "Fairly fine, showers likely", "Rather
unsettled clearing later", "Unsettled, probably improving",
"Showery, bright intervals", "Showery, becoming less settled",
"Changeable, some rain", "Unsettled, short fine intervals",
"Unsettled, rain later", "Unsettled, some rain", "Mostly very
unsettled", "Occasional rain, worsening", "Rain at times, very
unsettled", "Rain at frequent intervals", "Rain, very unsettled",
"Stormy, may improve", "Stormy, much rain"};
// Συσχετισμός με τον δίσκο A του Zambretti forecaster. Ανάλογα
με την βαρομετρική τάση επιλέγεται και το ανάλογο αποτέλεσμα
    intrise_options[] =
    {25,25,25,24,24,19,16,12,11,9,8,6,5,2,1,1,0,0,0,0,0} ;
    intsteady_options[] =
    {25,25,25,25,25,25,23,23,22,18,15,13,10,4,1,1,0,0,0,0,0} ;
    intfall_options[] =
    {25,25,25,25,25,25,25,25,23,23,21,20,17,14,7,3,1,1,1,0,0} ;
    float range = maxPressure - minPressure;
    float hPa = hPaz;
    float constan = (range / 22);
    bool summer;
    int m=month(); //an einai kalokairi
    if (m>=4 && m<=9)
        summer=true;
    else
        summer=false;
    switch (windDirection) {
    case 0 : //Βορράς
        hPa += (6.0 / 100.0) * range ;
    break;
        case 45 : //Βορειο Ανατολικός
            hPa += (5.0/100.0) * range;
```



```
break;
    case 90 : //Ανατολικός
        hPa -= (0.5 / 100.0) * range ;
break;
    case 135 : //ΝοτιόΑνατολικός
        hPa -= (5.0 / 100.0) * range ;
break;
    case 180 : //Νότος
        hPa -= (12.0 / 100.0) * range ;
break;
    case 225 : //ΝότιοΔυτικός
        hPa -= (6.0 / 100.0) * range ;
break;
    case 270 : //Δυτικός
        hPa -= (3.0 / 100.0) * range ;
break;
    case 315 : //ΒορείοΔυτικός
        hPa += (1.5 / 100.0) * range ;
break;
    }
    if (summer == 1) { // Αν είναι καλοκαίρι
        if (trend == 1) { // και αυξάνεται η τάση
            hPa += (7.0 / 100.0) * range;
        } else if (trend == 2) { // falling
            hPa -= (7.0 / 100.0) * range;
        }
    }

    if (hPa==maxPressure)
hPa=maxPressure-1;
int option= (int)((hPa - minPressure)/constan);
if (option < 0) {
option=0;
}else if (option > 21) {
option=21;
}
char* s;
switch (trend){
case 1: // αύξηση βαρομετρικής τάσης
s = forecast[rise_options[option]];
break;
case 2: // μείωση βαρομετρικής τάσης
s = forecast[fall_options[option]];
break;
case 0: // αμετάβλητος
s = forecast[steady_options[option]];
break;
default:
s=" ";
}
returns;
}
```



Παράρτημα Ι – συνάρτηση καταχώρησης στην κάρτα SD

```
File myFile;

void saveData2Disk(){
    //όνομα αρχείου προς αποθήκευση

    char filenameZ[20];
    sprintf(filenameZ, "%02d%02d%02d.csv",year(), month(), day());
    char* filename=filenameZ;

    char dateZ[20];
    sprintf(dateZ, "%02d/%02d/%02d %02d:%02d:00",year(), month(),
    day(),hour(), minute());
    char* date=dateZ;
    Serial.println(filename);

    File myFile;
    //εάν το αρχείο δεν υπάρχει δημιουργήσει το
    if(!SD.exists(filename)){
        myFile = SD.open(filename, FILE_WRITE);
        //δημιουργήσει τις κεφαλίδες του αρχείου csv
        myFile.println("Date,Pressure,Temperature,Humidity,WindDirecti
        on");
        myFile.close();
    }
    myFile = SD.open(filename, FILE_WRITE);
    // if the file opened okay, write to it:
    if (myFile) {
        //η δομή του αρχείου είναι
        //εξεεμμη ωω:λλ,πιεση,θερμοκρασια,υγρασια,ανεμος
        //
        myFile.print(date);
        myFile.print(",");
        myFile.print(wd.pressure);
        myFile.print(",");
        myFile.print(wd.temperature);
        myFile.print(",");
        myFile.print(wd.humidity);
        myFile.print(",");
        myFile.println(wd.windDirection);

        // close the file:
        myFile.close();
    } else {
    }
}
```



Παράρτημα ΙΑ – Συνάρτηση λειτουργίας εξυπηρετητή διαδικτύου.

```
void createWebPage(){
EthernetClient client = server.available();
if (client) { // αν ο υπαρχει συνδεση
boolean currentLineIsBlank = true;
while (client.connected()) {
if (client.available()) { // αν τα δεδομενα του πελατη διαβαζονται
charc = client.read(); // διαβασε το 1 byte (χαρακτηρα) του πελατη
HTTP_req += c; // αποθηκευσε το HTTP request 1 char at a time
// αν η τελευταια γραμμη του αιτηματος ειναι κενη και τελειωνει με
\n
// να απαντησεις εφοσον παρεις την κενη γραμμη
if (c == '\n' && currentLineIsBlank) {
// απαντησε ενανstandard http response header
client.println("HTTP/1.1 200 OK");
// ελεγχοςαντισεισερχομενοαιτημαειναιxmlηαρχειοhtm
if (HTTP_req.indexOf("ajax_inputs") > -1) {
// αποστολη της υπολοιπης HTTP κεφαλιδας
client.println("Content-Type: text/xml");
client.println("Connection: keep-alive");
client.println();
// αποστολη του XML το οποιο περιεχει τα
δεδομενα για προβολη στην ιστοσελιδα
XML_response(client);
}
else { // αιτημα για προβολη ιστοσελιδας
// Αιτημα αρχικης σελιδα
if ((HTTP_req.indexOf("GET / ") > -1)
|| (HTTP_req.indexOf("GET /index.htm") > -1)) {
client.println("Content-Type: text/html");
client.println("Connection: keep-alive");
client.println();
webFile = SD.open("newide.htm"); // ανοιγμα αρχειου απο καρτα
sd
}
//αιτημα προβολης στατιστικων
else if (HTTP_req.indexOf("GET /stats.htm") > -1) {
client.println("Content-Type: text/html");
client.println("Connection: keep-alive");
client.println();
webFile = SD.open("stats.htm"); // ανοιγμα αρχειου απο καρτα
sd
}
//αιτημα προβολης αρχειου temp.csv
else if (HTTP_req.indexOf("GET /temp.csv") > -1) {

webFile = SD.open("temp.csv");
if (webFile) {
client.println("HTTP/1.1 200 OK");
client.println();}
}
else if (HTTP_req.indexOf("GET /pres.csv") > -1) {

webFile = SD.open("pres.csv");
if (webFile) {
```



```
client.println("HTTP/1.1 200 OK");
client.println();}
else if (HTTP_req.indexOf("GET /humi.csv") > -1) {

webFile = SD.open("humi.csv");
if (webFile) {
client.println("HTTP/1.1 200 OK");
client.println();}
//αιτημα προβολης αρχειου csv τρεχουσας
ημερας
else if (HTTP_req.indexOf("GET /today.csv") > -1) {
//οι μετρησεις αποθηκευονται σε αρχειο csv με την μορφη εεεεεμμηη.csv
για να μπορεί καθε ημερα το αρχειο
//να διαβαζεται απο το javascript
"εξαπατω" το αιτημα. οταν μου ζηταει το today.csv εγω αποστειλω το
αρχειο με την
//τρεχουσα ημερομηνια
char filenameZ[20];
sprintf(filenameZ, "%02d%02d%02d.csv",year(), month(), day());
char* filename=filenameZ;
webFile = SD.open(filename);
if (webFile) {
client.println("HTTP/1.1 200 OK");
client.println();}

if (webFile) {
while(webFile.available()) {
client.write(webFile.read()); // ανοιγμα αρχειου απο καρτα sd
}
webFile.close();
}
}
HTTP_req = ""; // finished with
request, empty string
break;
}
// every line of text received from the client ends
with \r\n
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
currentLineIsBlank = false;
}
}
}
delay(1); // αναμονη 1 ms ωστε να δημιουργηθει η σελίδα
client.stop(); // κλεισιμο σύνδεσης
}
```

ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΣΧΟΛΗ ΒΙΟΤΕΧΝΙΚΩΝ ΕΡΕΥΝΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ



HELLENIC OPEN UNIVERSITY
SCHOOL OF SCIENCES AND TECHNOLOGY

WWW.EAP.GR