



ΕΘΝΙΚΟ ΚΑΙ  
ΚΑΠΟΔΙΣΤΡΙΑΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΓΕΩΛΟΓΙΑΣ ΚΑΙ  
ΓΕΩΠΕΡΙΒΑΛΛΟΝΤΟΣ

ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ  
ΣΕΡΡΩΝ

ΤΜΗΜΑ  
ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ  
& ΤΟΠΟΓΡΑΦΙΑΣ



ΔΙ-ΙΔΡΥΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
«ΠΡΟΛΗΨΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΦΥΣΙΚΩΝ ΚΑΤΑΣΤΡΟΦΩΝ»

## ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ ΕΙΔΙΚΕΥΣΗΣ

**Αναστασάκης Κωνσταντίνος Α.Μ. 21314**

**Μελέτη και υλοποίηση δικτύου καταγραφής μετεωρολογικών  
παραμέτρων χαμηλού κόστους για την πρόληψη και αντιμετώπιση  
των δασικών πυρκαγιών.**

### ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ

Δρ. Παναγιώτης Νάστος  
Δρ. Γαβριήλ Ξανθόπουλος  
Δρ. Ευστάθιος Χατζηευσθυμιάδης

Καθηγητής ΕΚΠΑ (Επιβλέπων)  
Ερευνητής Γ ΕΘΙΑΓΕ  
Αναπλ. Καθηγητής ΕΚΠΑ

**ΑΘΗΝΑ**  
**ΙΟΥΛΙΟΣ 2016**



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ



ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ  
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



**Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ**  
Επιχειρησιακό Πρόγραμμα  
Εκπαίδευσης και Αρχικής  
Επαγγελματικής Κατάρτισης



NATIONAL AND  
KAPODISTRIAN UNIVERSITY  
OF ATHENS

SCHOOL OF SCIENCES  
FACULTY OF GEOLOGY AND  
GEOENVIRONMENT

TECHNOLOGICAL  
EDUCATIONAL INSTITUTE  
OF SERRES

FACULTY OF  
GEOINFORMATICS AND  
SURVEYING ENGINEERING



POSTGRADUATE PROGRAM  
"PREVENTION AND MANAGEMENT OF NATURAL DISASTERS"

MASTER THESIS

**Anastasakis Konstantinos R.N. 21314**

**Design and implementation of low cost meteorological recording  
network system for the prevention and management of forest fires.**

THREE-MEMBER COMMITTEE

Dr. Panagiotis Nastos	Professor (supervisor)
Dr. Gavriil Xanthopoulos	Assoc. Researcher
Dr. Efstathios Chatziefthymiadis	Assoc. Professor

ATHENS  
JULY 2016



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ

ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ  
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ  
Επιχειρησιακό Πρόγραμμα  
Εκπαίδευσης και Αρχικής  
Επαγγελματικής Κατάρτισης



## Δήλωση

Δηλώνω ότι, το σύνολο του τεκμηρίου αποτελεί πρωτότυπο έργο παραχθέν από εμένα, εκτός εάν γίνεται ειδική αναφορά στο έργο άλλων και δεν παραβιάζει τα δικαιώματα άλλου δημιουργού καθ' οιονδήποτε τρόπο. Το περιεχόμενο αυτής της διατριβής είναι πρωτότυπο και δεν έχει υποβληθεί στο σύνολό της ή εν μέρει προς εξέταση, για οποιοδήποτε άλλο πτυχίο ή άλλο δίπλωμα, σε αυτό ή οποιοδήποτε άλλο πανεπιστήμιο. Αυτή η διατριβή είναι δικό μου έργο και δεν περιέχει τίποτα το οποίο να είναι το αποτέλεσμα εργασίας που πραγματοποιήθηκε σε συνεργασία με άλλους, εκτός αν διευκρινίζεται στο κείμενο και στις ευχαριστίες.

Η διατριβή περιέχει λιγότερες από 16533 λέξεις, συμπεριλαμβανομένων των προσαρτημάτων, βιβλιογραφία, υποσημειώσεις, πίνακες και εξισώσεις και έχει λιγότερα από 41 στοιχεία.

## Ευχαριστίες

Η παρούσα διπλωματική διατριβή εκπονήθηκε από τον μεταπτυχιακό φοιτητή Κωνσταντίνο Αναστασάκη του Δι-Ιδρυματικού Προγράμματος Μεταπτυχιακών Σπουδών «Πρόληψη και Διαχείριση Φυσικών Καταστροφών» του Εθνικού Και Καποδιστριακού Πανεπιστήμιου Αθηνών, σχολής Θετικών Επιστημών, τμήματος Γεωλογίας και Γεωπεριβάλλοντος.

*Θα ήθελα εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή της εργασίας μου κ. Παναγιώτη Νάστο, για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου αυτή την εργασία, για την καθοδήγησή του καθ' όλη τη διάρκειά της και κυρίως για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα ενδιαφέρον αντικείμενο. Επίσης, θα ήθελα να ευχαριστήσω τον κ. Δρ. Γαβριήλ Ξανθόπουλο για την συμμετοχή του στην τριμελή επιτροπή, για το βιβλιογραφικό υλικό που μου παρείχε και τις απολύτως απαραίτητες διορθώσεις του κειμένου της διπλωματικής.*

## **Περίληψη**

Στη παρούσα διπλωματική διατριβή, γίνεται μια προσπάθεια ανάδειξης εναλλακτικών οικονομικών τρόπων συλλογής, επεξεργασίας περιβαλλοντικών παραμέτρων και αξιοποίησης αυτών για μελέτες πρόληψης και αντιμετώπισης φυσικών καταστροφών. Χρησιμοποιηθήκαν ηλεκτρονικά κυκλώματα ευρείας χρήσης και υλοποιήθηκε με αυτά δίκτυο καταγραφής μετεωρολογικών παραμέτρων χαμηλού κόστους για την πρόληψη και αντιμετώπιση των δασικών πυρκαγιών.

Λέξεις Κλειδιά: δίκτυο καταγραφής, μετεωρολογικοί παράμετροι, δασικές πυρκαγιές.

## **Abstract**

This dissertation, is an attempt to highlight alternative economic methods of collection and processing environmental parameters for natural disaster prevention and response studies. Used low cost electronic circuits widely used and implemented with these meteorological recording systems for the prevention and management of forest fires.

Keywords: recording network, meteorological parameters, forest fires.

## Πίνακας περιεχομένων

Ευρετήριο Εικόνων.....	viii
Ευρετήριο Πινάκων.....	ix
1 Εισαγωγή.....	1
1.1 Δασικές πυρκαγιές.....	1
1.2 Αντικείμενο διπλωματικής.....	1
1.2.1 Συνεισφορά.....	2
1.3 Οργάνωση κειμένου.....	2
2 Σχετικές εργασίες.....	3
2.1 Firementor.....	3
2.2 Forest Cities.....	4
3 Θεωρητικό υπόβαθρο.....	7
3.1 Εκτίμηση κινδύνου δασικών πυρκαγιών.....	7
3.2 Τεχνολογίες μετρήσεων περιβαλλοντικών παραμέτρων.....	8
3.3 Ελεύθερο λογισμικό, λογισμικό ανοιχτού κώδικα.....	12
3.3.1 Άδειες ελεύθερου λογισμικού.....	14
3.4 Κόστος συστημάτων καταγραφής.....	15
4 Υλοποίηση.....	16
4.1 Επιλογή υλικών.....	16
4.1.1 Επιλογή αισθητήρων.....	16
4.1.2 Επιλογή ηλεκτρονικών.....	26
4.2 Κατασκευή αυτόνομων συστημάτων καταγραφής μετεωρολογικών παραμέτρων.....	28
4.2.1 Κατασκευή αυτόνομου συστήματος καταγραφής για τετρακόπτερο (Drone).....	34
4.3 Κατασκευή κόμβου καταγραφής και απεικόνισης μετεωρολογικών παραμέτρων πραγματικού χρόνου.....	35
5 Εφαρμογή.....	38
5.1 Δοκιμαστικές μετρήσεις.....	38
5.2 Εφαρμογή στο πεδίο.....	41
5.2.1 Αποτελέσματα μετρήσεων πεδίου.....	42
5.2.2 Μετρήσεις τετρακοπτερου (Drone).....	44
5.2.3 Σύνοψη συμπερασμάτων δοκιμών.....	45
6 Τεχνικές λεπτομέρειες.....	46
6.1 Πλατφόρμες και προγραμματιστικά εργαλεία.....	46
6.1.1 Εγκατάσταση προγραμμάτων.....	46
6.2 Διαχείριση δεδομένων πραγματικού χρόνου.....	48
7 Επίλογος.....	49
7.1 Σύνοψη και συμπεράσματα.....	49
7.2 Μελλοντικές επεκτάσεις.....	50
Βιβλιογραφία.....	51
Παράρτημα.....	53

## Ευρετήριο Εικόνων

Σχήμα 2.1: Το σύστημα GIS στο Δήμο Ηλιούπολης, που απεικονίζει την επικινδυνότητα περιοχών σε πραγματικό χρόνο για πρόκληση πυρκαγιάς. (Λυβιάκης 2013).....	5
Σχήμα 3.1: Πινακίδα Κινδύνου Πυρκαγιάς. ( <a href="http://smokeyzone.com/fire-danger-signs/">http://smokeyzone.com/fire-danger-signs/</a> ).....	8
Σχήμα 3.2: Ημερήσιο δελτίο πρόβλεψης κινδύνου πυρκαγιάς. ( <a href="http://www.gscp.gr/">http://www.gscp.gr/</a> ).....	8
Σχήμα 3.3: Αισθητήρες θερμοκρασίας DS18B20.....	10
Σχήμα 3.4: Υγρόμετρα, AM2320 (αριστερά), HS1101 (δεξιά).....	10
Σχήμα 3.5: Βαρόμετρα, BMP280 (αριστερά), BME280 (δεξιά).....	10
Σχήμα 3.6: Κυπελλοφόρο ανεμόμετρο (EAA).....	12
Σχήμα 3.7: Το WindObserver 75 της Gill Instruments.....	12
Σχήμα 3.8: Το βροχόμετρο AEPG της Belfort Instrument.....	12
Σχήμα 4.1: Σύγκριση των διαφόρων ψηφιακών αισθητήρων. (Kandrsmith 2016).....	17
Σχήμα 4.2: Γραφικές παραστάσεις από τις τιμές της υγρασίας από κάθε αισθητήρα συναρτήσει της πραγματικής υγρασίας. (Kandrsmith 2016).....	20
Σχήμα 4.3: Γραφικές παραστάσεις που δείχνουν την μεταβολή των τιμών υγρασίας από γνωστές τιμές αναφοράς. (Kandrsmith 2016).....	20
Σχήμα 4.4: Γραφήματα που δείχνουν την απόκλιση των μετρήσεων θερμοκρασίας του κάθε αισθητήρα από το μέσο όρο όλων. (Kandrsmith 2016).....	21
Σχήμα 4.5: Γραφήματα που δείχνουν την απόκλιση των μετρήσεων θερμοκρασίας του κάθε αισθητήρα από το μέσο όρο όλων. (Kandrsmith 2016).....	22
Σχήμα 4.6: Γραφήματα που δείχνουν την απόκλιση των μετρήσεων θερμοκρασίας του κάθε αισθητήρα σε συνάρτηση με την υγρασία.....	23
Σχήμα 4.7: Επιφάνειες που δείχνουν την απόκλιση του κάθε αισθητήρα από την πραγματική σχετική υγρασία ως συνάρτηση της θερμοκρασίας και της υγρασίας.(Kandrsmith 2016).....	24
Σχήμα 4.8: Υστέρηση του κάθε αισθητήρα.....	25
Σχήμα 4.9: Arduino Pro Mini 3.3v.....	27
Σχήμα 4.10: Mini DS3231(RTC).....	27
Σχήμα 4.11: Micro SD Module.....	27
Σχήμα 4.12: BME280 Module.....	27
Σχήμα 4.13: WiFi Module ESP8266 v1.....	27
Σχήμα 4.14: Micro USB 1A Lithium Battery Charger.....	27
Σχήμα 4.15: FTDI FT232RL.....	27
Σχήμα 4.16: Σχεδιάγραμμα Συνδεσμολογίας.....	29
Σχήμα 4.17: Κόμβος καταγραφής μετεωρολογικών παραμέτρων.....	29
Σχήμα 4.18: Τρόπος λειτουργίας προγράμματος.....	30
Σχήμα 4.19: Σχεδιάγραμμα Συνδεσμολογίας συστήματος για Drone.....	34
Σχήμα 4.20: Σχεδιάγραμμα Συνδεσμολογίας ασύρματου συστήματος πραγματικού χρόνου.....	35
Σχήμα 4.21: Απεικόνιση των δεδομένων στο διαδίκτυο.....	37
Σχήμα 5.1: Πρώτες δοκιμαστικές μετρήσεις θερμοκρασίας διάρκειας δυο ημερών.....	39



Σχήμα 5.2: Πρώτες δοκιμαστικές μετρήσεις υγρασίας διάρκειας δυο ημερών.....	39
Σχήμα 5.3: Δεύτερες δοκιμαστικές βελτιωμένες μετρήσεις θερμοκρασίας διάρκειας δυο ημερών.....	40
Σχήμα 5.4: Δεύτερες δοκιμαστικές βελτιωμένες μετρήσεις σχετικής υγρασίας διάρκειας δυο ημερών.....	40
Σχήμα 5.5: Περιοχή μελέτης με τα σημεία τοποθέτησης των σταθμών.....	41
Σχήμα 5.6: Μετρούμενη θερμοκρασία (°C) των σταθμών στη περιοχή μελέτης.....	42
Σχήμα 5.7: Μετρούμενη σχετική υγρασία (RH) των σταθμών στη περιοχή μελέτης.....	42
Σχήμα 5.8: Μέση ελάχιστη θερμοκρασία επτά ημερών.....	43
Σχήμα 5.9: Μέση μέγιστη θερμοκρασία επτά ημερών.....	43
Σχήμα 5.10: Μέση ελάχιστη σχετική υγρασία (%) επτά ημερών.....	43
Σχήμα 5.11: Μέση μέγιστη σχετική υγρασία (%) επτά ημερών.....	43
Σχήμα 5.12: Μετρήσεις τετρακόπτερου (Drone) σε σχέση με τον χρόνο.....	44
Σχήμα 5.13: Μετρήσεις τετρακόπτερου κατά την άνοδο σε σχέση με το υψόμετρο.....	45
Σχήμα 5.14: Μετρήσεις τετρακόπτερου κατά την κάθοδο σε σχέση με το υψόμετρο.....	45

## Ευρετήριο Πινάκων

Πίνακας 1: Αισθητήρες χαμηλού κόστους. (*Κόστος κατά προσέγγιση μέση λιανική τιμή αγοράς).....	17
Πίνακας 2: Πίνακας χαρακτηριστικών με βάση τις προδιαγραφές των εταιριών για την υγρασία (Kandrsmith 2016).....	19
Πίνακας 3: Πίνακας χαρακτηριστικών με βάση τις προδιαγραφές των εταιριών για την θερμοκρασία.....	21
Πίνακας 4: Υλικά που προμηθεύτηκαν για της ανάγκες κατασκευής των σταθμών.....	26
Πίνακας 5: Υλικά κατασκευής αυτόνομου συστήματος καταγραφής περιβαλλοντικών παραμέτρων.....	28
Πίνακας 6: Προτεινόμενες από τον κατασκευαστή ρυθμίσεις λειτουργίας.....	31
Πίνακας 7: Υλικά κατασκευής αυτόνομου συστήματος καταγραφής περιβαλλοντικών παραμέτρων για τετρακόπτερο (Drone).....	34
Πίνακας 8: Υλικά κατασκευής συστήματος πραγματικού χρόνου καταγραφής περιβαλλοντικών παραμέτρων.....	35
Πίνακας 9: Συντεταγμένες σταθμών.....	41
Πίνακας 10: Πλατφόρμες και προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν.....	46
Πίνακας 11: Οι ρυθμίσεις της βάσης του εξυπηρετητή data.sparkfun.com όπως διαμορφώθηκαν για τις ανάγκες της διπλωματικής.....	48

# 1

## Εισαγωγή

### 1.1 Δασικές πυρκαγιές

Οι δασικές πυρκαγιές αποτελούν την βασικότερη απειλή και ένα από τα σημαντικότερα προβλήματα για τα δάση της χώρας. Κάθε καλοκαίρι, οι πυρκαγιές προκαλούν πολύ μεγάλες καταστροφές στα δασικά οικοσυστήματα αφήνοντας στο πέρασμά τους στάχτες και αποκαΐδια. Υπάρχουν αρκετοί παράγοντες που συμβάλουν στην αύξηση του κινδύνου δασικών πυρκαγιών όπως είναι οι υψηλές θερμοκρασίες και η παρατεταμένη ξηρασία, οι δυνατοί άνεμοι, οι μεγάλες κλίσεις των δασικών εδαφών και η εύφλεκτη ξηροφυτική βλάστηση. Ένας άλλος επίσης σημαντικός παράγοντας είναι και η συσσώρευση βιομάζας, λόγω της εγκατάλειψης πολλών αγροτικών περιοχών. Η μαζική έξοδος των κατοίκων των πόλεων για αναψυχή στα δάση και ο αυξανόμενος τουρισμός, συμβάλουν επίσης σημαντικά στην αύξηση του κινδύνου πρόκλησης πυρκαγιών.

Πολλές χώρες που αντιμετωπίζουν το πρόβλημα των δασικών πυρκαγιών έχουν αναπτύξει τα τελευταία χρόνια με τη χρήση Γεωγραφικών Συστημάτων Πληροφοριών (GIS), συστήματα εκτίμησης κινδύνου πυρκαγιών, έτσι ώστε οι αρμόδιες υπηρεσίες να μπορούν να προσδιορίζουν τις περιοχές υψηλού κινδύνου και να σχεδιάζουν τις απαιτούμενες προληπτικές και προκατασταλτικές ενέργειες. Η πλειονότητα των συστημάτων αυτών στηρίζονται σε μετεωρολογικά δεδομένα, τα οποία συλλέγονται από αυτόματους μετεωρολογικούς σταθμούς που είναι εγκαταστημένοι στις δασικές περιοχές. Επίσης, με τη βοήθεια των Γεωγραφικών Συστημάτων Πληροφοριών είναι δυνατή η χωροθέτηση σταθμών πυρανίχνευσης τόσο επανδρωμένων, όσο και αυτόματων.

### 1.2 Αντικείμενο διπλωματικής

Ένα πολύ σημαντικό μέτρο για τη πρόληψη των δασικών πυρκαγιών είναι η ψηφιακή χαρτογράφηση του κινδύνου πυρκαγιών με τη χρήση Γεωγραφικών Συστημάτων Πληροφοριών, με στοιχεία που συλλέγονται από δίκτυα αυτόματων μετεωρολογικών σταθμών που μπορούν να εγκατασταθούν για το σκοπό αυτό, σε διάφορες δασικές περιοχές. Το κόστος της συλλογής μετεωρολογικών δεδομένων και της επεξεργασίας αυτών με ψηφιακές εφαρμογές Γεωγραφικών Συστημάτων Πληροφοριών, είναι αρκετά υψηλό και λόγω της οικονομικής κατάστασης της χώρας, δεν δίνεται μεγάλη έμφαση από πλευράς πολιτείας στο τομέα αυτό.

Η διπλωματική έχει σκοπό να μελετήσει και να αναδείξει εναλλακτικές οικονομικές μεθόδους συλλογής και επεξεργασίας μετεωρολογικών δεδομένων και θα επικεντρωθεί στη χρήση αυτών των συστημάτων για τη πρόληψη και αντιμετώπιση των δασικών πυρκαγιών. Οι τεχνικές που θα αναλυθούν στη διπλωματική μπορούν να χρησιμοποιηθούν και για την

πρόληψη και την αντιμετώπιση άλλων κινδύνων, όπως για παράδειγμα την έγκυρη ενημέρωση εμφάνισης πλημμυρικών ή και άλλων ακραίων καιρικών φαινομένων.

Στο δεύτερο κεφάλαιο της διπλωματικής αναφέρονται σχετικά προγράμματα που έχουν υλοποιηθεί για την πρόληψη και αντιμετώπιση των δασικών πυρκαγιών. Η διπλωματική δεν έχει σκοπό να αντικαταστήσει ή να «ανταγωνιστεί» τα προγράμματα αυτά παρά μόνο να αναδείξει ένα εναλλακτικό οικονομικό τρόπο συλλογής και επεξεργασίας δεδομένων για εκπαιδευτικούς σκοπούς ή και τη χρήση τους για μελέτες από εθελοντικούς οργανισμούς ή και μικρές κοινότητες.

### 1.2.1 Συνεισφορά

Η διπλωματική προσπαθεί να συνεισφέρει και να δώσει λύσεις στο πρόβλημα του μεγάλου κόστους εξοπλισμού (λογισμικού και υλικού) για τη συλλογή και επεξεργασία δεδομένων για τις μελέτες που αφορούν τις δασικές πυρκαγιές. Η μεθοδολογία που ακολουθήθηκε στα πλαίσια της διπλωματικής ήταν:

1. Μελετήθηκαν οι τεχνολογίες καταγραφής μετεωρολογικών παραμέτρων.
2. Σχεδιάστηκαν και υλοποιήθηκαν συστήματα καταγραφής μετεωρολογικών παραμέτρων χαμηλού κόστους, στηριζόμενα σε Ελεύθερες Τεχνολογίες υλικού και λογισμικού (open source software and hardware).
3. Αξιολογήθηκαν τα δεδομένα και η επίδοση των συστημάτων.
4. Επεξεργαστήκαν τα δεδομένα και ενσωματώθηκαν σε Γεωγραφικά Συστήματα Πληροφοριών με τη χρήση εφαρμογών ανοιχτού κώδικα.
5. Αξιολογήθηκαν τα αποτελέσματα και η συνεισφορά τους στη πρόληψη και αντιμετώπιση δασικών πυρκαγιών.

## 1.3 Οργάνωση κειμένου

Εργασίες σχετικές με το αντικείμενο της διπλωματικής παρουσιάζονται στο Κεφάλαιο 2. Το Κεφάλαιο 3 αναφέρει το Θεωρητικό υπόβαθρο που σχετίζεται με το θέμα της διπλωματικής. Στο Κεφάλαιο 4 αναπτύσσεται ο τρόπος μοντελοποίησης και υλοποίησης. Η αξιολόγηση και τα αποτελέσματα αναφέρονται στο 5 Κεφάλαιο. Στο 6 Κεφάλαιο αναφέρονται οι τεχνικές λεπτομέρειες υλοποίησης και οι πλατφόρμες και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν. Τέλος στο 7 Κεφάλαιο αναφέρονται τα συμπεράσματα και οι μελλοντικές επεκτάσεις που θα μπορούσε να έχει η διπλωματική.

# 2

## Σχετικές εργασίες

Στο κεφάλαιο αυτό παρουσιάζονται ολοκληρωμένα συστήματα που έχουν υλοποιηθεί για την ανίχνευση, την εποπτεία και την πρόβλεψη της εξέλιξης, καθώς και την υποστήριξη της επιχειρησιακής διαχείρισης των δασικών πυρκαγιών. Τα συστήματα αυτά έχουν τεθεί σε λειτουργία σε συγκεκριμένες περιοχές. Το σύστημα Firementor ανιχνεύει τη πυρκαγιά και προβλέπει την εξέλιξη της με τη χρήση δικτύου αναλώσιμων αισθητήρων. Το σύστημα Forest Cities είναι ένα σύστημα τηλεματικής για τη δυναμική αποτίμηση του κινδύνου εκδήλωσης δασικών πυρκαγιών σε πραγματικό χρόνο, βασισμένο σε Γεωγραφικά Συστήματα Πληροφοριών.

### 2.1 Firementor

Το Firementor είναι ένα ολοκληρωμένο σύστημα για την ανίχνευση την εποπτεία και την υποστήριξη της επιχειρησιακής διαχείρισης των δασικών πυρκαγιών. Αποτελείται από ένα αυτο-διοργανούμενο δίκτυο αναλώσιμων αισθητήρων, οι οποίοι σε πραγματικό χρόνο μεταδίδουν πληροφορίες που αφορούν στη θερμοκρασία και τις μεταβολές της μέσα στο δάσος. Οι πληροφορίες μεταδίδονται μέσα από μια σειρά διαδοχικών βημάτων από αισθητήρα σε αισθητήρα και στη συνέχεια σε κεντρικούς κόμβους που με τη σειρά τους διαβιβάζουν την κατάσταση ολόκληρου του δικτύου στο κέντρο διαχείρισης (Μαρκάτος, κ.α. 2008).

Όπως εξηγεί ο κ. Βασίλης Βεσκούκης, επίκουρος καθηγητής στο ΕΜΠ και υπεύθυνος επικοινωνίας και διασφάλισης ποιότητας του έργου: *«Παρακολουθούμε σε πραγματικό χρόνο την κατανομή της θερμοκρασίας στο δάσος. Όταν εκδηλωθεί πυρκαγιά, γίνεται άμεσα αντιληπτή από το σύστημα, κάτι που είναι χρήσιμο για μεγάλες εκτάσεις που η τοπογραφία δεν επιτρέπει την επιτήρησή τους από τον άνθρωπο. Επιπλέον, εάν η πυρκαγιά ξεσπάσει σε δύο ή περισσότερα σημεία, μπορούμε να γνωρίζουμε με βεβαιότητα ότι πρόκειται για εμπρησμό»* (Σουλιώτης 2006).

Σύμφωνα με τους κ. Μαρκάτος κ.α. (2008) το σύστημα Firementor έχει επίσης τη δυνατότητα προσομοίωσης της εξέλιξης δασικής πυρκαγιάς σε πραγματικό χρόνο χρησιμοποιώντας κατάλληλα μαθηματικά μοντέλα. Το σύστημα προσομοιώνει την εξέλιξη πυρκαγιάς με βάση το σημείο όπου εκδηλώνεται πυρκαγιά, το οποίο εντοπίζεται μέσω του δικτύου αισθητήρων, σε συνδυασμό με τις τρέχουσες μετεωρολογικές συνθήκες και το χάρτη καύσιμης ύλης της περιοχής.

Οι υπεύθυνοι του προγράμματος (Μαρκάτος, κ.α., 2008) επισημαίνουν πως τα δεδομένα που απαιτούνται για τις υπηρεσίες που παρέχει το σύστημα, μπορούν να συγκεντρωθούν από διάφορες πηγές, η δε δυνατότητα χρήσης κάθε συγκεκριμένης λειτουργίας του λογιστικού καθώς και η ποιότητα των αποτελεσμάτων καθορίζεται από την

ποιότητα των ψηφιακών δεδομένων της περιοχής την οποία εποπτεύει (πληρότητα και ακρίβεια).

Το σύστημα αυτό λειτουργεί με τη χρήση δικτύου αναλώσιμων αισθητήρων. Κάθε αισθητήρας διαθέτει αισθητήρα θερμοκρασίας, μικροϋπολογιστή και σύστημα ασύρματης επικοινωνίας. Οι αισθητήρες τοποθετούνται πάνω στα δέντρα και σε τακτά χρονικά διαστήματα μεταδίδουν ένα μήνυμα όπου και γίνεται η επεξεργασία των δεδομένων στο κέντρο διαχείρισης. Έτσι, είναι γνωστή σε πραγματικό χρόνο η κατανομή της θερμοκρασίας στο δάσος. Σε περίπτωση πυρκαγιάς, ο αισθητήρας καταστρέφεται έχοντας πρώτα στείλει στον αμέσως κοντινότερο αισθητήρα ένα μήνυμα κινδύνου. Στη συνέχεια, το δίκτυο αναδιοργανώνεται αυτόματα καλύπτοντας το κενό που άφησε ο κατεστραμμένος αισθητήρας έτσι ώστε τα μηνύματα να συνεχίσουν να φτάνουν στο κέντρο διαχείρισης (Μαρκάτος, κ.α., 2008).

Όπως αναφέρει ο συντάκτης της εφημερίδας «Η Καθημερινή» Γ. Σουλιώτης (2006) το κόστος για την πρώτη φάση ανάπτυξης ανήλθε στα 2,5 εκατ. ευρώ. Στη συνέχεια αναφέρει πως με δεδομένο ότι η ιδανική απόσταση μεταξύ των αισθητήρων είναι τα 40 - 50 μέτρα, το κόστος για την εγκατάσταση του συστήματος σε μια έκταση 1.000 στρεμμάτων όπου χρειάζονται περίπου 5.000 αισθητήρες, ανέρχεται στα 100.000 ευρώ, όπως όμως αναφέρει, ότι επισημαίνουν οι επιστήμονες, με τον επανασχεδιασμό του αισθητήρα το κόστος θα μειωθεί έως και 2,5 φορές.

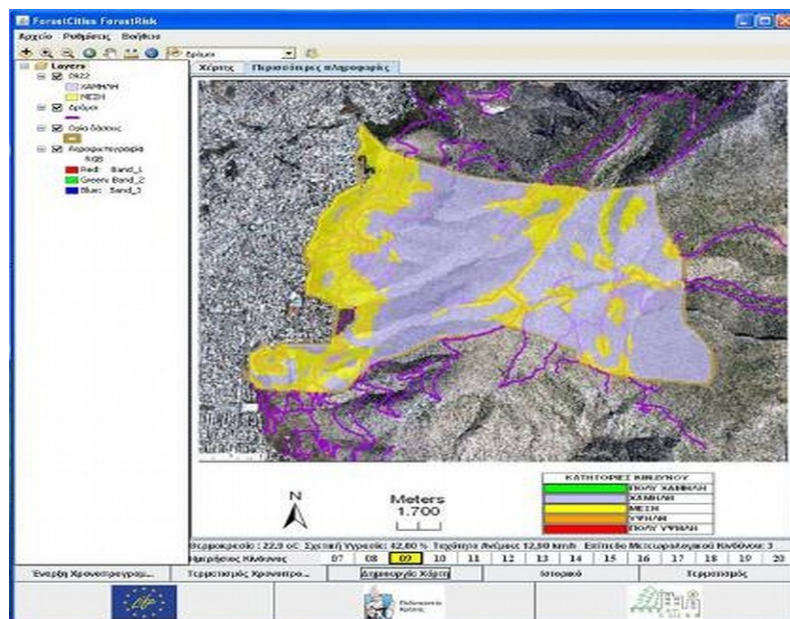
Στα πλαίσια της διπλωματική εργασίας γίνεται μια προσπάθεια δημιουργίας παρομοίου συστήματος, με κάποιες σημαντικές διαφορές, μια εκ των οποίων είναι το ότι, οι αισθητήρες δεν θα είναι αναλώσιμοι αυτό έχει ως αποτέλεσμα αυξημένο κόστος του κάθε αισθητήρα, επίσης το σύστημα δεν θα έχει τη δυνατότητα της ανίχνευσης της πυρκαγιάς σε πραγματικό χρόνο. Το σημαντικότερο πλεονέκτημα του συστήματος που μελετάει η διπλωματική είναι η πολύ μεγαλύτερη ακρίβεια δεδομένων και ο μικρότερος, συγκριτικά με το προαναφερθέν σύστημα, αριθμός των απαιτούμενων αισθητήρων που πρέπει να τοποθετηθούν στην υπο μελέτη περιοχή, που έχουν ως αποτέλεσμα πολύ μικρότερο κόστος συνολικά. Ως μελλοντικό σχέδιο θα μπορούσε το σύστημα που μελετάει η παρούσα διπλωματική να συμβάλει στη δημιουργία ακριβέστερων και πληρέστερων μοντέλων για το σύστημα Firementor.

## 2.2 Forest Cities

Το Forest Cities είναι ένα σύστημα τηλεματικής «για τη δυναμική αποτίμηση του κινδύνου εκδήλωσης δασικών πυρκαγιών σε πραγματικό χρόνο, βασισμένο σε Γεωγραφικά Συστήματα Πληροφοριών (GIS)», αναπτύχθηκε στο Πολυτεχνείο Κρήτης και τα τελευταία 2 χρόνια εφαρμόζεται στους Δήμους Ηλιούπολης Αττικής και Σερρών. Το σύστημα βασίζεται στην παρακολούθηση συγκεκριμένων σημαντικών παραμέτρων για την εκδήλωση δασικών πυρκαγιών (μετεωρολογικές και μικροκλιματικές παράμετροι, φυσικό ανάγλυφο, καύσιμη ύλη, χρονική περίοδο του έτους κ.ά.).

Όπως αναφέρεται στη Κυριακάτικη Ελευθεροτυπία (Λυβιάκης, 2013) το πληροφορικό αυτό σύστημα αναπτύχθηκε στο πλαίσιο του ευρωπαϊκού προγράμματος LIFE+ Local

Authorities for Forest Fire Prevention - Forest Cities (<http://www. forestcities.gr>) από τον καθηγητή στο Τμήμα Ηλεκτρονικών Μηχανικών και Μηχανικών Υπολογιστών (ΗΜΜΥ) του Πολυτεχνείου Κρήτης Γεώργιο Σταυρακάκη, ο οποίος είναι και ο επιστημονικός υπεύθυνος, την επίκουρη καθηγήτρια μηχανικών Περιβάλλοντος (ΜΗΠΕΡ) Διονυσία Κολοκοτσά, την αναπληρώτρια καθηγήτρια του Γενικού Τμήματος Ευπραξία-Αίθρα Μαριά και τον φυσικό-MSc Κωνσταντίνο Γομπάκη. Οι καθηγητές επισημαίνουν όπως αναφέρει ο Γ. Λυβιάκης (2013) ότι: «το πληροφορικό αυτό σύστημα βασίζεται στην παρακολούθηση συγκεκριμένων παραμέτρων κρίσιμων για την εκδήλωση δασικών πυρκαγιών, όπως μετεωρολογικές και μικροκλιματικές παράμετροι της συγκεκριμένης δασικής περιοχής εφαρμογής του, το φυσικό ανάγλυφο της δασικής περιοχής, την καύσιμη ύλη της, τη χρονική περίοδο του έτους (ουσιαστικά τη θερινή υψηλής επικινδυνότητας) κ.ά. Τα κύρια υποσυστήματά του είναι: συλλογή μετεωρολογικών και μικροκλιματικών δεδομένων σε πραγματικό χρόνο, επεξεργασία, ανάλυση και σύγκριση των δεδομένων αυτών με παλαιότερα για τη δημιουργία του μοντέλου, αναγνώριση κρίσιμων παραμέτρων και μοντέλο λήψης αποφάσεων σχετικά με τον κίνδυνο εκδήλωσης πυρκαγιάς, γραφική απεικόνιση αποτελεσμάτων και αποτίμηση του κινδύνου έναρξης δασικής πυρκαγιάς στην περιοχή εφαρμογής σε πραγματικό χρόνο.



Σχήμα 2.1: Το σύστημα GIS στο Δήμο Ηλιούπολης, που απεικονίζει την επικινδυνότητα περιοχών σε πραγματικό χρόνο για πρόκληση πυρκαγιάς. (Λυβιάκης 2013)

Στον πυρήνα του συστήματος βρίσκεται το μοντέλο εκτίμησης σε πραγματικό χρόνο του κινδύνου εκδήλωσης δασικών πυρκαγιών. Το μοντέλο δημιουργήθηκε από το συγκερασμό δύο ανεξάρτητων επί μέρους δεικτών επικινδυνότητας: 1ο του FHI (Fire Hazard Index), ο οποίος ταξινομεί τις περιοχές με βάση την επικινδυνότητά τους λόγω των τοπολογικών τους χαρακτηριστικών, και 2ο του FDI (Fire Danger Index), ο οποίος αντιπροσωπεύει την πιθανότητα εκδήλωσης πυρκαγιάς βάσει του τρέχοντος μετεωρολογικού κινδύνου.

Για καθένα από τους δύο επί μέρους δείκτες το επίπεδο κινδύνου έχει χωριστεί σε πέντε υποκατηγορίες: Πολύ Χαμηλή, Χαμηλή, Μέση, Υψηλή, Πολύ Υψηλή. Ο χωρισμός των πέντε υποκατηγοριών-κλάσεων του μοντέλου FDI πραγματοποιήθηκε με τη χρήση αλγόριθμων ασαφούς λογικής (fuzzy logic), χρησιμοποιώντας ιστορικά δεδομένα δασικών πυρκαγιών για κάθε περιοχή εφαρμογής, μετεωρολογικά δεδομένα, καθώς και δεδομένα που εξήχθησαν από τους ημερήσιους χάρτες κινδύνου πυρκαγιών της Γενικής Γραμματείας Πολιτικής Προστασίας για όλη τη χώρα.

Για τη λειτουργία του παραπάνω μοντέλου είναι απαραίτητη η ύπαρξη ψηφιακών χωρικών δεδομένων-χαρτών σε Σύστημα Γεωγραφικών Πληροφοριών και συγκεκριμένα επί μέρους ψηφιακοί χάρτες, που αφορούν: τύπο υφιστάμενης βλάστησης, κλίση εδάφους, προσανατολισμό πρανών, εγγύτητα σε δρόμους ή μονοπάτια, εγγύτητα σε γραμμές μεταφοράς ηλεκτρικής ενέργειας, εγγύτητα σε οικισμούς και σε χώρους ανεξέλεγκτης διάθεσης απορριμμάτων (ΧΑΔΑ).»

Οι υπεύθυνοι του προγράμματος επισημαίνουν πως η ύπαρξη μετεωρολογικού σταθμού στη δασική περιοχή εφαρμογής είναι απαραίτητη για τη συνεχή λήψη των τιμών θερμοκρασίας, σχετικής υγρασίας και ταχύτητας ανέμου.

Επίσης επισημαίνουν όπως αναφέρει ο Γ. Λυβιάκης (2013) στην εφημερίδα Κυριακάτικη Ελευθεροτυπία, ότι: «αναπτύχθηκε λογισμικό με ένα προσαρμοσμένο γραφικό περιβάλλον με κύριο γνώμονα την ευκολία εκμάθησης και χρήσης του συστήματος από τον εκάστοτε χρήστη. Το γραφικό περιβάλλον παρουσιάζει γραφικά όλες τις απαραίτητες πληροφορίες στο χρήστη, όπως τα διάφορα επίπεδα χωρικών πληροφοριών, τρέχουσες μετεωρολογικές συνθήκες, καθώς και την εξέλιξη του κινδύνου εκδήλωσης δασικής πυρκαγιάς κατά τη διάρκεια της τρέχουσας ημέρας. Δίνεται επίσης η δυνατότητα αποστολής e-mail ή SMS, όταν ο εκτιμώμενος κίνδυνος εκδήλωσης πυρκαγιάς ξεπεράσει ένα καθορισμένο όριο, δηλαδή όταν ο συνολικός δείκτης εκτίμησης της επικινδυνότητας έναρξης πυρκαγιάς λάβει την τιμή 4 ή 5 κάποια δεδομένη στιγμή της ημέρας».

Όσον αφορά το κόστος υλοποίησης, στον επίσημο δικτυακό τόπο του προγράμματος (forestcities.gr) αναφέρει πως ο συνολικός προϋπολογισμός του έργου ανέρχεται στα 698.323€ χωρίς να αναφέρει περαιτέρω λεπτομέρειες για το κόστος του προγράμματος αυτού.

# 3

## Θεωρητικό υπόβαθρο

### 3.1 Εκτίμηση κινδύνου δασικών πυρκαγιών

Η εκτίμηση κινδύνου δασικών πυρκαγιών λόγω της πολυπλοκότητας του φαινομένου, βασίζεται σε διαφορετικές μεθοδολογίες, που έχουν αναπτυχθεί χρησιμοποιώντας συνδυασμό παραμέτρων, ανάλογα με τη διαθεσιμότητα τους και συνήθως εφαρμόζονται με διαφορετικό τρόπο για κάθε περιοχή μελέτης. Σύμφωνα με Γενική Γραμματεία Πολιτικής Προστασίας (GSCP.gr 2008), οι βασικοί παράμετροι που καθορίζουν τον κίνδυνο εκδήλωσης πυρκαγιάς, είναι το κλίμα, η μορφολογία εδάφους, τα χαρακτηριστικά βλάστησης, οι ανθρώπινες δραστηριότητες και τα φυσικά αίτια (κεραυνοί).

Οι μεθοδολογίες για την εκτίμηση κινδύνου δασικών πυρκαγιών παράγουν δείκτες επικινδυνότητας. Οι δείκτες επικινδυνότητας έχουν ως βασική επιδίωξη τον χαρακτηρισμό μιας περιοχής με περισσότερο ή λιγότερο κίνδυνο εκδήλωσης πυρκαγιών. Στην ιστοσελίδα της Γενικής Γραμματείας Πολιτικής Προστασίας (GSCP.gr 2008) αναφέρει, ότι επικρατέστερος δείκτης εκτίμησης κινδύνου πυρκαγιάς είναι ο Αμερικάνικος δείκτης NFDRS (Ηνωμένων Πολιτειών - NFDRS και Καναδά – NFFDRS) ο οποίος προσαρμόζεται στις ιδιαιτερότητες της κάθε χώρας. Για τα συστήματα αυτά ο κ. Ξανθόπουλος (2012) αναφέρει: *«Στόχος ενός τέτοιου συστήματος είναι η δυνατότητα πρόβλεψης του κινδύνου εκδήλωσης και εξέλιξης πυρκαγιών για την επόμενη χρονική περίοδο (ημέρα ή ημέρες). Αποτελεί στοιχείο στο οποίο βασίζεται το δυναμικό μέρος του αντιπυρικού σχεδιασμού. Με την κατάλληλη αξιοποίησή του επιτυγχάνεται αυξημένη επιφυλακή και μέτρα κατά τις κρίσιμες ημέρες, ενώ εξοικονομούνται δυνάμεις και πόροι όταν ο κίνδυνος είναι σχετικά ήπιος. Έτσι, αυξάνεται η αποτελεσματικότητα ενώ, ταυτόχρονα, περιστέλλονται οι δαπάνες. Ακόμη, η πρόγνωση του κινδύνου είναι ιδιαίτερα χρήσιμη για την ενημέρωση των πολιτών, ώστε να επιδεικνύουν αυξημένη προσοχή.»*

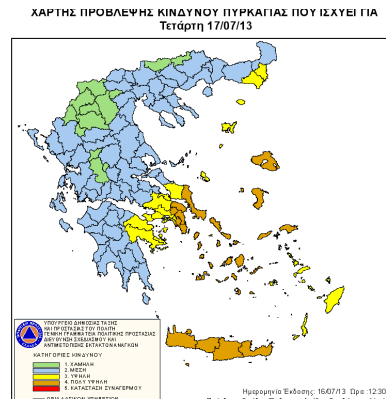
Οι δείκτες εκτίμησης κινδύνου απεικονίζονται συνήθως σε ταμπέλες (Σχήμα 3.1) που τοποθετούνται σε διάφορες περιοχές ή παρουσιάζονται σε μορφή θεματικού χάρτη (Σχήμα 3.2). Η Γενική Γραμματεία Πολιτικής Προστασίας, κατά τη διάρκεια της αντιπυρικής περιόδου, συντάσσει καθημερινά χάρτη πρόγνωσης κινδύνου πυρκαγιάς για την επόμενη ημέρα ο οποίος αναρτάται σε ψηφιακή μορφή στον δικτυακό της τόπο.

Όπως αναφέρεται στο GSCP.gr (2008) λαμβάνονται κυρίως υπόψη για τη σύνταξη των χαρτών αυτών, προβλέψεις των σχετικών με τις πυρκαγιές καιρικών φαινομένων για το επόμενο 24ωρο, η κατάσταση της βλάστησης καθώς και κάθε άλλη διαθέσιμη πληροφορία που συμβάλει στον προσδιορισμό της επικινδυνότητας μιας περιοχής σε δεδομένη χρονική στιγμή. Αξίζει να αναφερθεί πως η ακρίβεια του χάρτη εξαρτάται σε πολύ μεγάλο βαθμό από την ακρίβεια των μετεωρολογικών προβλέψεων (GSCP.gr 2008).





Σχήμα 3.1: Πινακίδα Κινδύνου Πυρκαγιάς.  
(<http://smokeyzone.com/fire-danger-signs/>)



Σχήμα 3.2: Ημερήσιο δελτίο πρόβλεψης κινδύνου πυρκαγιάς. (<http://www.gscp.gr/>)

Ένας άλλος σημαντικός παράγοντας που συμβάλει στη εκτίμηση του κίνδυνου πυρκαγιάς είναι η κατάσταση της βλάστησης, η οποία χαρακτηρίζεται από μη συνεχή μεταβολή, αντίθετα με τις μετεωρολογικές συνθήκες που μεταβάλλονται συνεχώς (GSCP.gr 2008). Ο πιο διαδεδομένος δείκτης υπολογισμού της κατάσταση της βλάστησης είναι ο κανονικοποιημένος δείκτης βλάστησης NDVI (Normalized Difference Vegetation Index). Ο δείκτης βλάστησης NDVI υπολογίζεται βάση του επιπέδου ανάκλασης του κόκκινου τμήματος του ηλεκτρομαγνητικού φάσματος και του εγγύς υπέρυθρου φωτός (ESA).

### 3.2 Τεχνολογίες μετρήσεων περιβαλλοντικών παραμέτρων

Η παρακολούθηση και μέτρηση φυσικών μεγεθών γίνονται με τη βοήθεια αισθητήρων (sensors). Οι αισθητήρες μπορούν να είναι ξεχωριστές συσκευές ή περίπλοκες κατασκευές, που διαθέτουν κάποια κατάλληλη ιδιότητα, η οποία μεταβάλλεται ως συνάρτηση του μετρούμενου φυσικού μεγέθους. Έτσι η μέτρηση της ιδιότητας του αισθητήρα επιτρέπει τον άμεσο ποσοτικό υπολογισμό της τιμής του φυσικού μεγέθους.

Όταν οι αισθητήρες δεν δίδουν στην έξοδό τους αναγνωρίσιμο ηλεκτρικό σήμα απαιτείται η χρήση ενός επιπρόσθετου ηλεκτρονικού κυκλώματος, το οποίο να αναλαμβάνει την αποκωδικοποίηση του σήματος σε αναγνωρίσιμο σήμα, σύμφωνα με τις απαιτήσεις των επόμενων βαθμίδων. Σύμφωνα με τον Elgar (2003) το κύκλωμα αυτό ονομάζεται μονάδα ρύθμισης σήματος όπου το τροποποιημένο σήμα μεταφέρεται στη μονάδα εμφάνισης ή απεικόνισης. Για παράδειγμα, υπάρχουν αισθητήρες μέτρησης απόστασης με λέιζερ, που λειτουργούν μετρώντας το χρόνο που απαιτείται για να ανακλαστεί το φωτεινό σήμα από τη μετρούμενη επιφάνεια και να επιστρέψει στο σημείο από όπου εκπέμφθηκε. Σε αυτούς πρέπει να υπάρχει κατάλληλο κύκλωμα για τη μετατροπή των τιμών χρόνου σε ανάλογες αναγνωρίσιμες τιμές.

Στις μέρες μας υπάρχει μια πολύ μεγάλη ποικιλία αισθητήρων που στηρίζονται σε διαφορετικές αρχές λειτουργίας, παρόλα αυτά έχουν κάποια κοινά βασικά χαρακτηριστικά.

Ο Elgar P. (2003) στο βιβλίο του “Αισθητήρες μέτρησης και Έλεγχου” (σελ. 27-33) αναφέρει τα ακόλουθα χαρακτηριστικά: Ακρίβεια (accuracy), Βαθμονόμηση (calibration), Νεκρή ζώνη (dead zone, dead band), Διαστάσεις (dimensions), Ολίσθηση (drift), Σφάλμα (error), Υστέρηση (hysteresis), Καθυστέρηση (lag), Γραμμικότητα (linearity), Χρόνος λειτουργίας (operating life), Επαναληψιμότητα (precision), Εύρος (operating range), Ονομαστική τιμή (rating), Αξιοπιστία (reliability), Απόκριση (response), Διακριτική Ικανότητα (resolution), Ευαισθησία (sensitivity), Ευστάθεια (stability), Στατικό σφάλμα (static error) και Ανοχή (tolerance).

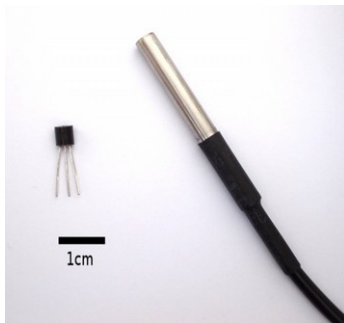
Στις παραγράφους που ακολουθούν περιγράφονται τα κυριότερα είδη αισθητήρων καταγραφής περιβαλλοντικών παραμέτρων. Έτσι περιγράφονται αισθητήρες θερμοκρασίας, υγρασίας, βροχής, πίεσης και άνεμου. Σε κάθε είδος αισθητήρα περιγράφεται η αρχή λειτουργίας του και στη συνέχεια δίνονται παραδείγματα προϊόντων του εμπορίου.

**Αισθητήρες θερμοκρασίας:** Η παράμετρος που μετριέται πιο συχνά είναι η θερμοκρασία. Αυτό συμβαίνει γιατί τα περισσότερα φυσικά, βιολογικά, χημικά, μηχανικά, και ηλεκτρονικά συστήματα επηρεάζονται από τη θερμοκρασία. Όπως διευκρινίζει ο Tarun Agarwal (elprocus.com) ορισμένες χημικές αντιδράσεις, βιολογικές διεργασίες, ακόμη και ηλεκτρονικά κυκλώματα αποδίδουν καλύτερα μέσα σε περιορισμένο εύρος θερμοκρασιών. Υπάρχουν πολλοί τρόποι για να μετρηθεί η θερμοκρασία. Η μέτρηση της θερμοκρασίας μπορεί να γίνει είτε μέσω της άμεσης επαφής με την πηγή θέρμανσης, είτε εξ αποστάσεως με την χρήση υπέρυθρων (IR). Οι βασικότεροι τύποι αισθητήρων θερμοκρασίας αυτών είναι: θερμοζευγών (Thermocouples), μεταβαλλόμενης θερμικής αντίστασης (RTDs), θερμοαντιστάτες (Thermistors), υπέρυθρων (IR), και αισθητήρες ημιαγωγών (Semiconductor). Οι νέου τύπου ψηφιακοί αισθητήρες θερμοκρασίας παρέχουν υψηλή ακρίβεια μετρήσεων με χαμηλή κατανάλωση ενέργειας και μεγάλη ευκολία χρήσης. Είναι έτοιμοι προς χρήση ρυθμισμένοι και βαθμονομημένοι από το εργοστάσιο παραγωγής επομένως δεν χρειάζονται πολύπλοκα κυκλώματα για τη ρύθμιση του αισθητήρα. Ένας ευρέως διαδεδομένος ψηφιακός αισθητήρας θερμοκρασίας είναι ο DS18B20 της εταιρείας Maxim Integrated (Σχήμα 3.3).

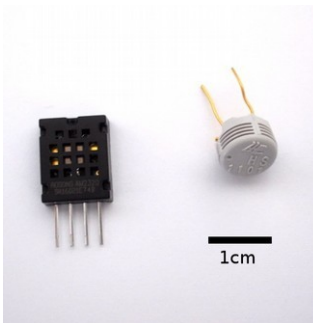
**Υγρασιόμετρο (υγρόμετρο):** Υγρόμετρο ονομάζεται το όργανο με το οποίο μετράμε τη σχετική υγρασία του περιβάλλοντος αέρα (EAA). Υπάρχουν διάφοροι τύποι υγρομέτρων όπως το υγρόμετρο τρίχας, το υγρόμετρο Αλλυάρ αλλά και το υγρόμετρο δρόσου. Η μέτρηση της υγρασίας μπορεί να γίνει και με το ξηρό-υγρό θερμομέτρο αλλά η διαδικασία επιτυγχάνεται πιο εύκολα με τα υγρόμετρα τα οποία παρέχουν απευθείας την τιμή της σχετικής υγρασίας. Στο εσωτερικό οργάνων παλιού τύπου συνήθως υπάρχουν τρίχες ουράς αλόγου των οποίων το μήκος μεταβάλλεται ανάλογα με τα ποσοστά της σχετικής υγρασίας, με χαμηλή υγρασία συστέλλονται και διαστέλλονται με υψηλή υγρασία. Μέσω κατάλληλης διάταξης μοχλών η κίνηση αυτή μεταδίδεται σε ένα δείκτη ο οποίος πάνω σε μία βαθμονομημένη κλίμακα μας δείχνει το ακριβή ποσοστά της σχετικής υγρασίας (EAA).

Για εφαρμογές όπου το κόστος, το μέγεθος και η ευθραυστότητα έχουν σημασία, χρησιμοποιούνται ηλεκτρονικού τύπου αισθητήρων υγρασίας. Οι ηλεκτρονικού τύπου αισθητήρες υγρασίας έχουν συνήθως χαμηλότερη ακρίβεια. Πολύ γνωστός αισθητήρας υγρασίας είναι ο HS1101 (Σχήμα 3.4) της εταιρίας Humirel ο οποίος είναι ηλεκτρονικός χωρητικός (Capacitive) αισθητήρας. Σε χωρητικού τύπου υγρόμετρα, μετριέται η επίδραση της υγρασίας στην διηλεκτρική σταθερά του πολυμερούς ή μεταλλικού οξειδίου (Roveti 2001).

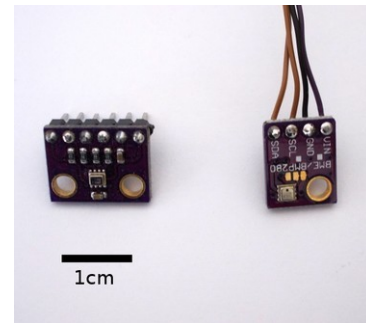
Η θερμοκρασίας περιβάλλοντος επηρεάζει σε μεγάλο βαθμό τις μετρήσεις υγρασίας, για αυτό το λόγο, οι εταιρείες παράγουν αισθητήρες που μετράνε και τις δυο παραμέτρους, ώστε να υπάρχει η δυνατότητα αυτόματης ρύθμισης. Τέτοιου τύπου αισθητήρας είναι και ο AM2320 (Σχήμα 3.4) της εταιρίας Aosong, ο οποίος έχει τη δυνατότητα μέτρησης της θερμοκρασίας και της σχετικής υγρασίας.



Σχήμα 3.3: Αισθητήρες θερμοκρασίας DS18B20



Σχήμα 3.4: Υγρόμετρα, AM2320 (αριστερά), HS1101 (δεξιά)



Σχήμα 3.5: Βαρόμετρα, BMP280 (αριστερά), BME280 (δεξιά)

**Βαρόμετρο:** Το βαρόμετρο είναι το μετεωρολογικό όργανο το οποίο μετράει τη ατμοσφαιρική πίεση. Τα βαρόμετρα διακρίνονται σε υδραργυρικά, μεταλλικά και αυτογραφικά. Η βαρομετρική πίεση αποτελεί ένα από τα σημαντικότερα μετεωρολογικά στοιχεία. Η μονάδα μέτρησης είναι το 1 hPa το οποίο είναι το ίδιο ακριβώς με το 1 mb το οποίο χρησιμοποιείται παλαιότερα. Δυο βασικά είδη βαρομέτρων είναι το υδραργυρικό και το μεταλλικό. Το υδραργυρικό υπάρχει σε όλους τους μετεωρολογικούς σταθμούς και θεωρείται μεγάλης ακρίβειας. (ΕΑΑ). Το μειονέκτημά είναι το υψηλό κόστος του και το γεγονός ότι η ένδειξη που δίνει δεν είναι σε hPa οπότε απαιτούνται μετατροπές. Άλλα βαρόμετρα λειτουργούν με τη χρήση ενός μεταλλικού τυμπάνου το οποίο κάμπτεται με οποιαδήποτε μεταβολή της ατμοσφαιρικής πίεσης. Αυτή η απειροελάχιστη κίνηση πολλαπλασιάζεται και μεταφέρεται μέσω ενός συστήματος μοχλών οι οποίοι συνδέονται με έναν δείκτη ο οποίος παρέχει την τιμή της ατμοσφαιρικής πίεσης. Το βασικό τους μειονέκτημα είναι ότι θέλουν συχνά βαθμονόμηση. Σήμερα τα περισσότερα βαρόμετρα είναι ηλεκτρονικά και λειτουργούν συνήθως πιεζοηλεκτρικά ή με πιεζοαντίσταση. Στους αισθητήρες πιεζοαντίστασης (piezoresistive) μετριέται η αλλαγή στην ηλεκτρική αντίσταση ενός ημιαγωγικού υλικού που οφείλεται σε μηχανική πίεση, ενώ στους πιεζοηλεκτρικούς μετριέται ένα φορτίο ή μια τάση (Kon κ.α. 2007).

Η κύρια χρήση του βαρόμετρου είναι για προβλέψεις μετεωρολογικών φαινομένων που πρόκειται να συμβούν στο άμεσο μέλλον (συνήθως μέσα στο επόμενο 24ωρο). μετρώντας για παράδειγμα, μια υψηλή βαρομετρική πίεση σε τοπικό επίπεδο, υποδηλώνει αίθριο καιρό,

ενώ μια χαμηλή σημαίνει ότι πλησιάζει καταιγίδα. Δίκτυα τέτοιων αισθητήρων δημιουργούν χάρτες βαρομετρικής πίεσης κάνοντας χρήση των ισοβαρικών γραμμών (γραμμές ίδιας πίεσης). Μέσω αυτών των γραμμών γίνεται η πρόβλεψη της κίνησης μετεωρολογικών συστημάτων σε ευρύτερες περιοχές. Ένα βαρομετρικό υψηλό αποτελεί ένα φραγμό στην μετατόπιση ενός τέτοιου συστήματος, ενώ μια γραμμή βαρομετρικού χαμηλού αντιπροσωπεύει την πιθανότερη κίνηση του μετεωρολογικού συστήματος, από τη στιγμή που αποτελεί την διαδρομή με την μικρότερη αντίσταση.

Μια μικρή μεταβολή της θερμοκρασίας και του υψομέτρου επηρεάζει σημαντικά την προς μέτρηση βαρομετρική πίεση. Για τον λόγο αυτό, για να γίνει σωστή η μέτρηση της βαρομετρικής πίεσης, απαιτείται ακριβής μέτρηση της θερμοκρασίας αλλά και του υψομέτρου στο οποίο είναι τοποθετημένο το βαρόμετρο.

Στα μέσα του έτους 2015 η εταιρεία Bosch διάθεσε στη αγορά το ηλεκτρονικό βαρόμετρο BMP280 (Σχήμα 3.5). Ο αισθητήρας αυτός στηρίζεται σε τεχνολογία πιεζοαντίστασης και συμπεριλαμβάνει αισθητήρα θερμοκρασίας, ο οποίος είναι απαραίτητος για τις διορθώσεις που απαιτούνται για να υπολογίζεται η πίεση με μεγαλύτερη ακρίβεια.

Παράλληλα με τον αισθητήρα BMP280 η εταιρεία Bosch διάθεσε στη αγορά και τον αισθητήρα BME280 (Σχήμα 3.5) ο οποίος συμπεριλαμβάνει και αισθητήρα μέτρησης σχετικής υγρασίας. Με τον αισθητήρα αυτόν γίνεται δυνατή η αξιόπιστη και με μεγάλη ακρίβεια μέτρηση των βασικών περιβαλλοντικών παραμέτρων (θερμοκρασία, υγρασία, πίεση).

**Βροχόμετρο:** Το βροχόμετρο χρησιμοποιείται για τη μέτρηση της ποσότητας του νερού που έχει πέσει μέσα σε ένα συγκεκριμένο χρονικό διάστημα σε μία περιοχή. Πρόκειται για ένα από τα τυπικά όργανα που χρησιμοποιούνται στους μετεωρολογικούς σταθμούς.

Συνήθως τα βροχόμετρα μετράνε την ποσότητα νερού σε χιλιοστά αν και πολλές φορές η μέτρηση γίνεται σε ίντσες ή και εκατοστά (Σχήμα 3.8). Η καταγραφή των δεδομένων αυτού του οργάνου γίνεται πλέον με ηλεκτρονικό τρόπο, αν και παλαιότερα γινόταν χειροκίνητα. Η συχνότητα καταγραφής των δεδομένων εξαρτάται από το μικροκλίμα της περιοχής αλλά και από τις απαιτήσεις της μετεωρολογικής υπηρεσίας.

**Ανεμόμετρο:** Ανεμόμετρο είναι το όργανο με το οποίο μετράμε την ένταση και τη διεύθυνση του ανέμου. Είναι ίσως το πιο ευρέως διαδεδομένο όργανο στους μετεωρολογικούς σταθμούς. Ο πιο γνωστός τύπος ανεμομέτρου είναι το κυπελλοφόρο ανεμόμετρο (cup anemometer, Σχήμα 3.6) το οποίο αποτελείται από ημισφαιρικά κύπελλα τα οποία περιστρέφονται λόγω του ανέμου. Η ταχύτητα μετριέται συνήθως σε μέτρα/δευτερόλεπτο.



Σχήμα 3.6: Κυπελλοφόρο ανεμόμετρο (EAA)



Σχήμα 3.7: Το WindObserver 75 της Gill Instruments



Σχήμα 3.8: Το βροχόμετρο AEPG της Belfort Instrument

Υπάρχουν και άλλοι τύποι ανεμόμετρων οι οποίοι χρησιμοποιούν διαφορετικές τεχνικές από τις προαναφερθείσες για να πετύχουν την μέτρηση αυτήν. Νέας τεχνολογίας ανεμόμετρα χρησιμοποιούν υπερήχους (Sonic anemometer). Η ταχύτητα του ανέμου σε αυτόν τον τύπο ανεμομέτρου γίνεται μέσω της μέτρησης του χρόνου που απαιτείται για να φτάσει ένας παλμός υπερήχου από τον πομπό στον δέκτη. Στην Εικόνα 3.7 φαίνεται ο αισθητήρας WindObserver 75 της εταιρείας Gill Instruments.

### 3.3 Ελεύθερο λογισμικό, λογισμικό ανοιχτού κώδικα

Η διπλωματική αυτή δημιουργήθηκε και στηρίχθηκε εξ ολοκλήρου σε λογισμικά και τεχνολογίες ανοιχτού κώδικα και για τον λόγο αυτό θεωρείται σκόπιμο να γίνει μια σύντομη αναφορά σε αυτή τη κατηγορία λογισμικών και τεχνολογιών γενικότερα.

Το “Ελεύθερο λογισμικό” στη αγγλική γλώσσα αναφέρεται ως “free software” όπου και επικρατεί σύγχυση όσον αφορά τη σημασία της λέξης “free”, για το λόγο αυτό πάντα διευκρινίζεται η σημασία της λέξης με την εξής έκφραση: “free for free speech not for free beer” (free για ελευθερία λόγου και όχι για δωρεάν μπίρα), δηλαδή αφορά την ελευθερία και όχι το κόστος. Στην ελληνική γλώσσα δεν υπάρχει τέτοια σύγχυση από τη στιγμή που υπάρχει διαφορετική λέξη για κάθε έννοια.

Τα οφέλη Χρήσης Ελεύθερου Λογισμικού / Λογισμικού Ανοιχτού Κώδικα όπως διευκρινίζεται στον επίσημο διαδικτυακό τόπο του ΕΛ/ΛΑΚ (ellak.gr) είναι τα εξής:

- Απόλυτα νόμιμο λογισμικό το οποίο διατίθεται χωρίς κόστος και έχει πολύ μικρότερο κόστος συντήρησης
- Λογισμικό που ενσωματώνει άμεσα τις πιο σύγχρονες τεχνολογικές εξελίξεις
- Συχνές ενημερώσεις με νέες δυνατότητες
- Μεγάλη και φιλική κοινότητα ανάπτυξης και υποστήριξης του λογισμικού
- Πληθώρα πρόσθετων προγραμμάτων με εύκολη και δωρεάν εγκατάσταση
- Σταθερότητα και ασφάλεια

- Σημαντικά ασφαλέστερο και αξιόπιστο σε σχέση με ιδιόκτητο λογισμικό.
- Δυνατότητα να εξερευνησης, μάθησης και προσαρμογής του τρόπου λειτουργίας του λογισμικού.

Όπως διευκρινίζεται στο επίσημο διαδικτυακό τόπο του Ιδρύματος Ελεύθερου Λογισμικού “GNU.com” (2014) το Ελεύθερο λογισμικό παρέχει στους χρήστες την ελευθερία να εκτελούν, αντιγράφουν, διανέμουν, μελετούν, τροποποιούν και βελτιώνουν το Ελεύθερο λογισμικό. Για την ακρίβεια, αναφέρεται σε τέσσερις βασικές ελευθερίες:

- Την ελευθερία να εκτελείτε το πρόγραμμα, για οποιονδήποτε σκοπό (ελευθερία 0).
- Την ελευθερία να μελετάτε τον τρόπο λειτουργίας του προγράμματος και να το προσαρμόζετε στις ανάγκες σας (ελευθερία 1). Η πρόσβαση στον πηγαίο κώδικα είναι προϋπόθεση για να ισχύει κάτι τέτοιο.
- Την ελευθερία να αναδιανέμετε αντίγραφα του προγράμματος ώστε να βοηθάτε το συνάνθρωπο σας (ελευθερία 2).
- Την ελευθερία να βελτιώνετε το πρόγραμμα και να δημοσιεύετε τις βελτιώσεις που έχετε κάνει στο ευρύ κοινό, ώστε να επωφεληθεί ολόκληρη η κοινότητα (ελευθερία 3). Η πρόσβαση στον πηγαίο κώδικα είναι προϋπόθεση για να ισχύει κάτι τέτοιο.

Επίσης αναφέρει πως: *«Ένα πρόγραμμα θεωρείται ελεύθερο λογισμικό όταν οι χρήστες του έχουν όλες τις παραπάνω ελευθερίες. Επομένως, θα πρέπει να υπάρχει η ελευθερία αναδιανομής αντιγράφων, με ή χωρίς τροποποιήσεις, δωρεάν ή χρεώνοντας για την διανομή, στον οποιονδήποτε και οπουδήποτε. Το να υπάρχει η ελευθερία για όλα τα παραπάνω σημαίνει, μεταξύ άλλων, πως δεν χρειάζεται εξουσιοδότηση ώστε να λάβει κάποιος την ανάλογη άδεια»* (GNU 2014).

Στο διαδικτυακό τόπο του Ιδρύματος Ελεύθερου Λογισμικού “GNU.com” (2014) αναφέρει πως πρέπει επίσης να υπάρχει η ελευθερία τροποποίησης χρήσης και αναδιανομής, ιδιωτικά στην δουλειά ή για διασκέδαση, χωρίς να χρειάζεται να γίνει κάποια αναφορά. Η ελευθερία της χρήσης ενός προγράμματος σημαίνει πως δίδεται η ελευθερία σε κάθε άτομο ή επιχείρηση να χρησιμοποιήσει Ελεύθερο Λογισμικό σε κάθε είδους υπολογιστικό σύστημα, για κάθε είδος εργασίας χωρίς να υπάρχει υποχρέωση ειδοποίησης συγκεκριμένου ατόμου η υπηρεσίας. Το Ίδρυμα Ελεύθερου Λογισμικού διευκρινίζει πως: *«σε αυτή την ελευθερία, είναι η άποψη του χρήστη που έχει σημασία, και όχι η άποψη του κατασκευαστή.»*

Οι άδειες ελεύθερων λογισμικών προϋποθέτουν πως στην αναδιανομή αντιγράφων θα πρέπει να περιλαμβάνονται εκτελέσιμες μορφές του προγράμματος (executable), καθώς και ο πηγαίος κώδικας, τόσο για την τροποποιημένη όσο και για την αρχική έκδοση του προγράμματος. Η πρόσβαση στον πηγαίο κώδικα είναι απαραίτητη προϋπόθεση στο ελεύθερο λογισμικό για να ισχύουν πρακτικά οι τέσσερις βασικές ελευθερίες. Επίσης το Ελεύθερο λογισμικό θα πρέπει να είναι διαθέσιμο για εμπορική χρήση, εμπορική ανάπτυξη ή εμπορική διανομή. Η εμπορική ανάπτυξη ελεύθερου λογισμικού ή ελεύθερων τεχνολογιών δεν είναι ασυνήθιστη.

Τέλος, αξίζει να σημειωθεί πως στην λογική των ελεύθερων λογισμικών συμπεριλαμβάνονται και σχέδια για υλικές κατασκευές που συνήθως αναφέρονται ως “υλικό ανοιχτού κώδικα”. Τα σχέδια υλικού ανοιχτού κώδικα, είναι δημοσιευμένα και διαθέσιμα για το κοινό, επιτρέποντας σε οποιονδήποτε να αντιγράψει, τροποποιήσει και να αναδιανέμει το υλικό και τον πηγαίο κώδικα χωρίς να πληρώσει πνευματικά δικαιώματα ή αμοιβή. Στη παρούσα διπλωματική αναφέρονται τα υλικά ανοιχτού κώδικα και τα ελεύθερα λογισμικά γενικά ως Ελεύθερες Τεχνολογίες.

### 3.3.1 Άδειες ελευθέρου λογισμικού

Κάθε λογισμικό δημοσιεύεται με συγκεκριμένη άδεια χρήσης, σε ορισμένες περιπτώσεις ένα “πακέτο” λογισμικών μπορεί να έχει πολλές διαφορετικές άδειες χρήσης για κάθε επιμέρους τμήμα του λογισμικού. Για να θεωρηθεί ένα λογισμικό ελεύθερο πρέπει να διανεμηθεί κάτω από μια ελεύθερη άδεια λογισμικού. Η άδεια που χρησιμοποιείται πιο συχνά είναι η Γενική Άδεια Δημόσιας Χρήσης GNU (GNU GPL).

Όπως αναφέρει ο κ. P. Schmitz (2009), η κοινοχρησία λογισμικού είναι μία από τις βασικές αρχές του «Ελεύθερου Λογισμικού ή Λογισμικού Ανοικτού Κώδικα» (ΕΛ/ΛΑΚ) και είναι από τους αποτελεσματικότερους τρόπους αύξησης της χρήσης του, βελτίωσης της ποιότητάς του (παρέχοντας τη δυνατότητα σε άλλους προγραμματιστές να ελέγχουν, να βελτιώνουν ή να αυξάνουν τις λειτουργικές δυνατότητές του) και μείωσης του κόστους του λογισμικού, με την αποφυγή επανάληψης εργασιών. Τέτοιου είδους κοινοχρησίας λογισμικού προτάθηκε με υπουργική διακήρυξη από τους ευρωπαίους υπουργούς, υπεύθυνους για την πολιτική της ηλεκτρονικής διακυβέρνησης και εγκρίθηκε ομόφωνα στις 24 Νοεμβρίου του 2005 στο Ηνωμένο Βασίλειο (Schmitz, 2009).

Οι άδειες χρήσης ελεύθερων λογισμικών συνήθως αναφέρονται ως άδειες copyleft σε αντίθεση με τους περιορισμούς που δίνει ο νόμος για τα πνευματικά δικαιώματα “copyright”. Ο τύπος άδειας copyleft επιχειρεί να δώσει επιπλέον δικαιώματα στον χρήστη και παραχωρεί ελευθερίες στην χρήση, τροποποίηση, και διανομή ενός πνευματικού έργου. Συνήθως έχουν τον περιορισμό, ότι κάθε αντίγραφο ή παράγωγο θα διανέμεται με την ίδια άδεια χρήσης παραχωρώντας τις ίδιες ελευθερίες.

Όπως αναφέρει το Ίδρυμα Ελεύθερου Λογισμικού “GNU.com” (2015) ο απλούστερος τρόπος να κάνει κανείς ένα πρόγραμμα ελεύθερο είναι να το θέσει σε δημόσια κτήση, ελεύθερο δικαιωμάτων. Αυτό επιτυγχάνεται πρακτικά με το να διανεμηθεί το λογισμικό με άδεια χρήσης Ελεύθερου Λογισμικού. Μερικές από τις πιο διαδεδομένες άδειες χρήσης με βάση το επίσημο διαδικτυακό τόπο Λογισμικού Ανοιχτού κώδικα (opensource.org) είναι:

- GNU General Public License (GPL)
- GNU Library or "Lesser" General Public License (LGPL)
- MIT license
- Apache License 2.0
- Mozilla Public License 2.0

- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- Common Development and Distribution License
- Eclipse Public License

Τα περισσότερα λογισμικά και τα σχέδια υλικών κατασκευών που χρησιμοποιούνται στη διπλωματική διανέμονται με την άδεια χρήσης GNU General Public License Version 2.

### 3.4 Κόστος συστημάτων καταγραφής

Υπάρχει ένας πολύ μεγάλος αριθμός διαφορετικών συστημάτων καταγραφής περιβαλλοντικών παραμέτρων, το κόστος αυτών, κυμαίνεται από λίγα ευρώ μέχρι εκατοντάδες ή και χιλιάδες ευρώ σε ορισμένες περιπτώσεις. Το κόστος διαμορφώνεται κυρίως από την ποιότητα κατασκευής τους, τις δυνατότητες και τα χαρακτηριστικά των αισθητήρων που φέρουν.

Τα πολύ χαμηλού κόστους, εμπορικά συστήματα μέτρησης περιβαλλοντικών παραμέτρων κάτω των πενήντα ευρώ, δεν λήφθηκαν υπόψη για τη χρήση τους στην παρούσα μελέτη, λόγω της πολύ χαμηλής ποιότητας και της πολύ χαμηλής αξιοπιστίας των αισθητήρων που διαθέτουν. Επίσης τα πολύ χαμηλού κόστους συστήματα δεν έχουν τη δυνατότητα αποθήκευσης ή μετάδοσης δεδομένων, δεν έχουν καμία δυνατότητα ρύθμισης, βαθμονόμησης, παραμετροποίησης και το σημαντικότερο καμία δυνατότητα επέκτασης δηλαδή προσθήκης άλλων αισθητήρων και δυνατοτήτων.

Συστήματα μέτρησης περιβαλλοντικών παραμέτρων που το κόστος τους κυμαίνεται μεταξύ πενήντα και εκατό ευρώ, παρέχουν αρκετά αξιόπιστες μετρήσεις με βάση τις προδιαγραφές των κατασκευαστών, ωστόσο και αυτά δεν έχουν καμία δυνατότητα προσθήκης άλλων αισθητήρων και δυνατοτήτων.

Συστήματα του εμπορίου που έχουν τη δυνατότητα παραμετροποίησης και επέκτασης ξεπερνάνε κατά πολύ τα εκατό ευρώ. Αν αναλογιστεί κανείς, πως μια μελέτη μπορεί να απαιτεί να χρησιμοποιηθούν πάνω από δέκα σταθμοί, το κόστος αυξάνεται κατά πολύ και πολύ εύκολα μπορεί να ξεπεραστεί ο αρχικός προϋπολογισμός.

Το κόστος κατασκευής των συστημάτων καταγραφής περιβαλλοντικών παραμέτρων, που δημιουργήθηκαν στα πλαίσια της διπλωματικής διατριβής, κυμαίνεται από δεκατέσσερα ως εικοσιένα ευρώ περίπου (Πίνακας 5) και τα χαρακτηριστικά αυτών ξεπερνάνε κατά πολύ αυτά των συστημάτων από πενήντα ευρώ και άνω, ανταγωνιζόμενα αυτά των εκατό ευρώ. Σαφώς δεν υπάρχει καμία αμφιβολία πως είναι η καλύτερη επιλογή όσον αφορά τη σχέση τιμής – απόδοσης. Το κόστος των συστημάτων αυτών διαμορφώνεται κυρίως ανάλογα με τον αισθητήρα και το σύστημα τροφοδοσίας που θα επιλεγεί.



# 4

## Υλοποίηση

Για τις ανάγκες της διπλωματικής εργασίας κατασκευάστηκαν σταθμοί μέτρησης περιβαλλοντικών παραμέτρων. Στο κεφάλαιο αυτό θα αναλυθεί ο τρόπος κατασκευής και προγραμματισμού των σταθμών. Τα ηλεκτρονικά εξαρτήματα που χρησιμοποιήθηκαν για την κατασκευή των σταθμών στηρίζονται σε αρχιτεκτονικές ανοικτού κώδικα, εκ των οποίων τα σχέδια των εξαρτημάτων αυτών διατίθενται στο διαδίκτυο, από όπου μπορεί κανείς να τα προμηθευτεί σε ηλεκτρονική μορφή χωρίς καμιά οικονομική επιβάρυνση. Τα ολοκληρωμένα ηλεκτρονικά εξαρτήματα, δηλαδή οι ολοκληρωμένες ηλεκτρονικές πλακέτες με όλα τα απαραίτητα ηλεκτρονικά εξαρτήματα τοποθετημένα στις πλακέτες, αποκτήθηκαν μέσω διαδικτύου από διαφορετικούς προμηθευτές, συναρμολογήθηκαν ανάλογα με το σκοπό χρήσης, όπως περιγράφεται στη συνέχεια και τοποθετήθηκαν σε κουτιά ηλεκτρονικών.

Κατασκευάστηκαν τρεις διαφορετικοί σταθμοί μέτρησης περιβαλλοντικών παραμέτρων. Ο πρώτος σταθμός που κατασκευάστηκε, είναι ένα αυτόνομο σύστημα καταγραφής των δεδομένων σε κάρτα μνήμης SD που με τη χρήση μπαταρίας 2000mAh έχει τη δυνατότητα καταγραφής περισσότερων των 200.000 μετρήσεων. Ο δεύτερος σταθμός κατασκευάστηκε με γνώμονα την ελαχιστοποίηση του βάρους, για να τοποθετηθεί σε τηλεκατευθυνόμενο τετρακόπτερο (Drone). Για να ελαχιστοποιηθεί το βάρος χρησιμοποιήθηκαν μόνο οι απολύτως απαραίτητες μονάδες για τη λειτουργία του σταθμού και μπαταρία μικρής χωρητικότητας. Αυτό έχει ως αποτέλεσμα τον περιορισμό του αριθμού μετρήσεων που μπορεί να καταγράψει. Ο τρίτος σταθμός που κατασκευάστηκε έχει τη δυνατότητα σύνδεσης στο διαδίκτυο, μέσω της μονάδας ασύρματου δικτύου ESP8266.

### 4.1 Επιλογή υλικών

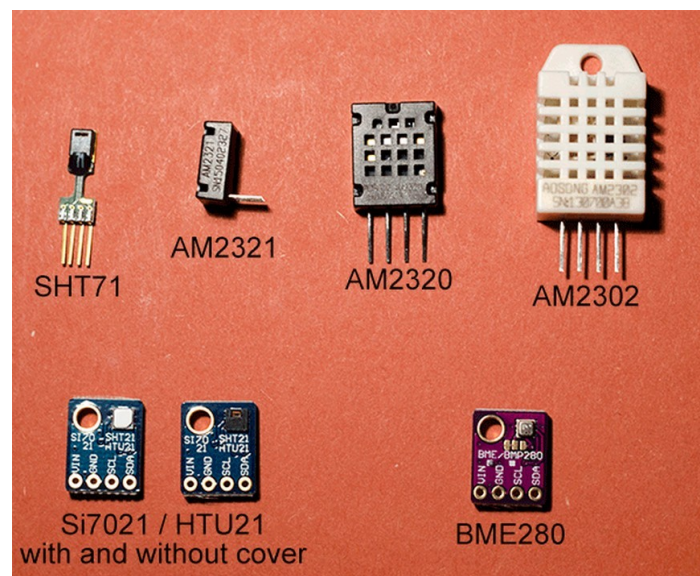
#### 4.1.1 Επιλογή αισθητήρων

Στην αγορά μπορεί κανείς να βρει ένα μεγάλο αριθμό διαφορετικών αισθητήρων καταγραφής βασικών μετεωρολογικών παραμέτρων (θερμοκρασία και υγρασία). Σε γενικές γραμμές, οι περισσότεροι αισθητήρες εμπίπτουν σε μία από τις δύο κατηγορίες είτε ψηφιακοί είτε αναλογικοί αισθητήρες. Στους ψηφιακούς αισθητήρες πραγματοποιείται μια μετατροπή των δεδομένων μετάδοσης σε ψηφιακό σήμα. Οι ψηφιακοί αισθητήρες αντικαθιστούν σταδιακά τους αναλογικούς, γιατί είναι ικανοί να ξεπεράσουν τα μειονεκτήματα των αναλογικών αισθητήρων (εκτός από περιπτώσεις που είναι αναγκαία η χρήση άλλου τύπου αισθητήρων).

Οι πιο διαδεδομένοι ψηφιακοί αισθητήρες χαμηλού κόστους που καταγράφουν τη θερμοκρασία και την υγρασία είναι οι AM2302 (DHT22), AM2320, AM2321, SHT71,

HTU21D, Si7021, και ο BME280 (Σχήμα 4.1). Αξίζει να σημειωθεί πως ο αισθητήρας BME280 καταγράφει επίσης και την ατμοσφαιρική πίεση. Στον πίνακα 1 αναφέρονται οι αισθητήρες και οι εταιρείες παραγωγής αυτών, η τιμές είναι κατά προσέγγιση η μέση λιανική τιμή αγοράς.

Ένας επίσης πολύ γνωστός και ευρέως διαδεδομένος αισθητήρας χαμηλού κόστους είναι ο αισθητήρας DHT11 της εταιρείας Aosong. Ο αισθητήρας αυτός χρησιμοποιείται σε όλες τις συσκευές χαμηλού κόστους που απαιτείται η μέτρηση της θερμοκρασίας και της υγρασίας. Το κόστος του αισθητήρα δεν ξεπερνάει το 1€ αλλά είναι εντελώς αναξιόπιστος για να χρησιμοποιηθεί για επιστημονικές μετρήσεις. Για το λόγο αυτό δεν λαμβάνεται υπόψη ούτε στην αρχική σύγκριση για την επιλογή αισθητήρα, για τις μετρήσεις που απαιτούνται στη παρούσα διπλωματική εργασία.



Σχήμα 4.1: Σύγκριση των διαφόρων ψηφιακών αισθητήρων. (Kandrsmith 2016)

Αισθητήρες	Εταιρεία	Κόστος* (€)
AM2302 (DHT22)	Aosong	3
AM2320	Aosong	2.5
AM2321	Aosong	3
HTU21	Measurement Specialties	4.5
Si7021	Silicon Labs	3
SHT71	Sensirion	40
BME280	Bosch Sensortec	5.5

Πίνακας 1: Αισθητήρες χαμηλού κόστους. (\*Κόστος κατά προσέγγιση μέση λιανική τιμή αγοράς)

Ο αισθητήρας AM2302 (DHT22) είναι μια βελτιστοποιημένη έκδοση του DHT11 με ελαφρώς καλύτερες επιδόσεις. Όπως αναφέρει ο καθηγητής R. Kandrsmith (2016) του

πανεπιστημίου Liverpool John Moores στη μελέτη που έκανε με έξι αισθητήρες AM2302, ο κάθε αισθητήρας έδινε διαφορετικές μετρήσεις με μεγάλες αποκλίσεις μεταξύ τους. Αν αναλογιστεί κανείς πως η τιμή αγοράς του αισθητήρα AM2302 είναι ελαφρώς μεγαλύτερη ή ίση με άλλους αισθητήρες με καλύτερες επιδόσεις, δεν υπάρχει λόγος να χρησιμοποιηθεί αυτός ο αισθητήρας.

Ο αισθητήρας AM2320 είναι ουσιαστικά η πιο πρόσφατη έκδοση του AM2302 με την δυνατότητα επικοινωνίας μέσω του πρωτοκόλλου I2C, είναι λίγο μικρότερος αλλά με πανομοιότυπη συνδεσμολογία (pin-out). Ο αισθητήρας AM2321 φαίνεται να είναι ίδιος με τον AM2320, αλλά με πολύ μικρότερες φυσικές διαστάσεις. Θα μπορούσε να είναι ηλεκτρικά ταυτόσημες, όπως αναφέρει ο R. Kandrsmith (2016), επίσης αναφέρει πως στην ιστοσελίδα του κατασκευαστή υπάρχει αναφορά διακοπής παραγωγής αυτού του αισθητήρα και ότι θα αντικατασταθεί από τον AM2322.

Οι αισθητήρες HTU21D και Si7021 παρόλο που κατασκευάζονται από διαφορετικές εταιρίες έχουν απόλυτη συμβατότητα. Οι αισθητήρες αυτοί χρησιμοποιούν ίδιο πρωτόκολλο επικοινωνίας και έχουν ακόμα και την ίδια διεύθυνση I2C, για το λόγο αυτό μπορεί εύκολα ο ένας αισθητήρας να αντικαταστήσει τον άλλον χωρίς την ανάγκη παραμετροποίησης του συστήματος.

Ο αισθητήρας SHT71 όπως αναφέρει ο R. Kandrsmith (2016) είναι πιο γραμμικός, πιο σταθερός στην πάροδο του χρόνου και αναμφοισβήτητα ο αισθητήρας με τις μικρότερες αποκλίσεις από τις αναφερόμενες τιμές των προδιαγραφών σε σύγκριση με τους προαναφερόμενους αισθητήρες όπου και δικαιολογείται το αυξημένο κόστος του.

Ο αισθητήρας BME280 της εταιρίας Bosch πέρα από τη θερμοκρασία και την υγρασία καταγράφει επίσης και τη ατμοσφαιρική πίεση. Πιθανό ο αισθητήρας αυτός να λειτουργεί με τελείως διαφορετικό τρόπο από τους άλλους αισθητήρες. Η μόνη πληροφορία που δίνει η εταιρεία, είναι πως, είναι συσκευή τύπου MEMS (μικρο ηλεκτρομηχανικές συσκευές), χωρίς να αναφέρει λεπτομέρειες για το πως λειτουργεί ο συγκεκριμένος αισθητήρας. Όπως θα δούμε στη συνέχεια είναι εξαιρετικά αξιόπιστος παρέχοντας μεγάλη ακρίβεια μετρήσεων, για το λόγο αυτό επιλέχθηκε για τις μετρήσεις της συγκεκριμένης διπλωματικής εργασίας.

Τα χαρακτηριστικά που μελετούνται στη συνέχεια είναι η ακρίβεια της θερμοκρασίας, της υγρασίας και η σχέση αυτών, δηλαδή κατά πόσο επηρεάζεται η θερμοκρασία από την αλλαγή της υγρασίας και το αντίστροφο, επίσης μελετάται η υστέρηση (Hysteresis) και η ταχύτητα απόκρισης (Response Speed).

Οι τιμές του πίνακα 2 λαμβάνονται από τα δελτία προδιαγραφών της κάθε εταιρίας όσον αφορά τη σχετική υγρασία. Όπως αναφέρει ο R. Kandrsmith. Τα δελτία προδιαγραφών της εταιρείας Aosong δεν καθορίζουν μέγιστα όρια ανοχής για τις περισσότερες παραμέτρους, απλά «τυπικές» τιμές. Φαίνεται συνεπώς ότι οι συγκεκριμένοι αισθητήρες δεν είναι εγγυημένοι για να πληρούν αυτές τις προδιαγραφές. Για όλους τους άλλους αισθητήρες, αναφέρει ο ίδιος, ότι οι τιμές αντιστοιχούν στις μέγιστες ανοχές και γενικά προσφέρουν καλύτερες «τυπικές» προδιαγραφές.

<b>Προδιαγραφές Κατασκευαστών (Υγρασίας)</b>						
	AM2302	AM2320/21	SHT71	HTU21D	Si7021	BME280
Operating Range	0–100	0–100	0–100	0–100	0–100	0–100
Absolute accuracy (%RH, 25°C)	±3% (10-90%) ±5% (<10, >90%)	±3% (10-90%) ±5% (<10, >90%)	±3% (20-80%) ±5% (<20, >80%)	±3% (20-80%) ±5% (<20, >80%)	±3% (0-80%) ±5% (>80%)	±3% (20-80%)
Repeatability (%)	±0.3	±0.1	±0.1	-	±0.025	-
Long term stability (%/year)	0.5	0.5	0.5	0.5	0.25	0.5
1/e Response(sec)	5	5	8	5	18	1
Voltage supply(V)	3.3–5.5	3.1–5.5	2.4–5.5	1.5–3.6	1.9–3.6	1.71–3.6

Πίνακας 2: Πίνακας χαρακτηριστικών με βάση τις προδιαγραφές των εταιριών για την υγρασία (Kandrsmith 2016)

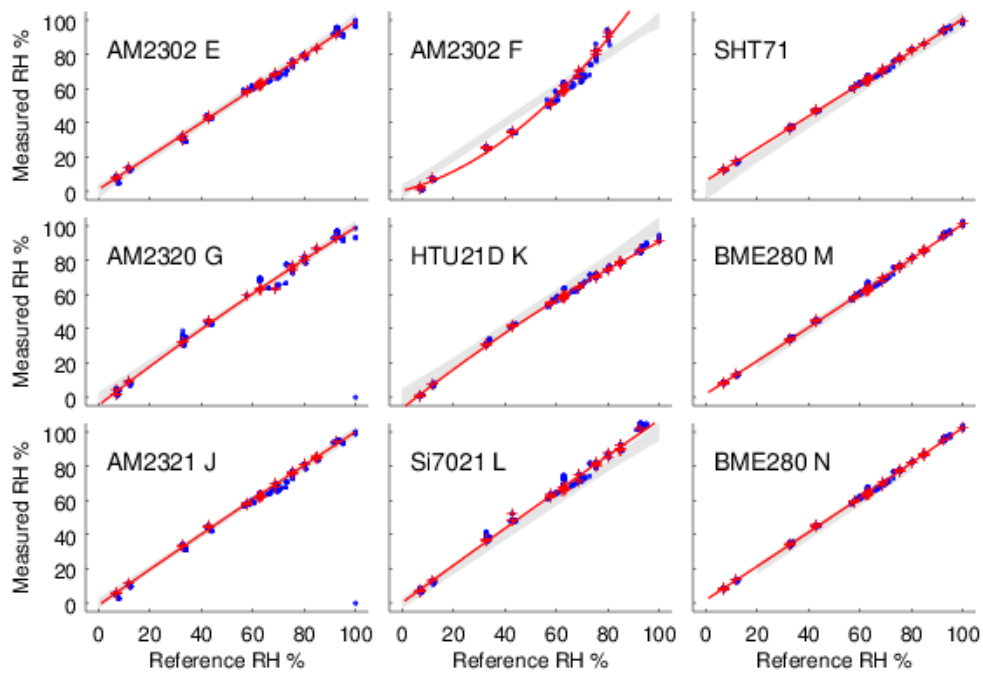
O R. Kandrsmith στη μελέτη που έκανε, χρησιμοποίησε τους αισθητήρες σε έντεκα κορεσμένα διαλύματα και σε αποσταγμένο νερό. Οι αισθητήρες αφέθηκαν να σταθεροποιηθούν για λίγες ώρες με κάθε διάλυμα. Στη μελέτη του χρησιμοποίησε σύστημα όπου η θερμοκρασία μεταβαλλόταν αυτόματα με βήματα σε ένα προκαθορισμένο εύρος, επιτρέποντας τους αισθητήρες να “προσαρμοστούν” σε κάθε θερμοκρασία και στη συνέχεια γινόταν η καταγραφή των μετρήσεων σε μια βάση δεδομένων, πριν από την μεταβίβαση στην επόμενη θερμοκρασία.

Στα διαγράμματα του R. Kandrsmith (2016) που ακολουθούν, παρατηρούμε πως τα δεδομένα για τον αισθητήρα AM2320 G είναι λίγο πιο αραιά από τους άλλους. Όπως διευκρινίζει ο ίδιος στη μελέτη του, αυτό συνέβη διότι και οι δύο συσκευές AM2320 αποτύχαιναν να δώσουν σταθερές μετρήσεις κατά τα διαστήματα όπου η σχετική υγρασία ήταν πάνω από 50% RH. Τα δεδομένα που λάμβανε ο μικροελεγκτής τις περισσότερες φορές ήταν “κενά” δηλαδή 00 ή FF. Για τον λόγο αυτό ο Kandrsmith αναγκάστηκε να προγραμματίσει τον μικροελεγκτή να απορρίπτει τα “κενά” δεδομένα. Όπως αναφέρει ο ίδιος όλοι οι άλλοι αισθητήρες συνέχισαν ανεπηρέαστοι, αλλά οι τιμές καταγράφονταν από τον προαναφερθέντα αισθητήρα μόνο όταν ήταν λογικές.

Παρομοίως, ο αισθητήρας AM2321 J απέτυχε να επιστρέψει τιμές όταν η υγρασία ήταν πάνω από 90%. Όπως αναφέρει ο R. Kandrsmith (2016), ο αισθητήρας επαναλειτούργουσε μόλις η σχετική υγρασία έπεφτε κάτω από 90%. Επίσης κάθε φορά που χρησιμοποιούσε διαλύματα H<sub>2</sub>O (100%) ή K<sub>2</sub>NO<sub>3</sub> (92%), ο αισθητήρας σταματούσε να λειτουργεί μετά από περίπου 10 με 15 δευτερόλεπτα, αρνούμενος να ανταποκριθεί στα αιτήματα επικοινωνίας μέσω του πρωτοκόλλου I2C.

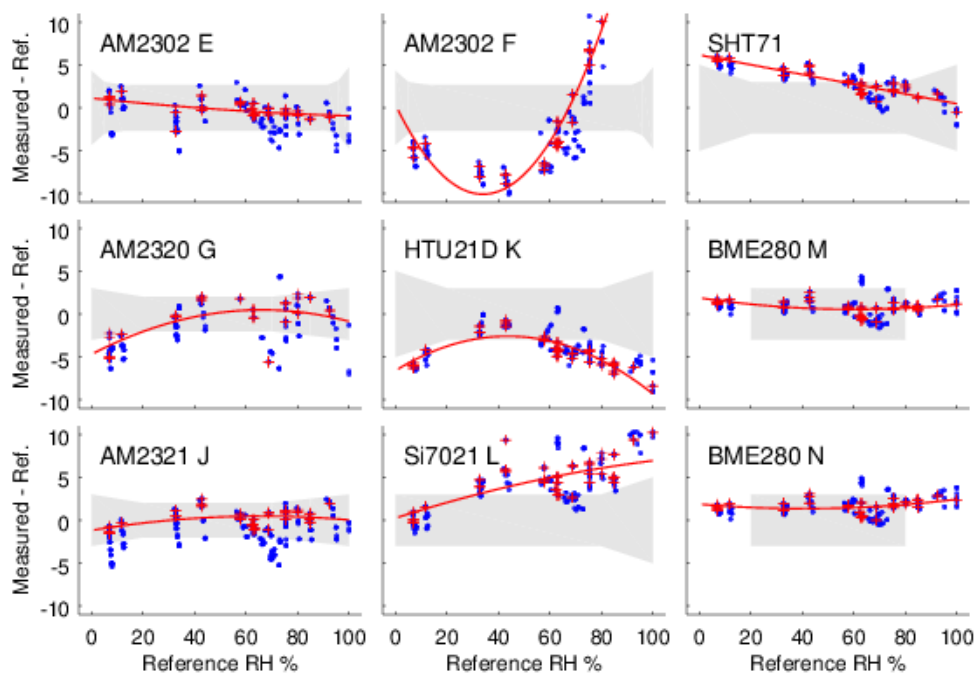
Στο πρώτο διάγραμμα (4.2) φαίνεται η απόκριση των αισθητήρων σε διαφορετικές υγρασίες αναφοράς 0-100%. Τα διαγράμματα 4.2 και 4.3 περιέχουν τα ίδια δεδομένα όμως απεικονίζονται με διαφορετικούς άξονες. Το διάγραμμα 4.2 δείχνει τις πραγματικές τιμές σχετικής υγρασίας που μετριοούνται και το διάγραμμα 4.3 δείχνει τις αποκλίσεις από μια τέλεια βαθμονόμηση. Τα δεδομένα με κόκκινο χρώμα είναι σε θερμοκρασία 25 ° C. Σε αυτή τη θερμοκρασία ορίζουν όλοι οι κατασκευαστές τις μέγιστες και ελάχιστες τιμές για την σχετική υγρασία. Τα δεδομένα που λαμβάνονται σε άλλες θερμοκρασίες εμφανίζονται με μπλε χρώμα. Η σκιασμένη γκρι περιοχή είναι η καθορισμένη ανοχή από τις προδιαγραφές

των κατασκευαστών. Όποια τιμή που βρίσκεται έξω από τη σκιασμένη περιοχή δεν πληρεί τις προδιαγραφές.



Σχήμα 4.2: Γραφικές παραστάσεις από τις τιμές της υγρασίας από κάθε αισθητήρα συναρτήσει της πραγματικής υγρασίας. (Kandrsmith 2016)

Οι γραφικές παραστάσεις των διαγραμμάτων που ακολουθούν δείχνουν το σφάλμα του κάθε αισθητήρα ως συνάρτηση της υγρασίας.

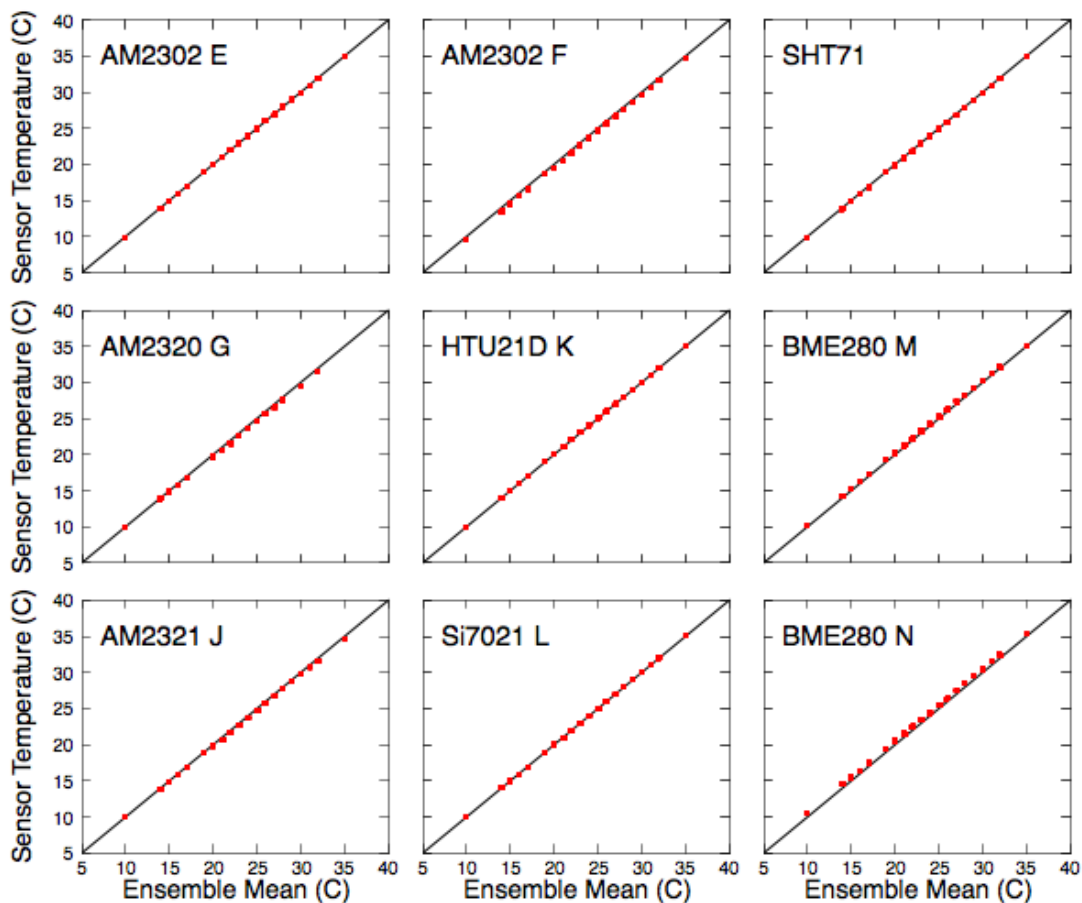


Σχήμα 4.3: Γραφικές παραστάσεις που δείχνουν την μεταβολή των τιμών υγρασίας από γνωστές τιμές αναφοράς. (Kandrsmith 2016)

Πέρα από την υγρασία σημαντικό είναι και η μέτρηση της ακριβείας της θερμοκρασίας που δίνει ο κάθε αισθητήρας. Ο Kandrsmith (2016) στη μελέτη του δεν βαθμονόμησε τους αισθητήρες με κάποια σταθερή πηγή μέτρησης αλλά άπλα σύγκρινε τις διαφορές των μετρήσεων του κάθε αισθητήρα με το μέσο όρο όλων των αισθητήρων που συμπεριέλαβε στις δόκιμες της μελέτης του (Σχήμα 4.4).

<b>Προδιαγραφές Κατασκευαστών (Θερμοκρασίας)</b>					
	AM2302	AM2320/21	HTU21D	Si7021	BME280
Operating Range	-40 to 80 °C	-40 to 80 °C	-40 to +125 °C	-40 to +125 °C	-40 to +85 °C
Absolute accuracy	±0.5°C	±0.5°C	±0.4°C	±0.4 –10< tA < 85 °C (max)	±0.5-25°C, ±1.0-0..65°C)
Repeatability (°C)	±0.2	±0.2	0.04	0.08	0.005
Output resolution	0.1°C	0.1°C	0.04°C	0.04°C	0.01°C
1/e Response(sec)	5	<5	5	5.1	1
Voltage supply(V)	3.3–5.5	3.1–5.5	1.5–3.6	1.9–3.6	1.71–3.6

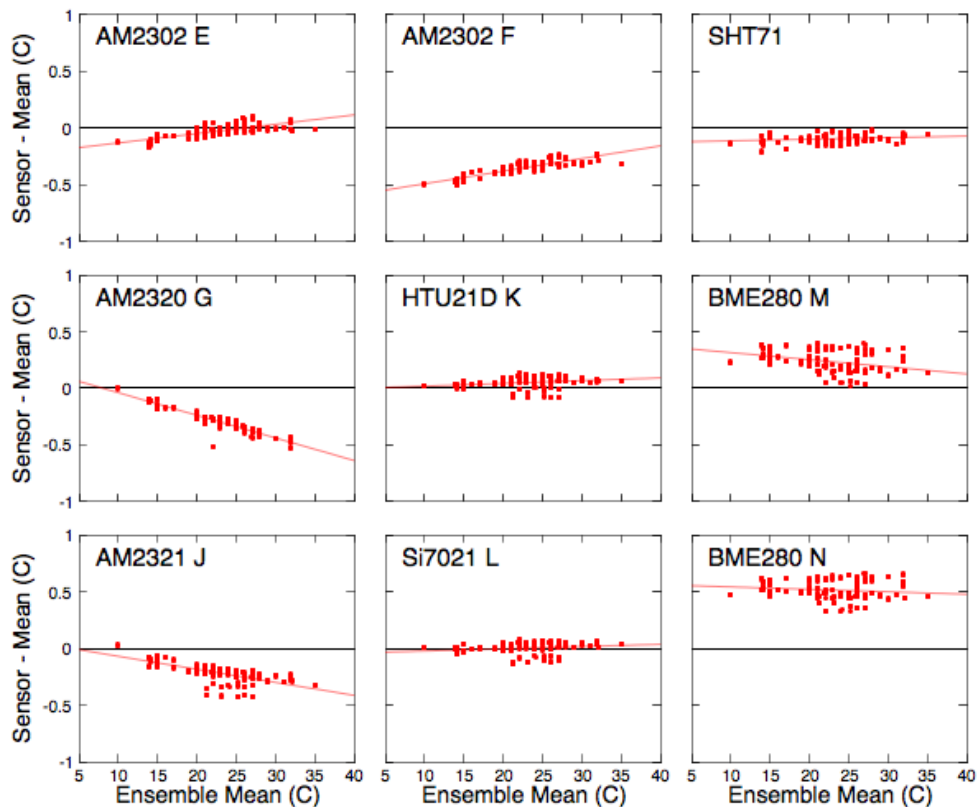
Πίνακας 3: Πίνακας χαρακτηριστικών με βάση τις προδιαγραφές των εταιριών για την θερμοκρασία.



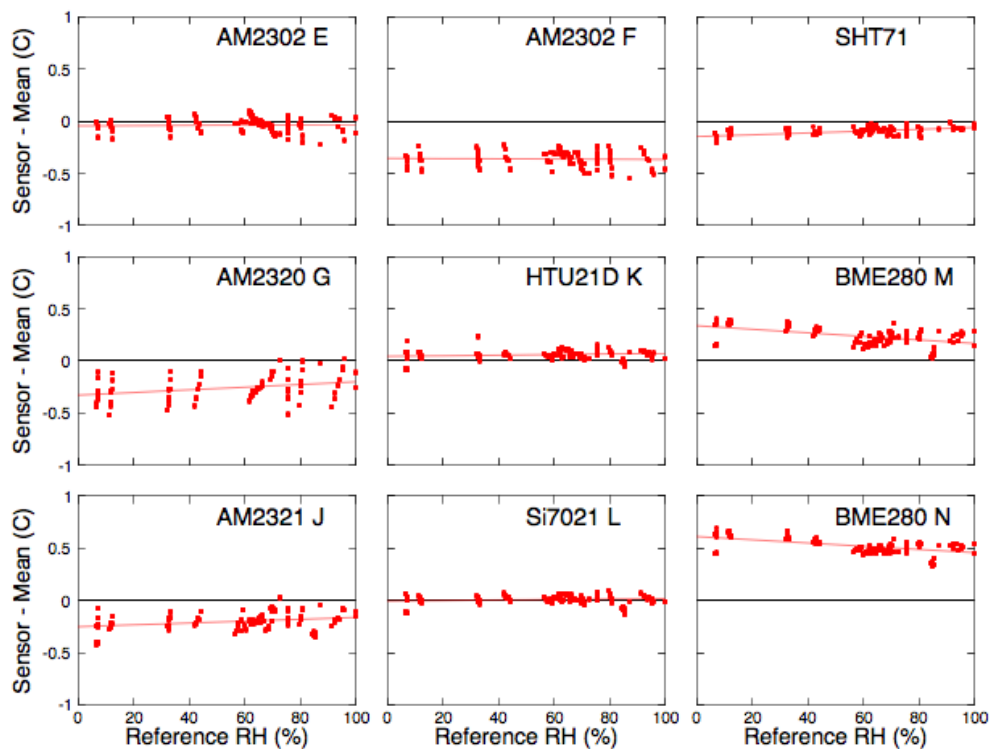
Σχήμα 4.4: Γραφήματα που δείχνουν την απόκλιση των μετρήσεων θερμοκρασίας του κάθε αισθητήρα από το μέσο όρο όλων. (Kandrsmith 2016)

Όπως φαίνεται στα διαγράμματα (4.5 και 4.6) όλοι οι αισθητήρες δεν υπερβαίνουν τις τιμές των προδιαγραφών που δίνουν οι κατασκευαστές δηλαδή την ανεκτικότητα των  $\pm 0,5^{\circ}\text{C}$ . Οι αισθητήρες της Bosch έδωσαν τις υψηλότερες τιμές, οπού στο διάγραμμα 4.5 δείχνουν τη μεγαλύτερη διασπορά, γεγονός που υποδηλώνει κακή επαναληψιμότητα. Ωστόσο, στη μελέτη του R. Kandrsmith (2016) παρατηρούμε πως οι μετρήσεις της θερμοκρασίας των αισθητήρων της Bosch επηρεάζονται πολύ λιγότερο από την υγρασία (Σχήμα 4.6). Οι διαφορές στην επαναληψιμότητα με σταθερή υγρασία είναι πολύ μικρές της τάξης των  $0,1^{\circ}\text{C}$ , οπού είναι εντός των προδιαγραφών του κατασκευαστή ( $0,5^{\circ}\text{C}$ ).

Σε αντίθεση, ο αισθητήρας AM2320 G έχει μεγάλο σφάλμα ως συνάρτηση της υγρασίας (Σχήμα 4.6), αλλά έχει το μικρότερο σφάλμα ως συνάρτηση της θερμοκρασίας. (Σχήμα 4.5).



Σχήμα 4.5: Γραφήματα που δείχνουν την απόκλιση των μετρήσεων θερμοκρασίας του κάθε αισθητήρα από το μέσο όρο όλων. (Kandrsmith 2016)

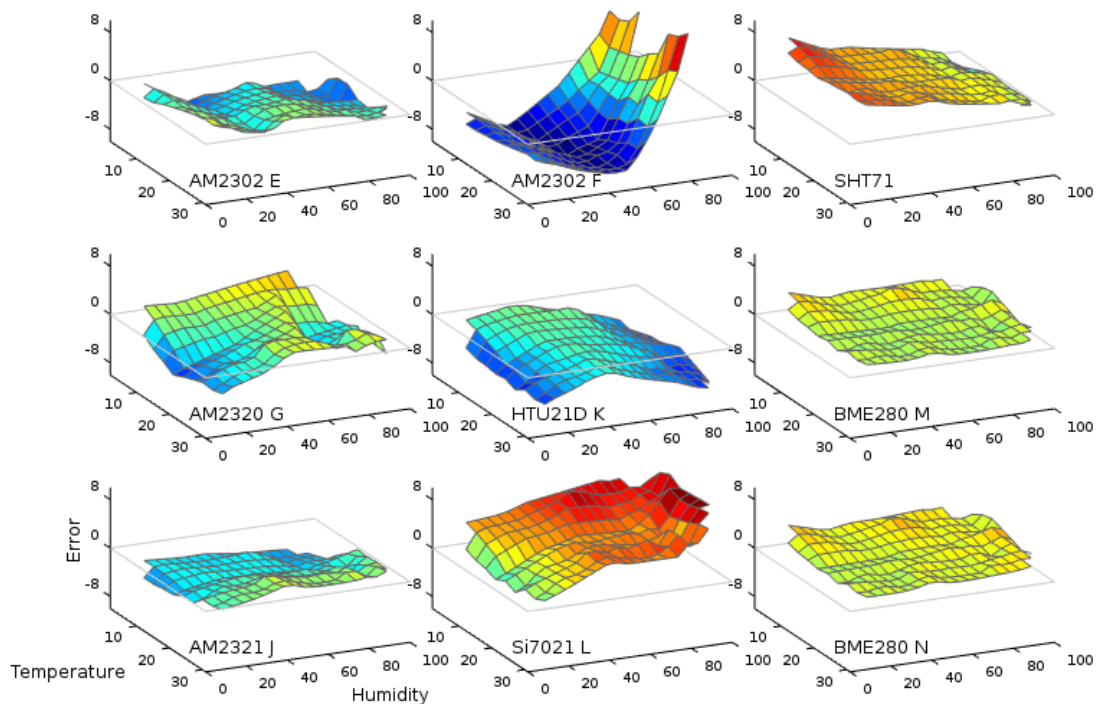


Σχήμα 4.6: Γραφήματα που δείχνουν την απόκλιση των μετρήσεων θερμοκρασίας του κάθε αισθητήρα σε συνάρτηση με την υγρασία.

Ο R. Kandrsmith (2016) από το πανεπιστήμιο του Liverpool John Moores, δημιούργησε επίσης διαγράμματα με επιφάνειες που δείχνουν την απόκλιση του κάθε αισθητήρα από την πραγματική σχετική υγρασία ως συνάρτηση της θερμοκρασίας και της υγρασίας (Σχήμα 4.7). Ένας τέλειος αισθητήρας θα παρουσίαζε μια επίπεδη επιφάνεια στο σημείο μηδέν. Το μπλε παρουσιάζει χαμηλές τιμές και το κόκκινο ψηλές.

Οι αισθητήρες AM2302, AM2320 και AM2321 δίνουν αρκετά αξιόπιστα δεδομένα με μικρο σφάλμα και μικρή απόκλιση με την μεταβολή της θερμοκρασίας. Θα πρέπει όμως να σημειωθεί πως οι συγκεκριμένοι αισθητήρες έδωσαν τα καλύτερα αποτελέσματα ανάμεσα άλλων ίδιων αισθητήρων σε προηγούμενες μελέτες που είχε κάνει ο ίδιος. Οι μετρήσεις των άλλων αισθητήρων ίδιου τύπου ήταν πολύ διαφορετικές με μεγάλες αποκλίσεις από τις προδιαγραφές του κατασκευαστή. Αυτό σημαίνει πως οι συγκεκριμένοι αισθητήρες δεν είναι αξιόπιστοι και πως απαιτείται να βαθμονομηθεί κάθε αισθητήρας χωριστά χωρίς όμως εγγυημένα αποτελέσματα. Αυτός είναι και ένας από τους σημαντικότερους λόγους που δεν χρησιμοποιήθηκαν οι συγκεκριμένοι αισθητήρες.





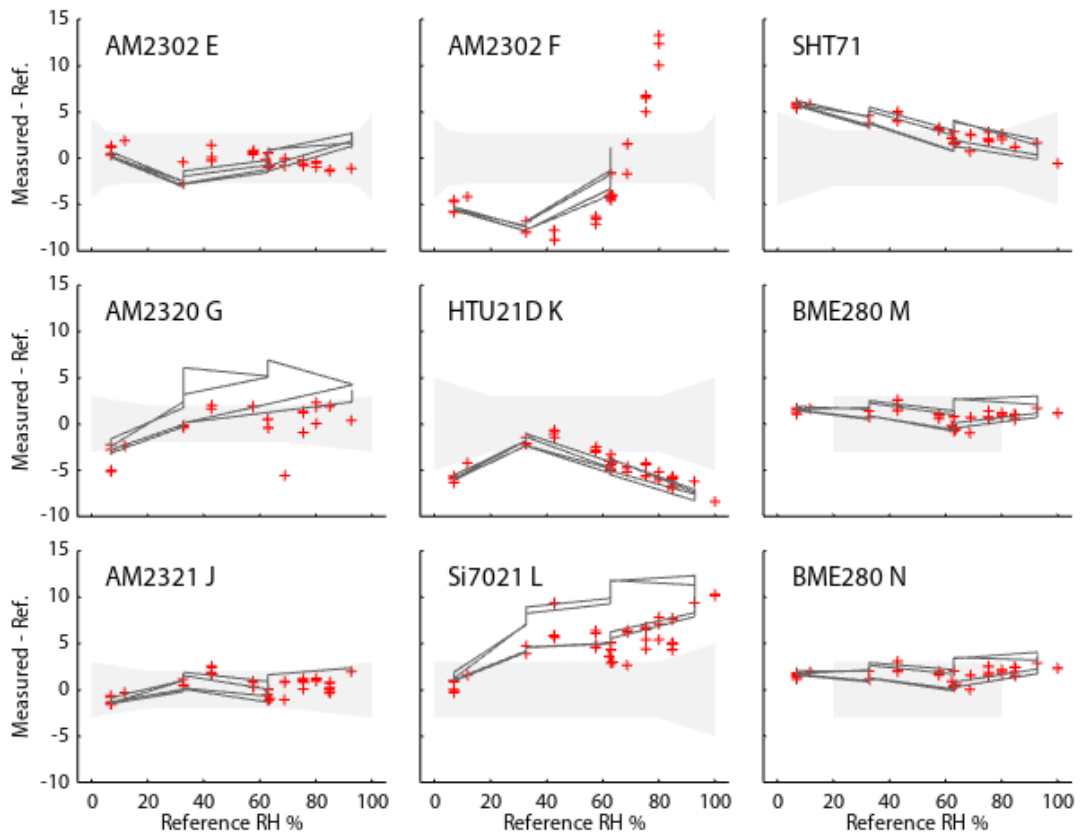
Σχήμα 4.7: Επιφάνειες που δείχνουν την απόκλιση του κάθε αισθητήρα από την πραγματική σχετική υγρασία ως συνάρτηση της θερμοκρασίας και της υγρασίας. (Kandrsmith 2016)

Ένα ακόμα σημαντικό χαρακτηριστικό που θα πρέπει να ληφθεί υπόψη, είναι η υστέρηση που έχει ο κάθε αισθητήρας, δηλαδή, η τάση που έχει ένας αισθητήρας να εξαρτάται από την προηγούμενη κατάσταση του. Ένας αισθητήρας με υστέρηση θα τείνει να δώσει χαμηλές ενδείξεις αν έχει πρόσφατα εκτεθεί σε ξηρό αέρα και θα τείνει να δώσει υψηλές όταν έχει πρόσφατα εκτεθεί σε υγρό αέρα.

Ο Kandrsmith (2016) στη μελέτη του για να μετρήσει την υστέρηση είχε όλους τους αισθητήρες σε σταθερή θερμοκρασία 25 ° C, και τους εκθέτεται με σειρά στα διαλύματα  $K_2NO_3$  (92%),  $NH_4NO_3$  (60%),  $MgCl$  (33%),  $NaOH$  (7%),  $MgCl$  (33%),  $NH_4NO_3$  (60%) και  $K_2NO_3$  (92%). Κάθε επίπεδο υγρασίας πραγματοποιήθηκε για δώδεκα ώρες και η μετρούμενη τιμή καταγραφόταν κάθε μισή ώρα. Έγιναν δύο πλήρεις κύκλοι από υγρή (92% σχετική υγρασία) σε ξερή ατμοσφαίρα (7% σχετική υγρασία).

Στο σχήμα 4.8 φαινόνταν οι αποκλίσεις από τη γνωστή τιμή αναφοράς, όπου η υγρασία μεταβάλετε από υψηλά σε χαμηλά και πίσω σε υψηλά επίπεδα (δύο πλήρεις κύκλοι όπου η σχετική υγρασία μεταβαλλόταν από 92%, 60%, 33%, 7%, 33%, 60%, 92%). Για τους περισσότερους αισθητήρες οι μετρήσεις από την άνοδο και την πτώση της υγρασίας είναι αρκετά κοντά μεταξύ τους, δείχνοντας μικρή υστέρηση. Μια σημαντική επίδραση δείχνουν οι αισθητήρες AM2302 και AM2321. Ο αισθητήρας AM2302 F έδινε τιμές 99,9% στο διάλυμα  $K_2NO_3$  έτσι οι τιμές αποκλειστήκαν και δεν απεικονίζονται στο διάγραμμα. Ο AM2321 απέτυχε γενικά να δώσει οποιαδήποτε έξοδο με το διάλυμα  $K_2NO_3$ , στα άλλα διαλύματα παρείχε περιστασιακά μια διακοπτόμενη ανάγνωση, αλλά αποτύγχανε σε μεγάλο βαθμό όταν η υγρασία ήταν πάνω από 90%. Ο αισθητήρας AM2320 απέτυχε να δώσει οποιαδήποτε έξοδο με το διάλυμα  $NH_4NO_3$ , σε γενικές γραμμές με τα άλλα διαλύματα

παρείχε επίσης περιστασιακά μια διακοπτόμενη ανάγνωση, αλλά αποτύγχανε σε μεγάλο βαθμό για υγρασίες περίπου 50-70%.



Σχήμα 4.8: Υστέρηση του κάθε αισθητήρα.

Με βάση τη μελέτη του R. Kandrsmith φαίνεται καθαρά η υπεροχή του αισθητήρα BME280. Όπως αναφέρει ο ίδιος, ο αισθητήρας αυτός δίνει με διαφορά τα καλύτερα αποτελέσματα. Ο αισθητήρας BME280 είναι ο πιο ακριβές στη προβλεπόμενη θερμοκρασία 25°C και δείχνει τη λιγότερη εξάρτηση από την υγρασία. Η καμπύλη υστέρησης φανερώνει ένα περιορισμένο εύρος μετατόπισης τιμών το οποίο ήταν σταθερό και στους δύο κύκλους μετρήσεων που εκτελέστηκαν. Είναι επίσης η μόνη περίπτωση όπου κάθε αισθητήρας παρείχε μετρήσεις εντός των προδιαγραφών. Ο καθηγητής Kandrsmith αναφέρει χαρακτηριστικά πως: «Η διαφορά είναι τόσο έντονη ώστε να με κάνει να πιστεύω ότι το αισθητήριο στοιχείο (του αισθητήρα) είναι κατά κάποιο τρόπο θεμελιωδώς διαφορετικό από τους άλλους αισθητήρες». Είναι προφανές με βάση όλα τα παραπάνω πως για τις μετρήσεις που απαιτούνται για τις ανάγκες εκπλήρωσης της μελέτης αυτής της διπλωμάτης εργασίας, η καλύτερη επιλογή με βάση τη σχέση τιμής - ποιότητας είναι ο αισθητήρας της εταιρείας Bosch Sensortec ο BME280, αν αναλογιστεί κανείς πως παρέχει εκτός των άλλων και μετρήσεις πίεσης. Πέρα από τους αισθητήρες BME280 χρησιμοποιήθηκε και ένας αισθητήρας Si7021 της εταιρείας Silicon Labs ο οποίος έδωσε εξίσου αξιόπιστες μετρήσεις.

#### 4.1.2 Επιλογή ηλεκτρονικών

Για την ελαχιστοποίηση του κόστους κατασκευής επιλέχθηκαν και χρησιμοποιήθηκαν ηλεκτρονικά εξαρτήματα και ολοκληρωμένα κυκλώματα ανοιχτής αρχιτεκτονικής ευρείας χρήσης που παράγονται μαζί, τύπου Arduino. Τα ολοκληρωμένα κυκλώματα προμηθεύτηκαν από κατασκευαστές που εδρεύουν στην Ιαπωνία και στην Κίνα ενώ τα ηλεκτρονικά εξαρτήματα από καταστήματα της Ελληνικής αγοράς. Στον πίνακα 4 που ακολουθεί αναφέρονται όλα τα υλικά που απαιτήθηκαν για την υλοποίηση όλων των σταθμών. Το συνολικό κόστος όλων των υλικών ανέρχεται στο ποσό των 110 ευρώ.

<u>Υλικά</u>	<u>Χαρακτηριστικά</u>	<u>Τεμάχια</u>	<u>Τιμή (€)/Τεμ</u>
<b>Ολοκληρωμένα κυκλώματα</b>			
Microcontroller board	Arduino Pro Mini 328 3.3v	6	2
WiFi Module	ESP8266 v1	1	3
SD Module	Mini Micro SD	6	0.75
Real Time Clock	Mini DS3231(RTC)	5	1.25
Sensor	BME280 (Temperature, Humidity, Pressure)	5	5
	Si7021 (Temperature, Humidity)	1	3
USB to Serial Converter	FTDI FT232RL	1	2
<b>Άλλα ηλεκτρονικά εξαρτήματα</b>			
Battery Charger	Micro USB 1A Lithium Battery Charger	6	0.75
Battery Holder	18650 Battery Clip Holder	5	0.5
Battery	18650 2000mAh 3.7v	5	3
	Li-Po 250mAh 3.7v	1	2
Switch	Cap Self-locking Type Square Button	5	0.25
Fuse	0.65A Resettable Fuse	5	0.25
PCB	2x8cm Double-Side	5	0.75
Enclosure	100x60x25mm	5	0.75
Other Electronics	Wires, Connections atc.	7	0.75
SD Card	Micro SD 8gb	5	3

Πίνακας 4: Υλικά που προμηθεύτηκαν για της ανάγκες κατασκευής των σταθμών.

Ο “εγκέφαλος” των κόμβων καταγραφής μετεωρολογικών παραμέτρων, είναι ο μικροελεγκτής ATmega328 της εταιρείας Atmel, ο οποίος βρίσκεται στη πλακέτα ηλεκτρονικών Arduino Pro Mini (Σχήμα 4.9). Ο μικροελεγκτής είναι στη πραγματικότητα ένας μικρός υπολογιστής που η έκδοσή των 3.3V “τρέχει” στα 8 MHz (η ταχύτητα του επεξεργαστή της έκδοσης των 5V είναι 16 Mhz), έχει 2 KB sram και 32 KB αποθηκευτικό χώρο. Επιλέχθηκε η έκδοση των 3.3V λόγω της μικρότερης κατανάλωσης σε σύγκριση με την έκδοση των 5V.

Για τη καταγραφή της πραγματικής ώρας και ημερομηνίας της κάθε μέτρησης, χρησιμοποιήθηκε το ρολόι υψηλής ακρίβειας DS3231 (Σχήμα 4.10) της εταιρείας Maxim Integrated.

Για την αποθήκευση των δεδομένων χρησιμοποιήθηκε ένα απλό υποσύστημα υποδοχής καρτών Micro SD (Σχήμα 4.11). Οι κάρτες SD λειτουργούν με τάση 3.3V, λόγω της χρήσης συστήματος που λειτουργεί με τάση 3.3V. Δεν απαιτείται περίπλοκο υποσύστημα μετασχηματισμού τάσης, όπως συνήθως χρησιμοποιείται για την υποδοχή αυτού του είδους καρτών.

Χρησιμοποιήθηκαν για την τροφοδοσία των κόμβων μπαταρίες τύπου 18650 οι οποίες έχουν χωρητικότητα περίπου 2000MAh. Οι μπαταρίες αυτού του τύπου δεν φέρουν σύστημα ελέγχου φόρτισης και αποφόρτισης, για τον λόγο αυτό είναι αναγκαία η χρήση υποσυστήματος για αυτό το είδος ελέγχου (Σχήμα 4.14).

Η επικοινωνία των κόμβων αυτών με υπολογιστή επιτυγχάνεται με τη χρήση υποσυστήματος μετατροπής συριακής επικοινωνίας στο δίαυλο επικοινωνίας USB και αντίστροφα (Σχήμα 4.15).

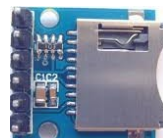
Για τον κόμβο με τη δυνατότητα μετάδοσης δεδομένων σε πραγματικό χρόνο χρησιμοποιήθηκε το ολοκληρωμένο κύκλωμα που φέρει τον μικροελεγκτή με δυνατότητες ασύρματης επικοινωνίας WiFi ESP8266 της εταιρείας Espressif (Σχήμα 4.13). Το ESP8266 προσφέρει μια πλήρη και αυτοδύναμη λύση δικτύωσης Wi-Fi, έχει ισχυρές δυνατότητες επεξεργασίας και αποθήκευσης που επιτρέπουν την άμεση σύνδεση με αισθητήρες και άλλες ειδικές συσκευές. Ο υψηλός βαθμός ολοκλήρωσης “on-chip” επιτρέπει την ελαχιστοποίηση της χρήσης εξωτερικών κυκλωμάτων. Το ESP8266 έχει ισχυρό επεξεργαστή των 80 Mhz, έχει μνήμη RAM 96 KiB και αποθηκευτικό χώρο 512 KiB με δυνατότητα επέκτασης ως 16 MiB.



Σχήμα 4.9: Arduino Pro Mini 3.3v



Σχήμα 4.10: Mini DS3231(RTC)



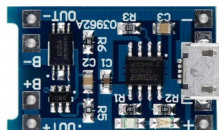
Σχήμα 4.11: Micro SD Module



Σχήμα 4.12: BME280 Module



Σχήμα 4.13: WiFi Module ESP8266 v1



Σχήμα 4.14: Micro USB 1A Lithium Battery Charger



Σχήμα 4.15: FTDI FT232RL

Οι αισθητήρες που χρησιμοποιήθηκαν είναι ο BME280 και ο Si7021. Για ακόμα καλύτερες μετρήσεις θερμοκρασίας μπορεί να χρησιμοποιηθούν άλλοι αισθητήρες όπως ο ψηφιακός αισθητήρας θερμοκρασίας DS18B20.

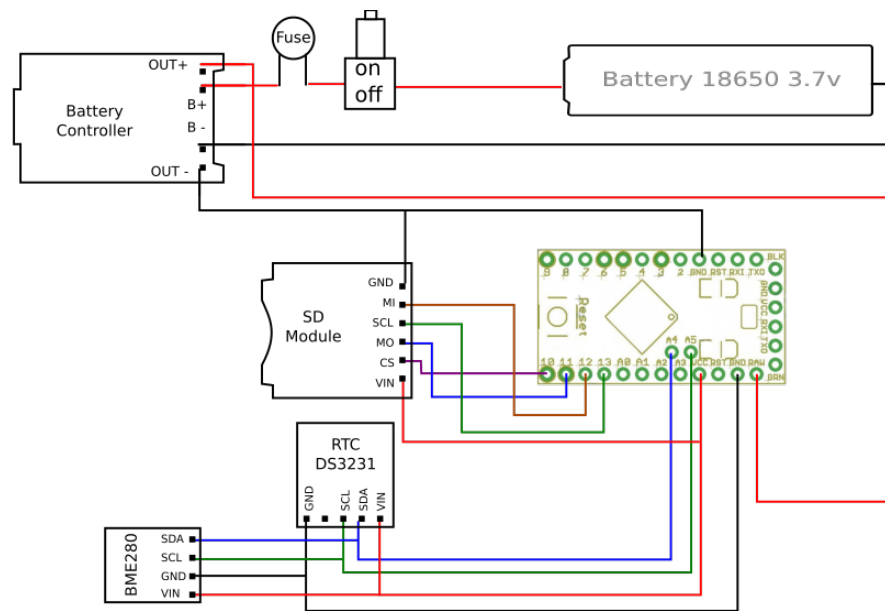
## 4.2 Κατασκευή αυτόνομων συστημάτων καταγραφής μετεωρολογικών παραμέτρων

Οι σταθμοί που κατασκευάστηκαν είναι αυτόνομοι κόμβοι καταγραφής των δεδομένων που λαμβάνουν από τους αισθητήρες σε κάρτα μνήμης SD, φέρουν τον μικροελεγκτή ATmega328, έχουν μονάδα μέτρησης πραγματικού χρόνου (Real Time Clock, RTC) η οποία στηρίζεται στο chip DS3231 και υποσύστημα ελέγχου φόρτισης και αποφόρτισης της μπαταρίας. Με τη χρήση μπαταρίας 2000mAh έχουν τη δυνατότητα καταγραφής πάνω από 250.000 μετρήσεις. Χρησιμοποιήθηκαν πέντε αισθητήρες BME280 οι οποίοι καταγράφουν τη θερμοκρασία, την υγρασία και τη πίεση και ένας αισθητήρας Si7021. Η επικοινωνία με τον υπολογιστή για τον προγραμματισμό του μικροελεγκτή, γίνεται μέσω της μονάδας USB to Serial Converter η οποία στηρίζεται στο chip FTDI FT232RL. Αξίζει να σημειωθεί πως η μονάδα επικοινωνίας δεν παραμένει στο σύστημα, είναι μόνο για τον έλεγχο και προγραμματισμό του σταθμού. Μια μονάδα επικοινωνίας μπορεί να χρησιμοποιηθεί για όλους τους κόμβους. Στον πίνακα 5 φαίνονται αναλυτικά όλα τα υλικά που απαιτήθηκαν για την κατασκευή του κόμβου.

<u>Υλικά</u>	<u>Χαρακτηριστικά</u>	<u>Τιμή (€)</u>
Microcontroller board	Arduino Pro Mini 328 3.3v	2
SD Module	Mini Micro SD	0.75
Real Time Clock (RTC)	Mini DS3231	1.25
Sensor	BME280 (Temperature, Humidity, Pressure)	5
	Si7021 (Temperature, Humidity)	3
Battery Charger	Micro USB 1A Lithium Battery Charger	0.75
Battery Holder	18650 Battery Clip Holder	0.5
Battery	18650 2000mAh 3.7v	3
Switch	Cap Self-locking Type Square Button	0.25
Fuse	0.65A Resettable Fuse	0.25
PCB	2x8cm Double-Side	0.75
Enclosure	100x60x25mm	0.75
Other Electronics	Wires, Connections atc.	0.75
SD Card	Micro SD 8gb	3
USB to Serial Converter	FTDI FT232RL	2
Σύνολο=		14-21

Πίνακας 5: Υλικά κατασκευής αυτόνομου συστήματος καταγραφής περιβαλλοντικών παραμέτρων.

Η συνδεσμολογία των παραπάνω υλικών φαίνεται στο Σχήμα 4.16. Ιδιαίτερη προσοχή απαιτείται στη πολικότητα της τροφοδοσίας. Για την ασφάλεια του κυκλώματος προς αποφυγή βραχυκυκλώματος η άλλου προβλήματος τοποθετήθηκε αυτόματη ασφάλεια 0,65A (Resettable Fuse). Αφότου συναρμολογήθηκαν τα εξαρτήματα τοποθετήθηκαν σε κουτί ηλεκτρονικών με διαστάσεις 100mm x 60mm x 25mm (Σχήμα 4.17). Στο κουτί δημιουργήθηκε μια οπή 7mm και τοποθετήθηκε σε αυτή ο αισθητήρας.



Σχήμα 4.16: Σχεδιάγραμμα Συνδεσμολογίας.

Δημιουργήθηκαν συνολικά πέντε αυτόνομοι κόμβοι καταγραφής μετεωρολογικών παραμέτρων, εκ των οποίων οι τέσσερις φέρουν τον αισθητήρα BME280 και ο ένας τον αισθητήρα Si7021. Οι αισθητήρες τοποθετήθηκαν έξω από το κουτί ηλεκτρονικών όπως φαίνεται στο σχήμα 4.17.

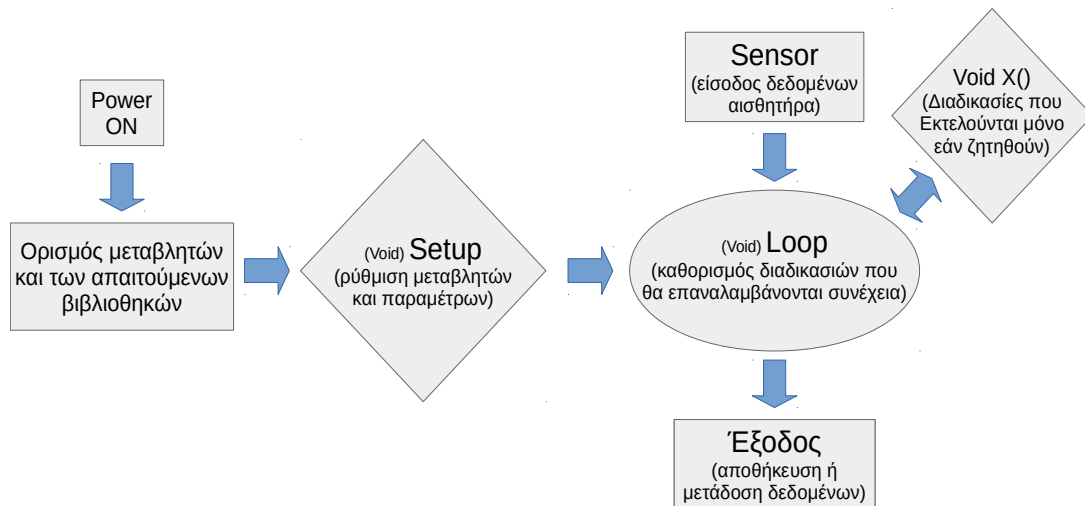


Σχήμα 4.17: Κόμβος καταγραφής μετεωρολογικών παραμέτρων.

Ο προγραμματισμός του μικροελεγκτή είναι η σημαντικότερη και δυσκολότερη διαδικασία που εκτελέστηκε για την υλοποίηση των συστημάτων καταγραφής περιβαλλοντικών παραμέτρων. Στη συνέχεια περιγράφεται η διαδικασία του προγραμματισμού.

Ο κώδικας δημιουργήθηκε στο ολοκληρωμένο περιβάλλον ανάπτυξης Arduino IDE (Integrated Development Environment) σε γλώσσα προγραμματισμού C++ (Παράρτημα 1). Στη συνέχεια περιγράφεται η διαδικασία προγραμματισμού του μικροελεγκτή και οι βασικότερες εντολές του κώδικα που δημιουργήθηκε για τη λειτουργία του συστήματος.

Ο τρόπος λειτουργίας του προγράμματος φαίνεται στο σχήμα 4.18. Με το που ενεργοποιηθεί ο μικροελεγκτής ορίζονται οι βιβλιοθήκες που θα συμπεριληφθούν στο κώδικα από όπου θα αντλούνται εντολές και επίσης ορίζονται οι μεταβλητές που θα χρησιμοποιηθούν. Στο τμήμα Loop καθορίζονται οι διαδικασίες που θα επαναλαμβάνονται συνέχεια.



Σχήμα 4.18: Τρόπος λειτουργίας προγράμματος

Οι βιβλιοθήκες είναι αρχεία που περιλαμβάνουν σύνολα εντολών, ορίζονται στην αρχή του κώδικα με την εντολή “#include” και περιλαμβάνουν έτοιμα σύνολα εντολών ώστε όταν χρειαστούν να αντλούνται με μια και μόνο εντολή. Οι βιβλιοθήκες που χρησιμοποιήθηκαν για τον σταθμό μέτρησης περιβαλλοντικών παραμέτρων είναι οι εξής:

```

#include <DS3232RTC.h> //Βιβλιοθήκη χρόνου (DS3232RTC)
// (http://github.com/JChristensen/DS3232RTC)
#include <Time.h> //Βιβλιοθήκη χρόνου (Arduino)
// (http://www.arduino.cc/playground/Code/Time)
#include <Wire.h> //Βιβλιοθήκη επικοινωνίας i2c
// (http://arduino.cc/en/Reference/Wire)
#include <SPI.h> //Βιβλιοθήκη επικοινωνίας SPI
// (http://arduino.cc/en/Reference/SPI)
#include <SD.h> //Βιβλιοθήκη κάρτας SD
// (http://arduino.cc/en/Reference/SD)
#include <BME280.h> //Βιβλιοθήκη αισθητήρα BME280
// (https://github.com/finitespace/BME280)
#include <VoltageReference.h> //Βιβλιοθήκη μέτρησης τάσης
// (https://github.com/rlogiacco/VoltageReference)
#include <LowPower.h> //Βιβλιοθήκη εξοικονόμησης ενέργειας
// (https://github.com/rockscream/Low-Power)
  
```



Τα κείμενα που ακολουθούνται μετά από δυο κάθετες “//” είναι σημειώσεις και δεν λαμβάνονται υπόψη από τον μεταγλωττιστή (compiler).

Ο κώδικας που δημιουργήθηκε για τους σταθμούς μέτρησης και καταγραφής περιβαλλοντικών παραμέτρων δημιουργήθηκε βάση τις οδηγίες και τα παραδείγματα των παραπάνω βιβλιοθηκών.

Στη συνέχεια του κώδικα ορίζονται οι ρυθμίσεις των αισθητήρων. Οι προτεινόμενες από τον κατασκευαστή, ρυθμίσεις λειτουργίας του αισθητήρα BME280, για την καταγραφή μετεωρολογικών παραμέτρων, περιγράφονται στον πίνακα που ακολουθεί (Πίνακας 6).

<b>Προτεινόμενες ρυθμίσεις για την καταγραφή μετεωρολογικών παραμέτρων</b>	
Sensor mode	forced mode, 1 sample / minute
Oversampling settings	pressure ×1
	temperature ×1
	humidity ×1
IIR filter settings	filter off
Performance for suggested settings	
Current consumption	0.16 $\mu$ A
RMS Noise	3.3 Pa / 30 cm, 0.07 %RH
Data output rate	1/60 Hz

Πίνακας 6: Προτεινόμενες από τον κατασκευαστή ρυθμίσεις λειτουργίας.

Οι ρυθμίσεις ορίζονται με την εντολή “BME280 bme(1, 1, 1, 3, 5, 0, false, 0x76);”, κάθε αριθμός αντιστοιχεί σε συγκεκριμένη ρύθμιση. Ο πρώτος αριθμός ορίζει τον ρυθμό υπερδειγματοληψίας για την θερμοκρασία, ο δεύτερος για την υγρασία και ο τρίτος για τη πίεση. Ο τέταρτος αριθμός ορίζει τον τρόπο λειτουργίας, ο πέμπτος τον χρόνο αναμονής, ο έκτος το είδος του φίλτρου και τέλος ορίζεται η διεύθυνση επικοινωνίας με τον μικροελεγκτή. Στον κώδικα που ακολουθεί περιγράφονται με τη σειρά, οι παράμετροι που μπορεί να οριστούν για τον συγκεκριμένο αισθητήρα.

```
// (Temp Oversampling Rate, Hum Ov/ing Rate, Pres Ov/ing Rate, Mode,
// Standby Time, Filter,SPI Enable, BME280 Address)
// * Temperature Oversampling Rate (tosr): uint8_t, default = 0x1
// values: 0 = Skipped, 1 = x1, 2 = x2, 3 = x4, 4 = x8, 5 = x16
// * Humidity Oversampling Rate (hosr): uint8_t, default = 0x1
// values: 0 = Skipped, 1 = x1, 2 = x2, 3 = x4, 4 = x8, 5 = x16
// * Pressure Oversampling Rate (posr): uint8_t, default = 0x1
// values: 0 = Skipped, 1 = x1, 2 = x2, 3 = x4, 4 = x8, 5 = x16
// * Mode: uint8_t, default = forced
// values: Sleep = 0, Normal = 1 and 2, Forced = 3
// * Standby Time (st): uint8_t, default = 1000ms
// values: 0 = 0.5ms, 1 = 62.5ms, 2 = 125ms, 3 = 250ms, 4 = 250ms,
// 5 = 1000ms, 6 = 10ms, 7 = 20ms
// * Filter: uint8_t, default = None
// values: 0 = off, 1 = 2, 2 = 4, 3 = 8, 4 = 16
// * SPI Enable: bool, default = false
// values: true = enable, false = disable
// * BME280 Address: uint8_t, default = 0x76
// values: any uint8_t
```



Για να ορίσουμε ώστε οι μετρήσεις να λαμβάνονται σε μετρικό σύστημα θα πρέπει να εισαγάγουμε τον κώδικα:

```
bool metric = true;
```

Εντός του τμήματος του κώδικα “void setup(void) {...}” ορίζονται όλες οι ρυθμίσεις που χρειάζονται προτού ξεκινήσει η λειτουργία του βασικού κώδικα. Μια πολύ σημαντική παράμετρος οπου ορίζεται στο τμήμα αυτό του κώδικα, είναι η ταχύτητα επικοινωνίας του μικροελεγκτή. Η εντολή αυτή ρυθμίζει το ρυθμό μετάδοσης δεδομένων σε bits ανά δευτερόλεπτο (baud) για τη σειριακή μετάδοση δεδομένων. Για την επικοινωνία με τον υπολογιστή, χρησιμοποιούνται μία από τις ακόλουθες τιμές: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, ή 115200. Η πιο διαδεδομένη ταχύτητα επικοινωνίας είναι “9600b/s” και ορίζεται με την εντολή:

```
Serial.begin(9600);
```

Επίσης στο τμήμα αυτό δίνονται οι εντολές εκκίνησης του υποσυστήματος πραγματικού χρόνου, του υποσυστήματος καταγραφής σε αποθηκευτικό μέσο και του αισθητήρα. Προτού ξεκινήσει ο βασικός κώδικας είναι σημαντικό να δοθούν επεξηγήσεις για το τι είναι η κάθε τιμή που λαμβάνεται. Αυτό επιτυγχάνεται δίνοντας μια εντολή (η οποία θα εκτελεστεί μόνο μια φορά κατά την εκκίνηση του μικροελεγκτή) όπου θα “εκτυπώνει” κεφαλίδες για την κάθε τιμή που θα λαμβάνεται. Η εντολή “Serial.println(...)” στέλνει τις επικεφαλίδες μέσω σειριακής επικοινωνίας σε περίπτωση που είναι συνδεδεμένο σε υπολογιστικό σύστημα. Ενώ η εντολή dataFile.println(...) γράφει τις επικεφαλίδες στη κάρτα αποθήκευσης. Ο κώδικας αυτός έχει την εξής μορφή:

```
Serial.println("Time Date TempC RH% hPa Alt(m) Dew p mV");
dataFile.println("Time Sec Date TempC RH% hPa Alt(m) Dew_p mV");
delay(100);
```

Ο βασικός κώδικας βρίσκεται εντός του τμήματος “void loop(void) {...}”. Ο κώδικας που βρίσκεται εντός αυτού του τμήματος επαναλαμβάνεται συνέχεια. Στη συνέχεια περιγράφονται οι απαραίτητες εντολές για την καταγραφή των δεδομένων. Η καταγραφή στο μέσο αποθήκευσης γίνεται με την εντολή “dataFile.print(...);” όπου μέσα στις παρενθέσεις εισάγονται οι παράμετροι του πραγματικού χρόνου και της τάσης ρεύματος του συστήματος. Με λίγο διαφορετικό τρόπο εισάγονται οι εντολές καταγραφής των μετεωρολογικών παραμέτρων, όπου γίνεται με τις εντολές “printBME280Data(&dataFile);” και “printBME280CalculatedData(&dataFile);”, οι εντολές αυτές περιέχουν στη πραγματικότητα έτοιμη τη σειρά εντολών “dataFile.print(...);” για τις μετρήσεις του αισθητήρα. Ο τελικός κώδικας για την καταγραφή όλων των απαραίτητων παραμέτρων έχει τη παρακάτω μορφή:

```
dataFile.print(tm.Hour);
dataFile.print(':');
dataFile.print(tm.Minute);
dataFile.print(' ');
dataFile.print(tm.Second);
```

```

dataFile.print(' ');
dataFile.print(tm.Day);
dataFile.print('/');
dataFile.print(tm.Month);
dataFile.print('/');
dataFile.print(tm.Year);
dataFile.print(' ');
printBME280Data(&dataFile);
printBME280CalculatedData(&dataFile);
dataFile.print(' ');
dataFile.print(vcc);
dataFile.print(' ');
dataFile.print(millis());
dataFile.println();

```

Σημαντικό είναι μετά τις εντολές καταγραφής των παραμέτρων να αποθηκεύεται το αρχείο στο μέσο αποθήκευσης δεδομένων δηλαδή στη κάρτα SD. Αυτό επιτυγχάνεται με την εξής εντολή:

```
dataFile.flush();
```

Για να δημιουργηθεί καθυστέρηση λήψης δεδομένων ανά συγκεκριμένη χρονική περίοδο συνήθως εισάγουμε την εντολή:

```
delay(1000); //Καθυστέρηση 1000 milliseconds (1000ms = 1sec)
```

Το πρόβλημα με τη συγκεκριμένη εντολή είναι ότι, ο μικροελεγκτής συνεχίζει να λειτουργεί όλο το χρονικό διάστημα που ορίζουμε για τη καθυστέρηση με συνέπεια να καταναλώνει ενέργεια χωρίς λόγο. Για να αντιμετωπιστεί το πρόβλημα αυτό είναι σημαντικό να προγραμματίσουμε τον μικροελεγκτή να μπαίνει σε κατάσταση “ύπνου” (sleep mode) κατά το χρονικό διάστημα της καθυστέρησης που ορίζουμε. Η κατάσταση ύπνου μπορεί να διαρκέσει μέχρι οκτώ δευτερόλεπτα, για να φτάσουμε στη επιθυμητή καθυστέρηση των 300 δευτερολέπτων (πέντε λεπτά), απαιτείται η δημιουργία ενός αλγόριθμου επανάληψης όπως φαίνεται στον παρακάτω κώδικα

```

// Εισαγωγή σε λειτουργία sleep συνολικά 300 δευτερόλεπτα (5 λεπτά)
int i;
for (i = 0; i <35; i++)
{
    // Εισαγωγή σε λειτουργία sleep για 8s με το ADC και το BOD module
    // απενεργοποιημένα
    LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
// SLEEP_15MS, SLEEP_30MS, SLEEP_60MS, SLEEP_120MS, SLEEP_250MS,
// SLEEP_500MS, SLEEP_1S, SLEEP_2S, SLEEP_4S, SLEEP_8S, SLEEP_FOREVER
}
LowPower.powerDown(SLEEP_500MS, ADC_OFF, BOD_OFF);
LowPower.powerDown(SLEEP_250MS, ADC_OFF, BOD_OFF);

```

Ο παραπάνω κώδικας δημιουργήθηκε για καθυστέρηση 280.750ms, το υπόλοιπο της καθυστέρησης που απαιτείται δημιουργείται από τον κώδικα που εκτελείται πριν αυτού του αλγόριθμου και από τον ίδιο τον μικροελεγκτή.

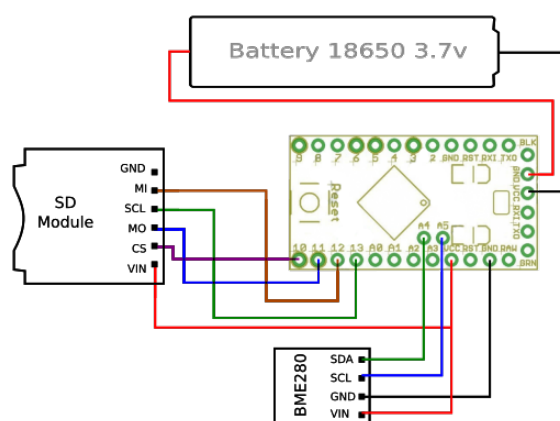
#### 4.2.1 Κατασκευή αυτόνομου συστήματος καταγραφής για τετρακόπτερο (Drone)

Το δεύτερο σύστημα που κατασκευάστηκε, είναι ένα αυτόνομο μικρών διαστάσεων σύστημα καταγραφής δεδομένων σε κάρτα μνήμης SD. Φέρει επίσης τον μικροελεγκτή ATmega328. Για να διατηρηθεί μικρο το μέγεθος και το βάρος του. Δεν χρησιμοποιήθηκε μονάδα μέτρησης πραγματικού χρόνου ούτε σύστημα ελέγχου φόρτισης και αποφόρτισης της μπαταρίας. Με τη χρήση μπαταρίας 240mAh η οποία περιεχί ενσωματωμένο σύστημα ελέγχου φόρτισης και αποφόρτισης, έχει τη δυνατότητα συνεχόμενης καταγραφής ανά 250 χιλιοστά του δευτερολέπτου (milliseconds), έως και 80 λεπτών. Χρησιμοποιήθηκε ο αισθητήρας BME280 ο οποίος καταγράφει τη θερμοκρασία, την υγρασία και τη πίεση. Η επικοινωνία με τον υπολογιστή για τον προγραμματισμό του μικροελεγκτή, γίνεται επίσης μέσω της μονάδας “USB to Serial Converter”. Στον πίνακα 7 φαίνονται αναλυτικά όλα τα υλικά που απαιτήθηκαν για την κατασκευή του κόμβου.

Εξαρτήματα	Χαρακτηριστικά	Price €
Microcontroller board	Arduino Pro Mini 328 3.3v	2
SD Module	Mini Micro SD	0.75
SD Card	Micro SD 8gb	3
Sensor	BME280 (Temperature, Humidity, Pressure)	5
Battery	250mAh 3.7v	2
Fuse	0.05A Resettable Fuse	0.25
Other electronics	Wires, Connections atc.	0.25
Σύνολο=		13

Πίνακας 7: Υλικά κατασκευής αυτόνομου συστήματος καταγραφής περιβαλλοντικών παραμέτρων για τετρακόπτερο (Drone).

Η συνδεσμολογία των παραπάνω υλικών φαίνεται στο Σχέδιο 4.19. Ιδιαίτερη προσοχή απαιτείται στη πολικότητα της τροφοδοσίας.



Σχήμα 4.19: Σχεδιάγραμμα Συνδεσμολογίας συστήματος για Drone.

Ο κώδικας αυτού του κόμβου (Παράρτημα 2) είναι παρόμοιος με τον κώδικα που δημιουργήθηκε για το πρώτο κόμβο, όπου αφαιρέθηκε η βιβλιοθήκη και οι εντολές που αφορούν το υποσύστημα πραγματικού χρόνου και η βιβλιοθήκη μέτρησης τάσης και οι αντίστοιχες εντολές αυτής.

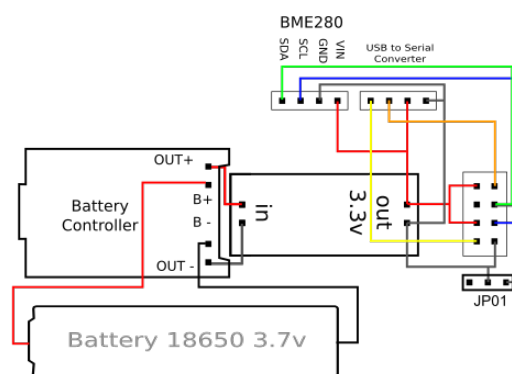
### 4.3 Κατασκευή κόμβου καταγραφής και απεικόνισης μετεωρολογικών παραμέτρων πραγματικού χρόνου

Για τις ανάγκες της διπλωματικής διατριβής, έγινε μια προσπάθεια κατασκευής κόμβου καταγραφής περιβαλλοντικών παραμέτρων σε πραγματικό χρόνο και όχι αποθήκευση αυτών σε κάποιο μέσο αποθήκευσης όπως γίνεται με τους προηγούμενους σταθμούς που κατασκευάστηκαν. Για την επίτευξη του στόχου χρησιμοποιήθηκε το νέο σύστημα ασύρματης επικοινωνίας WiFi χαμηλού κόστους ESP8266. Στον πίνακα 8 φαίνονται αναλυτικά όλα τα υλικά που απαιτήθηκαν για την κατασκευή του σταθμού.

Εξαρτήματα	Χαρακτηριστικά	Price €
WiFi, Microcontroller board	ESP8266 Esp-01	2
Sensor	BME280 (Temperature, Humidity, Pressure)	5
Step Down Converter	AMS1117 3.3v	0.25
Battery	18650 2000mAh 3.7v	3
Battery Charger	Micro USB 1A Lithium Battery Charger	0.75
Battery Holder	18650 Battery Clip Holder	0.25
Fuse	0.65A Resettable Fuse	0.25
Enclosure	100x60x25mm	0.75
Other Electronics	Wires, PCB, Connections atc.	1
Σύνολο =		13

Πίνακας 8: Υλικά κατασκευής συστήματος πραγματικού χρόνου καταγραφής περιβαλλοντικών παραμέτρων.

Το σύστημα ασύρματης επικοινωνίας WiFi ESP8266 λειτουργεί με 3.3v ενώ η μπαταρία έχει τάση περίπου από 3v έως 4.2v (σε πλήρη φόρτιση), για το λόγο αυτό είναι απαραίτητη η χρήση ενός υποσυστήματος ρύθμισης τάσης με έξοδο 3.3v. Στο σχεδιάγραμμα που ακολουθεί 4.20 φαίνεται ο τρόπος συνδεσμολογίας των εξαρτημάτων του σταθμού πραγματικού χρόνου.



Σχήμα 4.20: Σχεδιάγραμμα Συνδεσμολογίας ασύρματου συστήματος πραγματικού χρόνου.

Ο σταθμός πραγματικού χρόνου μπορεί να είναι πολύ πιο εύκολος στη κατασκευή αλλά είναι πολύ πιο δύσκολος και περίπλοκος ο προγραμματισμός του. Εκτός από τον προγραμματισμό του μικροελεγκτή του κόμβου αυτού, απαιτείται η δημιουργία εξυπηρετητή για τα δεδομένα που αποστέλλονται και παράλληλα η δημιουργία διαδικτυακής σελίδας

απεικόνισης των δεδομένων. Ο κώδικας δημιουργήθηκε με βάση τις οδηγίες του Brian T.S (2015).

Για τις ανάγκες της διπλωματικής χρησιμοποιήθηκε εξυπηρετητής δεδομένων `data.sparkfun.com`. Ο εξυπηρετητής δεδομένων `data.sparkfun.com` χρησιμοποιεί τη μηχανή καταγραφής δεδομένων “phant”. Η μηχανή καταγραφής δεδομένων “phant” είναι ένα εργαλείο καταγραφής ανοιχτού κώδικα που αναπτύχθηκε από τη SparkFun Electronics για τη συλλογή δεδομένων από το διαδίκτυο των πραγμάτων (Internet of Things).

Αφότου δημιουργηθεί ο εξυπηρετητής δεδομένων όπως περιγράφεται στο κεφάλαιο 6 γίνεται ο προγραμματισμός του μικροελεγκτή (Παράρτημα 3) με βάση τα κλειδιά που λήφθηκαν κατά τη διάρκεια δημιουργίας του εξυπηρετητή. Οι βιβλιοθήκες που συμπεριλαμβάνονταν στο προγραμματισμό αυτού του κόμβου είναι οι εξής:

```
#include <Wire.h> //Βιβλιοθήκη επικοινωνίας i2c
(http://arduino.cc/en/Reference/Wire)
#include <BME280.h> //Βιβλιοθήκη αισθητήρα BME280
(https://github.com/finitespace/BME280)
#include <ESP8266WiFi.h> //Βιβλιοθήκη WiFi (ESP8266)
(https://github.com/esp8266/Arduino)
```

Για να σταλούν τα δεδομένα στον εξυπηρετητή θα πρέπει αρχικά να συνδεθεί ο κόμβος σε κάποιο δίκτυο. Αυτό επιτυγχάνεται ορίζοντας το όνομα και τον κωδικό του δικτύου στο τμήμα του κώδικα όπως φαίνεται παρακάτω. Στη συνέχεια θα πρέπει να οριστεί το όνομα του εξυπηρετητή όπου φιλοξενεί τα δεδομένα και τα απαιτούμενα κλειδιά πρόσβασης που βρίσκονται στο αρχείο .JSON.

```
const char* ssid = "....."; //Εδώ εισάγουμε το όνομα του δικτύου
const char* password = "....."; //Εδώ εισάγουμε το κωδικό του δικτύου

const char* host = "data.sparkfun.com";
const char* publicKey = "wpq2GgjAa9Ulg4dbEGGL";
const char* privateKey = "wzW7VqEb01iRke8dD22p";
```

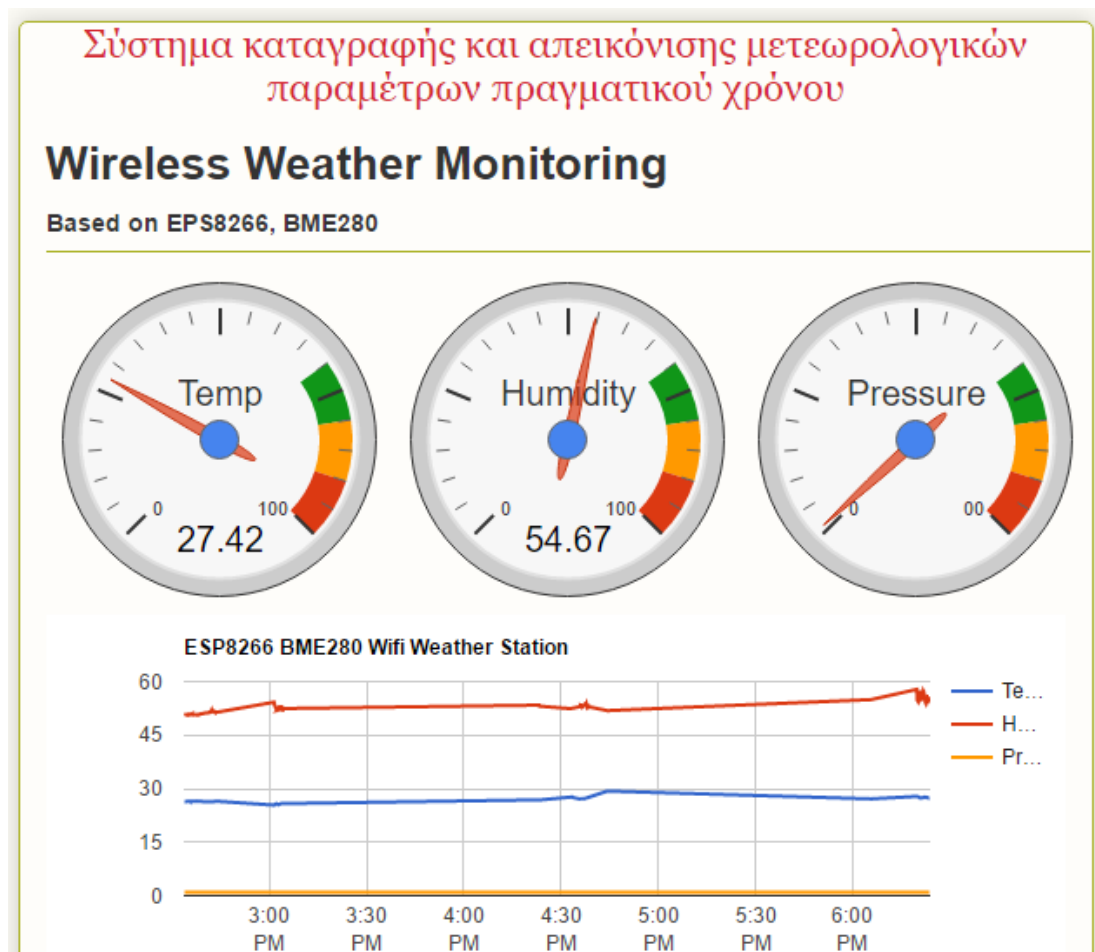
Τα δεδομένα στέλνονται στον εξυπηρετητή με τη μορφή διεύθυνσης url. Η διεύθυνση url δημιουργείται με των κώδικα που παρατίθεται στη συνέχεια και έχει την εξής μορφή: `http://data.sparkfun.com/input/[publicKey]?private_key=[privateKey]&hum=[Σχετική υγρασία]&pres=[ατμοσφαιρική πίεση]&temp=[θερμοκρασία]&volt=[τάση μπαταρίας]`

```
String url = "/input/";
url += publicKey;
url += "?private_key=";
url += privateKey;
url += "&temp=";
url += t;
url += "&hum=";
url += h;
url += "&pres=";
url += p;
url += "&volt=";
url += v;
```

Η απεικόνιση των δεδομένων επιτυγχάνεται δημιουργώντας κώδικα HTML (Παράρτημα 3.1) με εντολές javascript που αντλούν τα δεδομένα του εξυπηρετητή και τα απεικονίζουν σε γραφήματα ή άλλους τρόπους απεικόνισης. Ο κώδικας .html που δημιουργήθηκε βασίστηκε στον κώδικα του Brian (2015). Το μόνο που απαιτείται να γνωρίζουμε είναι το publicKey (wpq4GgjAa0Ulg2dbEGGL) από το αρχείο .JSON. Στον κώδικα αυτό απαιτείται η εισαγωγή του ονόματος του εξυπηρετητή και του κλειδιού “public\_key”.

```
var cloud_url = 'https://data.sparkfun.com/output/';
var public_key = 'wpq2GgjAa9Ulg4dbEGGL';
```

Η απεικόνιση των δεδομένων στο διαδίκτυο (Σχήμα 4.21) γίνεται διαμέσου των εργαλείων Google chart tools. Τα Google chart παρέχουν διάφορους τρόπους για την απεικόνιση δεδομένων σε ιστοσελίδες, από απλά γραμμικά γραφήματα μέχρι και πολύπλοκα διαγράμματα. Παρέχει ένα μεγάλο αριθμό έτοιμων προς χρήση διαφορετικών γραφημάτων, τα οποία μπορούν να χρησιμοποιηθούν ενσωματώνοντας απλές εντολές javascript μαζί με τις απαραίτητες βιβλιοθήκες.



Σχήμα 4.21: Απεικόνιση των δεδομένων στο διαδίκτυο.

# 5

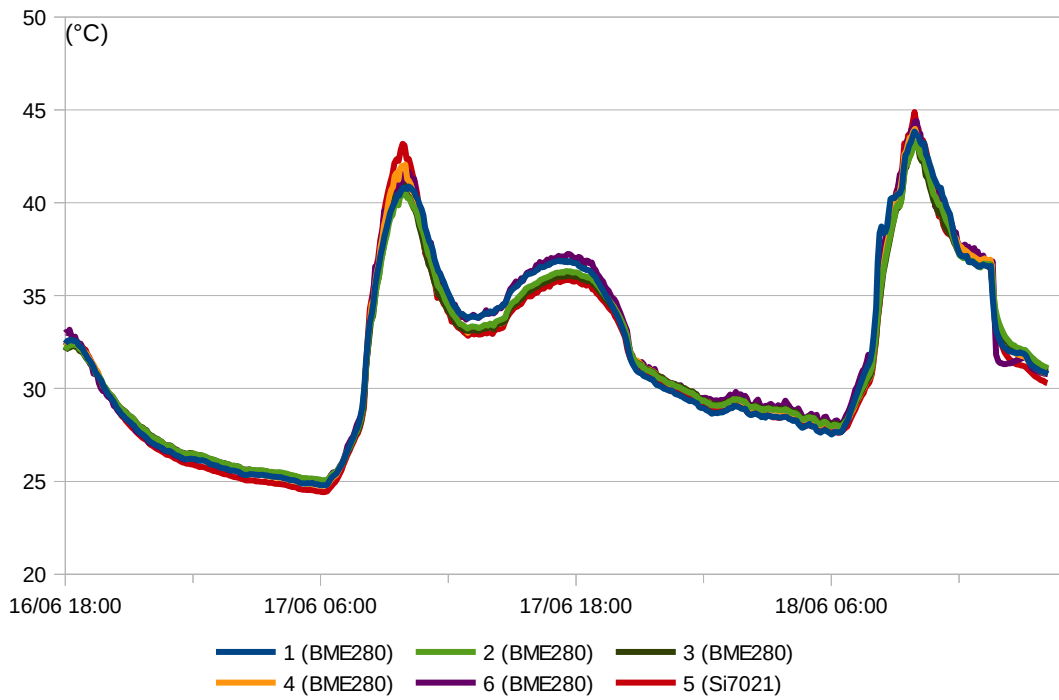
## Εφαρμογή

Σε μια ιδανική περίπτωση για την πρόληψη και διαχείριση δασικών πυρκαγιών, οι περιβαλλοντικοί παράμετροι που θα καταγράφονταν και θα παρακολουθούνταν σε πραγματικό χρόνο, είναι η θερμοκρασία, η υγρασία, η ταχύτητα και η διεύθυνση άνεμου. Λόγω του υψηλού κόστους δημιουργίας δικτύου καταγραφής αυτών των παραμέτρων, στα πλαίσια της διπλωματικής διατριβής καταγράφηκαν μόνο η υγρασία και θερμοκρασία και σε ορισμένες περιπτώσεις και η πίεση. Το πρώτο στάδιο των δοκιμαστικών μετρήσεων ήταν η σύγκριση των τιμών του κάθε αισθητήρα όταν βρίσκονταν κάτω από τις ίδιες συνθήκες. Στη συνέχεια οι σταθμοί μετρήσεις που κατασκευαστήκαν τοποθετήθηκαν σε δασική περιοχή για να παρθούν μετρήσεις σε φυσικές πραγματικές συνθήκες. Τα τελικά δεδομένα που παρήχθησαν επεξεργαστήκαν με λογισμικές εφαρμογές ανοιχτού κώδικα με σκοπό την ενσωμάτωση αυτών σε μοντέλα κινδύνου εκδήλωσης δασικών πυρκαγιών.

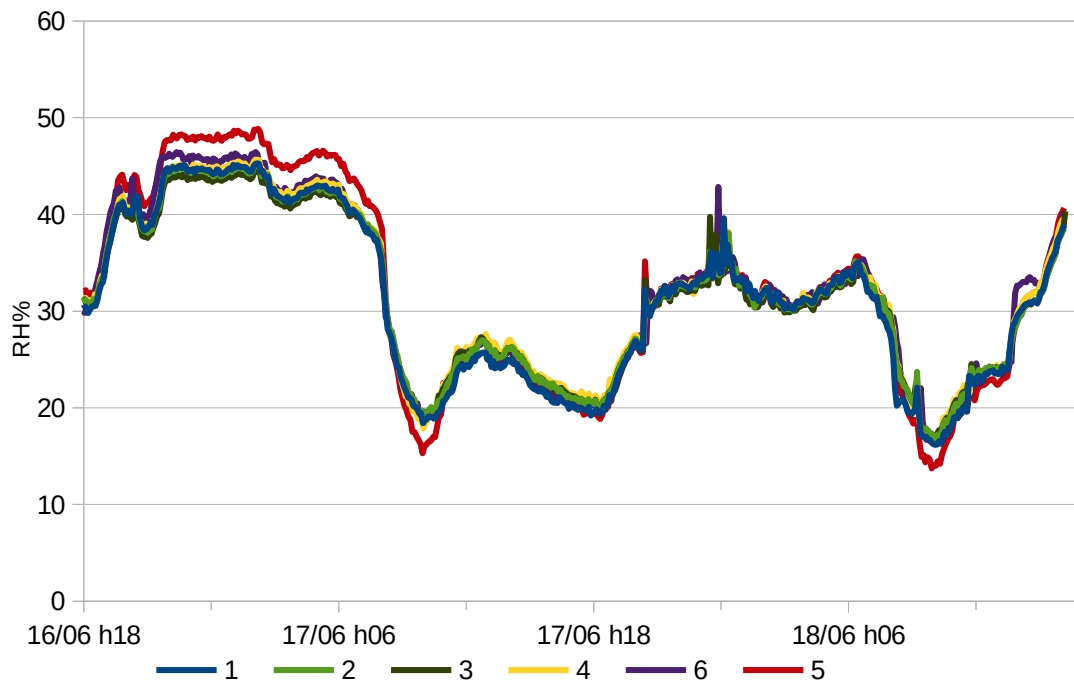
### 5.1 Δοκιμαστικές μετρήσεις

Σε μια ιδανική περίπτωση αν και εφόσον οι αισθητήρες υπάγονταν κάτω από τις ίδιες κλιματολογικές συνθήκες, θα παρείχαν όλοι τις ίδιες τιμές. Στο διάγραμμα θα παρουσιαζόταν μια ενιαία γραμμή χωρίς διακυμάνσεις. Μια πολύ σημαντική διαδικασία προτού τοποθετηθούν οι σταθμοί στο πεδίο, είναι η παρακολούθηση των μετρήσεων που δίνουν οι αισθητήρες, όταν βρίσκονται σε ίδιες περιβαλλοντικές συνθήκες. Για το λόγο αυτό θεωρήθηκε σκόπιμο η τοποθέτηση των σταθμών σε ίδιο περιβάλλον και την καταγραφή των τιμών που δίνουν οι αισθητήρες υπό αυτές τις συνθήκες.

Η πρώτη δοκιμή μετρήσεων όπου οι αισθητήρες υπάγονταν θεωρητικά κάτω από τις ίδιες κλιματολογικές συνθήκες παρείχε αρκετά ικανοποιητικά αποτελέσματα (Σχήμα 5.1 και 5.2). Παρόλα αυτά, παρατηρηθήκαν διακυμάνσεις μεταξύ των αισθητήρων, όπως όμως αποδείχτηκε στις επόμενες δοκιμαστικές μετρήσεις, οι διακυμάνσεις των πρώτων μετρήσεων οφείλονταν στη κακή τοποθέτηση των σταθμών και τον περιορισμένο χώρο γύρω από τους ίδιους τους αισθητήρες. Αυτό είχε ως αποτέλεσμα να δημιουργείται διαφορετικό μικρόκλιμα ανά αισθητήρα και κατά συνέπεια εσφαλμένες μετρήσεις. Επίσης στο σχήμα 5.1 παρατηρούνται μη φυσιολογικές μετρήσεις κατά τη χρονική περίοδο 8:00-12:00. Αυτές οι μη φυσιολογικές μετρήσεις οφείλονταν στην ανάκλαση της ηλιακής ακτινοβολίας, λόγω το ότι οι σταθμοί αρχικά τοποθετήθηκαν εσφαλμένα σε χώρο με τσιμεντένιο δάπεδο, παρόλο που βρίσκονταν οι ίδιοι υπό σκιά. Διορθώνοντας τα αρχικά λάθη οι αισθητήρες παρείχαν εντυπωσιακά καλύτερες μετρήσεις (Σχήμα 5.3 και 5.4).

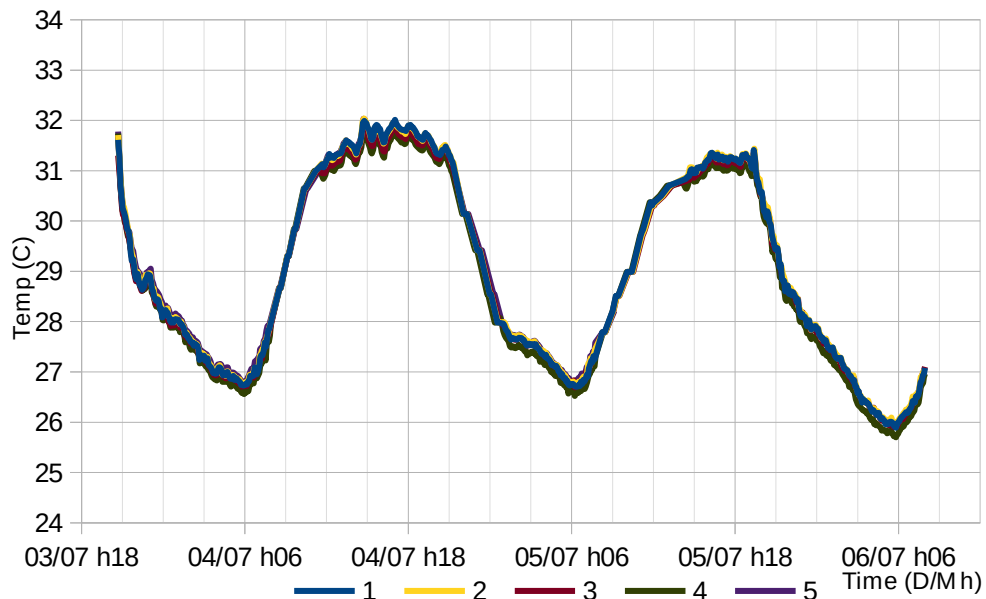


Σχήμα 5.1: Πρώτες δοκιμαστικές μετρήσεις θερμοκρασίας διάρκειας δυο ημερών.

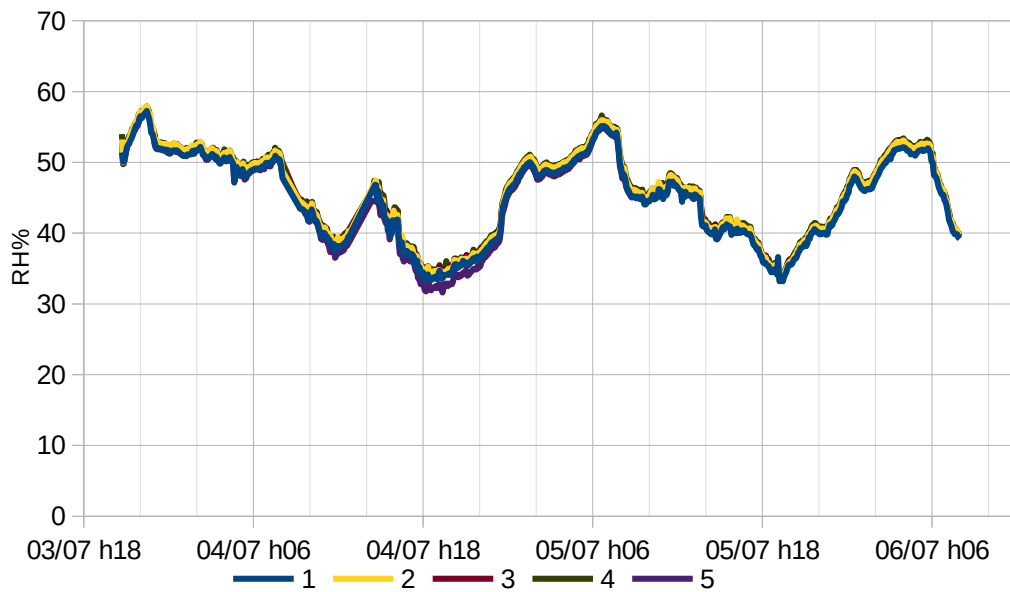


Σχήμα 5.2: Πρώτες δοκιμαστικές μετρήσεις υγρασίας διάρκειας δυο ημερών.





Σχήμα 5.3: Δεύτερες δοκιμαστικές βελτιωμένες μετρήσεις θερμοκρασίας διάρκειας δυο ημερών.



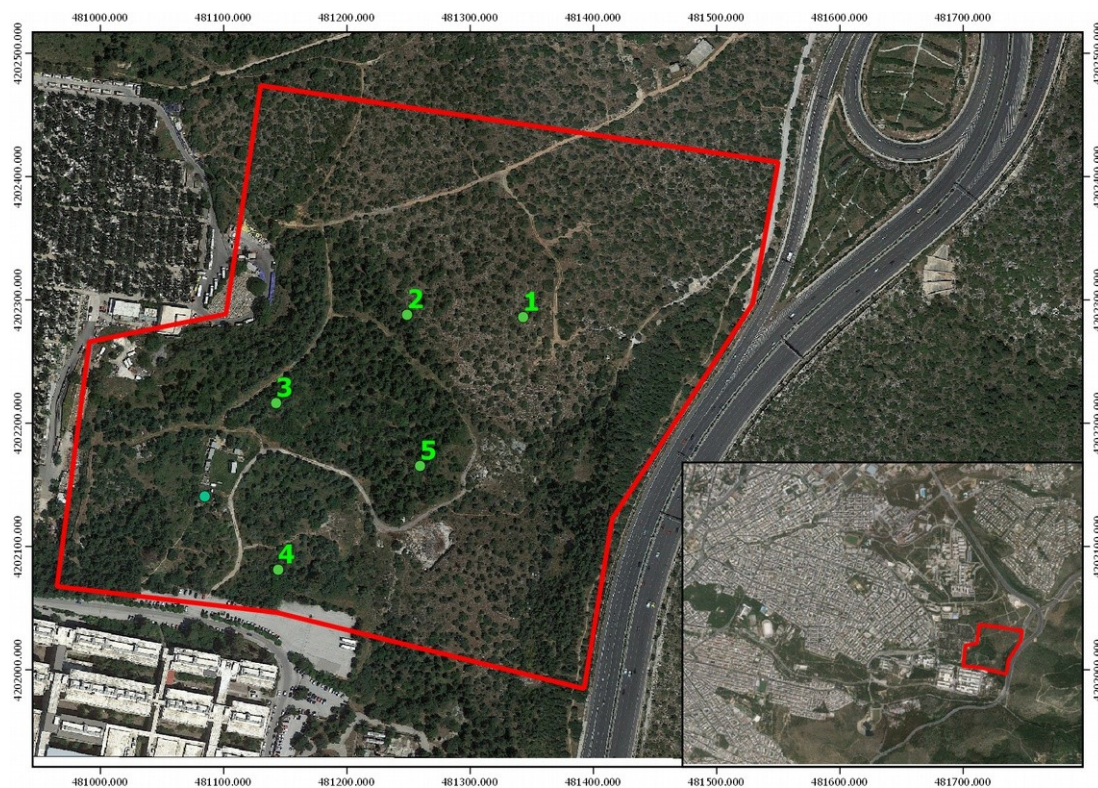
Σχήμα 5.4: Δεύτερες δοκιμαστικές βελτιωμένες μετρήσεις σχετικής υγρασίας διάρκειας δυο ημερών.

Τα αποτελέσματα από τις δοκιμαστικές μετρήσεις των αισθητήρων εντός κοινών περιβαλλοντικών συνθηκών, αποδεικνύουν πως οι τυχόν διακυμάνσεις των μετρήσεων πεδίου, δεν θα οφείλονται σε σφάλματα των αισθητήρων αλλά θα αντιπροσωπεύουν τις πραγματικές περιβαλλοντικές συνθήκες στις οποίες θα βρίσκονται οι σταθμοί.

## 5.2 Εφαρμογή στο πεδίο

Θεωρήθηκε σκόπιμο πέρα από τη κατασκευή των σταθμών, στα πλαίσια της διπλωματικής διατριβής, να τοποθετηθούν οι σταθμοί σε δασική έκταση, ώστε να καταγράφουν πραγματικές μετρήσεις πεδίου. Ιδανική περίπτωση θα ήταν η τοποθέτηση των σταθμών σε μεγάλες αποστάσεις μεταξύ τους, άνω του ενός χιλιομέτρου και σε περιοχές που διαφοροποιούνται οι κλιματολογικές συνθήκες. Στα πλαίσια της διπλωματικής δεν καθίσταται αυτό δυνατόν. Παρόλα αυτά, οι σταθμοί τοποθετήθηκαν σε μικρή δασική περιοχή εντός του χώρου του πανεπιστημίου Αθηνών. Το χρονικό διάστημα καταγραφής ήταν επτά ημέρες, από τις 21/06/2016 ώρα 16:00 ως τις 28/06/2016 ώρα 16:00.

Η περιοχή μελέτης όπου τοποθετήθηκαν οι σταθμοί είναι το Άλσος Ηριδανού το οποίο βρίσκεται βορειανατολικά του πανεπιστημίου Αθηνών (Σχήμα 5.5).



Σχήμα 5.5: Περιοχή μελέτης με τα σημεία τοποθέτησης των σταθμών

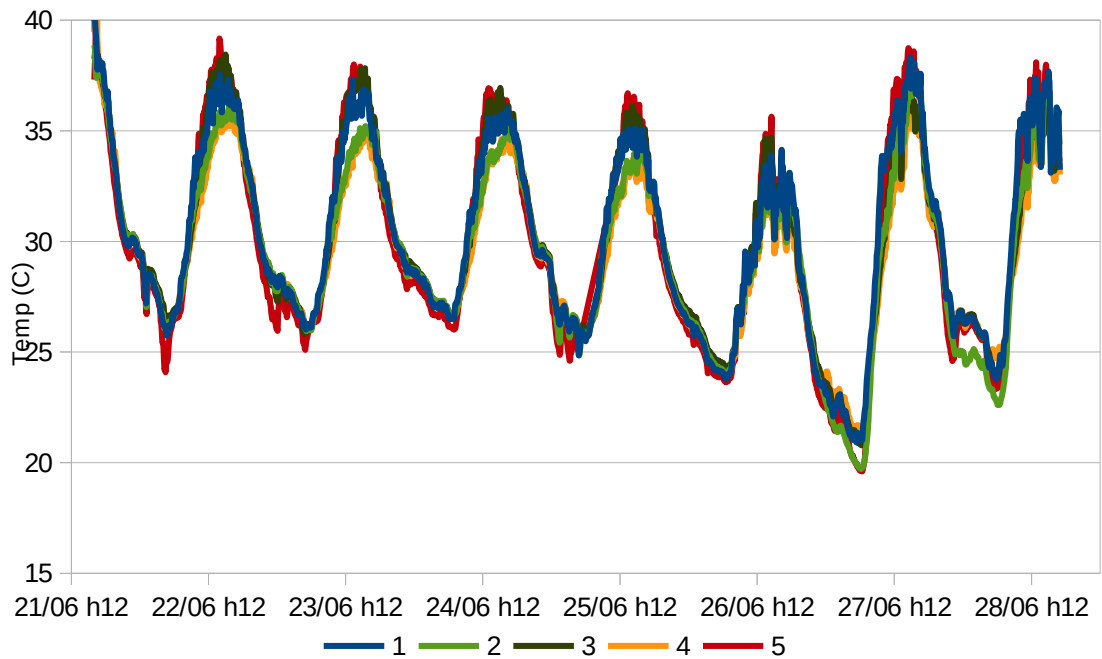
Οι σταθμοί τοποθετήθηκαν σε δέντρα εντός της περιοχής μελέτης στα σημεία όπως φαίνονται στο σχήμα 5.5. Οι μετρήσεις συγκριθήκαν με τις μετρήσεις του μετεωρολογικού σταθμού του εργαστηρίου κλιματολογίας του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών.

Κόμβος	Συντεταγμένες (ΕΓΣΑ87)		
	x	y	ζ
1	481343.0	4202286.0	235
2	481248.8	4202287.9	225
3	481142.5	4202216.2	221
4	481144.2	4202081.0	227
5	481259.3	4202165.2	218

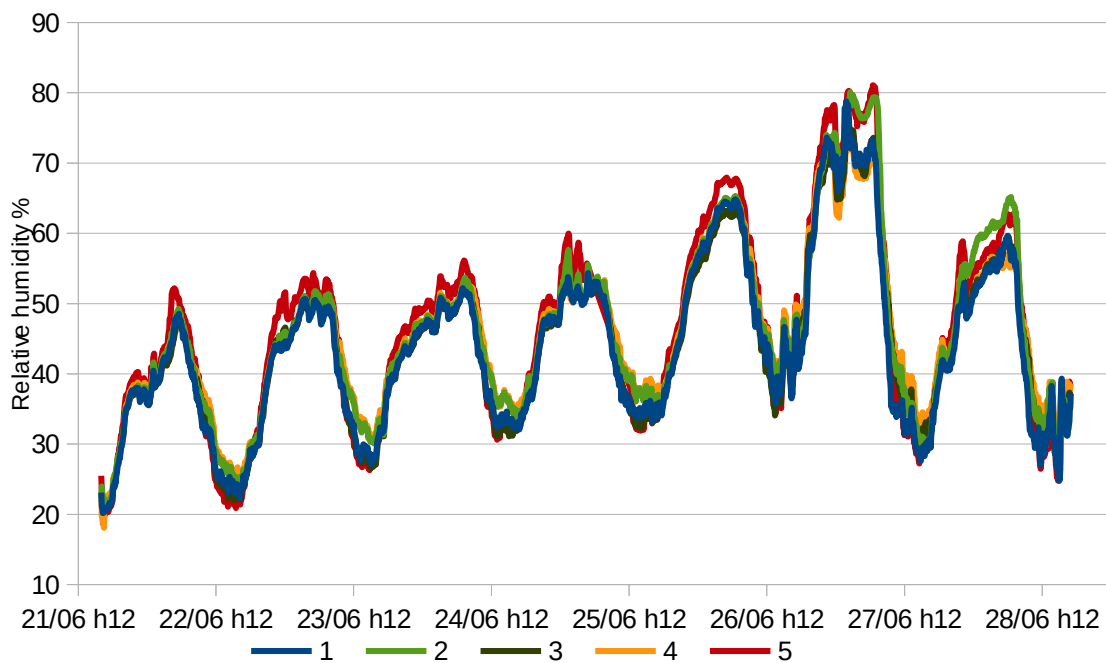
Πίνακας 9: Συντεταγμένες σταθμών.

### 5.2.1 Αποτελέσματα μετρήσεων πεδίου

Οι μετρήσεις των σταθμών ήταν αρκετά ικανοποιητικές. Όπως αναμενόταν, συγκρίνοντας τις μετρήσεις των σταθμών παρατηρούνται πολύ μικρές διαφοροποιήσεις (Σχήμα 4.6 και 4.7).



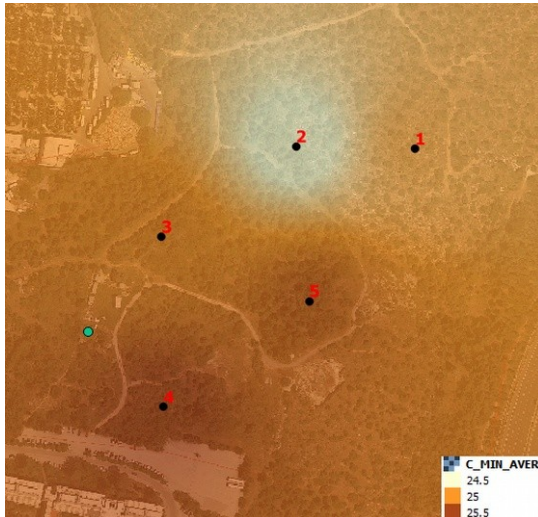
Σχήμα 5.6: Μετρούμενη θερμοκρασία (°C) των σταθμών στη περιοχή μελέτης.



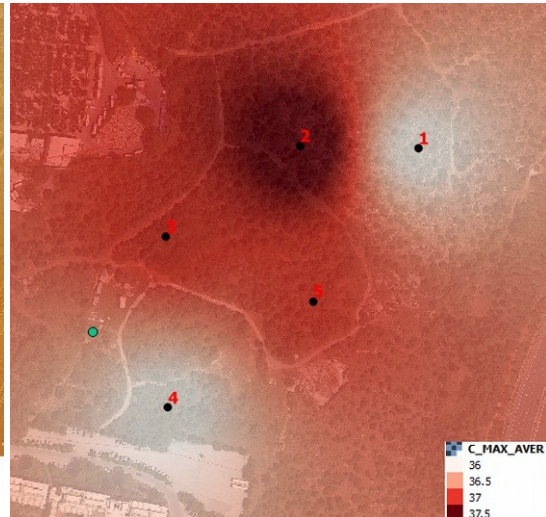
Σχήμα 5.7: Μετρούμενη σχετική υγρασία (RH) των σταθμών στη περιοχή μελέτης.



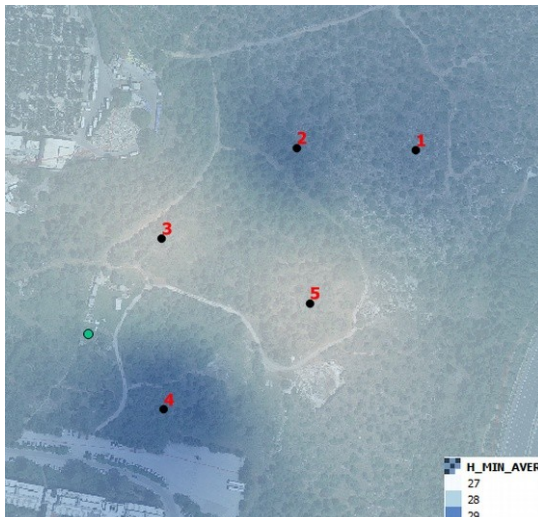
Από τις μετρήσεις των σταθμών δημιουργήθηκαν χάρτες θερμοκρασίας και σχετικής υγρασίας. Απώτερος σκοπός της δημιουργίας των χαρτών αυτών είναι η χρήση τους σε δείκτες επικινδυνότητας εκδήλωσης δασικών πυρκαγιών ή σε άλλα ψηφιακά μοντέλα για την πρόληψη και διαχείριση φυσικών καταστροφών. Οι χάρτες που ακολουθούν παρουσιάζουν τη μέση ελάχιστη και μέση μέγιστη σχετική υγρασία και θερμοκρασία. Οι ελάχιστες μετρήσεις θερμοκρασίας λαμβάνονταν κατά τη διάρκεια της νύχτας και οι μέγιστες κατά τη διάρκεια της μέρας ενώ οι ελάχιστες μετρήσεις της σχετικής υγρασίας λαμβάνονταν κατά τη διάρκεια της μέρας και οι μέγιστες κατά τη διάρκεια της νύχτας.



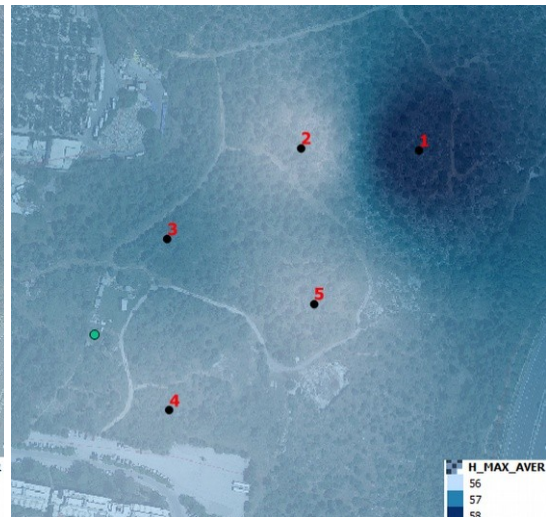
Σχήμα 5.8: Μέση ελάχιστη θερμοκρασία εφτά ημερών.



Σχήμα 5.9: Μέση μέγιστη θερμοκρασία εφτά ημερών.



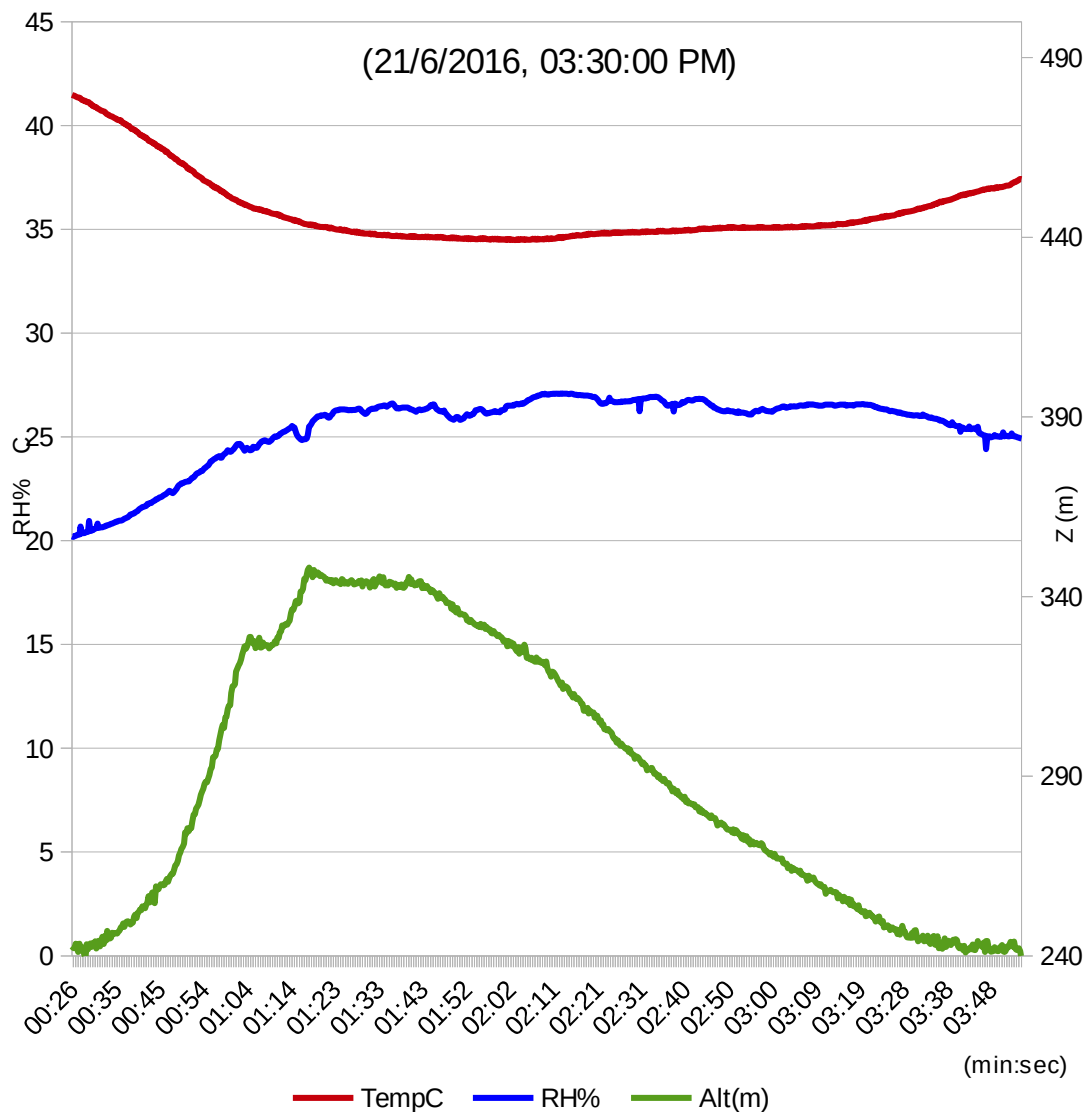
Σχήμα 5.10: Μέση ελάχιστη σχετική υγρασία (%) εφτά ημερών.



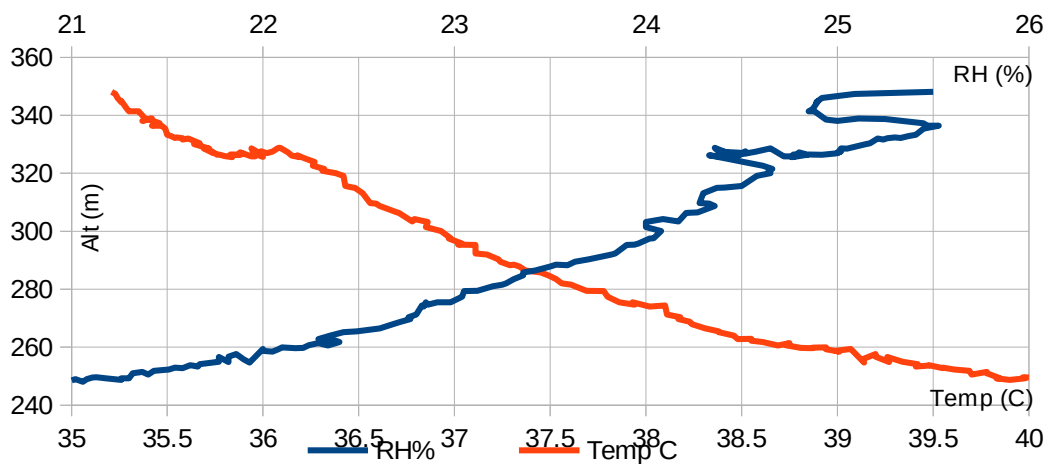
Σχήμα 5.11: Μέση μέγιστη σχετική υγρασία (%) εφτά ημερών.

### 5.2.2 Μετρήσεις τετρακοπτερου (Drone)

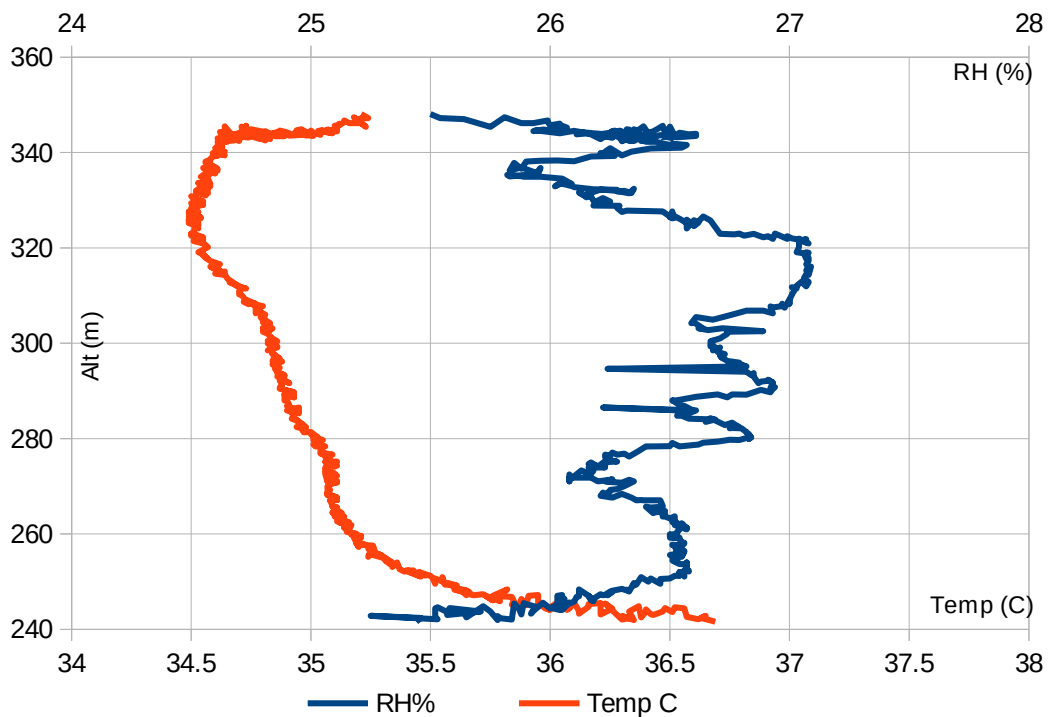
Στις 21/06/2016 πραγματοποιήθηκε πτήση τετρακοπτερου με τοποθετημένο πάνω σε αυτό τη συσκευή μέτρησης περιβαλλοντικών παραμέτρων. Η πτήση πραγματοποιήθηκε κοντά στο μετεωρολογικό σταθμό του εργαστηρίου κλιματολογίας του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών, σε υψόμετρο 240 μέτρα πάνω από την επιφάνεια της θάλασσας. Κατά τη διάρκεια της πτήσης το τετρακόπτερο έφτασε έως και 110 μέτρα πάνω από το έδαφος (Σχήμα 5.12). Η ελάχιστη τιμή μέτρησης της θερμοκρασίας κατά τη διάρκεια της πτήσης ήταν 34°C και της σχετικής υγρασίας 20%. Αντίστοιχα η μέγιστη τιμή μέτρησης της θερμοκρασίας ήταν 41°C και της σχετικής υγρασίας 27%. Η υψηλή μετρούμενη θερμοκρασία που παρατηρήθηκε κατά την εκκίνηση πιθανόν να οφείλεται λόγω της άμεσης έκθεσης του αισθητήρα στον ήλιο εκείνη την χρονική στιγμή.



Σχήμα 5.12: Μετρήσεις τετρακόπτερου (Drone) σε σχέση με τον χρόνο.



Σχήμα 5.13: Μετρήσεις τετρακόπτερου κατά την άνοδο σε σχέση με το υψόμετρο.



Σχήμα 5.14: Μετρήσεις τετρακόπτερου κατά την κάθοδο σε σχέση με το υψόμετρο.

### 5.2.3 Σύνοψη συμπερασμάτων δοκιμών

Οι μετρήσεις μετεωρολογικών παραμέτρων των σταθμών και του τετρακόπτερου ήταν αρκετά ικανοποιητικές. Απώτερος σκοπός είναι η εφαρμογή σε μεγαλύτερη έκταση και όχι σε έκταση που εφαρμόστηκε πειραματικά. Παρόλα αυτά οι δοκιμαστικές μετρήσεις αποδεικνύουν την λειτουργικότητα των σταθμών και την δυνατότητα άμεσης εφαρμογής σε πραγματικές μελέτες. Οι μετρήσεις του τετρακόπτερου μπορούν να συμβάλουν στην διαχείριση και πρόληψη δασικών πυρκαγιών παρέχοντας μετεωρολογικά δεδομένων και για μεγαλύτερα υψόμετρα.

# 6

## Τεχνικές λεπτομέρειες

### 6.1 Πλατφόρμες και προγραμματιστικά εργαλεία

Στο κεφάλαιο αυτό περιγράφονται τα χαρακτηριστικά της συγκεκριμένης υλοποίησης, όπως η πλατφόρμα ανάπτυξης και εκτέλεσης, τα προγραμματιστικά εργαλεία, οι απαιτήσεις της εφαρμογής σε hardware, κ.λ.π. Επίσης, περιγράφεται λεπτομερώς η διαδικασία εγκατάστασης της διπλωματικής σε υπολογιστή μαζί με όλες τις λεπτομέρειες, το απαραίτητο λογισμικό και τις αναγκαίες ρυθμίσεις.

Για την ολοκλήρωση της διπλωματικής χρησιμοποιήθηκαν αποκλειστικά ελεύθερα λογισμικά - λογισμικά ανοιχτού κώδικα ώστε να διατηρηθεί όσο το δυνατόν χαμηλότερο το κόστος υλοποίησης. Στον πίνακα που ακολουθεί (Πίνακας 10) περιγράφονται τα βασικότερα λογισμικά που χρησιμοποιήθηκαν τα οποία εγκαταστάθηκαν σε περιβάλλον Ubuntu Linux.

Λογισμικό	Έκδοση	
Ubuntu	16.04	Λειτουργικό σύστημα (GNU/Linux)
LibreOffice	5.1.2.2 (x64)	Εφαρμογή γραφείου (επεξεργασίας κειμένου, λογιστικών φύλλων κ.α.)
GIMP	2,8	Πρόγραμμα επεξεργασίας ψηφιδωτών γραφικών
Inkscape	0.91	Πρόγραμμα δημιουργίας και επεξεργασίας διανυσματικών γραφικών
Arduino IDE	1.6.7	Ολοκληρωμένο περιβάλλον ανάπτυξης Arduino
Fritzing	0.9.3b	Πρόγραμμα δημιουργίας και επεξεργασίας ολοκληρωμένων κυκλωμάτων
Brackets	1.6	Πλατφόρμα προγραμματισμού
QGIS	2.14.1	Πρόγραμμα γεωγραφικών συστημάτων πληροφοριών

Πίνακας 10: Πλατφόρμες και προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν

#### 6.1.1 Εγκατάσταση προγραμμάτων

Το λειτουργικό σύστημα που χρησιμοποιήθηκε είναι το Ubuntu Linux έκδοσης 16.04 LTS (Long Term Support). Το λειτουργικό αυτό εγκαταστάθηκε σε υπολογιστικό σύστημα παράλληλα με κλειστού τύπου λογισμικό Microsoft Windows 10 Home,

Προτού γίνει η εγκατάσταση του Ubuntu 16.04 LTS μαζί με Windows έπρεπε να προετοιμαστεί ο βασικός χώρος αποθήκευσης του αποθηκευτικού μέσου SSD. Αρχικά έγινε

έλεγχος ανασυγκρότησης και στη συνέχεια συρρίκνωση των τομέων του χώρου αποθήκευσης (partition). Έπρεπε παράλληλα να γίνει απενεργοποίηση της Γρήγορης Εκκίνησης (αυτό απαιτείται μόνο στα Windows 8 / 8.1 / 10).

Το λειτουργικό σύστημα είναι διαθέσιμο στο επίσημο δικτυακό τόπο του Ubuntu (www.ubuntu.com), αφότου έγινε η λήψη του αρχείου .iso (ubuntu-16.04-desktop-amd64.iso), έγινε στη συνέχεια εγγραφή του αρχείου αυτού σε φορητό αποθηκευτικό μέσο USB (μπορεί να αποθηκευτεί και σε cd, dvd, SD card ή άλλο αποθηκευτικό μέσο). Οι διαδικασίες που εκτελέστηκαν ώστε να γίνει η εγκατάσταση του λειτουργικού συστήματος είναι οι εξής:

- 1 Η εγκατάσταση Ubuntu 16.04
  - 1.1 Επιλογή εκκίνησης
  - 1.2 Είσοδος στο Live περιβάλλον
  - 1.3 Σύνδεση στο Internet - δοκιμή ήχου
  - 1.4 Εγκατάσταση Ubuntu 16.04
  - 1.5 Ρυθμίσεις ώρας, γλώσσας, και λογαριασμού
- 2 Επανεκκίνηση και οθόνη επιλογής λειτουργικού
- 3 Βασικές ρυθμίσεις μετά την εγκατάσταση Ubuntu

Στη συνέχεια εγκαταστάθηκαν τα προγράμματα: LibreOffice, GIMP, Inkscape, Arduino IDE, Fritzing, Brackets και QGIS από το περιβάλλον εγκατάστασης εφαρμογών (Κέντρο λογισμικού - Synaptics Pacage Manager) του λειτουργικού Ubuntu.

Το ολοκληρωμένο περιβάλλον προγραμματισμού Arduino (Arduino IDE) παρέχει τις βασικές βιβλιοθήκες. Η εγκατάσταση των βιβλιοθηκών που απαιτήθηκαν για των προγραμματισμό των σταθμών έγινε μέσα από την διαχείριση βιβλιοθηκών του περιβάλλοντος του IDE (Sketch > Include Library > Manage Libraries...).

Για την εγκατάσταση της τελευταίας έκδοσης του QGIS σε Ubuntu απαιτείται η προσθήκη εναλλακτικών αποθετηρίων λογισμικού, προσθέτοντας τις παρακάτω γραμμές στο αρχείο /etc/apt/sources.list file

```
deb http://qgis.org/ubuntu xenial main
deb-src http://qgis.org/ubuntu xenial main
```

Στην συνέχεια απαιτείται η εισαγωγή εντολών στη γραμμή εντολών (Terminal). Το Terminal εκτελείται ανοίγοντας το Dash (κάνοντας κλικ στο εικονίδιο του Ubuntu στο πάνω αριστερά) και στη συνέχεια πληκτρολογώντας Terminal, επιλέγουμε την εφαρμογή Terminal από τα αποτελέσματα που εμφανίζονται και πληκτρολογούμε σε αυτό τις εντολές:

```
sudo apt-get update

sudo apt-get install qgis python-qgis qgis-plugin-grass
```



## 6.2 Διαχείριση δεδομένων πραγματικού χρόνου

Στη συνέχεια περιγράφεται η διαδικασία δημιουργίας λογαριασμού στον εξυπηρετητή δεδομένων `data.sparkfun.com`. Στην ιστοσελίδα `data.sparkfun.com` επιλέγουμε “CREATE” (Create a free data stream immediately at `data.sparkfun.com`), όπου μας παραπέμπει στη φόρμα δημιουργίας καινούριας βάσης δεδομένων. Στη φόρμα αυτή εισάγουμε τον τίτλο (Title), την περιγραφή (Description), στην επιλογή “Show in Public Stream List?” το αφήνουμε Visible, στο πεδίο Fields εισάγουμε τα πεδία των δεδομένων (για παράδειγμα “temp” για τη θερμοκρασία, hum για την υγρασία και ούτω καθεξής), στο πεδίο Stream Alias εισάγουμε μια επιθυμητή υποδιεύθυνση και στα πεδία Tags και Location εισάγουμε λέξεις κλειδιά και την τοποθεσία αντίστοιχα. Στον πίνακα που ακολουθεί αναφέρονται όλες οι απαραίτητες πληροφορίες που συμπληρώθηκαν στα πεδία της φόρμας για τις ανάγκες της διπλωματικής διατριβής.

Πεδία	Τιμές
Title	Weather Data
Description	Weather Data from BME280 sensors
Show in Public Stream List?	Visible
Fields	hum, pres, temp, timestamp, volt
Stream Alias	weatherdata
Tags	Weather, BME280
Location	National and Kapodistrian University of Athens

Πίνακας 11: Οι ρυθμίσεις της βάσης του εξυπηρετητή `data.sparkfun.com` όπως διαμορφώθηκαν για τις ανάγκες της διπλωματικής.

Αφού αποθηκεύσουμε τις ρυθμίσεις (SAVE) μας παραπέμπει σε σελίδα με όλες τις απαραίτητες πληροφορίες διαχείρισης της βάσης δεδομένων που δημιουργήσαμε. Στη σελίδα αυτή υπάρχει η δυνατότητα αποθήκευσης των πληροφοριών διαχείρισης σε αρχείο .JSON (Download your keys as a JSON file.). Το περιεχόμενο του αρχείου .JSON περιέχει τη διεύθυνση της βάσης και τα κλειδιά διαχείρισης αυτής όπως φαίνεται παρακάτω.

```
{
  "title": "Weather Data",
  "outputUrl": "http://data.sparkfun.com/output/wpq4GgjAa9U1g2dbEGGL",
  "inputUrl": "http://data.sparkfun.com/input/wpq4GgjAa9U1g2dbEGGL",
  "manageUrl": "http://data.sparkfun.com/streams/wpq4GgjAa9U1g2dbEGGL",
  "publicKey": "wpq2GgjAa9U1g4dbEGGL",
  "privateKey": "wzW7VqEb01iRke8dD22p",
  "deleteKey": "Mv1pX60jwNfK5ZoN4kkn"}

```

# 7

## Επίλογος

### 7.1 Σύνοψη και συμπεράσματα

Όσον αφορά τα αυτόνομα συστήματα μέτρησης περιβαλλοντικών παραμέτρων που υλοποιήθηκαν στα πλαίσια της διπλωματικής εργασίας, αντιμετωπίστηκαν όλα τα προβλήματα που εμφανίστηκαν και λειτούργησαν με τρόπο που ξεπέρασαν κατά πολύ τις αρχικές προσδοκίες. Τα συστήματα μετάδοσης περιβαλλοντικών παραμέτρων σε πραγματικό χρόνο λειτούργησαν με αρκετά ικανοποιητικό τρόπο, ωστόσο απαιτούν βελτιστοποίηση. Καθιστούν όμως μια πολύ σημαντική βάση για μελλοντικές επεκτάσεις του συστήματος.

Οι μετρήσεις των κόμβων είναι αρκετά ικανοποιητικές, παρόλα αυτά λόγω το ότι οι κόμβοι τοποθετήθηκαν σε πολύ κοντινή απόσταση μεταξύ τους, οι μετρήσεις έχουν πολύ μικρές διαφοροποιήσεις, με αποτέλεσμα να μην προκύπτουν αξιόλογα συμπεράσματα, πέρα το ότι η συγκεκριμένη περιοχή που έγιναν οι δοκιμές μπορεί να θεωρηθεί ως ενιαία μετεωρολογικά.

Βασικός σκοπός είναι η τοποθέτηση των συστημάτων αυτών σε απομακρυσμένες περιοχές, όπου δεν υπάρχουν μετεωρολογικοί σταθμοί ή άλλα συστήματα μέτρησης και σε μεγάλες αποστάσεις μεταξύ τους, με απώτερο σκοπό τη δημιουργία αναλυτικότερων κλιματολογικών χαρτών για την χρήση τους στους δείκτες επικινδυνότητας εκδήλωσης δασικών πυρκαγιών. Με αυτά τα δεδομένα θα είναι δυνατόν να παράγονται χάρτες επικινδυνότητας ανά περιοχές με κοινές μετεωρολογικές συνθήκες και όχι ανά δήμο όπως παράγονται σήμερα από τη Γενική Γραμματεία Πολιτικής Προστασίας.

Επίσης τα συστήματα πραγματικού χρόνου μπορούν να αξιοποιηθούν για την παρατήρηση των πραγματικών κλιματολογικών συνθηκών που επικρατούν γύρω από μια δασική πυρκαγιά, εφόσον τοποθετηθούν σε οχήματα και τετρακόπτερα ή ενσωματωθούν στον εξοπλισμό των δασοπυροσβεστών.

Όπως αναλύεται στο επόμενο κεφαλαίο οι δυνατότητες επέκτασης των συστημάτων αυτών είναι πολύ μεγάλες. Τα συστήματα αυτά εδραιώνουν μια βάση ώστε να αξιοποιηθούν σε μελέτες συναφούς αντικείμενου, συνεισφέροντας ως εναλλακτική οικονομική πρόταση για επιστημονικές μελέτες.

## 7.2 Μελλοντικές επεκτάσεις

Στα πλαίσια της διπλωματικής εργασίας, κατασκευάστηκε μόνο ένας σταθμός με τη δυνατότητα μετάδοσης των δεδομένων σε πραγματικό χρόνο, οι υπόλοιποι σταθμοί δημιουργήθηκαν χωρίς τη δυνατότητα αυτή. Μια πολύ σημαντική επέκταση, θα ήταν η αναβάθμιση όλων των σταθμών, ώστε να υπάρχει δυνατότητα μετάδοσης δεδομένων σε πραγματικό χρόνο. Αυτό μπορεί να επιτευχθεί με την προσθήκη υποσυστήματος ασύρματης επικοινωνίας σε κάθε σταθμό.

Πρωταρχική ιδέα ήταν, η δημιουργία δικτύου σταθμών μετάδοσης δεδομένων βροχοπτώσεων πραγματικού χρόνου, για την έγκυρη ενημέρωση εκδήλωσης πλημμυρικών φαινομένων. Αυτό μπορεί να επιτευχθεί με την προσθήκη μονάδας μέτρησης του ύψους βροχής σε κάθε σταθμό. Οι σταθμοί αυτοί στη συνέχεια αν τοποθετηθούν εντός μιας λεκάνης απορροής θα λαμβάνονται δεδομένα βροχοπτώσεων σε πραγματικό χρόνο και με την χρήση αυτοματοποιημένων συστημάτων διαχείρισης των δεδομένων θα υπάρχει η δυνατότητα έγκυρης πρόβλεψης εκδήλωσης πλημμυρικών φαινομένων.

Επίσης παρόμοια συστήματα χρησιμοποιούνται και για πειραματικές γεωφυσικές μελέτες. Στη ιστοσελίδα [geocomputing.co.uk](http://geocomputing.co.uk) (2010) αναφέρεται ο τρόπος κατασκευής ενός μαγνητόμετρου με τη χρήση Arduino, και στη ιστοσελίδα [psychogeophysics.org](http://psychogeophysics.org) (2011) αναφέρει τον τρόπο κατασκευής συστήματος για γεω-ηλεκτρικές διασκοπήσεις.

Τα συστήματα που σχεδιάστηκαν και κατασκευάστηκαν στη παρούσα διπλωματική εργασία, μπορούν να αξιοποιηθούν και στη μετεωρολογία, για τη βελτιστοποίηση των μετεωρολογικών προβλέψεων. Αξίζει να αναφερθεί πως το σύστημα πραγματικού χρόνου που σχεδιάστηκε και κατασκευάστηκε στα πλαίσια της διπλωματικής εργασίας χρησιμοποιήθηκε στη παρουσίαση της ομάδας “Smart Weather Balloon” στο διαγωνισμό ActInSpace που οργάνωσε το Corallia και το si-Cluster στις 20-21 Μαΐου. Η ομάδα αυτή αποτελούνταν από τον Κωνσταντίνο Αναστασάκη, τον Κωνσταντίνο Δαβίδ και τον Ανδρέα Φράγκο. Στην Ελλάδα το “ActInSpace” διοργανώθηκε για πρώτη φορά, από το si-Cluster στην Αθήνα και την Πάτρα ταυτόχρονα. Ο εθνικός νικητής εκπροσώπησε την Ελλάδα στον παγκόσμιο τελικό του ActInSpace που διεξαχθεί στην Τουλούζη στις 29 Ιουνίου 2016.

Η ομάδα “Smart Weather Balloon” σχεδίασε και παρουσίασε ένα “έξυπνο” τηλεκατευθυνόμενο μετεωρολογικό μπαλόνι, που αξιοποιούσε πέρα των άλλων και το σύστημα πραγματικού χρόνου που υλοποιήθηκε στα πλαίσια της παρούσας διπλωματικής. Το μετεωρολογικό μπαλόνι θα έχει την δυνατότητα να συλλέγει δεδομένα τα οποία μπορεί να είναι εικόνες ηλεκτρομαγνητικού φάσματος πέρα από το ορατό και μετεωρολογικά δεδομένα για καλύτερες προβλέψεις καιρικών φαινομένων. Η ιδέα αυτή κέρδισε τη δεύτερη θέση στον διαγωνισμό αυτό και δημοσιεύτηκε στις εφημερίδες “Υπαιθρος Χώρα” (3.06.2016) και “Η Καθημερινή” (4.06.2016).

## Βιβλιογραφία

Brian T.S. (12 Ιουνίου 2015), ESP8266 based Wifi weather monitoring system, Embedded Systems Laboratory, Ανακτήθηκε 2 Μαρτίου 2016 από: <http://hpclab.blogspot.gr/2015/06/esp8266-based-wifi-weather-monitoring.html>

ΕΛ/ΛΑΚ, Τί είναι το Ελεύθερο Λογισμικό; Οφέλη Χρήσης Ελεύθερου Λογισμικού / Λογισμικού Ανοικτού Κώδικα, Ανακτήθηκε 10 Απριλίου 2016 από: <https://mathe.ellak.gr/?p=15>

EAA, Θησαυρός μετεωρολογικών Όρων, Εθνικό Αστεροσκοπείο Αθηνών - Ινστιτούτο Ερευνών Περιβάλλοντος, Διαθέσιμο στην ιστοσελίδα: <http://www.meteo.gr/meteorplus/dictionary.cfm>

Elgar P. (2003), Αισθητήρες μέτρησης και Έλεγχου, Εκδόσεις Τζιολα, ISBN 960-8050-31-6

ESA, NDVI - Βλάστηση από το διάστημα, Διαθέσιμο στην ιστοσελίδα: [https://www.esa.int/SPECIALS/Eduspace\\_Global\\_GR/SEM61KTWT1H\\_0.html](https://www.esa.int/SPECIALS/Eduspace_Global_GR/SEM61KTWT1H_0.html)

GNU.com (2014), Τι είναι το Ελεύθερο Λογισμικό;, Ίδρυμα Ελεύθερου Διαθέσιμο στον επίσημο δικτυακό τόπο του Ιδρύματος Ελεύθερου Λογισμικού: <http://www.gnu.org/philosophy/free-sw.el.html>

GNU.com (2015), Άδειες, Ίδρυμα Ελεύθερου Λογισμικού, Διαθέσιμο στον επίσημο δικτυακό τόπο του Ιδρύματος Ελεύθερου Λογισμικού: <http://www.gnu.org/licenses/licenses.el.html>

GSCP.gr (2008), Ημερήσιο δελτίο πρόβλεψης κινδύνου, Γενική Γραμματεία Πολιτικής Προστασίας, Διαθέσιμο στην ιστοσελίδα: <http://www.gscp.gr/ggpp/site/home/ws/promote/fisikes/pirkagies/deltio.csp>

Kandrsmith R. (5 Απριλίου 2016), Wide range of Hygrometers: DHT22, AM2302, AM2320, AM2321, SHT71, HTU21D, Si7021, BME280, Ανακτήθηκε 10 Απριλίου 2016 από: [http://www.kandrsmith.org/RJS/Misc/Hygrometers/calib\\_many.Html](http://www.kandrsmith.org/RJS/Misc/Hygrometers/calib_many.Html)

Kon, S., Oldham, K., & Horowitz, R. (2007). Piezoresistive and piezoelectric MEMS strain sensors for vibration detection. In The 14th International Symposium on: Smart Structures and Materials & Nondestructive Evaluation and Health Monitoring (pp. 65292V-65292V). International Society for Optics and Photonics.

Λυβιάκης Γ. (9 Ιουνίου 2013), Πρόβλεψη πυρκαγιών σε πραγματικό χρόνο, Κυριακάτικη Ελευθεροτυπία, Ανακτήθηκε 9 Απριλίου 2016 από: <http://www.enet.gr/?i=news.el.article&id=368330>

Μαρκάτος Ν., Βεσκούκης Β., Κυρανούδης Χ. (2008), Firementor: Επιχειρησιακό Σύστημα Σχεδιασμού Και Υποστήριξης Αποφάσεων Για Διαχείριση Δασικών Πυρκαγιών, ψηφιακή βιβλιοθήκη Τεχνικού Επιμελητηρίου Ελλάδος. Διαθέσιμο στον

επίσημο δικτυακό τόπο του Τεχνικού Επιμελητηρίου Ελλάδος:  
[http://library.tee.gr/digital/m2310/m2310\\_contents.htm](http://library.tee.gr/digital/m2310/m2310_contents.htm)

Σουλιώτης Γ. (24 Αυγούστου 2006), Τα σεκιούριτι των δασών, μέθانا με ειδήσεις, Ανακτήθηκε 11 Απριλίου 2016 από: <http://mymethana.gr/?q=node/185>

Σουλιώτης Γ. (25 Μαΐου 2006), Αισθητήρες σε δένδρα κατά πυρκαγιών, Η Καθημερινή, Ανακτήθηκε 12 Απριλίου 2016 από: <http://www.kathimerini.gr/251897/article/epikairothta/ellada/ais8hthres-se-dendra-kata-pyrkagiwn>

Patrice-Emmanuel Schmitz, European Union Public Licence (EURL) v.1.1 (2009), Διαθέσιμο στην ιστοσελίδα της Ευρωπαϊκής Επιτροπής: <http://ec.europa.eu/idabc/servlets/Docffc4.pdf?id=32399>

Roveti D. (2001), Choosing a Humidity Sensor: A Review of Three Technologies, Ανακτήθηκε 10 Απριλίου 2016 από: <http://www.sensorsmag.com/sensors/humidity-moisture/choosing-a-humidity-sensor-a-review-three-technologies-840>

Tarun Agarwal, Temperature Sensors – Types, Working & Operation, Ανακτήθηκε 2 Απριλίου 2016 από: <https://www.elprocus.com/temperature-sensors-types-working-operation/>

Ξανθόπουλος Γ. (2012), Η δασική πυρκαγιά ως πρόβλημα Σελ. 192 στο Α.Χ. Παπαγεωργίου, Γ. Καρέτσος και Γ. Κατσαδωράκης. Το δάσος: Μια ολοκληρωμένη προσέγγιση. WWF Ελλάς, Αθήνα, ISBN: 978-960-7506-28-3.

# Παράρτημα

## 1. Κώδικας αυτόνομου συστήματος καταγραφής μετεωρολογικών παραμέτρων

```
// Στην αρχει του κώδικα αντλούνται όλες οι απαραίτητες βιβλιοθήκες
#include <DS3232RTC.h> //Βιβλιοθήκη χρόνου (DS3232RTC)
#include <Time.h> //Βιβλιοθήκη χρόνου (Arduino)
#include <Wire.h> //Βιβλιοθήκη επικοινωνίας i2c
#include <SPI.h> //Βιβλιοθήκη επικοινωνίας SPI
#include <SD.h> //Βιβλιοθήκη κάρτας SD
#include <BME280.h> //Βιβλιοθήκη αισθητήρα BME280
#include <VoltageReference.h> //Βιβλιοθήκη μέτρησης τάσης

VoltageReference vRef;

const int chipSelect = 10; //SD
File dataFile; //SD

// Ρυθμίσεις αισθητήρα BME280
// Default values : BME280 bme(1, 1, 1, 3, 5, 0, false, 0x76);
// Οι αριθμοί αντιστοιχούν στους παράμετρος όπως φαίνεται παρακάτω
// (Temp Oversampling Rate, Hum Ov/ing Rate, Pres Ov/ing Rate, Mode,
// Standby Time, Filter, SPI Enable, BME280 Address)
/** Temperature Oversampling Rate (tosr): uint8_t, default = 0x1
// values: 0 = Skipped, 1 = x1, 2 = x2, 3 = x4, 4 = x8, 5 = x16
/** Humidity Oversampling Rate (hosr): uint8_t, default = 0x1
// values: 0 = Skipped, 1 = x1, 2 = x2, 3 = x4, 4 = x8, 5 = x16
/** Pressure Oversampling Rate (posr): uint8_t, default = 0x1
// values: 0 = Skipped, 1 = x1, 2 = x2, 3 = x4, 4 = x8, 5 = x16
/** Mode: uint8_t, default = forced
// values: Sleep = 0, Normal = 1 and 2, Forced = 3
/** Standby Time (st): uint8_t, default = 1000ms
// values: 0 = 0.5ms, 1 = 62.5ms, 2 = 125ms, 3 = 250ms, 4 = 250ms,
// 5 = 1000ms, 6 = 10ms, 7 = 20ms
/** Filter: uint8_t, default = None
// values: 0 = off, 1 = 2, 2 = 4, 3 = 8, 4 = 16
/** SPI Enable: bool, default = false
// values: true = enable, false = disable
/** BME280 Address: uint8_t, default = 0x76
// values: any uint8_t

BME280 bme(1, 1, 1, 3, 5, 0);

bool metric = true;

/* ===== Prototypes ===== */
/* === Print a message to stream with the temp, humidity and pressure.
=== */
void printBME280Data(Stream * client);
/* === Print a message to stream with the altitude, and dew point. ===
*/
void printBME280CalculatedData(Stream* client);
/* ===== END Prototypes ===== */

void setup(void)
{
  Serial.begin(9600);
  while (!Serial) {} // wait for serial port to connect. Needed for
Leonardo only
  while(!bme.begin()){
    Serial.println("Could not find BME280 sensor!");
    delay(10);
  }

  //Time
  setSyncProvider(RTC.get()); // the function to get the time from the
RTC
  if(timeStatus() != timeSet)
    Serial.println("Unable to sync with the RTC");
```

```

else
  Serial.println("RTC has set the system time");

//SD
Serial.print("Initializing SD card...");
// make sure that the default chip select pin is set to
// output, even if you don't use it:
pinMode(SS, OUTPUT);

// see if the card is present and can be initialized:
//if (!SD.begin(10, 11, 12, 13)) { //Only for Leonardo an Mega for
ather Boards:
if (!SD.begin(chipSelect)) {
  Serial.println("Card failed, or not present");
  // don't do anything more:
  while (1) ;
}
Serial.println("card initialized.");

// Open up the file we're going to log to!
dataFile = SD.open("datalog.txt", FILE_WRITE);
if (! dataFile) {
  Serial.println("error opening datalog.txt");
  // Wait forever since we cant write data
  while (1) ;
  delay(10);
}

//Voltage Reference begin
vRef.begin();

//Print Header
Serial.println("Time   Date   TempC RH%   hPa Alt(m) Dew p mV");
dataFile.println("Time Date TempC RH% hPa Alt(m) Dew_p mV");
delay(10);
}

void loop(void) {

  int vcc = vRef.readVcc();

  printDigits(hour());
  Serial.print(':');
  printDigits(minute());
  Serial.print(':');
  printDigits(second());
  Serial.print(' ');
  printDigits(day());
  Serial.print('/');
  printDigits(month());
  Serial.print('/');
  Serial.print(year());
  Serial.print(' ');

  printBME280Data(&Serial);
  printBME280CalculatedData(&Serial);

  Serial.print(' ');
  Serial.print(vcc); // Input voltage in mV
  Serial.println();

  dataFile.print(hour());
  dataFile.print(':');
  dataFile.print(minute());
  dataFile.print(':');
  dataFile.print(second());
  dataFile.print(' ');
  dataFile.print(day());

```



```

dataFile.print('/');
dataFile.print(month());
dataFile.print('/');
dataFile.print(year());
dataFile.print(' ');

printBME280Data(&dataFile);
printBME280CalculatedData(&dataFile);

dataFile.print(' ');
dataFile.print(vcc);
dataFile.println();
// The following line will 'save' the file to the SD card after every
// line of data - this will use more power and slow down how much
data
// you can read but it's safer!
// If you want to speed up the system, remove the call to flush() and
it
// will save the file only every 512 bytes - every time a sector on
the
// SD card is filled with data.
dataFile.flush();

// Take 1 measurement every x milliseconds 1000ms = 1s
//delay(1800000);
delay(9900);
}

void printDigits(int digits)
{
// prints preceding colon and leading 0 for digital clock display
if(digits < 10)
Serial.print('0');
Serial.print(digits);
}

/* ==== BME280 Functions ==== */
void printBME280Data(Stream* client){
float temp(NAN), hum(NAN), pres(NAN);
uint8_t pressureUnit(1); // unit: B000 = Pa, B001 = hPa, B010 = Hg,
B011 = atm,
//B100 = bar, B101 = torr, B110 = N/m^2, B111 = psi
bme.ReadData(pres, temp, hum, metric, pressureUnit);// Parameters:
(float& pressure,
//float& temp, float& humidity, bool hPa = true, bool celsius =
false)
client->print(temp);
client->print(' ');
client->print(hum);
client->print(' ');
client->print(pres);
client->print(' ');
}
void printBME280CalculatedData(Stream* client){
float altitude = bme.CalculateAltitude(metric);
float dewPoint = bme.CalculateDewPoint(metric);
client->print(altitude);
client->print(' ');
client->print(dewPoint);
}
/* ==== END BME280 Functions ==== */

```

## 2. Κώδικας αυτόνομου συστήματος καταγραφής μετεωρολογικών παραμέτρων για Drone

```
#include <Time.h> //Βιβλιοθήκη χρόνου (Arduino)
#include <Wire.h> //Βιβλιοθήκη επικοινωνίας i2c
#include <SPI.h> //Βιβλιοθήκη επικοινωνίας SPI
#include <SD.h> //Βιβλιοθήκη κάρτας SD
#include <BME280.h> //Βιβλιοθήκη αισθητήρα BME280

const int chipSelect = 10; //SD
File dataFile; //SD

// Default values : BME280 bme(1, 1, 1, 3, 5, 0, false, 0x76);
// (Temp Oversampling Rate, Hum Ov/ing Rate, Pres Ov/ing Rate, Mode,
Standby Time, Filter,
// SPI Enable, BME280 Address)
BME280 bme(1, 1, 1, 3, 0, 0);

bool metric = true;

void printBME280Data(Stream * client);
void printBME280CalculatedData(Stream* client);

void setup(void)
{
  Serial.begin(9600);
  while (!Serial) {} // wait for serial port to connect. Needed for
Leonardo only
  while(!bme.begin()){
    Serial.println("Could not find BME280 sensor!");
    delay(10);
  }

  //SD
  Serial.print("Initializing SD card...");
  pinMode(SS, OUTPUT);

  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    while (1) ;
  }
  Serial.println("card initialized.");

  // Open up the file we're going to log to!
  dataFile = SD.open("datalog.txt", FILE_WRITE);
  if (! dataFile) {
    Serial.println("error opening datalog.txt");
    // Wait forever since we cant write data
    while (1) ;
    delay(10);
  }

  //Print Header
  Serial.println("Time Date TempC RH% hPa Alt(m) Dew p");
  dataFile.println("Time Date TempC RH% hPa Alt(m) Dew_p");
  delay(10);
}

void loop(void) {

  printDigits(hour());
  Serial.print(':');
  printDigits(minute());
  Serial.print(':');
  printDigits(second());
  Serial.print(' ');
  printBME280Data(&Serial);
```

```

printBME280CalculatedData(&Serial);
Serial.println();

dataFile.print(hour());
dataFile.print(':');
dataFile.print(minute());
dataFile.print(':');
dataFile.print(second());
dataFile.print(' ');
printBME280Data(&dataFile);
printBME280CalculatedData(&dataFile);
dataFile.println();

dataFile.flush();

// Take 1 measurement every x milliseconds 1000ms = 1s
delay(100);
}

void printDigits(int digits)
{
  // prints preceding colon and leading 0 for digital clock display
  if(digits < 10)
    Serial.print('0');
  Serial.print(digits);
}

/* ==== BME280 Functions ==== */
void printBME280Data(Stream* client){
  float temp(NAN), hum(NAN), pres(NAN);
  uint8_t pressureUnit(1); // unit: B000 = Pa, B001 = hPa, B010 = Hg,
B011 = atm,
  //B100 = bar, B101 = torr, B110 = N/m^2, B111 = psi
  bme.ReadData(pres, temp, hum, metric, pressureUnit); // Parameters:
(float& pressure,
  //float& temp, float& humidity, bool hPa = true, bool celsius =
false)
  client->print(temp);
  client->print(' ');
  client->print(hum);
  client->print(' ');
  client->print(pres);
  client->print(' ');
}
void printBME280CalculatedData(Stream* client){
  float altitude = bme.CalculateAltitude(metric);
  float dewPoint = bme.CalculateDewPoint(metric);
  client->print(altitude);
  client->print(' ');
  client->print(dewPoint);
}

/* ==== END BME280 Functions ==== */

```

### 3. Κώδικας κόμβου μετάδοσης δεδομένων πραγματικού χρόνου

#### 3.1 Κώδικας html απεικόνισης δεδομένων πραγματικού χρόνου

```
<html>
  <head>
    <title>Wireless Weather</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js
"></script>
    <script src="https://www.google.com/jsapi"></script>

    <script>
      // onload callback
      function drawChart() {

        var cloud_url = 'https://data.sparkfun.com/output/';
        // Public Key from https://data.sparkfun.com
        var public_key = 'ZGom2qynArU5pj2YxWzp';

        // JSONP request
        var jsonData = $.ajax({
          url: cloud_url + public_key + '.json',
          data: {page: 1},
          dataType: 'jsonp',
        }).done(function (results) {
          var latest = results[0];
          var gauge = new google.visualization.Gauge($
('#gauge').get(0));
          var gaugeData = google.visualization.arrayToDataTable([
            ['Label', 'Value'],
            ['Temp', 0],
            ['Humidity', 0],
            ['Pressure', 0]
          ]);
          var options = {
            width: 600,
            redFrom: 90, redTo: 100,
            yellowFrom:80, yellowTo: 90,
            greenFrom:70, greenTo: 80,
            minorTicks: 5
          };

          // For animation purpose only
          gauge.draw(gaugeData,options);

          // Show real data
          gaugeData.setValue(0,1,parseFloat(latest.temp));
          gaugeData.setValue(1,1,parseFloat(latest.hum));
          gaugeData.setValue(2,1,parseFloat(latest.pres));

          gauge.draw(gaugeData,options);
          var data = new google.visualization.DataTable();

          data.addColumn('datetime', 'Time');
          data.addColumn('number', 'Temp (C)');
          data.addColumn('number', 'Humidity (RH%)');
          data.addColumn('number', 'Pressure (hPa*10)');

          $.each(results, function (i, row) {
            data.addRow([
              (new Date(row.timestamp)),
              parseFloat(row.temp),
              parseFloat(row.hum),
              parseFloat(row.pres*10)
            ]);
          });
        });
      }
    </script>
  </head>
  <body>
    <div id="gauge">
      <img alt="A gauge chart showing weather data: Temperature, Humidity, and Pressure." data-bbox="200 400 800 850"/>
    </div>
  </body>
</html>
```

```

        var chart = new google.visualization.LineChart($('#temperature').get(0));
        chart.draw(data, {
            title: 'ESP8266 BME280 Wifi Weather Station'
        });

    });
}
// load chart lib
google.load('visualization', '1', {
    packages: ['corechart','gauge']
});

// call drawChart once google charts is loaded
google.setOnLoadCallback(drawChart);
</script>

<script src="jQuery.js" type="text/javascript"></script>
<script type="text/javascript">

    $(document).ready(function(){
        // First load the chart once
        drawChart();
        // Set interval to call the drawChart again
        setInterval(drawChart, 3000);
    });

</script>
</head>
<body style="text-align: center;">
    <h1>
Anastasakis Konstantinos Thesis, R.N. 21314</h1>
<h2>
Design and implementation of low cost meteorological recording network
system for the prevention and management of forest fires.</h2>
<div id="gauge" style="margin-left: auto; margin-right: auto; width:
600px;">
</div>
<div id="temperature" style="width: 100%;">
</div>
POSTGRADUATE PROGRAM "PREVENTION AND MANAGEMENT OF NATURAL DISASTERS"
- NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS SCHOOL OF SCIENCES
FACULTY OF GEOLOGY AND GEOENVIRONMENT
</body>
</html>

```

### 3.2 Κώδικας μικροελεγκτή συστήματος πραγματικού χρόνου

```

#include <Wire.h> //Βιβλιοθήκη επικοινωνίας i2c
#include <BME280.h> //Βιβλιοθήκη αισθητήρα BME280
#include <ESP8266WiFi.h> //Βιβλιοθήκη WiFi (ESP8266)

extern "C" {
    #include "user_interface.h"
    //uint16 readvdd33(void);
}
ADC_MODE(ADC_VCC);

const char* ssid = "....."; //Εδώ εισάγουμε το όνομα του δικτύου
const char* password = "....."; //Εδώ εισάγουμε το κωδικό του
δικτύου

const char* host = "data.sparkfun.com";
const char* publicKey = "wpq2GgjAa9Ulg4dbEGGL";

```

```

const char* privateKey = "wzW7VqEb01iRke8dD22p";

/* ==== Global Variables ==== */
// Default values : BME280 bme(1, 1, 1, 3, 5, 0, false, 0x76);
// (Temp Oversampling Rate, Hum Ov/ing Rate, Pres Ov/ing Rate, Mode,
Standby Time, Filter,
// SPI Enable, BME280 Address)
BME280 bme(1, 1, 1, 3, 5, 0);

bool metric = true;
/* ==== END Global Variables ==== */

/* ==== Prototypes ==== */
/* === Print a message to stream with the temp, humidity and pressure.
=== */
void printBME280Data(Stream * client);
/* === Print a message to stream with the altitude, and dew point. ===
*/
void printBME280CalculatedData(Stream* client);
/* ==== END Prototypes ==== */

void setup(void)
{
  Serial.begin(9600);
  while (!Serial) {}// wait for serial port to connect. Needed for
Leonardo only
  delay(250);
  //Wire.pins(int sda, int scl)
  Wire.pins(0, 2); //on ESP-01.
  Wire.begin();

  while (!bme.begin()){
    Serial.println("Could not find BME280 sensor!");
    delay(250);

    // Connecting to a WiFi network
    Serial.println();
    Serial.println();
    Serial.print("ESP8266 Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
  }
}

void loop(void) {
  delay(100);

  // Read temperature as Celsius
  float t = bme.ReadTemperature();
  // Read Humidity
  float h = bme.ReadHumidity();
  // Read Pressure as atm*
  float p = bme.ReadPressure(3); // unit: 0 = Pa, 1 = hPa, 2 = Hg, 3 =
atm,
  //4 = bar, 5 = torr, 6 = N/m^2, 7 = psi
  // Input voltage in V (mV/1000=V)

```

```

float v = ESP.getVcc() / 1000.0;

float altitude = bme.CalculateAltitude(metric);
float dewPoint = bme.CalculateDewPoint(metric);

// Check if any reads failed and exit early (to try again).
if (isnan(h) || isnan(t) || isnan(p)) {
  Serial.println("Failed to read from BME280 sensor!");
  return;
}

// Compute heat index
Serial.print("Temperature: ");
Serial.print(t);
Serial.print(" *C");
Serial.print(", Humidity: ");
Serial.print(h);
Serial.print(", Pressure: ");
Serial.print(p);
Serial.print(", Voltage: ");
Serial.print(v); // Input voltage in mV
Serial.print("mV");

Serial.print(", connecting to ");
Serial.println(host);

// Use WiFiClient class to create TCP connections
WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
  Serial.println("connection failed");
  return;
}

// We now create a URI for the request
String url = "/input/";
url += publicKey;
url += "?private_key=";
url += privateKey;
url += "&temp=";
url += t;
url += "&hum=";
url += h;
url += "&pres=";
url += p;
url += "&volt=";
url += v;

Serial.print("Requesting URL: ");
Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
  "Host: " + host + "\r\n" +
  "Connection: close\r\n\r\n");
delay(10);

// Read all the lines of the reply from server and print them to
Serial
while(client.available()){
  String line = client.readStringUntil('\r');
  Serial.print(line);
}

Serial.println();
Serial.println("closing connection");
delay(2890); // Send data every x milliseconds (1000ms = 1s)
}

```

---

Μεταπτυχιακή διατριβή ειδίκευσης που εκπονήθηκε στα πλαίσια του μεταπτυχιακού δι-ιδρυματικού προγράμματος σπουδών «Πρόληψη και Διαχείριση Φυσικών Καταστροφών».

Επιμέλεια: Αναστασάκης Κωνσταντίνος (Α.Μ. 21314)

Τίτλος: Μελέτη και υλοποίηση δικτύου καταγραφής μετεωρολογικών παραμέτρων χαμηλού κόστους για την πρόληψη και αντιμετώπιση των δασικών πυρκαγιών.

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Αθήνα 2016

---



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ



ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ  
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ  
Επιχειρησιακό Πρόγραμμα  
Εκπαίδευσης και Αρχικής  
Επαγγελματικής Κατάρτισης