

ΤΕΙ ΠΕΙΡΑΙΑ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Π.Μ.Σ. “ΕΦΑΡΜΟΣΜΕΝΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ”

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και
αποστολή των μετρήσεων σε συσκευή Android**

Νικόλαος Γρηγορόπουλος

Εισηγητής: Δρ Ιωάννης Έλληνας

ΑΘΗΝΑ
ΝΟΕΜΒΡΙΟΣ 2017

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

**Νικόλαος Γρηγορόπουλος
Α.Μ. ais0053**

Εισηγητής:

Δρ Ιωάννης Έλληνας, Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος *Γρηγορόπουλος Νικόλαος*, του *Ιωάννη*, με αριθμό μητρώου *ais0053* φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

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

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

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της πλατφόρμας Arduino και του λειτουργικού συστήματος Android. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου Ιωάννης Έλληνας, τον οποίο θα ήθελα να ευχαριστήσω.

Ακόμα θα ήθελα να ευχαριστήσω την κυρία Λίνα Κλουσάκου για τις πολύτιμες συμβουλές της καθώς και για το γεγονός ότι αποτέλεσε πηγή έμπνευσης για μένα όλο αυτό το διάστημα. Παράλληλα θα ήθελα να ευχαριστήσω την οικογένειά μου για την στήριξή της καθόλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής.

ΠΕΡΙΛΗΨΗ

Για την παρατήρηση των καιρικών φαινομένων έχουν κατασκευαστεί ποικίλα μετεωρολογικά συστήματα τα οποία κυμαίνονται σε ένα ευρύ φάσμα τιμών και δυνατοτήτων. Αυτές οι λύσεις μπορεί να είναι από μια απλή απεικόνιση των δεδομένων, έως πολύπλοκα συστήματα. Το Arduino αποτελεί ένα χαμηλού κόστους υπολογιστικό σύστημα, με μεγάλο εύρος περιφερειακών συσκευών, το οποίο προγραμματίζεται εύκολα και υποστηρίζεται από ευρεία κοινότητα μελών.

Στην παρούσα διπλωματική εργασία παρουσιάζεται η διαδικασία υλοποίησης ενός αυτόνομου φορητού μετεωρολογικού σταθμού με χρήση της υπολογιστικής πλατφόρμας Arduino καθώς και ο προγραμματισμός της με σκοπό την λήψη μετεωρολογικών δεδομένων όπως η θερμοκρασία, η υγρασία, ο δείκτης θερμότητας, η ατμοσφαιρική πίεση και η ένταση του φωτός. Επίσης δημιουργήθηκε μια εφαρμογή σε περιβάλλον του λειτουργικού Android έτσι ώστε τα δεδομένα των μετρήσεων να αποστέλλονται μέσω Bluetooth σε κάποια κινητή συσκευή αυτού που χρησιμοποιεί αυτό το λειτουργικό (κινητό τηλέφωνο, tablet, κτλ.).

Αρχικά γίνεται μια ανάλυση των μετεωρολογικών φαινομένων και τονίζεται η σημασία τους για την πρόγνωση καιρού ενώ αναλύεται και η μεθοδολογία καταγραφής τους, σύμφωνα με τους αλγορίθμους που έχει θεσπίσει ο Παγκόσμιος Οργανισμός Μετεωρολογίας. Σε επόμενη παράγραφο γίνεται αναφορά στους αισθητήρες που χρησιμοποιήθηκαν για την λήψη των μετεωρολογικών δεδομένων, τα κριτήρια επιλογής αυτών καθώς και ο τρόπος συνδεσμολογίας τους με το Arduino. Στη συνέχεια πραγματοποιείται μια σύντομη αναφορά στο λειτουργικό Android προκειμένου να γίνουν κατανοητές οι πτυχές της δομής και της λειτουργίας του. Έπειτα παρουσιάζεται η διαδικασία προγραμματισμού του Arduino ενώ αναλύεται λεπτομερώς ο κώδικας που κατασκευάστηκε για τη λήψη των μετρήσεων από τους αισθητήρες. Τέλος παρατίθεται ολόκληρη η διαδικασία δημιουργίας εφαρμογής σε περιβάλλον Android ώστε οι μετρήσεις των αισθητήρων να παρουσιάζονται ανά πάσα στιγμή σε κάποια συμβατή με το λειτουργικό αυτό συσκευή.

ABSTRACT

In order to observe the weather phenomena, a variety of meteorological systems have been constructed, ranging in a wide area of values and features. These solutions can be from a simple depiction of data to complex systems. Arduino is a low-cost computing system with a wide range of peripheral devices that is easily programmable and supported by a wide community of members.

The present thesis concerns the process of implementing an autonomous portable meteorological station using the Arduino computing platform as well as its programming in order to obtain meteorological data such as temperature, humidity, heat index, atmospheric pressure and light intensity. Also, an application was created in a working Android environment so that measurement data is sent via Bluetooth to a mobile device that uses this operating system (mobile phone, tablet, etc.).

In the beginning, an analysis of meteorological phenomena is presented and their significance for weather forecasting is emphasized. Subsequently, the methodology of their recording according to the algorithms established by the World Meteorological Organization is analyzed. In the next chapter, a reference is made to the sensors used to obtain the meteorological data, to the criteria for selecting them and how to connect them to Arduino. A short reference is then made to the operating system Android in order to understand the aspects of its structure and operation. The Arduino programming process is then presented, while the code constructed to capture sensor readings is analyzed in detail. Finally, the entire Android implementation process is cited so that sensor measurements can be displayed at any time on any device compatible with this operating system

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Πλατφόρμα Arduino, Προγραμματισμός σε περιβάλλον Android, Μετεωρολογία

ΛΕΞΕΙΣ-ΚΛΕΙΔΙΑ: Arduino, Android, Καιρός, Μετεωρολογία

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1	11
1.1 ΕΙΣΑΓΩΓΙΚΑ ΓΙΑ ΤΗΝ ΜΕΤΕΩΡΟΛΟΓΙΑ ΚΑΙ ΤΟΝ ΚΑΙΡΟ	11
1.1.2 Θερμοκρασία ατμόσφαιρας	12
1.1.3 Ατμοσφαιρική Πίεση	13
1.1.4 Υγρασία	14
1.1.5 Σημείο δρόσου ή Σημείο υγροποίησης ή σημείο κόρου ατμόσφαιρας	15
1.1.6 Δείκτης Δυσφορίας ή Δείκτης Θερμότητας.....	15
1.1.7 Άνεμος	16
1.1.8 Αύρες	17
1.1.9 Βροχή και χιόνι	18
1.1.10 Καταιγίδες και χαλάζι.....	21
1.1.11 Είδη νεφών	22
1.2.Το κλίμα της Ελλάδας	23
1.3 Ο καιρός στην Ελλάδα	26
1.3.1 Το καλοκαίρι	26
1.3.2 Ο χειμώνας	26
1.4 Μετεωρολογικός Σταθμός	27
1.4.1 Ορισμός.....	27
1.5 Αυτόνομος Μετεωρολογικός Σταθμός	28
ΚΕΦΑΛΑΙΟ 2	30
2.1 Η ΠΛΑΤΦΟΡΜΑ ARDUINO	30
2.1.1 Εισαγωγικά για το Arduino	30
2.1.2 Μοντέλα μικροελεγκτών Arduino	31
2.1.3 Το Υλικό (Arduino Uno).....	32
2.1.4 Τροφοδοσία	34
2.1.5 Μνήμη	35
2.1.6 Είσοδος/Εξοδος.....	35
2.1.7 Επικοινωνία	36
2.1.8 Προγραμματισμός.....	37
2.1.9 Φυσικά Χαρακτηριστικά	37
2.1.10 Το λογισμικό λειτουργίας του Arduino	37
2.2 Αισθητήρες	39
2.2.1 Περιγραφή των αισθητήρων καταγραφής	39

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

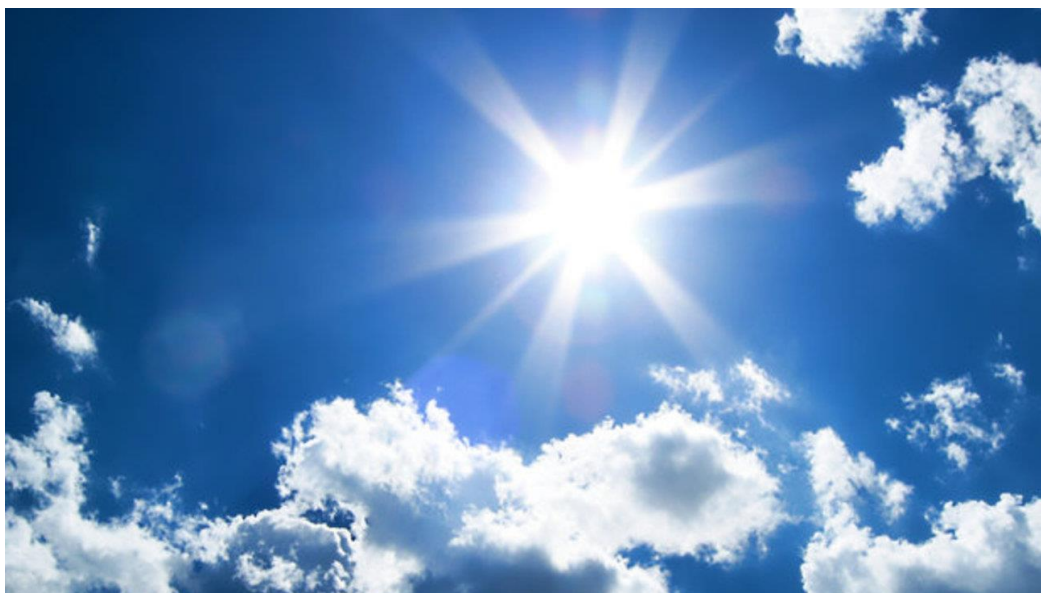
2.2.2 Αισθητήρας Θερμοκρασίας/Υγρασίας DHT-11	40
2.2.3 Αισθητήρας Βαρομετρικής Πίεσης BMP180	42
2.2.4 Αισθητήρας φωτός (Photo Resistor).....	45
ΚΕΦΑΛΑΙΟ 3	48
3.1 Λειτουργικό Σύστημα Android	48
3.1.1 Εισαγωγικά	48
3.1.2 Το υλικό.....	51
3.1.3 Η Αρχιτεκτονική.....	53
ΚΕΦΑΛΑΙΟ 4	56
4.1 Μετεωρολογικός Σταθμός με Arduino	56
4.1.1 Σκοπός	56
4.1.2 Περιγραφή.....	57
4.1.3 Σχηματικό.....	58
4.1.4 Προγραμματισμός.....	60
ΚΕΦΑΛΑΙΟ 5	68
5.1 Εφαρμογή Android για τον μετεωρολογικό σταθμό	68
5.1.1 Κύρια στοιχεία μιας εφαρμογής Android	68
5.1.2 Η διάταξη.....	69
5.1.3 Η ροή του προγράμματος.....	71
5.1.4 Άνοιγμα μιας σύνδεσης.....	71
5.1.5 Επεξεργασία του AndroidManifest.xml.....	72
5.1.6 Σύνδεση με τη συσκευή	72
5.1.7 Ανάγνωση εισερχόμενων δεδομένων	74
5.1.8 Μετάδοση δεδομένων	75
5.1.9 Δοκιμή της εφαρμογής	75
.....	77
ΚΕΦΑΛΑΙΟ 6	79
6.1 Κινητός Μετεωρολογικός Σταθμός	79
ΚΕΦΑΛΑΙΟ 7	81
7.1 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ	81
7.1.1 Σύνοψη της πτυχιακής εργασίας	81
7.1.2 Προοπτικές.....	82
ΚΕΦΑΛΑΙΟ 8	83
ΒΙΒΛΙΟΓΡΑΦΙΑ	83

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

ΚΕΦΑΛΑΙΟ 1

1.1 ΕΙΣΑΓΩΓΙΚΑ ΓΙΑ ΤΗΝ ΜΕΤΕΩΡΟΛΟΓΙΑ ΚΑΙ ΤΟΝ ΚΑΙΡΟ

1.1.1 Ορισμός του καιρού



Εικόνα 1.1: Άποψη ηλιοφάνειας

Καιρός ονομάζεται το σύνολο των μετεωρολογικών φαινομένων που παρατηρούνται στην ατμόσφαιρα της Γης σε συγκεκριμένο τόπο και χρόνο. Τέτοια φαινόμενα είναι η θερμοκρασία, η υγρασία, η ατμοσφαιρική πίεση, οι κινήσεις των ανέμων, η παρουσία νεφών κλπ. Ο καιρός προσδιορίζεται για μια συγκεκριμένη χρονική στιγμή και η πρόγνωσή του βασίζεται στα αποτελέσματα των παρατηρήσεών του. Στην πραγματικότητα είναι μια συνάρτηση στην οποία εισάγονται μετεωρολογικά δεδομένα, ο τόπος, ο χρόνος και άλλοι παράμετροι και με βάση τις καταγραφές και παρατηρήσεις που έχουν γίνει, εξάγεται ένα αποτέλεσμα. Το αποτέλεσμα αυτό αποτελεί την πρόγνωση του καιρού και η ακρίβειά του εξαρτάται από το μετεωρολογικό μοντέλο που θα χρησιμοποιηθεί.[1]

Οι "μηχανές" που παράγουν τις διάφορες ατμοσφαιρικές καταστάσεις ανά τον κόσμο είναι πρωταρχικά:

1. ο **Ήλιος**, ως πηγή ενέργειας που θερμαίνει την ατμόσφαιρα, αλλά και τις υγρές και στερεές μάζες στην επιφάνεια της Γης
2. η **Γη**, με τις κινήσεις της, γύρω από τον άξονά της και γύρω από τον Ήλιο και
3. η **Ατμόσφαιρα** που ως αέρινο κάλυμμα περιβάλλει τη Γη.

Οι τρεις αυτές οντότητες συγκροτούν ένα "θερμικό σύστημα". Όλα τα συναφή φαινόμενα, λεγόμενα και "καιρικά φαινόμενα" (άνεμος, βροχή, χιονόπτωση, καταιγίδα, τυφώνας, σίφωνας κ.λπ.) συμβαίνουν στην ατμόσφαιρα, και πιο συγκεκριμένα στο χαμηλότερο τμήμα της, την Τροπόσφαιρα. Τα φαινόμενα αυτά επιδρούν στην επιφάνεια της Γης και κατά συνέπεια στις ανθρώπινες δραστηριότητες όπως στη γεωργία, την κτηνοτροφία, την αλιεία, αλλά και σε πλείστες άλλες παραγωγικές δραστηριότητες, τη ναυσιπλοΐα, τα ταξίδια, τη κατασκευή έργων κ.α. εξ ου και οι χαρακτηρισμοί "καλοκαιρία" και "κακοκαιρία". Για τον λόγο αυτό, η μελέτη και η πρόγνυσή τους έχει τεράστιο οικονομικό και κοινωνικό ενδιαφέρον, γεγονός που και οδήγησε στην ανάπτυξη του κλάδου της Μετεωρολογίας.

Για να γίνει περισσότερο αντιληπτός ο μηχανισμός της δημιουργίας του καιρού, δηλαδή του θερμικού συστήματος Ήλιος - Γη - Ατμόσφαιρα, θα πρέπει να έχουν γίνει προηγουμένως κατανοητές οι φυσικές ιδιότητες και τα χαρακτηριστικά γνωρίσματα των παραπάνω συντελεστών. Όπως επίσης και ο ρόλος της Γης και της ατμόσφαιράς της, τόσο στη δημιουργία, όσο και στην εξέλιξη των καιρικών φαινομένων, των καλουμένων και μετεωρολογικών φαινομένων.[2]

Με την εξέλιξη των υπολογιστικών συστημάτων τα οποία πλέον εκτελούν τρισεκατομμύρια πράξεις ανά δευτερόλεπτο και τους δορυφόρους από τους οποίους μπορούμε να παρατηρούμε τα μετεωρολογικά φαινόμενα με μεγαλύτερη ακρίβεια και για μεγαλύτερο γεωγραφικό εύρος, τα μοντέλα αυτά έχουν εξελιχθεί και εξάγουν ασφαλέστερα και πιο μακροπρόθεσμα αποτελέσματα.[1]

1.1.2 Θερμοκρασία ατμόσφαιρας



Εικόνα 1.2: Θερμόμετρο

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

αυτής εκτάσεων. Συνεπώς η αναφορά της θερμοκρασίας σχετίζεται πάντα με κάποια περιοχή, είτε μικρή, είτε μεγάλη, στην περίπτωση μας στην περιοχή εμβέλειας του σταθμού. Η αναφορά σε παγκόσμια κλίμακα, ανάγεται σε αντικείμενο άλλης επιστήμης. Η θερμοκρασία της ατμόσφαιρας μετριέται με τα θερμόμετρα και υπάρχουν διάφορες κλίμακες μέτρησης, με συνηθισμένες κλίμακες τις Κελσίου (Celsius, σύμβολο C°), Κέλβιν (Kelvin, σύμβολο K°) και Φαρενάιτ (Fahrenheit, σύμβολο F°). Στην Ελλάδα χρησιμοποιείται η κλίμακα Κελσίου και ορίζεται ως « Το σημείο βρασμού του νερού είναι στους 100 C° και το σημείο παγιοποίησης του, στους 0 C°».[3]

1.1.3 Ατμοσφαιρική Πίεση



Εικόνα 1.3: Αναλογικό βαρόμετρο

Είναι γνωστό ότι η βαρύτητα είναι η αιτία που παραμένει το νερό και τα αέρια στην επιφάνεια της γης και δεν διαφεύγουν στο διάστημα. Η γη περιβάλλεται από ένα λεπτό στρώμα αερίων το οποίο ονομάζουμε ατμόσφαιρα. Η ατμόσφαιρα έχει βάρος το οποίο πιέζει όλα τα σώματα που βρίσκονται μέσα σε αυτή. Η πίεση αυτή ονομάζεται ατμοσφαιρική πίεση. Η ατμοσφαιρική πίεση όπως γνωρίζουμε από τη φυσική τη μετράμε με τα βαρόμετρα. Στην μετεωρολογία χρησιμοποιούμε και τους βαρογράφους, οι οποίοι καταγράφουν συνεχώς την πίεση σε κάποιο αποθηκευτικό μέσο. Η πίεση μετριέται σε χιλιοστόμετρα στήλης υδραργύρου (στην θάλασσα επικρατεί πίεση περίπου 760 χιλ. Hg). Στην μετεωρολογία χρησιμοποιούμε επίσης και τη μονάδα μιλιμπάρ (mb) και ισχύει 1000 mb ίσα προς 750 χιλ. Hg. Η τυπική πίεση που επικρατεί στη στάθμη της θάλασσας εκφρασμένη σε mb είναι 1013.25. Γνωρίζουμε βέβαια ότι στη φυσική μετράμε την πίεση σε Pascal (Pa), προς τιμή του Blaise Pascal (1623-1662). Ένα pascal ισούται με 0.01 μιλιμπάρ ή 0.00001 μπάρ. Στη μετεωρολογία η επικράτηση του μιλιμπάρ ως μονάδα ατμοσφαιρικής πίεσης έγινε το 1929 και λόγω συνήθειας διατηρήθηκε και μετά

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

το 1960 όταν αποφασίστηκε η αλλαγή των μονάδων και η εφαρμογή του συστήματος S.I. Άλλοι βέβαια μετεωρολόγοι υιοθέτησαν τη μονάδα hectopascal (hPa), (το «hecto» (h), αντιστοιχεί στο 100) και έτσι 1 hectopascal (hPa) ισούται με 100 Pa που ισούνται με 1 mb. 100,000 Pa ισούνται με 1000 hPa που ισούνται με 1000 mb. Εν κατακλείδι με το τρικ αυτό μπορεί οι δυο επικρατούσες μονάδες να έχουν διαφορετικό όνομα αλλά η τιμή που μετράνε είναι η ίδια. Για παράδειγμα η τυπική ατμοσφαιρική πίεση στην επιφάνεια της θάλασσας είναι 1013.25 mb ή 1013.25 Hp.[4]

1.1.4 Υγρασία



Εικόνα 1.4: Υγρασία πάνω σε τζάμι

Ο ατμοσφαιρικός αέρας περιέχει υδρατμούς με διαφορετική ποσότητα από τόπο σε τόπο και από ώρα σε ώρα. Ο αέρας όμως δεν είναι δυνατόν να περιέχει απεριόριστη ποσότητα υδρατμών, αλλά για κάθε θερμοκρασία υπάρχει μια μέγιστη δυνατή περιεκτικότητα υδρατμών. Όταν ο αέρας περιέχει τη μέγιστη τέτοια ποσότητα ονομάζεται κορεσμένος. Όσο ψυχρότερος είναι ο αέρας τόσο μικρότερη ποσότητα υδρατμών μπορεί να συγκρατήσει. Αν λοιπόν μια μάζα υγρού και θερμού αέρα ψυχθεί θα φτάσει σε μια θερμοκρασία όπου δεν είναι δυνατόν πλέον να συγκρατήσει άλλους τους υδρατμούς από τους οποίους περιέχει. Οι υδρατμοί που περισσεύουν θα συμπυκνωθούν ως σταγονίδια πάνω στα αιωρούμενα μικροσωματίδια και θα δημιουργήσουν το νέφος. Αν δε, συμπυκνωθούν πάνω σε ψύχρα αντικείμενα θα δημιουργήσουν τη δρόσο. Η θερμοκρασία στην οποία ο ακόρεστος αέρας καθώς ψύχεται φτάνει στο κορεσμό, ονομάζεται σημείο δρόσου. Η υγρασία παίζει σημαντικό ρόλο

στην μορφή του κλίματος, τη βλάστηση και τη ζωή ενός τόπου. Μετράμε την υγρασία με τα υγρόμετρα, τα οποία δείχνουν πόσους υδρατμούς περιέχει η ατμόσφαιρα επί της εκατό (%) (όπου 100 θεωρούνται οι υδρατμοί οι οποίοι θα περιέχονταν για την ίδια θερμοκρασία αν είχαμε κορεσμό). Για ορισμένη θερμοκρασία, η υγρασία παίζει σπουδαίο ρόλο στο αίσθημα της άνεσης ή της δυσφορίας το οποίο αισθανόμαστε. Το καλοκαίρι αισθανόμαστε άνετα αν κλιματίζουμε ένα χώρο με θερμοκρασία 27°C και υγρασία 55%. Μεγαλύτερη υγρασία προκαλεί το αίσθημα της δυσφορίας.[4]

1.1.5 Σημείο δρόσου ή Σημείο υγροποίησης

Σημείο δρόσου χαρακτηρίζεται το σημείο εκείνο της θερμοκρασίας που όταν οι υδρατμοί ψυχθούν δημιουργούν το φαινόμενο της δρόσου, δηλαδή τις σταγόνες δρόσου. Στη θερμοκρασία αυτή εξυπακούεται πως όταν ο αέρας είναι κορεσμένος και δεν μπορεί να συγκρατήσει άλλους υδρατμούς η σχετική υγρασία να είναι 100%. Σημειώνεται όμως ότι η θερμοκρασία κορεσμού της ατμόσφαιρας ή του "σημείου δρόσου" μπορεί να είναι οποιαδήποτε θερμοκρασία, πάνω από τους 0°C. Η θερμοκρασία αυτή εξαρτάται μόνο από την ποσότητα των υδρατμών που περιέχει 1 κυβικό μέτρο αέρος, συνεπώς εξαρτάται από την απόλυτη υγρασία. Η θερμοκρασία του σημείου δρόσου αποτελεί σπουδαίο μετεωρολογικό στοιχείο για ένα τόπο και γι' αυτό πάντοτε αναφέρεται στους μετεωρολογικούς χάρτες με τα σύμβολα D.P. από τα αρχικά του αγγλικού όρου Dew Point (Δρόσου Σημείο) [5]

1.1.6 Δείκτης Δυσφορίας ή Δείκτης Θερμότητας

Ο δείκτης δυσφορίας δεν αφορά ένα αμιγές μετεωρολογικό φαινόμενο, αλλά παρουσιάζει «πόσο ζέστη αισθανόμαστε πραγματικά» με το συνδυασμό της τρέχουσας θερμοκρασίας και της σχετικής υγρασίας. Εκφράζει δηλαδή το βαθμό δυσφορίας του ανθρώπου λόγω των επιδράσεων των θερμοϋγρομετρικών συνθηκών. Σε συνθήκες αυξημένης θερμοκρασίας το ανθρώπινο σώμα ρυθμίζει τη θερμοκρασία του μέσω της εφίδρωσης του σώματος. Αποβάλλει θερμότητα με την εξάτμιση του ιδρώτα. Σε συνθήκες αυξημένης σχετικής υγρασίας, η εξάτμιση του ιδρώτα περιορίζεται, με αποτέλεσμα να αισθανόμαστε τη θερμοκρασία υψηλότερη από ό,τι πραγματικά είναι. [6]

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

	ΣΧΕΤΙΚΗ ΥΓΡΑΣΙΑ (%)												
Θερμ.	40	45	50	55	60	65	70	75	80	85	90	95	100
47	58												
43	54	58											
41	51	54	58										
40	48	51	55	58									
39	46	48	51	54	58								
38	43	46	48	51	54	58							
37	41	43	45	47	51	53	57						
36	38	40	42	44	47	49	52	56					
34	36	38	39	41	43	46	48	51	54	57			
33	34	36	37	38	41	42	44	47	49	52	55		
32	33	34	35	36	38	39	41	43	45	47	50	53	56
31	31	32	33	34	35	37	38	39	41	43	45	47	49
30	29	31	31	32	33	34	35	36	38	39	41	42	44
29	28	29	29	30	31	32	32	33	34	36	37	38	39
28	27	28	28	29	29	29	30	31	32	32	33	34	35
27	27	27	27	27	28	28	28	29	29	29	30	30	31
Κατηγορία	Δείκτης Δυσφορίας		Πιθανές θερμικές επιπτώσεις στην υγεία ανθρώπων που ανήκουν σε ομάδες υψηλού κινδύνου										
Εξαιρετικός Κίνδυνος	(54°C ή ψηλότερα)		Μεγάλη πιθανότητα θερμοπληξίας ή ηλίας.										
Κίνδυνος	(41 - 54°C)		Πιθανή ηλίαση, μυϊκές κράμπες και/ή θερμική εξάντληση. Πιθανή και θερμοπληξία με παρατεταμένη έκθεση ή/και φυσική δραστηριότητα.										
Εξαιρετική Προσοχή	(32 - 41°C)		Πιθανή ηλίαση, μυϊκές κράμπες και/ή θερμική εξάντληση με παρατεταμένη έκθεση ή/και φυσική δραστηριότητα.										
Προσοχή	(27 - 32°C)		Πιθανό αίσθημα κόπωσης με παρατεταμένη έκθεση ή/και φυσική δραστηριότητα.										

Εικόνα 1.5: Κλίμακα τιμών δείκτη δυσφορίας

1.1.7 Άνεμος



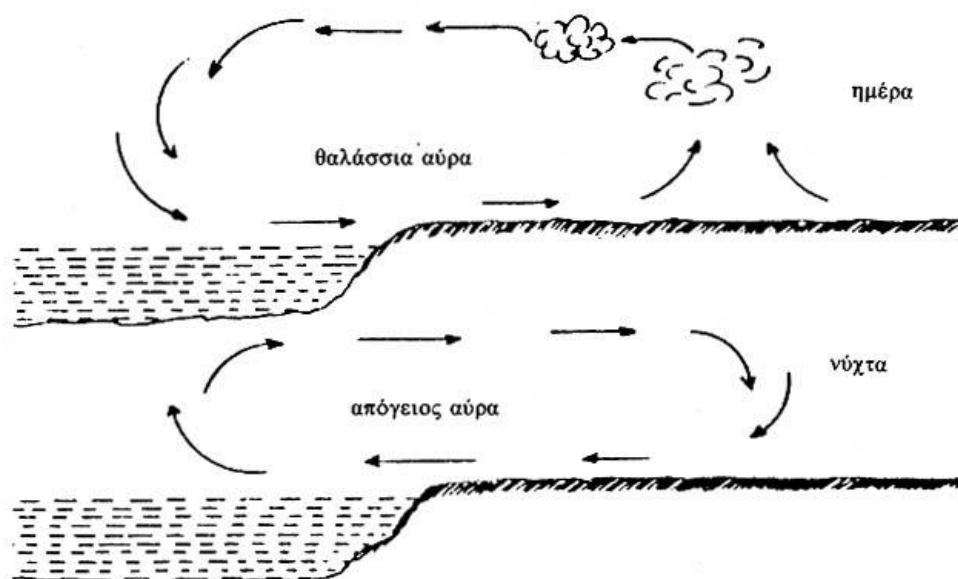
Εικόνα 1.6: Άνεμος σε φθινοπωρινό τοπίο

Ο άνεμος είναι η οριζόντια μετακίνηση του αέρα και προκαλείται από τη διαφορά της ατμοσφαιρικής πίεσης από τόπο σε τόπο. Ο άνεμος έχει μεγάλη σημασία για τον άνθρωπο και έχει μελετηθεί από την αρχαιότητα κυρίως όσο αφορά τη ναυσιπλοΐα. Δύο είναι τα κύρια χαρακτηριστικά του ανέμου, η διεύθυνση και η ένταση. Ως προς τη διεύθυνση, ο άνεμος ονοματίζεται όχι από το σημείο ορίζοντα στο οποίο κατευθύνεται, αλλά από αυτό εκ του οποίου προέρχεται. Η ένταση του ανέμου ορίζεται από την ταχύτητα του σε χιλιόμετρα ανά ώρα ή κόμβους (ναυτικά μίλια ανά ώρα). Στην πράξη επειδή οι ναυτικοί δεν διέθεταν ανεμόμετρα επικράτησε μια εμπειρική κλίμακα την οποία επινόησε ο Άγγλος Ναύαρχος Μποφόρ, η οποία βασίζεται σε οπτικές παρατηρήσεις.[4]

1.1.8 Αύρες

Ένας ειδικός τύπος ανέμου καθαρά τοπικός είναι οι αύρες. Οι αύρες οφείλονται σε τοπικές διαφορές θερμοκρασίας μεταξύ θάλασσας και ξηράς ή βουνών και κοιλάδων. Οι αύρες στην Ελλάδα είναι πολύ χαρακτηριστικό φαινόμενο και πολλές φορές εξαιτίας τους δημιουργούνται τοπικοί άνεμοι τελείως αντίθετοι από τους επικρατούντες στην υπόλοιπη χώρα. Γνωρίζουμε ήδη ότι η θάλασσα θερμαίνεται πιο αργά από τη ξηρά, και έτσι την ημέρα η ξηρά είναι θερμότερη από τη θάλασσα. Ο θερμός αραιός αέρας πάνω από τη ξηρά ανέρχεται, ο δε ψυχρός θαλάσσιος αέρας τείνει να καταλάβει τη θέση του. Έτσι δημιουργείται η θαλάσσια αύρα, η οποία κατευθύνεται από το άνοιγμα των κόλπων προς τη ξηρά και για αυτό ονομάζεται από τους ναυτικούς και μπουκαδούρα.

Το αντίθετο συμβαίνει τη νύχτα, οπότε η θάλασσα ψύχεται πιο αργά από την ξηρά και δημιουργείται η απόγειος αύρα η οποία έχει κατεύθυνση από τη ξηρά προς τη θάλασσα. Παρόμοια φαινόμενα δημιουργούνται μεταξύ ορέων και κοιλάδων. Σε αυτή την περίπτωση η κοιλάδα η οποία καλύπτεται από υδρατμούς και σκόνη θερμαίνεται την ημέρα ή ψύχεται τη νύχτα πιο αργά από ότι τα όρη. Η διαφορά της θέρμανσης αυτής δημιουργεί τις αύρες. [4]



Εικόνα 1.7: Τα δύο είδη αύρας

1.1.9 Βροχή και χιόνι



Εικόνα 1.8: Η βροχή

Όσο η υγραποίηση των υδρατμών αναπτύσσεται μέσα σε ένα νέφος τόσο δημιουργούνται μεγαλύτερα υδροσταγονίδια. Επίσης, στα μεγαλύτερα ύψη, όπου η θερμοκρασία είναι πολύ χαμηλότερη του μηδενός, τα υδροσταγονίδια γίνονται παγοκρυσταλλίδια τα οποία συνεχώς μεγαλώνουν.

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

Όταν δημιουργηθούν σταγόνες τόσο βαριές ώστε να έχουν ταχύτητα πτώσης μεγαλύτερη από την ανοδική ταχύτητα του ρεύματος αυτά πέφτουν προς τη γη ως βροχή. Οι σταγόνες που πέφτουν προς το έδαφος περνούν μέσα από θερμότερα στρώματα αέρα και μέρος τους εξατμίζεται και πάλι. Με αυτό τον τρόπο μια βροχή στα ανώτερα στρώματα είναι πιθανό να μη φτάσει ποτέ στο έδαφος. Στην περίπτωση που οι υδρατμοί συμπυκνώνονται σε θερμοκρασία λίγο κάτω από το μηδέν, σχηματίζονται παγοκρύσταλλοι με μορφή χιονιού. Στα μέσα γεωγραφικά πλάτη που βρίσκεται η Ελλάδα, το χιόνι είναι συχνό στα μεγάλα ύψη ενώ στη στάθμη της θάλασσας παρουσιάζεται σπάνια. Αυτό συμβαίνει γιατί κατά τη αργή πτώση του μέσω θερμότερου αέρα, το χιόνι τήκεται και φτάνει στα χαμηλά ύψη υπό τη μορφή μικρών σταγόνων (χιονόνερο).[4]



Εικόνα 1.9: Χιονισμένο τοπίο

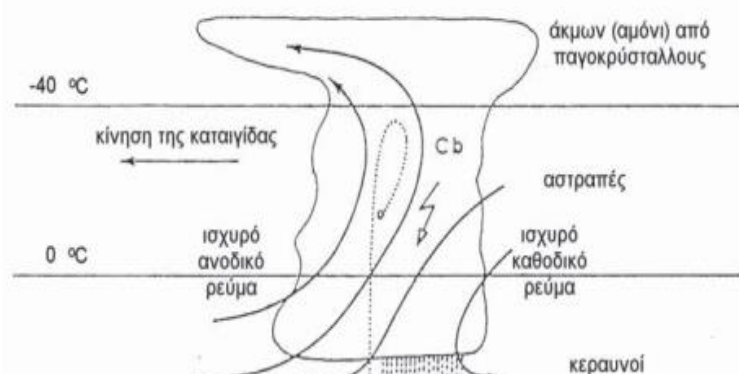
Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

<p>ΨΥΧΡΟ ΣΤΡΩΜΑ</p> <p>ΡΗΧΟ ΘΕΡΜΟ ΣΤΡΩΜΑ</p> <p>ΨΥΧΡΟ ΣΤΡΩΜΑ</p> <p>Επιφάνεια</p> <p>-3,9 °C 0 °C</p> <p>Θερμοκρασία ατμόσφαιρας</p> <p>↑ Ύψος</p>	<p>Στη φωτογραφία (αριστερά) η πράσινη γραμμή παριστάνει τη θερμοκρασία της ατμόσφαιρας σε συνάρτηση με το ύψος. Στην επιφάνεια λοιπόν η θερμοκρασία είναι $-3,9^{\circ}\text{C}$ και αυξάνει με το ύψος πριν αρχίσει να μειώνεται. Ωστόσο, αφού η θερμοκρασία σε κάθε περίπτωση παραμένει κάτω από το 0°C ο υετός που πέφτει θα παραμείνει με τη μορφή χιονιού.</p>
<p>ΒΑΘΥ ΨΥΧΡΟ ΣΤΡΩΜΑ</p> <p>Επιφάνεια</p> <p>-2,8 °C 0 °C</p> <p>Θερμοκρασία ατμόσφαιρας</p> <p>↑ Ύψος</p>	<p>Στην εικόνα αυτή, η επιφανειακή θερμοκρασία είναι υψηλότερη $-2,8^{\circ}\text{C}$. Επίσης, καθώς αυξάνει το ύψος, η θερμοκρασία αυξάνει σε σημείο όπου κάποιο ατμοσφαιρικό στρώμα έχει θερμοκρασία πάνω από το 0°C πριν ξαναπέσει και πάλι κάτω από το σημείο πήξης. Καθώς λοιπόν πέφτει το χιόνι μέσα σε αυτό το στρώμα όπου η θερμοκρασία είναι πάνω από το μηδέν, οι χιονονιφάδες λιώνουν μερικώς. Καθώς όμως ξαναμπαίνουν στο ψυχρό στρώμα που βρίσκεται χαμηλότερα, ξαναπαγώνουν και παίρνουν τη μορφή σβώλων πάγου που αναπηδούν στο έδαφος, σαν ψιλό χαλάζι.</p>
<p>ΨΥΧΡΟ ΣΤΡΩΜΑ</p> <p>ΒΑΘΥ ΘΕΡΜΟ ΣΤΡΩΜΑ</p> <p>ΨΥΧΡΟ ΣΤΡΩΜΑ</p> <p>Επιφάνεια</p> <p>-3,9 °C 0 °C</p> <p>Θερμοκρασία ατμόσφαιρας</p> <p>↑ Ύψος</p>	<p>Χιονόνερο θα έχουμε όταν το θερμό στρώμα στην ατμόσφαιρα είναι αρκετά βαθύ με μόνο ένα ρηχό ψυχρό στρώμα με θερμοκρασία κάτω του μηδενός κοντά στην επιφάνεια. Ο υετός μπορεί να ξεκινήσει σαν βροχή ή/και σαν χιόνι αλλά γίνεται βροχή στο θερμό στρώμα. Η βροχή μπορεί μεν να ξαναμπαίνει σε ψυχρό στρώμα με θερμοκρασία κάτω του μηδενός αλλά εφόσον το στρώμα αυτό είναι ρηχό, δεν προλαβαίνει να ξαναπαγώσει.</p>

Εικόνα 1.10: Πώς δημιουργούνται η βροχή, το χιόνι και το χιονόνερο

1.1.10 Καταιγίδες και χαλάζι

Ενδιαφέρον παρουσιάζει η ζωή της υδροσταγόνας μέσα σε ένα σωρειτομελανία. Επειδή τα ανοδικά ρεύματα είναι πολύ μεγάλα οι σταγόνες δεν πέφτουν αν δεν αποκτήσουν μεγάλο μέγεθος. Τότε προκαλούν ραγδαία βροχή και δημιουργούν ισχυρότατα καθοδικά ρεύματα. Εξάλλου οι σταγόνες παρασυρόμενες σε μεγαλύτερα ύψη δημιουργούν παγοσφαιρίδια, το χαλάζι, το οποίο είναι δυνατόν να πέσει και να ανέρθει επανειλημμένα εντός του ανοδικού ρεύματος αποκτώντας συνεχώς όλο και μεγαλύτερο μέγεθος μέχρι να πέσει στη γη.



Εικόνα 1.11: Διαδικασία δημιουργίας καταιγίδας

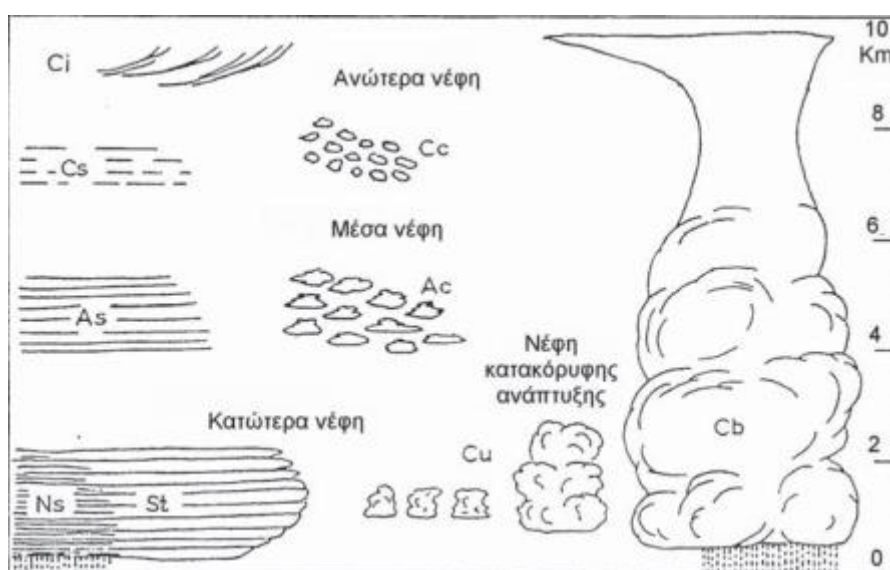
Τα ισχυρά ανοδικά ρεύματα μαζί με τα σταγονίδια μεταφέρουν στο σωρειτομελανία και ηλεκτρικά φορτία. Για την αποφόρτιση αυτών των φορτίων ξεσπούν ηλεκτρικές εκκενώσεις μεταξύ των τμημάτων του νέφους (αστραπές) ή μεταξύ νέφους και γης (κεραυνοί). Η διάμετρος μιας καταιγίδας είναι της τάξης των λίγων χιλιομέτρων και για αυτό μια καταιγίδα μπορεί να είναι ένα τελείως τοπικό φαινόμενο σε μια ζώνη μικρού πλάτους προς τη διεύθυνση κατά την οποία τη μεταφέρει ο άνεμος. Η καταιγίδα διαλύεται μετά από λίγο από τα καθοδικά ρεύματα τα οποία η ίδια δημιουργεί. Οι σωρειτομελανίες είναι δυνατόν να εμφανιστούν σε ομάδες, όπως επίσης και να είναι ενσωματωμένοι μέσα σε στρωματόμορφα νέφη ώστε να μην γίνονται αντιληπτοί παρά μόνο από τις αστραπές που παράγονται. Οι ισχυρές αναταράξεις και τα ρεύματα του σωρειτομελανία είναι επικίνδυνα για τα αεροπλάνα τα οποία αποφεύγουν τις καταιγίδες είτε επειδή γνωρίζουν τη θέση τους από μετεωρολογικές παρατηρήσεις, είτε γιατί τις ανιχνεύουν με το ραντάρ καιρού, το οποίο και διαθέτουν σήμερα τα περισσότερα αεροσκάφη.[4]



Εικόνα 1.12: Χαλάζι

1.1.11 Είδη νεφών

Τα νέφη χωρίζονται ανάλογα με το ύψος στο οποίο βρίσκονται σε χαμηλά, μέσα και ανώτερα. Επίσης ανάλογα με το σχήμα τους είναι σωρειτόμορφα ή στρωματόμορφα. Στον πίνακα σημειώνονται τα είδη των νεφών μαζί με τις διεθνείς ονομασίες τους και τα σύμβολά τους. Στην παρακάτω εικόνα (Εικόνα 1.13) εμφανίζονται αντιπροσωπευτικές μορφές νεφών.



Εικόνα 1.13: Είδη νεφών

Τα νέφη πέρα από τις βασικές τους ονομασίες φέρουν συχνά παρωνύμια ανάλογα του σχήματος ή της προέλευσης τους, όπως π.χ. φακοειδείς, υψισωρείτες ή σωρείτες καλοκαιρίας. Επίσης τα νέφη, κυρίως δε τα στρωματόμορφα, δεν παρουσιάζονται μεμονωμένα, αλλά με μορφή νεφικών συστημάτων.[4]



Εικόνα 1.14: Νέφη από ψηλά

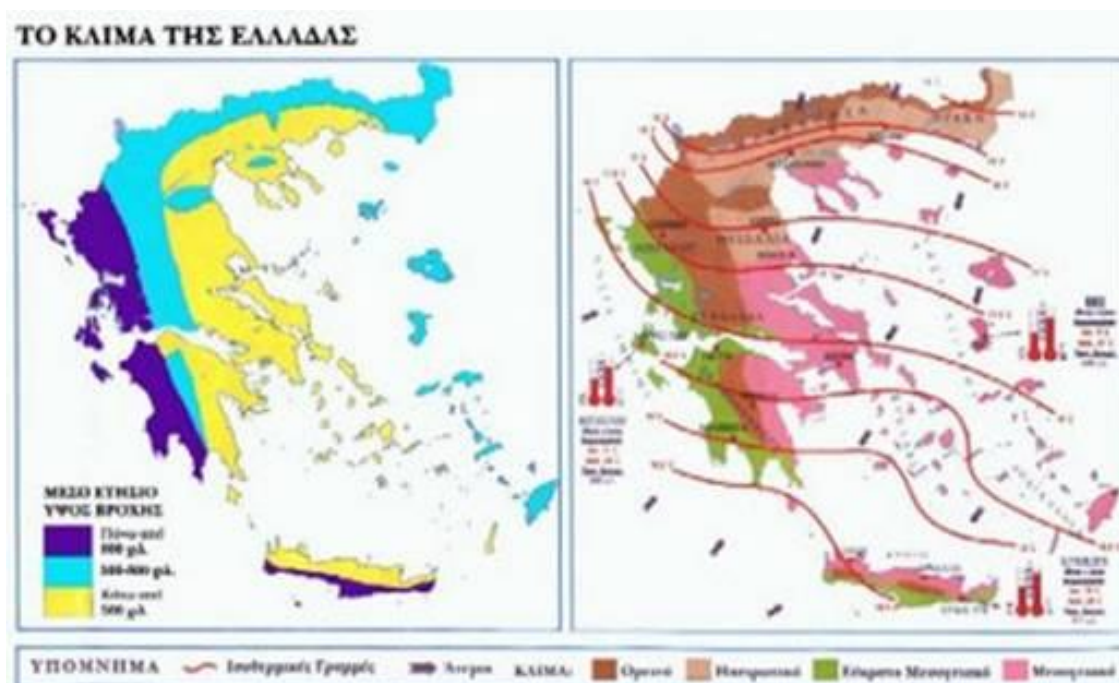
1.2.Το κλίμα της Ελλάδας

Η Ελλάδα χαρακτηρίζεται από το μεσογειακό τύπο του εύκρατου κλίματος και έχει ήπιους υγρούς χειμώνες και ζεστά ξηρά καλοκαίρια. Το κλίμα της χώρας μπορεί να διαιρεθεί σε τέσσερις βασικές κατηγορίες:

- 1. Υγρό μεσογειακό (δυτική Ελλάδα, δυτική Πελοπόννησος, πεδινά και ημιορεινά της Ηπείρου)**
- 2. Ξηρό μεσογειακό (Κυκλάδες, παραλιακή Κρήτη, Δωδεκάνησα, ανατολική Πελοπόννησος, Αττική, πεδινές περιοχές Ανατολικής Στερεάς)**
- 3. Ηπειρωτικό (δυτική Μακεδονία, εσωτερικά υψίπεδα ηπειρωτικής Ελλάδας, βόρειος Έβρος)**
- 4. Ορεινό (ορεινές περιοχές με υψόμετρο περίπου >1500 m στη Βόρεια Ελλάδα, >1800m στην Κεντρική Ελλάδα και >2000m στην Κρήτη).**

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

Οι θερμοκρασίες είναι σπάνια υπερβολικές στις παραθαλάσσιες περιοχές. Στις κλειστές εσωτερικές πεδιάδες και στα υψίπεδα της χώρας παρατηρούνται τα μεγαλύτερα θερμοκρασιακά εύρη -τόσο ετήσια όσο και ημερήσια. Οι χιονοπτώσεις είναι κοινές στα ορεινά από τα τέλη Σεπτεμβρίου (στη βόρεια Ελλάδα, τέλη Οκτωβρίου κατά μέσο όρο στην υπόλοιπη χώρα), ενώ στις πεδινές περιοχές χιονίζει κυρίως από το Δεκέμβριο μέχρι τα μέσα Μαρτίου. Έχει χιονίσει, πάντως, ακόμα και κατά μήνα Μάιο στη Φλώρινα. Στις παραθαλάσσιες περιοχές των νησιωτικών περιοχών οι χιονοπτώσεις συμβαίνουν σπανιότερα και δεν αποτελούν βασικό χαρακτηριστικό του κλίματος. Οι καύσωνες επηρεάζουν κυρίως τις πεδινές περιοχές και είναι συχνότεροι τον Ιούλιο και τον Αύγουστο. Σπάνια, πάντως, διαρκούν περισσότερες από 3 ημέρες.



Εικόνα 1.15: Χάρτης με το κλίμα της Ελλάδας

Η Ελλάδα βρίσκεται μεταξύ του 34ου και 42ου παραλλήλου του βορείου ημισφαιρίου και έχει μεγάλη ηλιοφάνεια όλο σχεδόν το χρόνο. Λεπτομερέστερα στις διάφορες περιοχές της Ελλάδας παρουσιάζεται μεγάλη ποικιλία κλιματικών τύπων, πάντα βέβαια μέσα στα πλαίσια του μεσογειακού κλίματος. Αυτό οφείλεται στην τοπογραφική διαμόρφωση της χώρας που έχει μεγάλες διαφορές υψομέτρου (υπάρχουν μεγάλες οροσειρές κατά μήκος της κεντρικής χώρας και άλλοι ορεινοί όγκοι) και εναλλαγή ξηράς και θάλασσας. Έτσι, από το ξηρό κλίμα της Αττικής και γενικά της ανατολικής Ελλάδας μεταπίπτουμε στο

υγρό της βόρειας και δυτικής Ελλάδας. Τέτοιες κλιματικές διαφορές συναντώνται ακόμη και σε τόπους που βρίσκονται σε μικρή απόσταση μεταξύ τους, πράγμα που παρουσιάζεται σε λίγες μόνο χώρες σε όλο τον κόσμο.

Από κλιματολογικής πλευράς το έτος μπορεί να χωριστεί κυρίως σε δύο εποχές: Την ψυχρή και βροχερή χειμερινή *περίοδο*, που διαρκεί από τα μέσα του Οκτωβρίου και μέχρι το τέλος Μαρτίου και τη θερμή και άνομβρη εποχή, που διαρκεί από τον Απρίλιο έως τον Οκτώβριο.

Κατά την πρώτη περίοδο οι ψυχρότεροι μήνες είναι ο Ιανουάριος και ο Φεβρουάριος, όπου κατά μέσον όρο η μέση ελάχιστη θερμοκρασία κυμαίνεται από 5-10 °C στις παραθαλάσσιες περιοχές, από 0-5 °C στις ηπειρωτικές περιοχές και σε χαμηλότερες τιμές κάτω από το μηδέν στις βόρειες περιοχές

Οι βροχές ακόμη και τη χειμερινή περίοδο δεν διαρκούν για πάρα πολλές ημέρες και ο ουρανός της Ελλάδας δεν μένει συνεφιασμένος καθ' όλη τη διάρκεια του χειμώνα, όπως συμβαίνει σε άλλες περιοχές της γης. Οι χειμερινές κακοκαιρίες διακόπτονται καμιά φορά κατά τον Ιανουάριο και το πρώτο δεκαπενθήμερο του Φεβρουαρίου από ηλιόλουστες ημέρες, τις γνωστές από την αρχαιότητα Αλκυονίδες ημέρες. Κατά αυτήν την περίοδο, λοιπόν, στα νησιά, κυρίως στο νότιο μέρος της χώρας, όπως για παράδειγμα στην Κρήτη, η θερμοκρασία μπορεί να ξεπεράσει τους 18-20 βαθμούς Κελσίου, στην Αττική τους 13-14 °C και στη Θεσσαλονίκη ο υδράργυρος μπορεί να ξεπεράσει τους 9 °C και πολλές φορές ακόμα και τους 10 °C. Σε άλλες πόλεις, όπως για παράδειγμα στην Αλεξανδρούπολη κατά τις Αλκυονίδες μέρες, η θερμοκρασία ξεπερνάει τους 7-8 °C, με αποτέλεσμα το χιόνι από τις χιονοπτώσεις του χειμώνα να λιώνει κατά τη διάρκεια της ημέρας.

Η χειμερινή εποχή είναι γλυκύτερη στα νησιά του Αιγαίου και του Ιονίου από ό,τι στη Βόρεια και Ανατολική ηπειρωτική Ελλάδα. Κατά τη θερμή και άνομβρη εποχή ο καιρός είναι σταθερός, ο ουρανός σχεδόν αίθριος, ο ήλιος λαμπερός και δε βρέχει εκτός από σπάνια διαστήματα με ραγδαίες βροχές ή καταιγίδες μικρής γενικά διάρκειας.

Η θερμότερη περίοδος είναι το τελευταίο δεκαήμερο του Ιουλίου και το πρώτο του Αυγούστου, οπότε η μέση μέγιστη θερμοκρασία κυμαίνεται από 30 °C μέχρι 35 °C. Κατά τη θερμή εποχή οι υψηλές θερμοκρασίες μετριάζονται από τη δροσερή θαλάσσια αύρα στις παράκτιες περιοχές της χώρας και από τους βόρειους ανέμους (*ετήσιες*) που φυσούν κυρίως στο Αιγαίο.

Η άνοιξη έχει μικρή διάρκεια, διότι ο μεν χειμώνας είναι όψιμος, το δε καλοκαίρι αρχίζει πρώιμα. Το φθινόπωρο είναι μακρύ και θερμό και πολλές φορές παρατείνεται στη νότια Ελλάδα και τα νησιά μέχρι τα μέσα του Δεκεμβρίου. Στην Αθήνα, την πρωτεύουσα, το κρύο γίνεται συνήθως αισθητό από το Νοέμβριο και μετά και από εκεί και πέρα συνεχίζεται έως και τα τέλη του Μάρτη. Μετά τα μέσα του Δεκεμβρίου, αναπτύσσεται ιδιαίτερο ψύχος στις

αστικές περιοχές, το οποίο συνεχίζεται έως και τα τέλη Φεβρουαρίου. Από τις αρχές του μήνα Μαρτίου η άνοιξη γίνεται αισθητή και η θερμοκρασία ανεβαίνει σταδιακά. Η ψυχρότερη εποχή του έτους στην πόλη της Αθήνας θεωρείται από την τελευταία εβδομάδα του Δεκεμβρίου έως και την τρίτη εβδομάδα του Ιανουαρίου. Αναφορικά, η χαμηλότερη θερμοκρασία που έχει καταγραφεί ποτέ στην Αθήνα είναι αυτή των $-17,1\text{ }^{\circ}\text{C}$ στις 28 Δεκεμβρίου του 1938 ενώ η χαμηλότερη που έχει παρατηρηθεί στην χώρα είναι $-30\text{ }^{\circ}\text{C}$. Η Ελλάδα κατέχει το ρεκόρ υψηλότερης καταγεγραμμένης θερμοκρασίας στην Ευρώπη με $48,0\text{ }^{\circ}\text{C}$ στην Αθήνα (Ελευσίνα και Τατόι) στις 10 Ιουλίου 1977. [7]

1.3 Ο καιρός στην Ελλάδα

Στην Ελλάδα παρουσιάζονται από μετεωρολογική άποψη δύο περίοδοι, η ξηρά θερινή και η ευμετάβλητη χειμερινή.

1.3.1 Το καλοκαίρι

Κατά το καλοκαίρι όπως έχουμε αναφέρει οι κλιματικές ζώνες της γης μετατοπίζονται βορειότερα ακολουθώντας την ετήσια μετατόπιση του ηλίου. Έτσι η Ελλάδα βρίσκεται κάτω από την επίδραση του καιρού της τροπικής ζώνης δηλαδή των αληγών ανέμων. Το μελέμι (ετησίας) είναι άνεμος ΒΑ κατεύθυνσης και μπορεί να θεωρηθεί μια επέκταση των αληγών ανέμων προς βορρά. Κατά τη νύχτα οπότε ο ήλιος δεν υπερθερμαίνει της ερήμους της Μέσης Ανατολής το χαμηλό γίνεται ηπιότερο και ο άνεμος σταματά για να ξεκινήσει και πάλι την επόμενη μέρα. Όταν το μελέμι είναι ασθενές ή σε περιοχές όπου δεν πνέει λόγω παρεμβαλλόμενων βουνών (π.χ. δυτική Ελλάδα) επικρατούν κατά το καλοκαίρι θαλάσσιες αύρες οι οποίες είναι δυνατόν να πνέουν σε εντελώς διαφορετική κατεύθυνση από τους ανέμους που πνέουν στα υψηλότερα στρώματα.

1.3.2 Ο χειμώνας

Κατά το χειμώνα η Ελλάδα βρίσκεται εντός της εύκρατης ζώνης με συνέπεια να διέρχονται διαδοχικές υφέσεις και αντικυκλώνες. Η Ελλάδα βρίσκεται μεταξύ του μόνιμου αντικυκλώνα του Ατλαντικού και του Σιβηρικού αντικυκλώνα, ο οποίος σχηματίζεται το χειμώνα πάνω από την παγωμένη Ρωσία. Μεταξύ των δυο αυτών αντικυκλώνων διέρχονται κινητοί αντικυκλώνες και υφέσεις από τους οποίους άλλοι δημιουργούνται στο βόρειο Ατλαντικό και άλλοι στις Άλπεις. Όταν προς την Ελλάδα κατευθύνεται μια ύφεση , στο μπροστινό της μέρος έχουμε νότιους έως ΝΑ ανέμους οι οποίοι φέρουν θερμές υγρές μάζες στον ελληνικό χώρο και προκαλούν ραγδαίες παρατεταμένες

βροχές. Παρομοίως βροχές προκαλεί και η διέλευση των μετώπων της υφέσεως. Αντίθετα η διέλευση αντικυκλώνα πάνω από την Ελλάδα προκαλεί νηνεμία και ηλιόλουστες ημέρες. Η μετάβαση από τη θερινή στη χειμερινή περίοδο, ή αντιστρόφως δεν γίνεται απότομα, αλλά με διαδοχικές μεταπτώσεις. Έτσι τον Μάιο και τον Ιούνιο έχουμε περιόδους πρώιμων μελτεμιών, μέχρι δε τα τέλη Οκτωβρίου εμφανίζονται μικρές περιόδους όψιμων μελτεμιών.[4]

1.4 Μετεωρολογικός Σταθμός

1.4.1 Ορισμός

Ο **Μετεωρολογικός σταθμός** είναι ένα επίγειο σημείο στο οποίο πραγματοποιούνται τακτικές μετεωρολογικές παρατηρήσεις. Πρόκειται για επανδρωμένη μόνιμη εγκατάσταση (κτιριακή) στην οποία φέρονται πολλά μετεωρολογικά όργανα, τόσο μέσα σε μετεωρολογικό κλωβό είτε εκτός αυτού στον πέριξ χώρο είτε και εντός αυτού, όπως επαναλήπτες μετεωρολογικών οργάνων. Η θέση ανέγερσης αυτών των σταθμών ορίζεται από τη κεντρική *Εθνική Μετεωρολογική Υπηρεσία* της κάθε χώρας έτσι ώστε στο σύνολό τους αυτοί ν' αποτελούν ένα ενιαίο δίκτυο μετεωρολογικής παρατήρησης. [8]



Εικόνα 1.16: Μετεωρολογικός σταθμός σε κλωβό

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

Καθένας μετεωρολογικός σταθμός φέρει διεθνή αριθμό ταυτότητας με τον οποίο και απεικονίζεται στους μετεωρολογικούς χάρτες. Οι Μετεωρολογικοί σταθμοί επανδρώνονται από επιστημονικό προσωπικό ή ειδικά εκπαιδευμένο για τις ανάγκες των παρατηρήσεων. Στην Ελλάδα Μετεωρολογικοί σταθμοί υπάρχουν στις κυριότερες πόλεις, στους μεγάλους λιμένες και σε όλα τα αεροδρόμια της Χώρας. Από τους Σταθμούς αυτούς μεταβιβάζονται σε τακτά χρονικά διαστήματα οι παρατηρούμενες ενδείξεις των φερομένων οργάνων με ειδικό κωδικοποιημένο τύπο σήματος. Οι σηματικές αυτές αναφορές των μετεωρολογικών σταθμών στη κεντρική υπηρεσία καταχωρούνται στους υπό σύνταξη μετεωρολογικούς χάρτες της ευρύτερης περιοχής, από τη μελέτη των οποίων εξάγονται συμπεράσματα πρόβλεψης καιρού.

Επίσης μετεωρολογικοί σταθμοί θεωρούνται πέραν των ειδικών επιστημονικής έρευνας πλοίων και όλα εκείνα που φέρουν μετεωρολογικό κλωβό και καταγράφουν τακτές μετεωρολογικές παρατηρήσεις. Και αυτά φέρουν ιδιαίτερους κωδικούς αριθμούς με τους οποίους και υποβάλουν "μέτεο-σήματα" στη κεντρική μετεωρολογική υπηρεσία της εγγύτερης χώρας του χώρου που διαπλέουν ή σε διεθνή μετεωρολογική υπηρεσία αν βρίσκονται σε Ωκεανούς.[9]

	Εισαγωγικοί απλου τυπου	Εισαγωγικοί αυτοματοι	Οικονομικοί αυτοματοι	Μεσαιου επιπεδου αυτοματοι	Επαγγελματικοι
Τιμη	100 €	100 € εως 200 €	200 € εως 500 €	500 € εως 1500 €	> 1500 €
Μαρκες	TechnoLine Oregon Scientific	Fine Offset Watson Technoline La Crosse Oregon Scientific	Oregon Scientific Davis Instruments Irox Ventus Nielsen Kestrel Gemini	Davis Instruments	Campbell Scientific Met One Instruments Vaisala Environmental Measurements Davis Instruments

Εικόνα 1.17: Μάρκες μετεωρολογικών σταθμών ανάλογα με τον τύπο τους

1.5 Αυτόνομος Μετεωρολογικός Σταθμός

Αυτόνομος μετεωρολογικός σταθμός (ΑΜΣ) ορίζεται ένας μετεωρολογικός σταθμός στον οποίο γίνονται λήψεις μετεωρολογικών δεδομένων με αυτοματοποιημένη διαδικασία και στην συνέχεια τοπικά ή σε κεντρικές μονάδες όπου αποστέλλονται, γίνεται η επεξεργασία των δεδομένων τους. Στους ΑΜΣ, οι μετρήσεις των οργάνων λαμβάνονται από μια κεντρική μονάδα δεδομένων, στην περίπτωση μας από το Arduino. Ο αυτόνομος σταθμός καιρού μπορεί να αποτελούν μέρος ενός δικτύου συνολικής καταγραφής, ονομάζονται δε συνήθως «Αυτοματοποιημένο Σύστημα

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

Παρατήρησης Καιρικών Συνθηκών» (Automated Weather Observing System - AWOS) ή «Αυτοματοποιημένο Σύστημα Παρατήρησης Επιφάνειας» (Automatic System Observation Surface - ASOS). Όμως η ευρεία χρήση αναφοράς σε ένα τέτοιο σύστημα είναι ΑΜΣ, αν και ο όρος "σταθμός" δεν ανταποκρίνεται πλήρως σε αυτόν τον ορισμό. Παρ'όλα αυτά, στην παρούσα εργασία η αναφορά στον όρο ΑΜΣ θα εννοεί ένα τέτοιο σύστημα.[10]



Εικόνα 1.18: Μετεωρολογικός σταθμός σε βουνό

ΚΕΦΑΛΑΙΟ 2

2.1 Η ΠΛΑΤΦΟΡΜΑ ARDUINO

2.1.1 Εισαγωγικά για το Arduino



Εικόνα 2.1: Το λογότυπο του Arduino

Το Arduino είναι μια πλακέτα που βασίζεται σε ανοιχτού υλικού μικροεπεξεργαστή, ο οποίος προγραμματίζεται σε γλώσσα Wiring, μια παραλλαγή της γνωστής γλώσσας C++[11]. Το Arduino χαρακτηρίζεται για το ελάχιστο κόστος κατασκευής του, την χαμηλή κατανάλωση σε ισχύ, κάτι που αξιοποιείται κυρίως σε φορητές ή και αυτόνομες λύσεις, το μεγάλο πλήθος από περιφερειακές κάρτες για πιο πολύπλοκες και εξειδικευμένες λύσεις, όπως η κάρτα Ardupilot η οποία περιέχει συναρτήσεις και αισθητήρες για την δημιουργία συστήματος αυτόματης πλοήγησης και χρησιμοποιείται για την κατασκευή τηλεκατευθυνόμενων αεροσκαφών. Η αρχιτεκτονική Arduino, αποτελείται από ένα σύνολο μικροελεγκτών υλικών συστημάτων που με την κατάλληλη παροχή λογισμικού, αυτοματοποιούν δραστηριότητες που επιθυμούν οι χρήστες. Στην ενότητα αυτή, παρουσιάζεται ο μικροελεγκτής Arduino που χρησιμοποιήθηκε, τα χαρακτηριστικά του, διαθέσιμα εξαρτήματα που μπορούν να συνδεθούν απ' ευθείας, αλλά και αυτά ανάλυση εκείνων που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος. Το σύστημα στήθηκε πάνω σε μικροελεγκτή της αρχιτεκτονικής Arduino. Πρόκειται για ανοικτού λογισμικού πλατφόρμα πρωτοτύπων ηλεκτρονικών συσκευών που βασίζονται στην ευελιξία και στην ευκολία χρήσης υλικού και λογισμικού. Οι Arduino συσκευές μπορούν να αλληλοεπιδρούν με το περιβάλλον κάνοντας λήψη

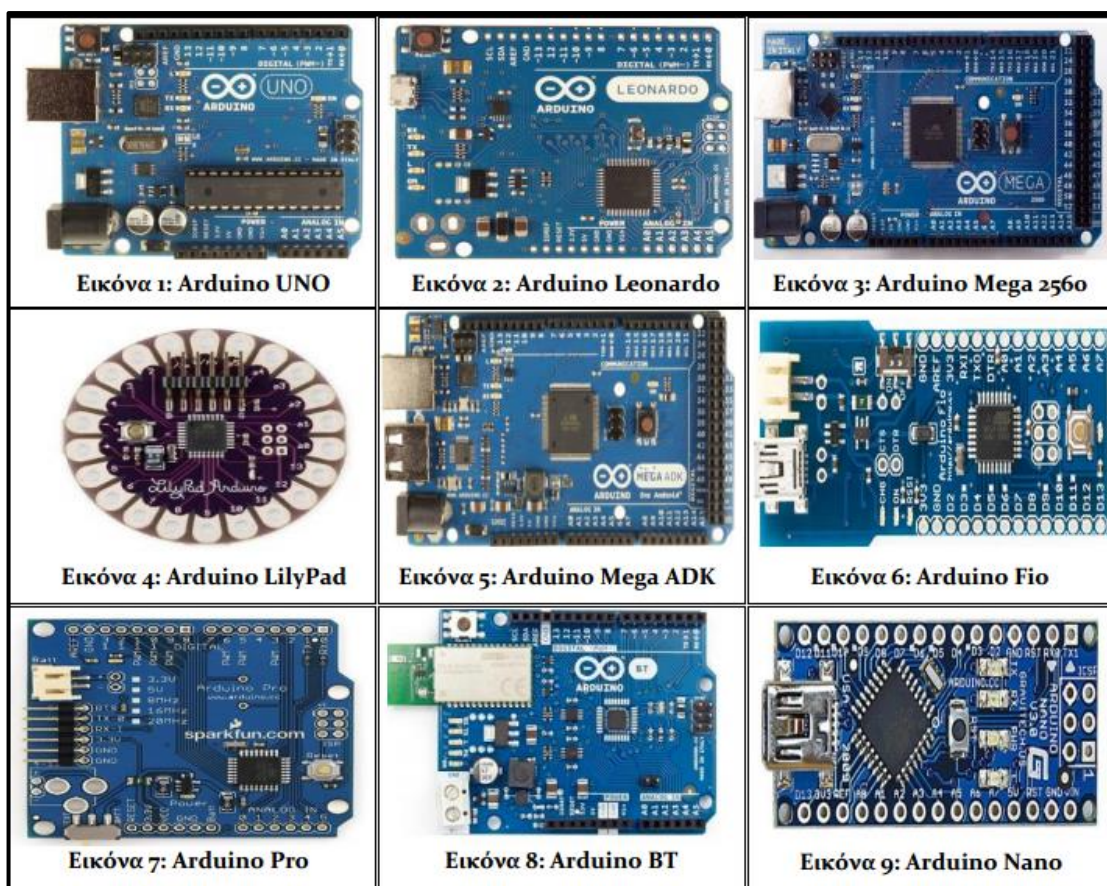
Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

σημάτων μέσα από μια ποικιλία αισθητήρων. Τα έργα που βασίζονται σε αυτούς τους μικροελεγκτές μπορούν να είναι αυτόνομα ή μπορούν να επικοινωνούν με το λογισμικό που τρέχει σε έναν υπολογιστή. [12]

2.1.2 Μοντέλα μικροελεγκτών Arduino

Στην παρακάτω λίστα παρουσιάζονται τα διάφορα μοντέλα μικροελεγκτών Arduino. Οι διαφορές τους συνήθως είναι στους ακροδέκτες (pins) που έχουν, στην τάση εισόδου και εξόδου, καθώς και στα χαρακτηριστικά των συστημάτων που υλοποιούν:

- ❖ Arduino UNO (Συνήθως για συστήματα έρευνας ή ερασιτεχνικά)
- ❖ Arduino Leonardo
- ❖ Arduino Mega 2560 (Συνήθως για βιομηχανικά συστήματα)
- ❖ Arduino LilyPad
- ❖ Arduino Mega ADK
- ❖ Arduino Fio
- ❖ Arduino Pro
- ❖ Arduino BT



- ❖ Arduino Nano

Εικόνα 2.2: Τα είδη του Arduino

2.1.3 Το Υλικό (Arduino Uno)

Αποτελείται από μια πλακέτα με πυρήνα, ένα μικροελεγκτή των 8-bit (Atmel AVR –ATmega328 ή ATmega168) και διαθέτει από οχτώ έως δεκαέξι σειριακές θύρες (RS232) και οχτώ ψηφιακές, αναλόγως την έκδοση της πλακέτας, με τις οποίες αλληλοεπιδρά με διάφορες συσκευές. Επίσης, υπάρχουν κάποιες επιπλέον κάρτες γνωστές ως “shields” , οι οποίες προσφέρουν περισσότερες δυνατότητες στον μικροελεγκτή. Αυτές οι κάρτες κατασκευάστηκαν έτσι ώστε να παρέχεται η δυνατότητα πρόσθετης διασύνδεσης του μικροελεγκτή, όπως με δίκτυο τύπου Ethernet (Ethernet shield), ασύρματο δίκτυο wifi, ασύρματη σειριακή επικοινωνία με τεχνολογία Bluetooth (BT Shield) κ.α. Λόγω της ευρείας χρήσης του Arduino, όλο και περισσότερες shields κατασκευάζονται από τρίτους κατασκευαστές, όπως δέκτης GPS για λήψη της τρέχουσας τοποθεσίας, εμφάνιση πληροφοριών σε οθόνη OLED κλπ. Υπάρχουν πολλών ειδών πλακέτες Arduino οι οποίες έχουν σχεδιαστεί έτσι ώστε να καλύψουν ένα μεγάλο εύρος απαιτήσεων.[11]



Εικόνα 2.3: Το Arduino Uno

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

Για την εργασία χρησιμοποιήθηκε η βασική πλακέτα Arduino UNO με τα εξής τεχνικά χαρακτηριστικά:

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

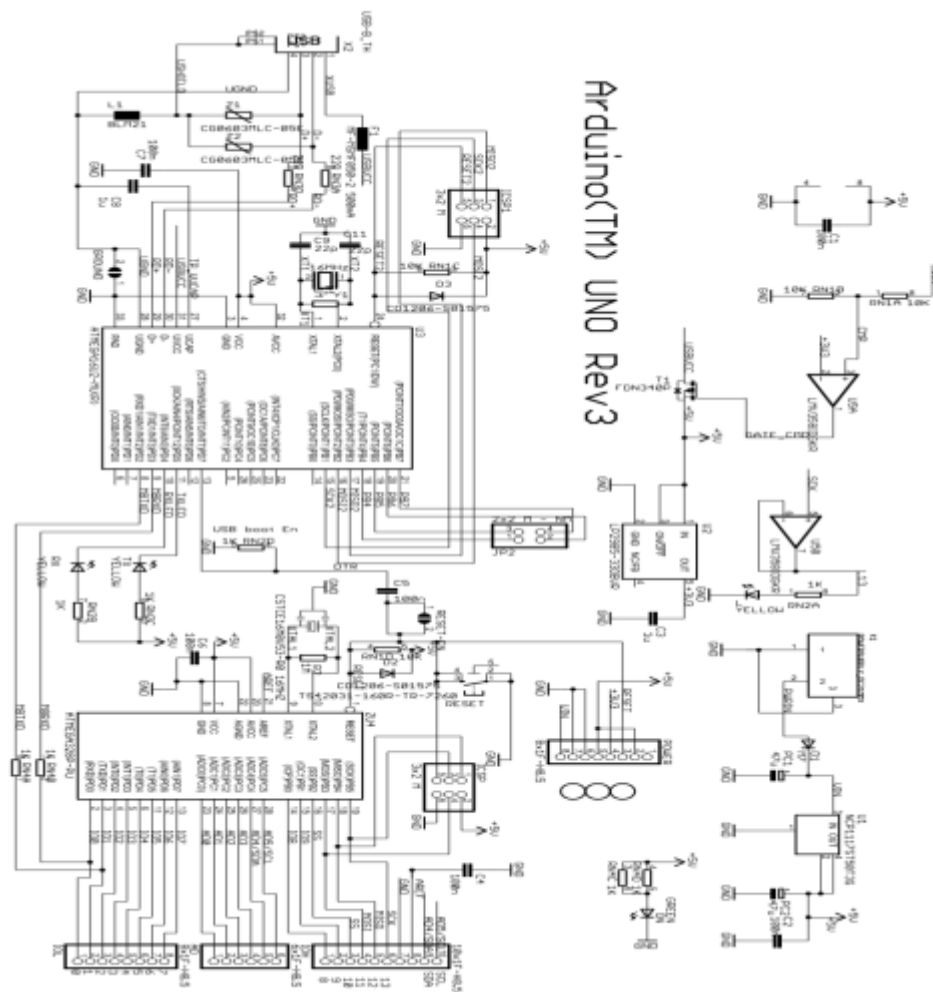
Εικόνα 2.4: Τα τεχνικά χαρακτηριστικά του Arduino Uno

Τα πλεονεκτήματα του καινούριου Revision 3 της πλακέτας που χρησιμοποιούμε είναι:

- ✓ Προστέθηκαν τα SDA και SCL ακροδέκτες που είναι κοντά στο AREF ακροδέκτες, και δύο καινούρια ακροδέκτες που βρίσκονται δίπλα στο RESET ακροδέκτες, το IOREF που επιτρέπει στις πλακέτες που μπαίνουν πάνω από το Uno (shields) να χρησιμοποιούν την ίδια τάση που προέρχεται από το Uno.
- ✓ Καλύτερο RESET κύκλωμα
- ✓ Ο ATmega16U2 αντικατέστησε τον 8U2.

Στην επόμενη σελίδα ακολουθεί το σχηματικό του Arduino Uno όπως το δίνει ο κατασκευαστής.[11]

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android



Εικόνα 2.5: Το σχηματικό του Arduino Uno

2.1.4 Τροφοδοσία

Το Arduino Uno μπορεί να τροφοδοτηθεί μέσω USB καλωδίωσης ή με εξωτερικό τροφοδοτικό. Η πηγή τροφοδοσίας επιλέγεται αυτόματα. Η εξωτερική τροφοδοσία χωρίς USB μπορεί να προέλθει είτε από έναν AC-to-DC αντάπτορα είτε από μπαταρία. Ο αντάπτορας μπορεί να ενωθεί με ένα 2.1mm κεντρικό θετικό βύσμα στην υποδοχή ρεύματος (power jack) της πλακέτας. Οι οδηγιοί από την μπαταρία μπορούν να τοποθετηθούν στο Gnd και Vin ακροδέκτες της πλακέτας.

Η πλακέτα μπορεί να λειτουργήσει από εξωτερική πηγή των 6 έως 20 volts. Παρ όλα αυτά αν τροφοδοτηθεί με λιγότερο από 7V, ο 5V ακροδέκτης ίσως τροφοδοτήσει με λιγότερο από πέντε volt και η πλακέτα θα είναι ασταθής. Αν χρησιμοποιηθούν περισσότερα από 12V, ο ρυθμιστής τάσης μπορεί να υπερθερμανθεί και να ζημιώσει την πλακέτα. Το προβλεπόμενο εύρος τιμών είναι 7 με 12 volts.

Οι ακροδέκτες της τροφοδοσίας είναι τα εξής:

•VIN. Η εισερχόμενη τάση του Arduino όταν αυτή τροφοδοτείται από εξωτερική πηγή. Μπορούμε να τροφοδοτήσουμε τάση μέσω αυτού του ακροδέκτη, αν τροφοδοτούμε μέσω της υποδοχής ρεύματος (power jack) μπορούμε να έχουμε πρόσβαση σε αυτήν την τάση μέσω του ακροδέκτη.

•5V. Αυτός ο ακροδέκτης εξάγει 5V από τον ρυθμιστή τάσης της πλακέτας. Η πλακέτα μπορεί να τροφοδοτηθεί είτε με τάση από τον DC υποδοχέα (7-12V), τον USB υποδοχέα (5V) είτε το VIN ακροδέκτη (7-12V). Τροφοδοτώντας τάση μέσω των 5V ή 3.3V ακροδεκτών παρακάμπτε τον ρυθμιστή και μπορεί να προκαλέσει βλάβη στην πλακέτα.

•3.3V. Μια 3.3 volt τάση παράγεται από τον ρυθμιστή που βρίσκετε πάνω στην πλακέτα. Η μέγιστη κατανάλωση ρεύματος είναι 50mA.

•GND. Ακροδέκτης γείωσης.

•IOREF. Αυτός ο ακροδέκτης παρέχει στην πλακέτα την τάση αναφοράς με την οποία ο μικροελεγκτής λειτουργεί. Μια πλακέτα που θα προσθέσουμε (shield) θα μπορεί να διαβάσει την τάση από τον IOREF ακροδέκτη και να επιλέξει την κατάλληλη πηγή τάσης ή να ενεργοποιήσει τους επιλογείς τάσης για τις εξόδους δουλεύοντας με 5V ή 3.3V.[11]

2.1.5 Μνήμη

Ο ATmega328 έχει 32 KB (με 0.5 KB να χρησιμοποιούνται για τον bootloader). Επίσης έχει 2KB SRAM και 1KB EEPROM (η οποία μπορεί να διαβαστεί και να γραφεί μέσω της EEPROM βιβλιοθήκης).[11]

2.1.6 Είσοδος/Εξοδος

Κάθε ένα από τους 14 ψηφιακούς ακροδέκτες στο Uno μπορούν να χρησιμοποιηθούν σαν είσοδοι και έξοδοι κάνοντας χρήση των μεθόδων pinMode(), digitalWrite() και digitalRead(). Λειτουργούν στα 5 volts. Κάθε ακροδέκτης μπορεί να διαθέσει ή να δεχτεί μέγιστο ρεύμα των 40mA και έχουν μια εσωτερική pull-up αντίσταση των 20- 50 kOhms. Σε αντίθεση κάποιοι ακροδέκτες έχουν διαφορετικές ιδιότητες- λειτουργίες:

•Serial: 0 (RX) and 1 (TX). Χρησιμοποιείτε για λήψη (RX) και εκπομπή (TX) TLL σειριακών δεδομένων. Αυτοί οι ακροδέκτες είναι συνδεδεμένοι με τα αντίστοιχους του ATmega8U2 USB-to-TTL Serial chip.

- External Interrupts: 2 and 3. Αυτοί οι ακροδέκτες μπορούν να τροποποιηθούν για να προκαλούν διακοπές σε μία χαμηλή τιμή, σε μία ανοδική η καθοδική ακμή ή σε μία αλλαγή μίας τιμής.
- PWM: 3, 5, 6, 9, 10, και 11. Παρέχουν 8-bit PWM έξοδο με την `analogWrite()` μέθοδο.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Αυτοί οι ακροδέκτες υποστηρίζουν SPI επικοινωνία μέσω της βιβλιοθήκης SPI.
- LED: 13. Υπάρχει ένα ενσωματωμένο led ενωμένο με το ψηφιακό ακροδέκτη 13. Όταν θέτουμε στον ακροδέκτη την τιμή HIGH, το led είναι ανοικτό, ενώ όταν είναι LOW, τότε είναι κλειστό.

Το UNO έχει 6 αναλογικές εισόδους, με τις ετικέτες A0 μέχρι A5, κάθε μια από αυτές παρέχει 10bit ανάλυση (π.χ 1024 διαφορετικές τιμές). Είναι προεπιλεγμένες να μετρούν από την γείωση μέχρι 5 volts, παρόλο που είναι εφικτό να αλλάξουμε την εμβέλεια (άνω των 5volts) χρησιμοποιώντας τον AREF ακροδέκτη και την `analogReference()` μέθοδο. Διαφορετικά κάποιοι άλλοι ακροδέκτες έχουν διαφορετική λειτουργικότητα:

- WI: A4 ή SDA και A5 ή SCL ακροδέκτες. Υποστηρίζουν TWI επικοινωνία χρησιμοποιώντας την Wire βιβλιοθήκη.

Υπάρχουν και άλλοι ακροδέκτες στην πλακέτα όπως:

- AREF. Τάση αναφοράς για τις αναλογικές εισόδους. Χρησιμοποιείτε με την `analogReference()`.
- Reset. Κατεβάζουμε αυτή την γραμμή LOW για να κάνουμε επανεκκίνηση του μικροελεγκτή. Τυπικά χρησιμοποιείται για να δημιουργούμε ένα κουμπί επανεκκίνησης όταν έχουμε κάποιο shield τοποθετημένο και δεν μπορούμε να χρησιμοποιήσουμε αυτό του UNO.[11]

2.1.7 Επικοινωνία

Το Arduino Uno έχει διάφορους τρόπους να επικοινωνήσει με έναν υπολογιστή, με ένα άλλο Arduino ή έναν άλλο μικροελεγκτή. Ο ATmega328 παρέχει UART TTL (5V) σειριακή επικοινωνία, η οποία είναι διαθέσιμη στους ψηφιακούς ακροδέκτες 0 (RX) και 1 (TX). Ένας ATmega16U2 μεταφέρει αυτή τη σειριακή επικοινωνία μέσω USB ενώ παρουσιάζετε σαν εικονική com πόρτα στο λογισμικό του υπολογιστή. Το 16U2 firmware χρησιμοποιεί τους κλασικούς USB COM οδηγούς και έτσι δεν χρειάζονται εξωτερικοί. Παρόλα αυτά στα

Windows ένα .inf αρχείο χρειάζεται. Το λογισμικό του Arduino περιέχει μία εικονική οθόνη η οποία απλά με εμφανίζει τα δεδομένα που στέλνονται από και προς την πλακέτα. Τα RX και TX leds αναβοσβήνουν όταν δεδομένα μεταφέρονται μέσω USB-to-serial chip και USB ένωσης στον υπολογιστή.

Η βιβλιοθήκη SoftwareSerial επιτρέπει την σειριακή επικοινωνία για κάθε ένα από τους ακροδέκτες της πλακέτας. Ο ATmega328 επίσης υποστηρίζει I2C (TWI) και SPI επικοινωνία. Το λογισμικό του Arduino περιέχει την βιβλιοθήκη Wire για να απλοποιήσει την χρήση του I2C διαύλου.[11]

2.1.8 Προγραμματισμός

Το Arduino Uno μπορεί να προγραμματιστεί μέσω του λογισμικού Arduino IDE. Ο ATmega328 έρχεται με προ εγκατεστημένο bootloader ο οποίος μας επιτρέπει να ανεβάσουμε το κώδικα μας χωρίς την χρήση εξωτερικού προγραμματιστή. Επικοινωνεί χρησιμοποιώντας το STK500 πρωτόκολλο. Μπορούμε επίσης να παρακάμψουμε τον bootloader και να προγραμματίσουμε τον μικροελεγκτή μέσω της ICSP (In-Circuit Serial Programming) επικεφαλίδας χρησιμοποιώντας το Arduino ISP ή κάτι παρόμοιο.

Ο firmware κώδικας του ATmega16U2 είναι διαθέσιμος. Ο ATmega16U2 είναι φορτωμένος με έναν DFU bootloader, ο οποίος μπορεί να ενεργοποιηθεί από την αντίσταση που τραβάει ο 8U2/16U2 HWB στην γείωση, κάνοντας το έτσι πιο εύκολο να μπαίνει σε DFU λειτουργία.

Μπορούμε να χρησιμοποιήσουμε το λογισμικό Flip της Atmel για Windows ή το DFU programmer για Mac OS X και Linux για να φορτώσουμε firmware. Διαφορετικά μπορούμε να κάνουμε χρήση της ISP επικεφαλίδας με εξωτερικό προγραμματιστή.[11]

2.1.9 Φυσικά Χαρακτηριστικά

Το μέγιστο μήκος και πλάτος της πλακέτας του Uno είναι 6.858 και 5.334 εκατοστά αντίστοιχα, με την USB πόρτα και την υποδοχή ρεύματος να εκτείνονται πέρα των αρχικών διαστάσεων. Τέσσερις τρύπες για βίδες επιτρέπουν στην πλακέτα να μπορεί να τοποθετηθεί σε μία ξεχωριστή επιφάνεια ή ένα κουτί.[11]

2.1.10 Το λογισμικό λειτουργίας του Arduino

Το Ολοκληρωμένο Περιβάλλον Ανάπτυξης (Intergraded Development Environment – IDE) με το οποίο θα προγραμματίσουμε το λογισμικό του μετεωρολογικού σταθμού, έχει υλοποιηθεί με την γλώσσα Java η οποία

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

εκτελείται ανεξάρτητα από το λειτουργικό ή υλικό του υπολογιστή, είναι ευρέως διαδεδομένη, και έχει πολλές ομοιότητες με την γλώσσα C. Κατά συνέπεια, το συγκεκριμένο IDE είναι πολλαπλού λειτουργικού (multi-platform) και μπορεί να δουλεύει άψογα σε κάθε λειτουργικό σύστημα (Windows, Linux, OSX). Η γλώσσα wiring είναι μια παραλλαγή της C++ και όπως και το IDE, είναι μια πλατφόρμα ανοιχτού λογισμικού που χρησιμοποιείται για προγραμματισμό μικροελεγκτών. Έχει επίσης σχεδιαστεί λαμβάνοντας υπόψη μια ευρεία γκάμα χρηστών, από αρχάριους μέχρι επαγγελματίες, με σκοπό την δημιουργία μιας κοινότητας η οποία θα ανταλλάσσει ιδέες, τεχνογνωσία και εμπειρία, με τελικό στόχο την κατασκευή “έξυπνων συσκευών” οι οποίες θα περιορίζονται μόνο από την φαντασία του εκάστοτε δημιουργού τους.



```
File Edit Sketch Tools Help
Blink
/*
 * BLINK
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13

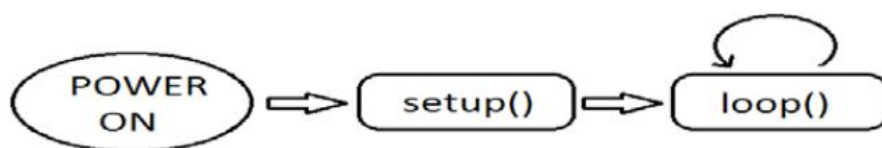
void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}

Done compiling.
```

Εικόνα 2.6: Το περιβάλλον ανάπτυξης του Arduino

Κύριο χαρακτηριστικό της wiring είναι ότι απαιτεί την ύπαρξη δυο τουλάχιστον συναρτήσεων. Της setup() και της loop(). Η setup() περιέχει όλες εκείνες τις μεταβλητές οι οποίες απαιτούνται για την αρχικοποίηση του προγράμματος. Η loop(), αφού έχει εκτελεστεί η setup(), περιέχει την δομή του υπόλοιπου προγράμματος και εκτελείται συνεχόμενα πλην της περίπτωσης να έχει δοθεί εντολή τερματισμού μετά από συγκεκριμένες συνθήκες.[11]



Εικόνα 2.7: Διάγραμμα ροής λειτουργίας της γλώσσας Wiring

Γενικά η λειτουργία του προγράμματος είναι πολύ εύκολη, αφού ο χρήστης γράψει το σκίτσο έχει τη δυνατότητα να το ελέγξει για τυχόν συντακτικά λάθη και στην συνέχεια να το στείλει στο μικροελεγκτή της εκάστοτε πλακέτας. Ακόμα δίνει την δυνατότητα με συγκεκριμένες συναρτήσεις που υπάρχουν να παρουσιάσουμε στο Serial Monitor που είναι ενσωματωμένο με το λογισμικό αποτελέσματα αισθητήρων ή γενικά εξόδους.

2.2 Αισθητήρες

2.2.1 Περιγραφή των αισθητήρων καταγραφής

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

- α) Αισθητήρας θερμοκρασίας / υγρασίας DHT-22**
- β) Αισθητήρας Ατμοσφαιρικής Πίεσης BMP-180**
- γ) Αισθητήρας φωτός (Photo Resistor)**

2.2.2 Αισθητήρας Θερμοκρασίας/Υγρασίας DHT-11

Ο αισθητήρας DHT-11 είναι ένας φθηνός και αξιόπιστος αισθητήρας. Χρησιμοποιεί έναν πυκνωτή για την υγρασία και ένα θερμίστορ για να μετρήσει τον περιβάλλοντα αέρα. Όλα αυτά συνδέονται με έναν υψηλής επίδοσης 8-bit μικροελεγκτή που προσφέρει πολύ καλή ποιότητα και ταχύτητα στα αποτελέσματα.



Εικόνα 2.7: Ο Αισθητήρας Θερμοκρασίας/Υγρασίας DHT-11

Γενικά ο αισθητήρας είναι μικρός σε μέγεθος, έχει αρκετά χαμηλή κατανάλωση ισχύος και είναι αρκετά ακριβής στις μετρήσεις. Έχει τρεις ακροδέκτες, έναν για την τροφοδοσία, για την γείωση και για τα δεδομένα.

Τεχνικά Χαρακτηριστικά:

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

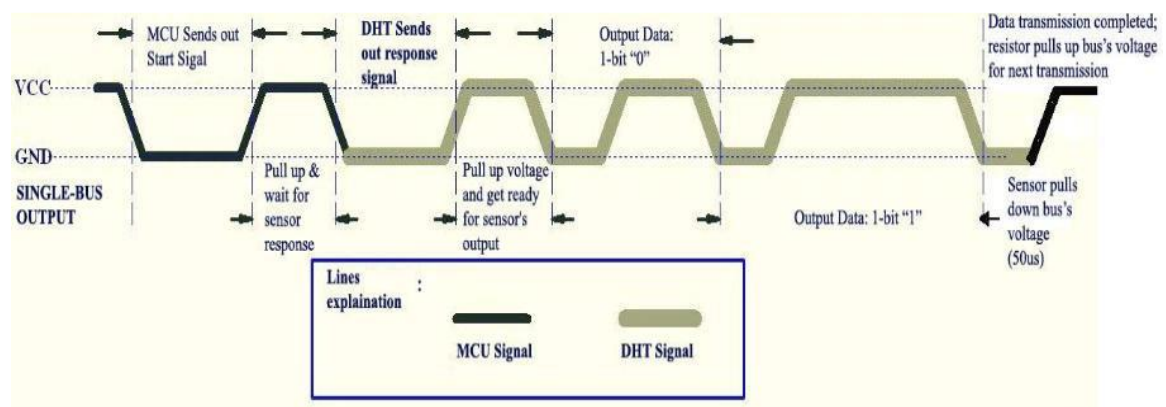
	Υγρασία	Θερμοκρασία
Τάση	3-5.5Volt	
Έξοδος	Ψηφιακή	
Αισθητήρας	Resistive-Type Humidity	NTC temperature component
Εύρος	20-80% Rh	0-50°C
Ακρίβεια	5%	±2°C
Δειγματοληψία	1Hz	

Εικόνα 2.8: Τα τεχνικά χαρακτηριστικά του αισθητήρα DHT-11

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

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

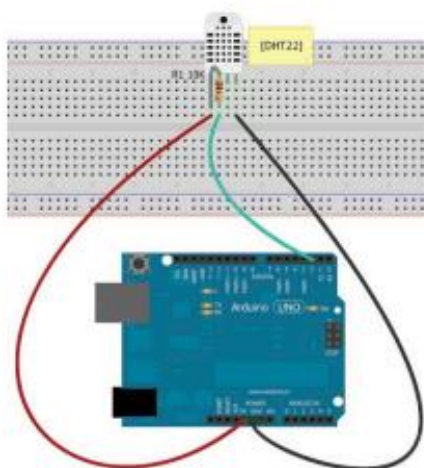
Χωρίς το αρχικό σήμα από το Arduino ο αισθητήρας δεν στέλνει δεδομένα. Όταν τα δεδομένα σταθούν από τον αισθητήρα αυτός μεταβαίνει και πάλι στην χαμηλής-κατανάλωσης λειτουργία και περιμένει νέο σήμα.[13]



Εικόνα 2.9: Διάγραμμα μεταφοράς δεδομένων DHT-11

Συνδεσμολογία με την πλακέτα Arduino

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



Εικόνα 2.10: Σύνδεση του αισθητήρα DHT-11 με το Arduino

2.2.3 Αισθητήρας Βαρομετρικής Πίεσης BMP180

Ο αισθητήρας βαρομετρικής πίεσης BMP180 της εταιρίας Bosch, υπολογίζει την βαρομετρική πίεση της ατμόσφαιρας και την θερμοκρασία. Επίσης επειδή η πίεση αλλάζει σύμφωνα με το υψόμετρο, με τον κατάλληλο κώδικα μπορεί να χρησιμοποιηθεί και σαν μετρητής υψομέτρου. Ο αισθητήρας είναι αρκετά μικρός σε μέγεθος και είναι κολλημένος σε PCB με ρυθμιστή τάσης 3.3V, I2C level shifter και pull-up αντίσταση στο I2C ακροδέκτη. Στο εσωτερικό του υπάρχει ένας αισθητήρας πιεζοηλεκτρικής αντίστασης, ένας αναλογικός σε ψηφιακός μετατροπέας και μία μονάδα ελέγχου με EEPROM και σειριακή I2C διεπαφή. Η EEPROM αποθηκεύει 176 bit από ξεχωριστά δεδομένα που υπάρχουν για το setup. (UP = δεδομένα πίεσης 16 έως 19 bit, UT = δεδομένα θερμοκρασίας 16 bit)

Τεχνικά χαρακτηριστικά:

- ✓ Vin: 3 έως 5VDC
- ✓ Λογικό: 3 με 5V συμβατότητα

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

- ✓ Εύρος Μετρήσιμης πίεσης: 300-1100 hPa (9000m to -500m above sea level)
- ✓ Πάνω από 0.03hPa / 0.25m ανάλυση
- ✓ -40 to +85°C εύρος λειτουργίας, +-2°C ακρίβεια θερμοκρασίας
- ✓ Η πλακέτα χρησιμοποιεί I2C 7-bit διεύθυνση 0x77.

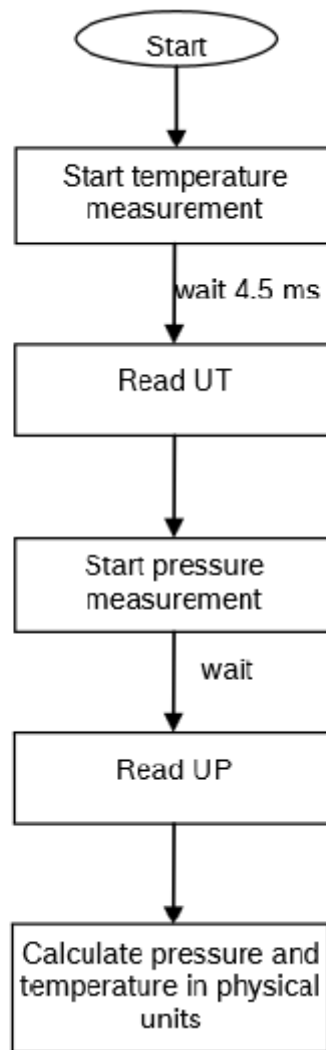


Εικόνα 2.11: Ο αισθητήρας πίεσης BMP180

Για όλες τις μετρήσεις του αισθητήρα παρέχετε από την εταιρία έτοιμη βιβλιοθήκη σε γλώσσα C.

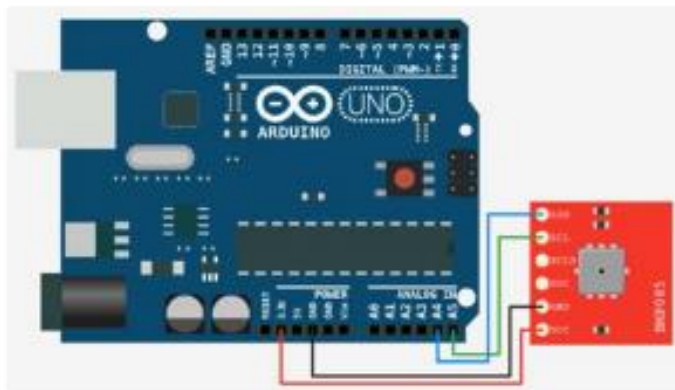
Το Arduino στέλνει μία αρχική ακολουθία για να ξεκινήσει ο αισθητήρας να μετράει την πίεση και την θερμοκρασία. Μετά από τον χρόνο μετατροπής, τα ορίσματα των αποτελεσμάτων (UP ή UT αντίστοιχα) μπορούν να διαβαστούν μέσω του I2C. Για να μετατρέψουμε τις μετρήσεις σε γνωστές μονάδες μέτρησης (Celsius την θερμοκρασία και hPa την πίεσης υπάρχουν έτοιμες βιβλιοθήκες. Ο χρόνος που παίρνουμε μετρήσεις μπορεί να αυξηθεί σε 128 μετρήσεις ανά δευτερόλεπτο.[14]

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android



Εικόνα 2.11: Γράφημα ακολουθίας BMP180

Συνδεσμολογία αισθητήρα με την πλακέτα Arduino

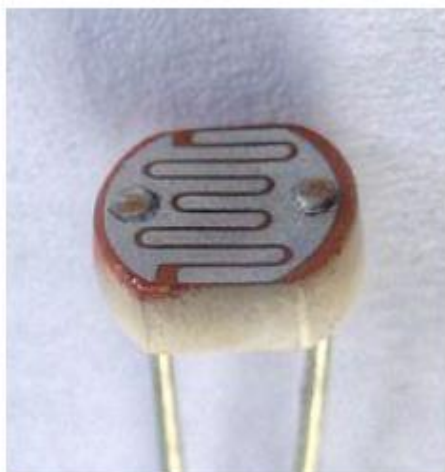


Εικόνα 2.12: Σύνδεση αισθητήρα BMP180 με Arduino

Στο σχήμα βλέπουμε την συνδεσμολογία του αισθητήρα με την πλακέτα. στον ακροδέκτη VCC συνδέεται η τάση 3.3V, στο GRD η γείωση. Ο ακροδέκτης SCL συνδέεται με το ρολόι του arduino για τον συγχρονισμό του αισθητήρα. Για το Arduino UNO αυτό βρίσκεται στην αναλογική είσοδο #5. Ο ακροδέκτης SDA συνδέεται με το πρωτόκολλο επικοινωνίας i2c και είναι υπεύθυνο για την ανάγνωση των δεδομένων του αισθητήρα

2.2.4 Αισθητήρας φωτός (Photo Resistor)

Ο αισθητήρας φωτός η αλλιώς photoresistor ή light-dependent resistor (LDR) ή photocell είναι μία μεταβλητή αντίσταση, ελεγχόμενη από το φως. Η αντίσταση του αισθητήρα μειώνεται με αυξανόμενη ένταση του φωτός, με άλλα λόγια είναι αγώγιμο στο φως.



Εικόνα 2.13: Ο φωτοαισθητήρας

Ο αισθητήρας φωτός είναι φτιαγμένος από υψηλής αντίστασης ημιαγωγό. Στο σκοτάδι ο αισθητήρας μπορεί να έχει αντίσταση της τάξης των megaohms (MΩ), σε αντίθεση με συνθήκες φωτεινότητας όπου ο αισθητήρας έχει αντίσταση της τάξης των λίγων εκατοντάδων ohms.

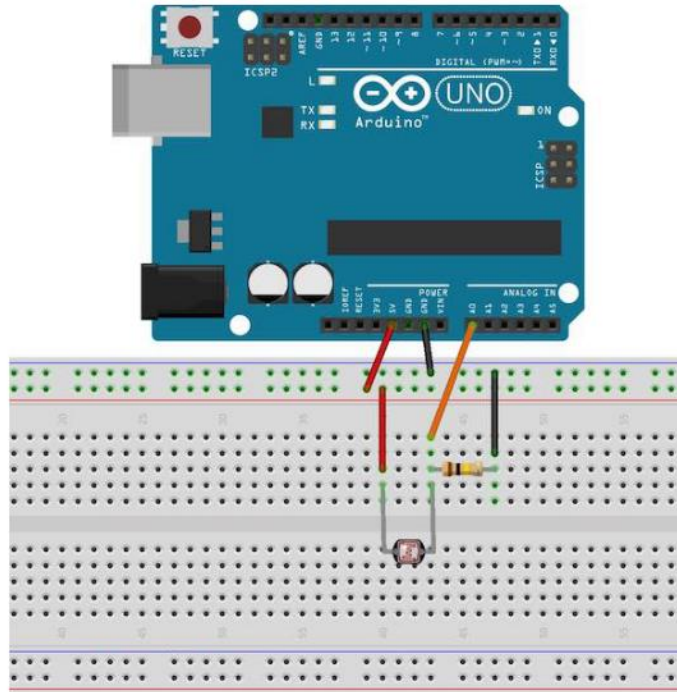
Γενικά ο αισθητήρας είναι πολύ μικρός σε μέγεθος, έχει γρήγορη απόκριση και μεγάλη ευαισθησία.[15]

Τεχνικά Χαρακτηριστικά:

Μέγεθος	Στρογγυλό, 5mm διάμετρος
Εύρος αντίστασης	200KΩ(σκοτάδι) έως 10KΩ(10 lux φωτεινότητα)
Εύρος ευαισθησίας	Απόκριση σε φως με φάσμα 400nm και 600nm με peak 520nm
Τάση τροφοδοσίας	Οτιδήποτε μέχρι τα 100V, χρησιμοποιεί λιγότερο από 1mA.

Εικόνα 2.14: Τα τεχνικά χαρακτηριστικά του φωτοαισθητήρα

Συνδεσμολογία αισθητήρα με την πλακέτα Arduino



Εικόνα 2.15: Σύνδεση του φωτοαισθητήρα με το Arduino

ΚΕΦΑΛΑΙΟ 3

3.1 Λειτουργικό Σύστημα Android

3.1.1 Εισαγωγικά



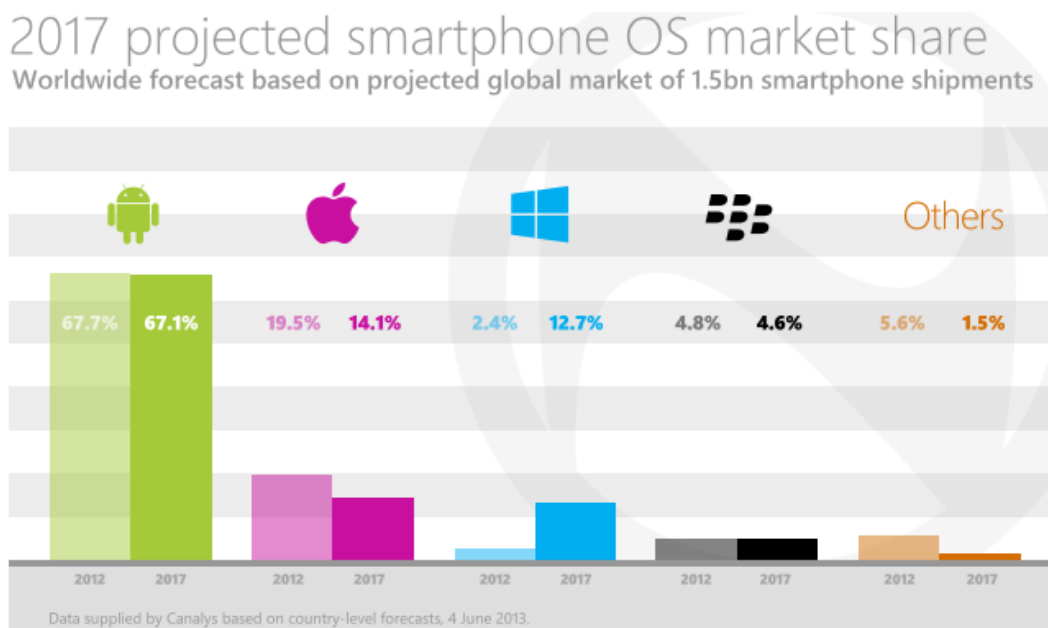
Εικόνα 3.1: Το λογότυπο του λειτουργικού Android

Το Android αποτελεί ένα λειτουργικό σύστημα που συναντάται κυρίως σε ενσωματωμένα συστήματα όπως είναι τα κινητά τηλέφωνα, τα tablet αλλά και άλλες κινητές συσκευές. Το γεγονός ότι αυτές οι συσκευές συνήθως τροφοδοτούνται από μια μπαταρία τις κάνει να απαιτούν από το λειτουργικό σύστημα που διαθέτουν να είναι σχεδιασμένο ώστε να μην χρησιμοποιεί σε μεγάλο βαθμό τους πόρους του υλικού καθώς και να μην είναι ενεργοβόρο. Το Android είναι σχεδιασμένο έχοντας λάβει υπόψιν αυτούς ακριβώς τους περιορισμούς. Δεδομένου ότι το λειτουργικό σύστημα λειτουργεί βάση αυτής της φιλοσοφίας το ίδιο πρέπει να κάνουν και οι εφαρμογές που τρέχουν σε αυτό. Έτσι λοιπόν, έχουμε την εισαγωγή ενός νέου τρόπου σκέψης, κατά την διαδικασία ανάπτυξης, που πριν δεν υπήρχε. Ενώ μέχρι τώρα, τα προγράμματα που εκτελούνταν στους προσωπικούς μας υπολογιστές, είχαν στην διάθεσή τους «απεριόριστη» μνήμη και επεξεργαστική ισχύ πλέον αυτό παύει να ισχύει. Ο προγραμματιστής οφείλει να είναι προσεκτικός με τους πόρους που χρησιμοποιούν οι εφαρμογές του καθώς και με την διάρκεια για την οποία τους χρησιμοποιούν. Χάριν αυτού του χαρακτηριστικού ο προγραμματιστής αναπτύσσει έναν περισσότερο συγκροτημένο και προσεκτικό τρόπο σκέψης (ο οποίος κανονικά θα έπρεπε να είναι ο

ενδεδειγμένος) για την ανάπτυξη κάθε είδους εφαρμογής ανεξαρτήτως υλικού και λειτουργικού συστήματος.

Σαν πλατφόρμα κινητών συσκευών, το Android, ανήκει σε μια κατηγορία συστημάτων που καταλαμβάνουν ολοένα μεγαλύτερο μερίδιο στην πύλα της πρόσβασης στο διαδίκτυο. Κατά την συγγραφή του παρόντος κειμένου έχουν μετρηθεί μέγιστες τιμές πρόσβασης στο διαδίκτυο από κινητές συσκευές, που αγγίζουν το δέκα τοις εκατό του συνολικού αριθμού συσκευών που συνδέονται σε αυτό. Αν αναλογιστεί κανείς τον αριθμό των συσκευών στις οποίες αντιστοιχεί το ποσοστό αυτό σίγουρα θα καταλήξει στο συμπέρασμα ότι οι κινητές συσκευές θα αποτελέσουν στο μέλλον το κύριο μέσο με το οποίο οι χρήστες θα συνδέονται στο διαδίκτυο. Αναπτύσσοντας για το Android έχει κανείς την ευκαιρία να είναι μέρος αυτής της εξέλιξης και γιατί όχι, να την οδηγήσει.

Ο πυρήνας του Android είναι βασισμένος στον πολύ γνωστό και αρκετά δοκιμασμένο πυρήνα του Linux. Κληρονομώντας όλα τα χαρακτηριστικά ασφαλείας του Linux και όλες τις διαχειριστικές τεχνικές μνήμης και επεξεργαστή που αυτό διαθέτει, το Android, καθίσταται ένα αρκετά αξιόπιστο λειτουργικό σύστημα. Το Android, παρότι νέο σε ηλικία, φέρει μαζί του εμπειρία χρόνων από την ανάπτυξη του πυρήνα του Linux. Επιπλέον, λόγω του ότι το Android αλλά και το Linux είναι λογισμικά ανοιχτού κώδικα, ο καθένας που το επιθυμεί, μπορεί να τα προσαρμόσει στις δικές του ανάγκες και στο δικό του υλικό. Αυτό είναι που κάνουν και οι διαφορετικές εταιρίες που παράγουν κινητές συσκευές που τρέχουν Android. Πέρα από την δυνατότητα προσαρμογής του Android σε διαφορετικό υλικό, δίνεται η δυνατότητα στον προγραμματιστή να επέμβει και στο εσωτερικό των ανώτερων επιπέδων του λειτουργικού συστήματος αλλά ακόμη και να χτίσει πάνω σε αυτό. Χαρακτηριστικό παράδειγμα είναι οι διεπαφές χρήστη που προσφέρει κάθε εταιρία που παράγει συσκευές Android. Όντας ένα λειτουργικό σύστημα ανοιχτού κώδικα, το Android, δεν θα μπορούσε παρά να διαθέτει και ένα σύνολο εργαλείων ανάπτυξης που επίσης είναι ανοιχτού κώδικα ή διατίθενται δωρεάν. Με μηδενικό κόστος εργαλείων ανάπτυξης δίνεται η δυνατότητα σε οποιονδήποτε να δοκιμάσει την ανάπτυξη εφαρμογών για Android με μοναδικό ρίσκο τον χρόνο που θα επενδύσει για την εξοικείωσή του με το περιβάλλον. Επιπλέον, λόγω της μεγάλης διάδοσης των χρησιμοποιούμενων εργαλείων και τεχνολογιών στον τομέα της ανάπτυξης εφαρμογών γενικότερα, υπάρχει στο διαδίκτυο μια μεγάλη βάση βοήθειας για τα περισσότερα προβλήματα που θα συναντήσει κανείς στα πρώτα, και όχι μόνο, βήματα του με το Android. Όπως αναφέραμε, το Android αποτελεί μια πλατφόρμα που συναντάται σε κινητές συσκευές. Το μερίδιο του Android στην αγορά κινητών συσκευών αγγίζει, όπως φαίνεται στην Εικόνα 3.2, (κατά την συγγραφή του παρόντος κειμένου) το υψηλότερο ποσοστό του 67,1% έναντι του iOS της Apple, του Blackberry και άλλων. Επομένως, με ασφάλεια μπορούμε να πούμε ότι αποτελεί το πιο δημοφιλές λειτουργικό σύστημα κινητών συσκευών σήμερα. Αξίζει να σημειωθεί ότι καθημερινά ενεργοποιούνται περίπου εννιακόσιες χιλιάδες συσκευές Android παγκοσμίως.



Εικόνα 3.2: Ποσοστά χρησιμοποίησης των διαφόρων λειτουργικών συστημάτων σε παγκόσμιο επίπεδο για το 2017

Τέλος, για όσους ενδιαφέρονται να βγάλουν κέρδος από την ανάπτυξη εφαρμογών για Android, αρκεί να σημειώσουμε ότι κάθε μήνα οι χρήστες των συσκευών Android εγκαθιστούν στις συσκευές τους ένα δισεκατομμύριο εφαρμογές συνολικά. Το Google Play, είναι ο βασικός πάροχος εφαρμογών για Android και σήμερα φιλοξενεί πάνω από μισό εκατομμύριο εφαρμογές. Το μεγάλο κοινό, ο εύκολος τρόπος δημοσίευσης εφαρμογών, τα εργαλεία διαφήμισης που προσφέρει η Google για τους προγραμματιστές εφαρμογών καθώς και το δελεαστικότερο ποσοστό πληρωμής του 70% (επί της τιμής της εφαρμογής) συνθέτουν ένα πακέτο που δύσκολα ένας έξυπνος και δημιουργικός προγραμματιστής θα του αντισταθεί. Η οικονομική επιτυχία σαφώς δεν είναι εγγυημένη αλλά η προσπάθεια μόνο θετικά αποτελέσματα μπορεί να έχει. Όσον αφορά στα αρνητικά στοιχεία, σαφώς και αυτά είναι υπαρκτά. Για προφανείς λόγους δεν θα επεκταθούμε σε αυτά αλλά θα αναφερθούμε συνοπτικά σε δυο. Το Android, όπως ήδη αναφέρθηκε είναι βασισμένο στον πυρήνα του Linux και αυτό του δίνει την δυνατότητα να τρέχει σε μια πληθώρα διαφορετικών συσκευών. Αυτό αυξάνει τον βαθμό δυσκολίας ανάπτυξης εφαρμογών γιατί πλέον ο προγραμματιστής πρέπει να λάβει υπ' όψιν του κατά την σχεδίαση της διεπαφής χρήστη την μεγάλη ποικιλία μεγεθών και αναλύσεων που μπορεί η εφαρμογή του να συναντήσει. Τέλος, το περιβάλλον πώλησης εφαρμογών της Apple είχε πού μεγαλύτερο ποσοστό κέρδους από ότι το αντίστοιχο της Google (ενός δις έναντι εκατό εκατομμυρίων). Αυτό σημαίνει ότι οι χρήστες του iOS παρότι λιγότεροι είναι περισσότερο πρόθυμοι να πληρώσουν για να αγοράσουν μια εφαρμογή. Αυτό καθιστά τις εφαρμογές για iPhone περισσότερο πιθανές να αποφέρουν κέρδη. Εδώ φυσικά δεν πρέπει να ξεχνάμε ότι τα νούμερα αυτά αλλάζουν μέρα με την μέρα προς όφελος του Android και μόνο ο χρόνος θα δείξει ποιο λειτουργικό

σύστημα θα επικρατήσει του άλλου. Προσωπική άποψη του συγγραφέα είναι ότι η μάχη θα συνεχίζεται επ' αόριστο με τα δυο λειτουργικά να αναπτύσσονται και να βελτιώνονται συνεχώς. [16]

3.1.2 Το υλικό

Όπως κάθε λειτουργικό σύστημα, έτσι και το Android, χρειάζεται κάποιο υλικό το οποίο θα του παράσχει τους απαραίτητους πόρους (μνήμη, επεξεργαστική ισχύ) για να μπορέσει να λειτουργήσει σωστά. Μπορούμε να σκεφτούμε το υλικό σαν το σώμα και το λογισμικό (στην περίπτωσή μας το λειτουργικό σύστημα) σαν το μυαλό του συστήματος. Τις πρώτες ημέρες των κινητών συσκευών οι διαθέσιμοι πόροι ήταν τρομερά περιορισμένοι. Οι πρώτες κινητές συσκευές παρείχαν μόνο την δυνατότητα πραγματοποίησης και λήψης τηλεφωνικών κλήσεων. Το μέγεθος τους ήταν αρκετά μεγαλύτερο απ' ό,τι έχουμε συνηθίσει σήμερα και περιείχαν διακριτά ηλεκτρονικά ειδικά για τον σκοπό των λειτουργιών της ασύρματης τηλεφωνίας. Είναι σαφές σε αυτό το σημείο ότι η έννοια του λειτουργικού συστήματος σε κινητή συσκευή δεν ήταν κάτι το εφικτό με τα τρέχοντα τεχνολογικά δεδομένα.

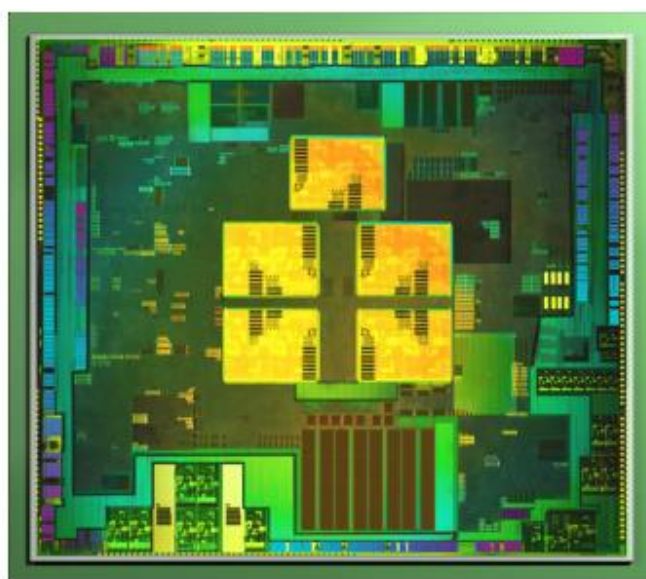


Εικόνα 3.3: Μια από τις πρώτες συσκευές κινητής

Με την συνεχή ανάπτυξη της μικροηλεκτρονικής και των τεχνικών σχεδίασης και κατασκευής ολοκληρωμένων κυκλωμάτων οι κινητές συσκευές άρχισαν σιγά-σιγά να αποκτούν χαρακτηριστικά που μέχρι τότε μόνο οι προσωπικοί υπολογιστές είχαν. Η εισαγωγή στις κινητές συσκευές μικροεπεξεργαστών και η ανάπτυξη λογισμικού για τον έλεγχο των λειτουργιών της συσκευής αποτέλεσαν την πρώτη μορφή λειτουργικού συστήματος που

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

«έτρεχε» σε κινητή συσκευή. Σήμερα, μετά από αρκετά χρόνια μιας συνεχούς εξελικτικής πορείας της μικροηλεκτρονικής αλλά και των υπολογιστών έχουμε φτάσει σε σημείο να έχουμε επεξεργαστές για κινητές συσκευές που έχουν επιδόσεις που θα ζήλευε ένας προσωπικός υπολογιστής ηλικίας πέντε ετών. Οι τεχνικές σχεδίασης ολοκληρωμένων συστημάτων σε ένα και μόνο ολοκληρωμένο κύκλωμα έχουν δώσει την δυνατότητα κατασκευής επεξεργαστών που φέρουν περισσότερους του ενός πυρήνες. Συνηθισμένοι είναι πλέον οι τετραπύρηνοι επεξεργαστές αλλά και οι επεξεργαστές που εκτός από τους βασικούς επεξεργαστικούς πυρήνες φέρουν και πολλούς περισσότερους πυρήνες ειδικού σκοπού (επεξεργασία βίντεο, 3D γραφικών κ.λ.π.) όπως το ολοκληρωμένο σύστημα Tegra 3 της nVidia.



Εικόνα 3.4: Το ολοκληρωμένο σύστημα Tegra 3 της NVidia

Έχοντας στην διάθεσή μας τόση επεξεργαστική ισχύ, μπορούμε πλέον να υποστηρίξουμε αρκετά πιο πολύπλοκα λειτουργικά συστήματα που παρέχουν λειτουργικότητες πέρα από την απλή λήψη και πραγματοποίηση κλήσεων. Επιπλέον, οι σύγχρονες κινητές συσκευές, φέρουν μια πληθώρα περιφερειακών στοιχείων όπως πυξίδες, επιταχυνσιόμετρα, αισθητήρες προσέγγισης, αισθητήρες φωτός, Wi-Fi, κάμερες, GPS και άλλα τα οποία μπορούν να αξιοποιηθούν από το λειτουργικό σύστημα. Εδώ γίνεται εμφανές ότι η υπηρεσία που μια εφαρμογή θα παρέχει περιορίζεται μόνο από την φαντασία του προγραμματιστή.[16]

3.1.3 Η Αρχιτεκτονική

Το Android είναι ένα λειτουργικό σύστημα. Σαν λειτουργικό σύστημα σκοπός του είναι να παρέχει ένα επίπεδο αφαιρετικότητας ανάμεσα στο υλικό και τον χρήστη. Με απλά λόγια ο ρόλος ενός λειτουργικού συστήματος είναι να δίνει την δυνατότητα στον χρήστη να χρησιμοποιεί τους πόρους του συστήματος προς όφελος του με μια διεπαφή περισσότερο κατανοητή προς τον άνθρωπο. Για να το καταφέρει αυτό, το Android, αποτελείται από μία στοιβή λογισμικών τμημάτων (software stack) με ξεκάθαρους και αυστηρά καθορισμένους ρόλους.



Εικόνα 3.5: Η δομή της αρχιτεκτονικής του λειτουργικού συστήματος Android

Στο κατώτερο επίπεδο βρίσκεται ο πυρήνας του Linux. Ο πυρήνας του Linux είναι γνωστός για την μεγάλη ποικιλία αρχιτεκτονικών επεξεργασιών με τις οποίες είναι συμβατός. Χτίζοντας, λοιπόν, τα υπόλοιπα τμήματα του Android πάνω από τον πυρήνα του Linux επιτυγχάνουμε την συμβατότητα του με μια ευρεία γκάμα αρχιτεκτονικών και άρα συσκευών που μπορούν να το υποστηρίξουν.

Στο επίπεδο ακριβώς πάνω από τον πυρήνα του Linux βρίσκονται οι βασικές βιβλιοθήκες του συστήματος. Εδώ βρίσκονται οι υλοποιήσεις για βιβλιοθήκες γραφικών, για συστήματα βάσεων δεδομένων, η βασική

βιβλιοθήκη της γλώσσας προγραμματισμού C και άλλα. Όλες αυτές οι βιβλιοθήκες “τρέχουν” στον πυρήνα του Linux. Το επίπεδο αυτό προσφέρει τα απαραίτητα στοιχεία ώστε να είναι δυνατή η υλοποίηση του Android Runtime, του βασικού μηχανισμού, δηλαδή, που απαιτείται για την εκτέλεση των εφαρμογών που αναπτύσσονται για το περιβάλλον του Android.

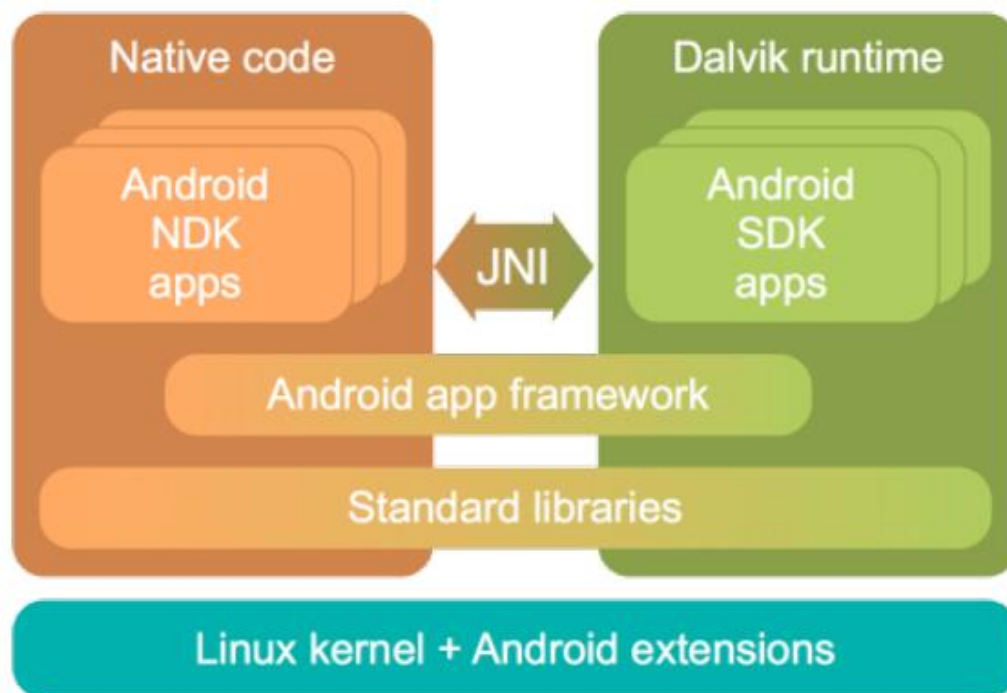
Το Android Runtime περιλαμβάνει την εικονική μηχανή (Virtual Machine) που διερμηνεύει τις εφαρμογές Android καθώς και την υλοποίηση των βασικών βιβλιοθηκών της Java που προσφέρονται στον προγραμματιστή κατά την διαδικασία ανάπτυξης των δικών του εφαρμογών.

Βασιζόμενο στο Android Runtime και στις βασικές βιβλιοθήκες του συστήματος, το προτελευταίο επίπεδο είναι το ονομαζόμενο Application Framework. Το επίπεδο αυτό προσφέρει στον προγραμματιστή μια πληθώρα δυνατοτήτων σχετιζόμενων με το λειτουργικό σύστημα και την συσκευή που το φιλοξενεί στο σύνολό τους, όπως πρόσβαση στα διάφορα περιφερικά της συσκευής, πρόσβαση σε λειτουργικότητες του ίδιου του λειτουργικού και άλλα.

Τέλος, το ανώτερο επίπεδο της στοίβας είναι το επίπεδο όπου βρίσκονται οι εφαρμογές για το περιβάλλον του Android. Σε αυτό το επίπεδο αναπτύσσονται οι εφαρμογές που περιγράφονται στο παρόν κείμενο και εδώ μπορούν να αξιοποιηθούν όλες οι δυνατότητες που μας προσφέρουν τα δυο προαναφερθέντα επίπεδα.

Το Android NDK Οι εφαρμογές που απαιτούν υψηλότερες επιδόσεις από αυτές που μπορεί να προσφέρει ο διερμηνευτής Dalvik μπορούν αξιοποιήσουν τις δυνατότητες του Android NDK (Native Development Kit). Όταν μια εφαρμογή χρησιμοποιεί το Android NDK, το ένα τμήμα της (κυρίως η διεπαφή χρήστη και κάποια λογική της εφαρμογής) συνεχίζει να διερμηνεύεται από τον Dalvik και το άλλο τμήμα της εκτελείτε απευθείας στον επεξεργαστή που έχει στο υλικό της η συσκευή.

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

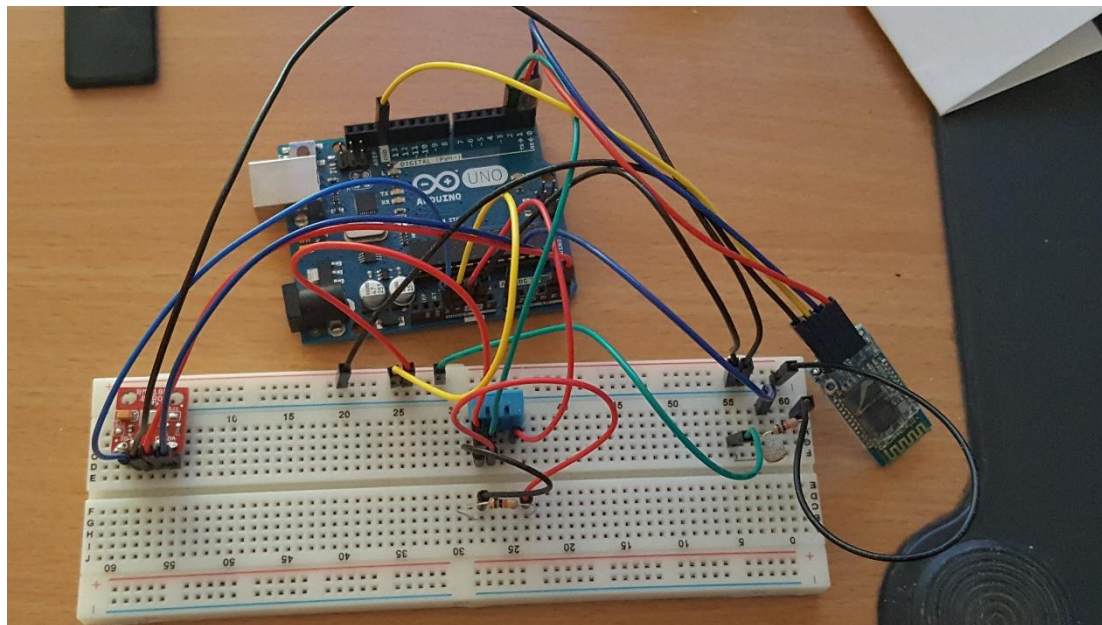


Εικόνα 3.6: Η αρχιτεκτονική του Android NDK

Τα τμήματα του κώδικα που εκτελούνται κατευθείαν στον επεξεργαστή μεταφράζονται σε βιβλιοθήκες και καλούνται από τα τμήματα του κώδικα που διερμηνεύονται από τον Dalvik μέσω του JNI (Java Native Interface) . Η βιβλιοθήκη μετά την μετάφρασή της πρέπει να συμπεριληφθεί στο .apk αρχείο της εφαρμογής και να φορτωθεί ρητά πριν χρησιμοποιηθεί. Εδώ αξίζει να παρατηρήσει κανείς ένα πιθανό πρόβλημα ασφάλειας. Ο κώδικας που εκτελείται στον επεξεργαστή δεν περνά από κανέναν από τους περιορισμούς για το ποιες λειτουργικότητες του συστήματος μπορεί να χρησιμοποιήσει. Ο κυριότερος λόγος για να χρησιμοποιήσει κανείς το Android NDK είναι για να επιτύχει καλύτερες επιδόσεις σε εφαρμογές που απαιτούν μεγάλο όγκο υπολογισμών (εξομοιώσεις κ.λπ.) ή εφαρμογές που κάνουν βαριά χρήση γραφικών (παιχνίδια κ.λπ.). Από δοκιμές που έχουν γίνει χρησιμοποιώντας μια εφαρμογή υπολογισμού των αριθμών της ακολουθίας Fibonacci έχει παρατηρηθεί μείωση του απαιτούμενου χρόνου εκτέλεσης έως και ενενήντα τοις εκατό! Οι μετρήσεις έγιναν σε συσκευή χωρίς JIT compiler διαθέσιμο και επειδή η ενσωμάτωση ενός τέτοιου μεταφραστή αυξάνει σημαντικά τις επιδόσεις του Dalvik, σε κάποιες περιπτώσεις πιθανότατα η χρήση του Android NDK να μην είναι απαραίτητη.[16]

ΚΕΦΑΛΑΙΟ 4

4.1 Μετεωρολογικός Σταθμός με Arduino



Εικόνα 4.1: Ο μετεωρολογικός μας σταθμός

4.1.1 Σκοπός

Σκοπός της παρούσης διπλωματικής εργασίας είναι η ανάπτυξη μιας συσκευής που συνδυάζει ένα αναπτυξιακό σύστημα(arduino) μαζί με αισθητήρες καθώς και μία εφαρμογή σε περιβάλλον Android τα οποία επικοινωνούν. Το αποτέλεσμα είναι μία “έξυπνη” εφαρμογή. Δημιουργήθηκε ένα μοντέλο, μικρού μεγέθους, μετεωρολογικού σταθμού με συγκεκριμένους αισθητήρες που προαναφέραμε στα παραπάνω κεφάλαια και αναπτυξιακό Arduino. Επίσης υλοποιήθηκε μια εφαρμογή σε λειτουργικό περιβάλλον Android ούτως ώστε να έχουμε γραφική απεικόνιση των αποτελεσμάτων των μετρήσεων του μετεωρολογικού σταθμού μας σε κάποια κινητή συσκευή (κινητό τηλέφωνο, tablet κτλ.). Η μεταξύ τους επικοινωνία πραγματοποιήθηκε μέσω μονάδας πομποδέκτη Bluetooth (HC-05). Απώτερος στόχος της όλης εφαρμογής είναι ο προγραμματισμός του Arduino, η συνδεσμολογία αισθητήρων και η μέτρηση-καταγραφή real-time τιμών που αφορούν μετεωρολογικά φαινόμενα, η δημιουργία εφαρμογής σε περιβάλλον Android, η ασύρματη επικοινωνία των δυο συσκευών, η αποστολή των δεδομένων των μετεωρολογικών τιμών καθώς και η γραφική παρουσίασή τους. Θα ήταν ελλιπές αν δεν αναφέραμε ότι η συγκεκριμένη εφαρμογή μπορεί να χρησιμοποιηθεί και από αγρότες για έλεγχο στα χωράφια τους κυρίως όταν

πρόκειται για μεγάλες εκτάσεις. Είναι γενικά παραδεκτό ότι μήνα με μήνα διαφορετικών ετών αλλά και μέρα με μέρα ακόμη και ώρα με ώρα τα χωράφια έχουν διαφορετικές απαιτήσεις όσων αφορά το πότισμα και γενικά την περιποίηση των εκτάσεων. Η καταγραφή των μετρήσεων υγρασίας , θερμοκρασίας αλλά και ηλιοφάνειας θα αποτελεί μια διευκόλυνση στην δουλειά τους.

4.1.2 Περιγραφή

Πιο συγκεκριμένα στο σύστημα χρησιμοποιήθηκαν τα εξής εξαρτήματα:

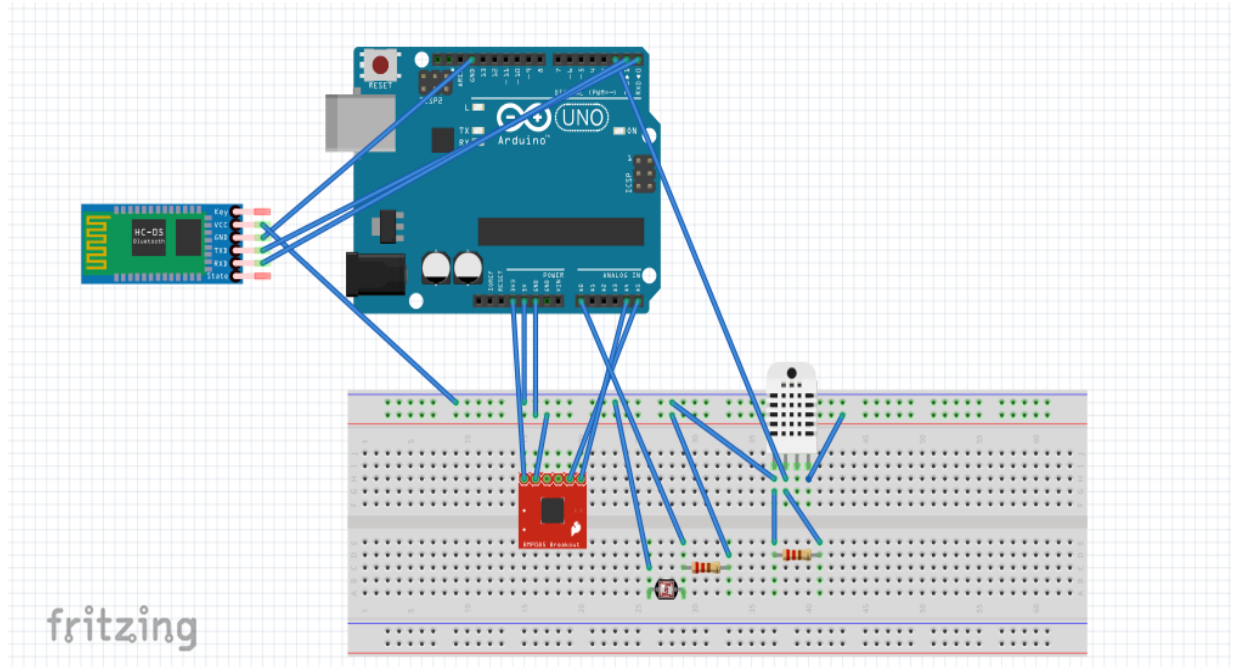
- Το αναπτυξιακό Arduino Uno
- Πλακέτα τοποθέτησης εξαρτημάτων (breadboard)
- Καλώδια συνδεσμολογίας
- Αισθητήρας Υγρασίας-Θερμοκρασίας DHT11
- Αισθητήρας Βαρομετρικής πίεσης BMP180
- Αισθητήρας φωτός μεταβλητής αντίστασης (photocell)
- Αντίσταση 10KΩ X 2
- Καλώδιο τροφοδοσίας

Οι αισθητήρες που χρησιμοποιήθηκαν είναι αρκετά χαμηλού κόστους καθώς και ενεργειακής κατανάλωσης χωρίς όμως αυτό να σημαίνει ότι δεν είναι ακριβής στις μετρήσεις.

Αφού οι αισθητήρες πάρουν τα δεδομένα, αυτά στέλνονται στο Arduino και αυτό με την σειρά του σύμφωνα με τον κώδικα μας τα στέλνει μέσω του Bluetooth module στην κινητή συσκευή Android που διαθέτουμε. Αυτό γίνεται τη στιγμή που, μέσω της εφαρμογής που έχουμε δημιουργήσει, ζητήσουμε αυτές τις μετρήσεις. Τέλος, οι τρέχουσες μετρήσεις εμφανίζονται στην οθόνη της κινητής συσκευής μας μέσα από το android app.

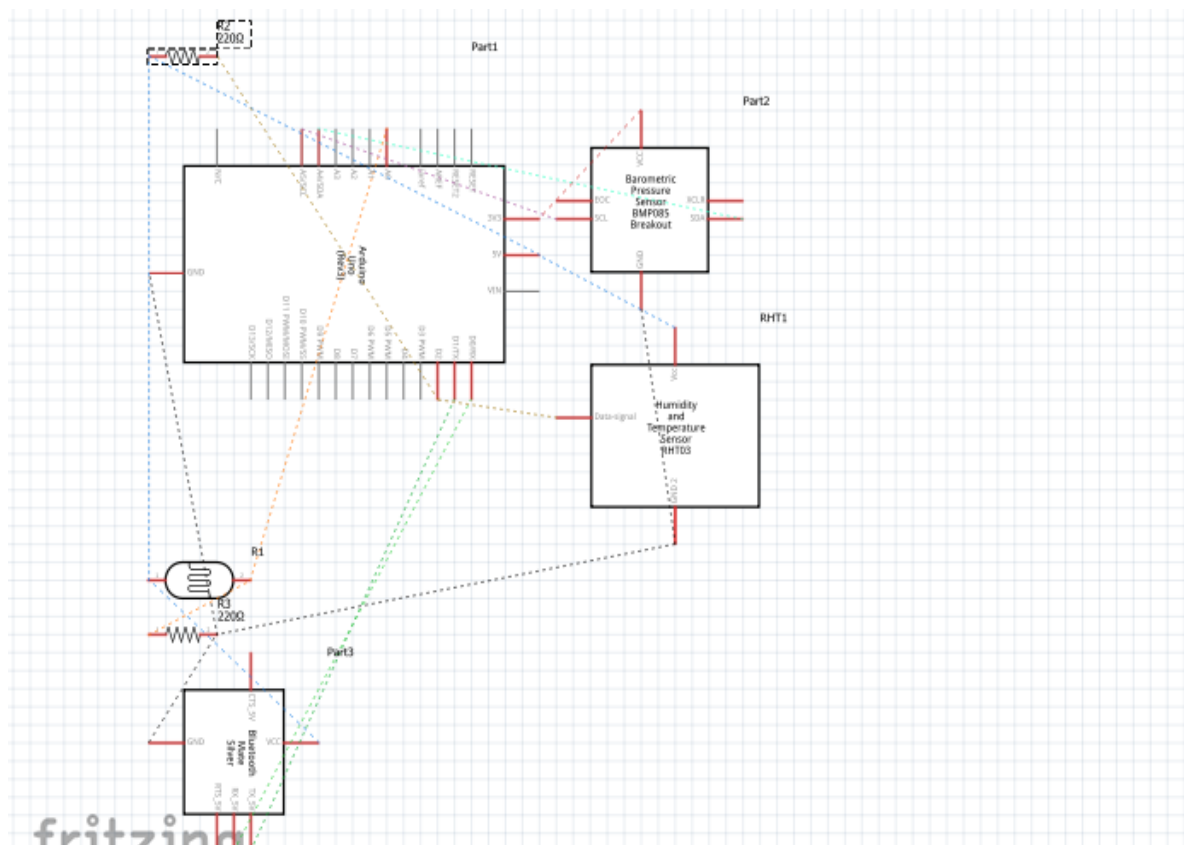
Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

4.1.3 Σχηματικό



Εικόνα 4.2: Προσομοίωση του κυκλώματος του μετεωρολογικού σταθμού με τους αισθητήρες.

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android



Εικόνα 4.3: Σχηματικό διάγραμμα ολόκληρου του κυκλώματος και των αισθητήρων του μετεωρολογικού σταθμού.

Όπως φαίνεται στο παραπάνω σχηματικό (Εικόνα 4.3), στην σειρά δύο συνδέουμε ένα καλώδιο από τον ακροδέκτη της γείωσης (GND) του Arduino. Έτσι κάθε φορά που θέλουμε να συνδέσουμε κάποιο εξάρτημα με γείωση θα ενώνουμε τον ακροδέκτη του εξαρτήματος με την σειρά δύο.

Ομοίως με την τροφοδοσία συνδέουμε ένα καλώδιο από τον ακροδέκτη 5V του Arduino με την σειρά ένα. Πάλι για τροφοδοσία των εξαρτημάτων θα ενώνουμε το εξάρτημα με την συγκεκριμένη σειρά. Αυτό γίνεται διότι υπάρχουν πολλοί αισθητήρες στο συγκεκριμένο κύκλωμα οι οποίοι χρειάζονται πάντα υποχρεωτικά τροφοδοσία και γείωση.

Αφού η τροφοδοσία και η γείωση έχουν συνδεθεί για κάθε εξάρτημα, καθένα από αυτά έχει ακροδέκτες και επικοινωνία με το Arduino. Συγκεκριμένα:

- Αισθητήρας υγρασίας/θερμοκρασίας: Τα δεδομένα του αισθητήρα είναι ψηφιακά σαφώς λοιπόν ο τρίτος ακροδέκτης (out) θα συνδεθεί σε μία από τις ψηφιακές υποδοχές 0-13 του αναπτυξιακού. Παράλληλα, μεταξύ των ακροδεκτών 1 και 4, έχει προστεθεί μια αντίσταση 10KΩ.
- Αισθητήρας βαρομετρικής πίεσης BMP180: Ο αισθητήρας χρησιμοποιεί το I2C πρωτόκολλο έτσι λοιπόν ενώνουμε τον ακροδέκτη SCL, που είναι

για τον χρονισμό, με τον αναλογικό ακροδέκτη A5 του Arduino και τον ακροδέκτη SDA που είναι για την μεταφορά δεδομένων στον αναλογικό ακροδέκτη A4 αντίστοιχα.

- Αισθητήρας Φωτός: Ο αισθητήρας φωτός ενώθηκε με τον αναλογικό ακροδέκτη A0 αφού πρώτα χρησιμοποιήθηκε μια αντίσταση 10KΩ

4.1.4 Προγραμματισμός

Για τον προγραμματισμό του συστήματος χρειαζόμαστε το λογισμικό της εταιρίας του Arduino IDE.

Στην αρχή του προγράμματος γίνονται κάποιες δηλώσεις βιβλιοθηκών, μεταβλητών και άλλες λειτουργίες. Στην συνέχεια το πρόγραμμά μας χωρίζεται σε δύο απλούστερες συναρτήσεις:

- I. Void Setup(): Η συνάρτηση καλείται μόλις το πρόγραμμα τρέχει για να αρχικοποιήσει μεταβλητές, ακροδέκτες, να χρησιμοποιηθούν οι βιβλιοθήκες. Αυτή η συνάρτηση τρέχει μόνο μια φορά και θα επαναληφθεί όταν επανεκκινήσουμε την συσκευή και τρέχει από την αρχή.
- II. Void Loop(): Η συνάρτηση καλείται αφού η setup έχει καλεστεί και ολοκληρωθεί, και πραγματοποιεί συνέχεια επαναλήψεις στον κώδικα που έχει αυτή δίνοντας την δυνατότητα στο πρόγραμμα να τρέχει συνεχώς και να δίνει ή παίρνει δεδομένα.

Ξεκινώντας:

- Ο κώδικας (sketch όπως ονομάζονται τα προγράμματα στο Arduino) όπως αναφέρθηκε ξεκινάει με την εισαγωγή των βιβλιοθηκών για κάθε ένα εξάρτημα ή λειτουργία που χρησιμοποιήσουμε.

```
#include <SFE_BMP180.h> //βιβλιοθήκη αισθητήρα βαρομετρικού
```

```
#include <DHT.h> //βιβλιοθήκη αισθητήρα υγρασίας
```

```
#include <Wire.h> //βιβλιοθήκη για την λειτουργία του I2C
```

Με την λέξη #include εισάγουμε βιβλιοθήκη βάζοντάς την σε εισαγωγικά ή αποσιωπητικά όταν αυτή βρίσκεται σε αρχείο. Οι βιβλιοθήκες είτε παρέχονται

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

από το λογισμικό για τις πιο γνωστές λειτουργίες είτε υπάρχουν στο διαδίκτυο για κάθε ένα από τα επιπρόσθετα υλικά χρησιμοποιούμε.

Πριν από τις βιβλιοθήκες, έχουμε δηλώσει 4 μεταβλητές δεκαδικού τύπου οι οποίες θα μας χρειαστούν στα πλαίσια του προγράμματος και αφορούν τη βαθμονόμηση (calibration) των αισθητήρων θερμοκρασίας, υγρασίας και πίεσης

```
float tempcorrection = 0; //temperature sensor correction(calibration) for
```

```
float tempcorrection2 = 0; //temperature sensor correction(calibration)
```

```
float humidcorrection = 0; //humidity sensor correction(calibration)
```

```
float presscorrection = 0; //pressure sensor correction(calibration)
```

- Στη συνέχεια δηλώνονται οι μεταβλητές που θα χρειαστούν στο πρόγραμμα:

```
#define DHTPIN 3 //ο ακροδέκτης του DHT11 ενώνεται με τον ψηφιακό ακροδέκτη 3 του Arduino
```

```
#define DHTTYPE DHT11 //χρησιμοποιείται τον DHT11 και όχι κάποιον άλλο τύπο
```

```
#define ALTITUDE 195.0 //Ορίζεται υψόμετρο που χρειάζεται σαν όρισμα για τον υπολογισμό του βαρομετρικού
```

```
SFE_BMP180 pressureSensor; // δηλώση του αισθητήρα BMP180
```

```
DHT dht(DHTPIN, DHTTYPE); //συνάρτηση DHT11 για ορισμό τύπου και σε ποιον ακροδέκτη βρίσκεται.
```

```
int LDR_Pin = A0;
```

- Αφού τελείωσαν οι δηλώσεις και αρχικοποιήσεις το πρόγραμμα συνεχίζει με τη συνάρτηση Void Setup():

```
void setup() //καλείται η συνάρτηση  
{
```

```
Serial.begin(9600); //ξεκινάει σειριακή επικοινωνία μεταξύ υπολογιστή και Arduino στα 9600 bits ανά δευτερόλεπτο.
```

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

```
dht.begin(); //έναρξη αισθητήρα dht
if (pressure.begin()){ // έναρξη του αισθητήρα βαρομετρικής πίεσης
}
else{
  Serial.println("BMP180 init fail\n\n"); // Σε διαφορετική περίπτωση θα
  εμφανίζεται μήνυμα ότι ο αισθητήρας απέτυχε να εκκινήσει.
  while(1); // Pause forever. // παύση
```

- Έπειτα ξεκινά η Void Loop() η οποία καταλαμβάνει και το μεγαλύτερο κομμάτι κώδικα. Υλοποιείται συνεχώς αφού παίρνει τις μετρήσεις και τις εμφανίζει.

```
void loop() // εκκίνηση της void loop()
{
  char c; // Δήλωση της μεταβλητής c τύπου χαρακτήρα.
  if(Serial.available())
  {
    c = Serial.read(); // αποθήκευση στην μεταβλητή c της τιμής που διαβάζεται
    σειριακά από την αναλογική είσοδο.
    if(c=='t'){ // Έλεγχος για τον αν ο χαρακτήρας που διαβάζεται είναι ο t.
      readSensor(); // Σε αυτή την περίπτωση διαβάζονται οι μετρήσεις από τον
      αισθητήρα DHT.
    }else if(c=='i'){ // Έλεγχος για τον αν ο χαρακτήρας που διαβάζεται είναι ο i.
      readIntensity(); // Σε αυτή την περίπτωση διαβάζονται οι μετρήσεις από τον
      αισθητήρα που μετρά την ένταση του φωτός.
    }else if (c=='p'){ // Έλεγχος για τον αν ο χαρακτήρας που διαβάζεται είναι ο
    p.
      BMP180(); // Σε αυτή την περίπτωση διαβάζονται οι μετρήσεις από τον
      αισθητήρα που μετρά την βαρομετρική πίεση (BMP180).
    }else if(c=='a'){ // // Έλεγχος για τον αν ο χαρακτήρας που διαβάζεται είναι ο
    a.
```

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

```
    readSensor();  
  
    readIntensity(); // Σε αυτή την περίπτωση διαβάζονται οι μετρήσεις από  
    όλους τους αισθητήρες.  
  
    BMP180();  
  
}else{  
  
    Serial.println("Failed to read sensors!"); // Τέλος σε οποιαδήποτε άλλη  
    περίπτωση εμφανίζεται μήνυμα σφάλματος ανάγνωσης τιμών από τους  
    αισθητήρες.  
  
    }  
  
}  
  
}
```

- Στο παρακάτω κομμάτι κώδικα δημιουργείται η συνάρτηση void readSensor με την οποία υπολογίζονται οι τιμές της θερμοκρασίας, της υγρασίας και του δείκτη θερμότητας.

```
void readSensor() { // καλείται η συνάρτηση void loop()  
  
    float h = dht.readHumidity(); // δηλώνεται η δεκαδική μεταβλητή h στην οποία  
    θα αποθηκεύεται η τιμή που επιστρέφει η συνάρτηση που μετρά την υγρασία  
    από τον αισθητήρα DHT11.  
  
    float t = dht.readTemperature(); // ομοίως για την μεταβλητή t στη οποία θα  
    αποθηκεύεται η τιμή της θερμοκρασίας.  
  
    if (isnan(h) || isnan(t)) {  
  
        Serial.println("Failed to read from DHT sensor!");  
  
        return;  
  
    } // Εντολές εμφάνισης μηνύματος σε περίπτωση που δεν μπορούν να  
    διαβαστούν μετρήσεις από τον αισθητήρα DHT.  
  
    float hic = dht.computeHeatIndex(t, h, false); // Δήλωση της μεταβλητής hic. Σε  
    αυτή τη μεταβλητή θα αποθηκεύονται οι τιμές της συνάρτησης  
    computeHeatIndex(). Η συνάρτηση αυτή υπολογίζει τον δείκτη θερμότητας  
    συνδυάζοντας την τρέχουσα τιμή θερμοκρασίας με την τρέχουσα τιμή της  
    υγρασίας που λαμβάνονται από τον αισθητήρα DHT.  
  
    Serial.print("Humidity: ");  
  
    Serial.print(h);  
  
    Serial.print(" %t");
```


Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

```
Serial.print("Temperature: ");
```

```
Serial.print(t);
```

```
Serial.print(" *C ");
```

```
Serial.print("Heat index: ");
```

```
Serial.print(hic);
```

```
Serial.print(" *C ");
```

// Σειρά εντολών παρουσίασης μηνυμάτων για την εμφάνιση των τιμών της υγρασίας, της θερμοκρασίας και του δείκτη θερμότητας.

- Στο παρακάτω κομμάτι κώδικα δημιουργείται η συνάρτηση `readIntensity` με την οποία υπολογίζεται η τιμή της έντασης του φωτός.

```
void readIntensity(){ // εκκίνηση της συνάρτησης readIntensity
```

```
int L = analogRead(LDR_Pin); // δήλωση της μεταβλητής L. Σε αυτήν θα αποθηκεύεται η τιμή της έντασης του φωτός η οποία έχει διαβαστεί από τον φωτοαισθητήρα μας.
```

```
Serial.print("Intensity: ");
```

```
Serial.print(L);
```

```
Serial.println(" LUX \t"); // Εντολές με τις οποίες εμφανίζεται με μήνυμα η τιμή της φωτεινότητας.
```

```
}
```

- Στο παρακάτω κομμάτι κώδικα δημιουργείται η συνάρτηση `BMP180` με την οποία υπολογίζεται της βαρομετρικής πίεσης.

```
void BMP180(){ // εκκίνηση της συνάρτησης BMP180.
```

```
char status; // δήλωση της μεταβλητής status τύπου χαρακτήρα
```

```
double T,P; // δήλωση των μεταβλητών T,P
```

```
status = pressure.startTemperature(); // εκκίνηση της συνάρτησης startTemperture και αποθήκευση των τιμών που επιστρέφει στην μεταβλητή status
```

```
if (status != 0)
```

```
{
```

```
    // αναμονή για την ολοκλήρωση των μετρήσεων
```

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

```
delay(status);

// ανάκτηση της ολοκληρωμένης μέτρησης της θερμοκρασίας
// παρατηρούμε ότι η μέτρηση καταχωρείται στην μεταβλητή T
// η συνάρτηση επιστρέφει την τιμή 1 όταν είναι επιτυχημένη η εκτέλεσή της
// και 0 σε περίπτωση σφάλματος.

status = pressure.getTemperature(T);
if (status != 0)
{
    // εκτύπωση της μέτρησης
    /* Serial.print("Temperature: ");
    Serial.print(T + tempcorrection,2);
    Serial.println(" deg C"); */

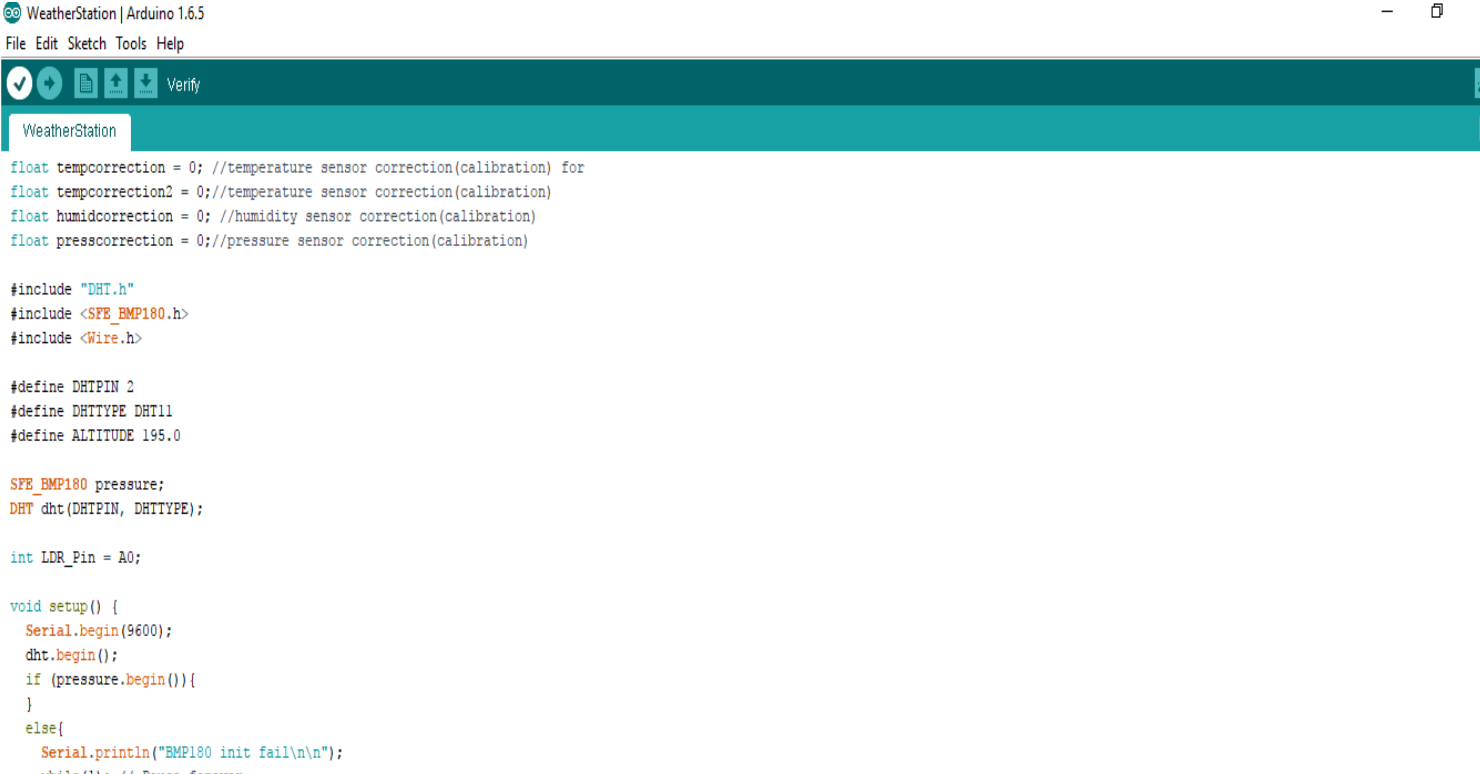
    status = pressure.startPressure(3);
    if (status != 0)
    {
        // αναμονή για την ολοκλήρωση της μέτρησης:
        delay(status);

        status = pressure.getPressure(P,T);
        if (status != 0)
        {
            // εκτύπωση της μέτρησης:
            Serial.print("Absolute pressurre: "); // εμφάνιση της τιμής της
            ατμοσφαιρικής πίεσης
            Serial.print(P + presscorrection,2);
            Serial.println(" mb");
        }
    }
}
```

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

```
    else Serial.println("error retrieving pressure measurement\n");
  }
  else Serial.println("error starting pressure measurement\n");
}
else Serial.println("error retrieving temperature measurement\n");
}
else Serial.println("error starting temperature measurement\n");
} // εκτύπωση μηνυμάτων σφάλματος σε περίπτωση προβλήματος στην
ανάκτηση των τιμών των μετρήσεων της θερμοκρασίας και της ατμοσφαιρικής
πίεσης.
```

Έτσι το πρόγραμμα είναι έτοιμο για να τρέξει στο Arduino. Πρώτα γίνεται η απαραίτητη αποσφαλμάτωση (debug). Η λειτουργία του debug γίνεται μέσα στο λογισμικό arduino ide πατώντας το κουμπί της επικύρωσης στα αριστερά της γραμμής εργαλείων.



```
WeatherStation | Arduino 1.6.5
File Edit Sketch Tools Help
WeatherStation
float tempcorrection = 0; //temperature sensor correction(calibration) for
float tempcorrection2 = 0; //temperature sensor correction(calibration)
float humidcorrection = 0; //humidity sensor correction(calibration)
float presscorrection = 0; //pressure sensor correction(calibration)

#include "DHT.h"
#include <SFE_BMP180.h>
#include <Wire.h>

#define DHTPIN 2
#define DHTTYPE DHT11
#define ALTITUDE 195.0

SFE_BMP180 pressure;
DHT dht(DHTPIN, DHTTYPE);

int LDR_Pin = A0;

void setup() {
  Serial.begin(9600);
  dht.begin();
  if (pressure.begin()){
  }
  else{
    Serial.println("BMP180 init fail\n\n");
  }
}
```

Εικόνα 4.4: Το πρόγραμμά μας έτοιμο για να το τρέξουμε

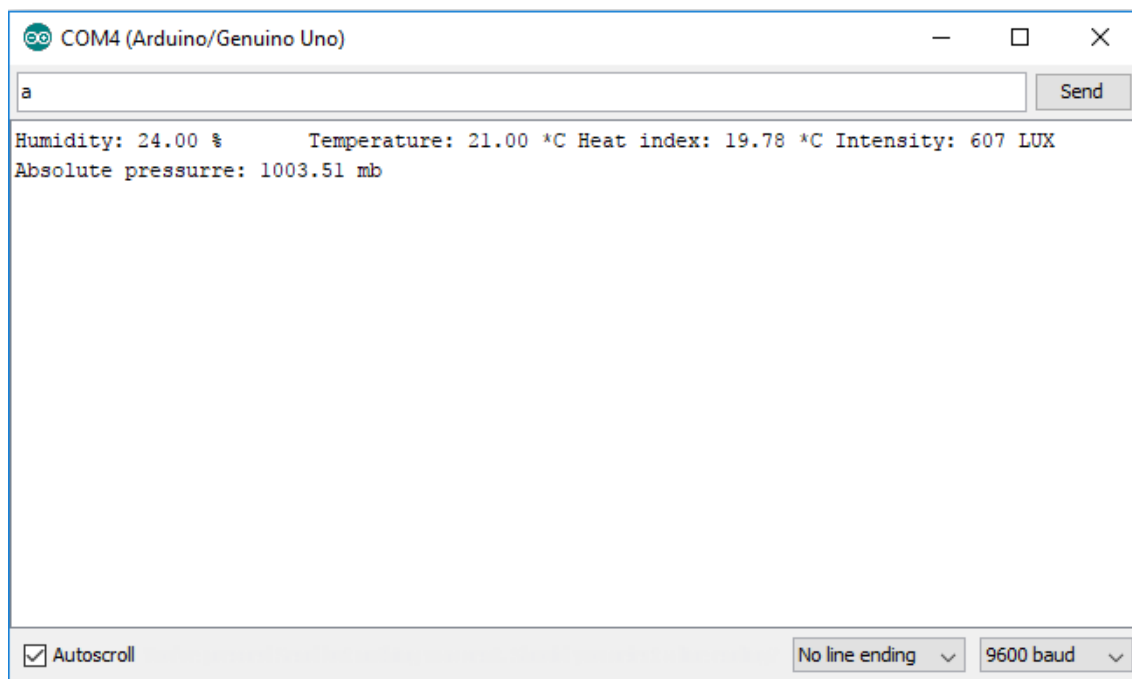
Το debug εκτελείται και εφόσον δεν υπάρχουν συντακτικά λάθη εμφανίζεται μήνυμα ολοκλήρωσης και πληροφοριών σχετικά με το πρόγραμμα.

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

```
Done compiling.  
Sketch uses 11,252 bytes (34%) of program storage space. Maximum is 32,256 bytes.  
Global variables use 831 bytes (40%) of dynamic memory, leaving 1,217 bytes for local variables. Maximum is 2,048 bytes.
```

Εικόνα 4.5: Το αποτέλεσμα του debug

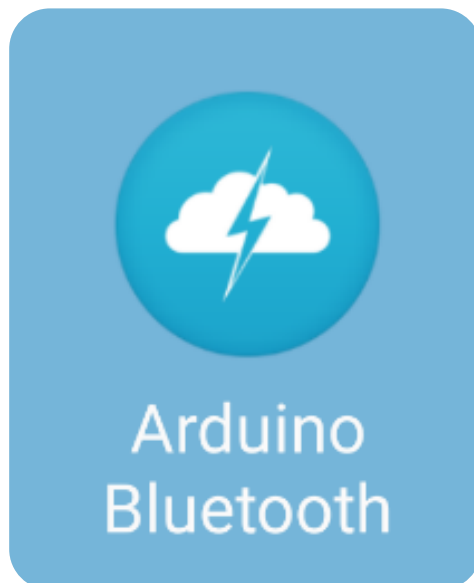
Αφού περάσουμε το πρόγραμμα στο Arduino, το σύστημα είναι έτοιμο να τρέξει. Κάθε φορά που το τοποθετείτε στο ρεύμα παίρνει μετρήσεις και καταγράφονται στην κάρτα μνήμης. Το σύστημα μπορεί να είναι συνδεδεμένο είτε με τον υπολογιστή και ταυτόχρονα να εμφανίζει τις μετρήσεις στην οθόνη αλλά και να τις καταγράφει είτε να παίρνει τροφοδοσία από εξωτερική πηγή και απλά να καταγράφει τις τιμές.



Εικόνα 4.6: Τα αποτελέσματα που μας επιστρέφει στο serial monitor

ΚΕΦΑΛΑΙΟ 5

5.1 Εφαρμογή Android για τον μετεωρολογικό σταθμό



Εικόνα 5.1: Το λογότυπο της εφαρμογής που δημιουργήσαμε

5.1.1 Κύρια στοιχεία μιας εφαρμογής Android

Υπάρχουν 3 σημαντικά αρχεία σε μια εφαρμογή Android:

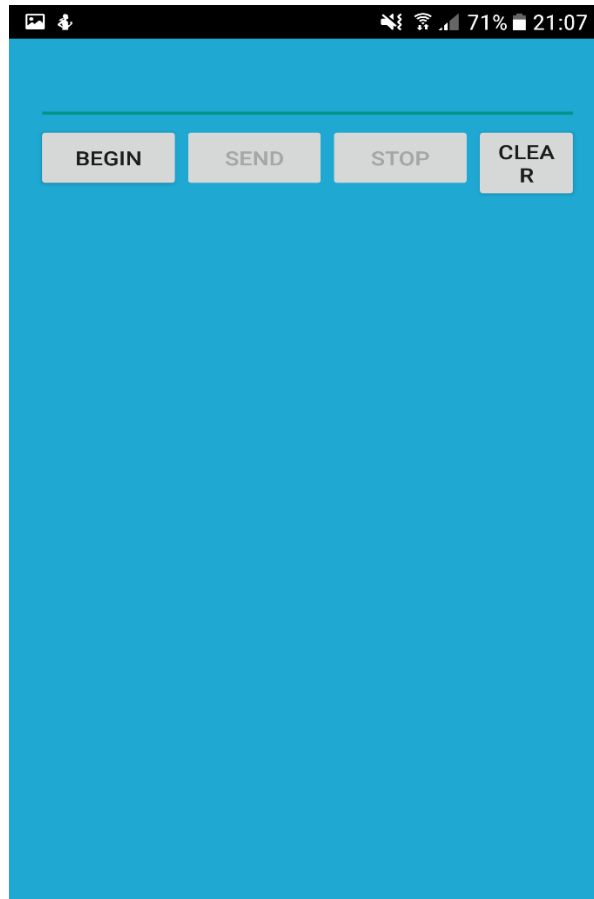
- **MainActivity.java** : Σε αυτό το αρχείο γράφεται και τοποθετείται ο κώδικας της Java. Ελέγχει τον τρόπο λειτουργίας της εφαρμογής.
- **activity_main.xml** : Περιέχει τη διάταξη της εφαρμογής, δηλαδή τα στοιχεία ή τα κουμπιά τύπου widget, Text Views κ.λπ.
- **AndroidManifest.xml** : Εδώ καθορίζεται πότε πρέπει να ξεκινήσει η εφαρμογή, ποια δικαιώματα χρειάζεται και ποιο υλικό χρειάζεται να έχει πρόσβαση σε αυτήν.

Υπάρχουν πολλά άλλα αρχεία, αλλά όλα συνδέονται μαζί με τη βοήθεια αυτών των τριών. Μια δραστηριότητα μπορεί να περιγραφεί ως μια οθόνη στην οποία ο χρήστης αλληλεπιδρά με το τηλέφωνο. Οι δραστηριότητες περιλαμβάνουν widgets όπως κουμπιά, πεδία κειμένου, εικόνες κλπ., τα οποία βοηθούν στη μεταφορά των πληροφοριών. Η εφαρμογή που δημιουργήσαμε χρησιμοποιεί μία δραστηριότητα, την κύρια δραστηριότητα

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

(MainActivity), η οποία παίρνει την είσοδο του χρήστη για να στείλει στο Arduino και επίσης εμφανίζει το ληφθέν κείμενο. [17]

5.1.2 Η διάταξη



Εικόνα 5.2: Η αρχική οθόνη της εφαρμογής μας

Είναι απλό και με τα ελάχιστα απαιτούμενα γραφικά στοιχεία για τη σύνδεση μεταξύ των συσκευών. Όπως φαίνεται, έχει ένα γραφικό στοιχείο EditText για εισαγωγή από τον χρήστη, κουμπιά για να ξεκινήσει η σύνδεση, να μεταδοθούν τα δεδομένα, να τερματιστεί η σύνδεση και το TextView να μείνει κενό. Τα παραληφθέντα δεδομένα εμφανίζονται στο TextView (είναι το κενό μέρος κάτω από τα κουμπιά). Στη συνέχεια παρατίθεται ένα μέρος του XML κώδικα.

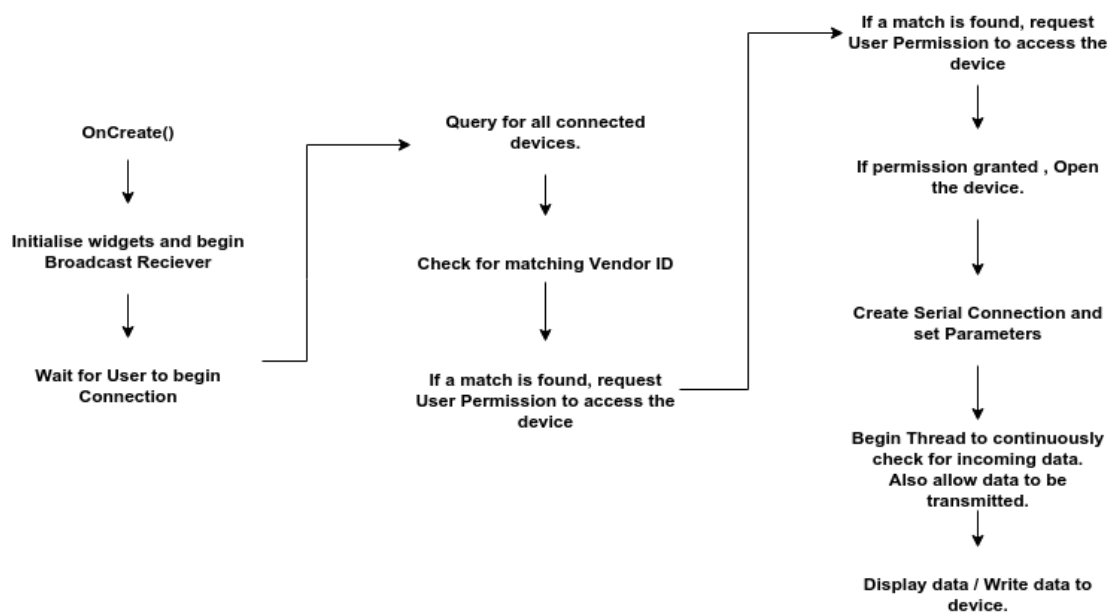
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent" android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
```

```
android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".MainActivity"> <EditText android:layout_width="wrap_content"
android:layout_height="wrap_content" android:id="@+id/editText"
android:layout_alignParentTop="true" android:layout_alignParentRight="true"
android:layout_alignParentEnd="true" android:layout_alignParentLeft="true"
android:layout_alignParentStart="true" /> <Button
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="Begin" android:id="@+id/buttonStart"
android:layout_below="@+id/editText" android:layout_alignParentLeft="true"
android:layout_alignParentStart="true" android:onClick="onClickStart"/>
<TextView android:layout_width="wrap_content"
android:layout_height="wrap_content" android:id="@+id/textView"
android:layout_below="@+id/buttonSend"
android:layout_alignParentLeft="true" android:layout_alignParentStart="true"
android:layout_alignRight="@+id/editText"
android:layout_alignEnd="@+id/editText"
android:layout_alignParentBottom="true" /> </RelativeLayout>
```

Έχουμε χρησιμοποιήσει ένα RelativeLayout εδώ, που σημαίνει ότι κάθε widget είναι διατεταγμένο σε σχέση με τα widgets γύρω από αυτό. Η διάταξη μπορεί να αναπαραχθεί εύκολα χρησιμοποιώντας την καρτέλα Σχεδίαση, όπου μπορείτε να μεταφέρετε και να αποθέσετε τα widget όπου θέλετε. Όταν κάνετε κλικ σε ένα κουμπί, θα πρέπει να περιγράψουμε τι πρέπει να συμβεί. Για αυτό, χρησιμοποιείται μια μέθοδος onClick. [17]

5.1.3 Η ροή του προγράμματος

Αυτή είναι μια σύντομη περιγραφή του τρόπου με τον οποίο θα προχωρήσουμε. Κάθε δραστηριότητα έχει μια μέθοδο onCreate () που είναι η εκτέλεση κατά τη δημιουργία της δραστηριότητας. Οποιοσδήποτε κώδικας θέλουμε να τρέξουμε στην αρχή πρέπει να τοποθετηθεί μέσα σε αυτό. Σημειώνουμε ότι η ανάγνωση από τη συσκευή είναι ασύγχρονη, πράγμα που σημαίνει ότι θα συνεχίσει να εκτελείται στο παρασκήνιο. Αυτό γίνεται ώστε τα δεδομένα να λαμβάνονται το συντομότερο δυνατό.



Εικόνα 5.3: Η ροή του προγράμματός μας

5.1.4 Άνοιγμα μιας σύνδεσης

Αρχικά καθορίζουμε την μέθοδο onClick για το κουμπί Begin. Όταν κάνουμε κλικ, θα πρέπει να μπορούμε να αναζητούμε όλες τις συνδεδεμένες συσκευές και, στη συνέχεια, να ελέγχουμε αν το αναγνωριστικό προμηθευτή του Arduino ταιριάζει με εκείνο μιας συνδεδεμένης συσκευής. Αν βρεθεί, πρέπει να ζητηθεί άδεια από το χρήστη.

```
public void onClickStart(View view) {  
    if(BTinit())  
    {  
        if(BTconnect())  
        {
```



```
        setUiEnabled(true);
        deviceConnected=true;
        beginListenForData();
        textView.append("\nConnection Opened!\n");
        textView.append("\n");
    }
}
}
```

5.1.5 Επεξεργασία του AndroidManifest.xml

Δεδομένου ότι το πρόσθετο υλικό που αυτή η εφαρμογή θα χρησιμοποιήσει είναι ο ενσωματωμένος προσαρμογέας Bluetooth, θα πρέπει να αναφέρεται στο αρχείο Manifest.xml

```
uses-permission android:name="android.permission.BLUETOOTH" />
```

5.1.6 Σύνδεση με τη συσκευή

Ελέγχουμε πρώτα για την παρουσία ενός προσαρμογέα Bluetooth.

```
BluetoothAdapter
bluetoothAdapter=BluetoothAdapter.getDefaultAdapter();    if
(bluetoothAdapter == null)
{    Toast.makeText(getApplicationContext(),"Device doesnt Support
Bluetooth",Toast.LENGTH_SHORT).show();    }
```

Εάν αναγνωρισθεί, ελέγχουμε αν είναι ενεργοποιημένη. Εάν δεν είναι ενεργοποιημένη, ζητάμε από τον χρήστη την άδεια να την ενεργοποιήσει.

```
if(!bluetoothAdapter.isEnabled())
{
    Intent enableAdapter = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(enableAdapter, 0);
    try {
        Thread.sleep(1000);
    }
}
```

```
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
}
```

Η συσκευή πρέπει να έχει αντιστοιχιστεί πριν η εφαρμογή μπορεί να την χρησιμοποιήσει. Τώρα που ο προσαρμογέας είναι ενεργοποιημένος, ελέγχουμε για τις συζευγμένες / συνδεδεμένες συσκευές. Χρησιμοποιώντας τις εντολές AT από την παραπάνω ενότητα, δίνουμε το κατάλληλο όνομα στη συσκευή. Επίσης, μαθαίνουμε την MAC διεύθυνση του. Εδώ, οι διευθύνσεις MAC συγκρίνονται για μία ταύτιση επειδή τα ονόματα ενδέχεται να αλλάζουν συχνά.[17]

```
Set<BluetoothDevice> bondedDevices =  
bluetoothAdapter.getBondedDevices();  
    if(bondedDevices.isEmpty())  
    {  
        Toast.makeText(getApplicationContext(), "Please Pair the Device  
first", Toast.LENGTH_SHORT).show();  
    }  
    else  
    {  
        for (BluetoothDevice iterator : bondedDevices)  
        {  
            if(iterator.getAddress().equals(DEVICE_ADDRESS))  
            {  
                device=iterator;  
                found=true;  
                break;  
            }  
        }  
    }  
    return found;  
}
```

Μετά τη λήψη του BluetoothDevice, πρέπει να δημιουργηθεί μια υποδοχή για να χειριστεί την εξερχόμενη σύνδεση. Εδώ χρησιμοποιείται μια υποδοχή RFCOMM. Το RFCOMM - επίσης γνωστό ως προφίλ σειριακής θύρας - είναι ουσιαστικά ένα πρωτόκολλο Bluetooth για την προσομοίωση ενός καλωδίου RS232.

```
socket = device.createRfcommSocketToServiceRecord(PORT_UUID);  
socket.connect();
```

Στη συνέχεια, δημιουργούμε τις ροές εισόδου και εξόδου της υποδοχής.

```
outputStream=socket.getOutputStream();  
  
inputStream=socket.getInputStream();
```

5.1.7 Ανάγνωση εισερχόμενων δεδομένων

Επειδή ότι τα δεδομένα μπορούν να ληφθούν σε οποιαδήποτε χρονική στιγμή, η εκτέλεση ενός νήματος για ακρόαση δεδομένων θα ήταν καλύτερο. Καταρχάς, η ροή εισόδου αναζητείται για τα διαθέσιμα δεδομένα. Στη συνέχεια, τα byte μετατρέπονται σε μορφότυπο UTF-8 αναγνώσιμο από τον άνθρωπο και το κείμενο αποστέλλεται σε έναν χειριστή για να τοποθετηθεί στο περιβάλλον χρήστη. Αυτό γίνεται επειδή το περιβάλλον εργασίας χρήστη δεν μπορεί να ενημερωθεί από τα νήματα του φόντου.

```
void beginListenForData()  
{  
    final Handler handler = new Handler();  
    stopThread = false;  
    buffer = new byte[1024];  
    Thread thread = new Thread(new Runnable())  
    {  
        public void run()  
        {  
            while(!Thread.currentThread().isInterrupted() && !stopThread)  
            {  
                try  
                {  
                    int byteCount = inputStream.available();  
                    if(byteCount > 0)  
                    {  
                        byte[] rawBytes = new byte[byteCount];  
                        inputStream.read(rawBytes);  
                        final String string=new String(rawBytes, "UTF-8");  
                        handler.post(new Runnable() {  
                            public void run()                        })  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        {  
            textView.append(string);  
        }  
    });  
    }  
    }  
    catch (IOException ex)  
    {  
        stopThread = true;  
    }  
    }  
    }  
});  
thread.start();  
}
```

5.1.8 Μετάδοση δεδομένων

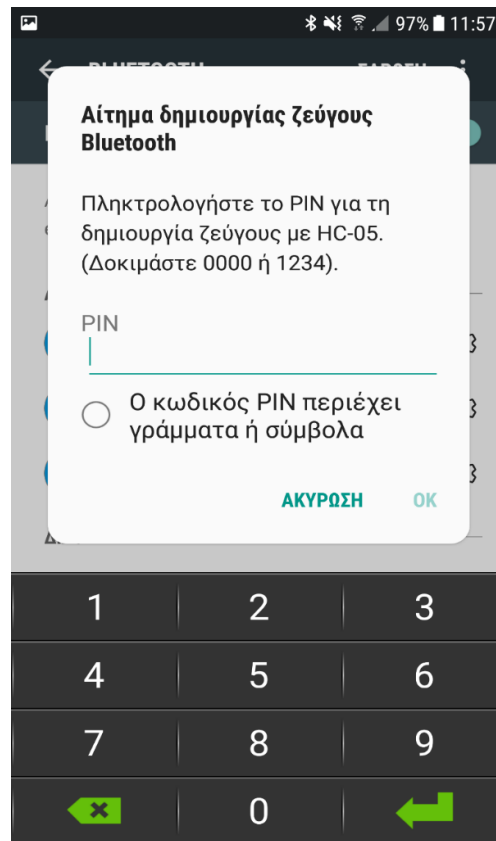
Για να στείλουμε δεδομένα, περνάμε το String στο OutputStream.

```
outputStream.write(string.getBytes());
```

5.1.9 Δοκιμή της εφαρμογής

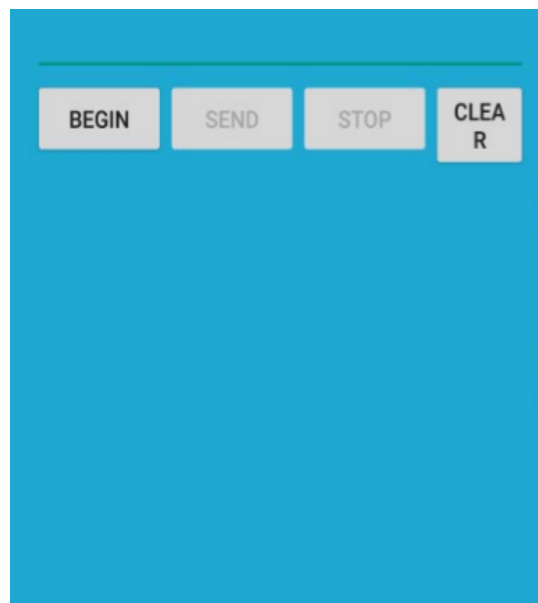
Πρώτα απενεργοποιούμε το ζεύγος του τηλεφώνου και της μονάδας σαρώνοντας νέες συσκευές στην καρτέλα Bluetooth των ρυθμίσεων συστήματος. Πρέπει να εμφανιστεί το όνομα που είχε οριστεί νωρίτερα. Τοποθετούμε τον καθορισμένο κωδικό πρόσβασης (Η προεπιλογή είναι είτε 0000 είτε 1234).

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android



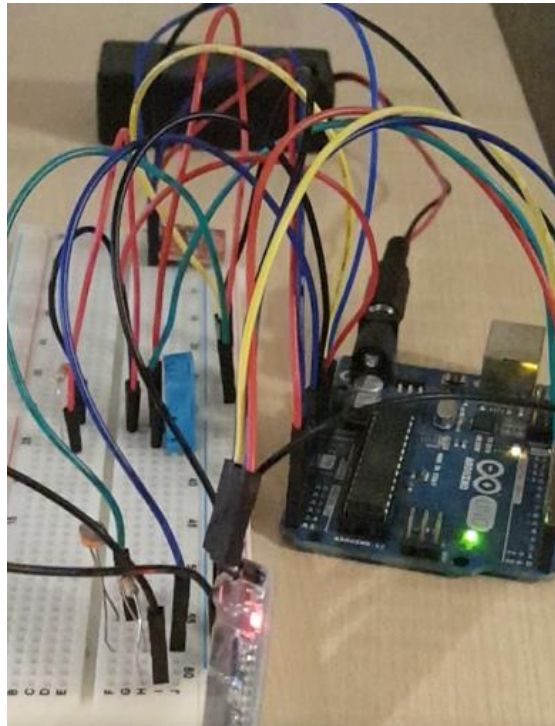
Εικόνα 5.4: Σύνδεση(pair) του κινητού με το Bluetooth του Arduino

Αφού ενεργοποιήσουμε το Arduino, ίσως παρατηρήσετε ότι το LED κατάστασης του Bluetooth αναβοσβήνει περιοδικά. Τώρα ανοίγουμε την εφαρμογή και πατάμε το κουμπί Begin. Η λυχνία κατάστασης πρέπει να σβήσει και η λυχνία LED Conn θα ανάψει. Αυτό μας δείχνει ότι έχει δημιουργηθεί μια σύνδεση.[17]



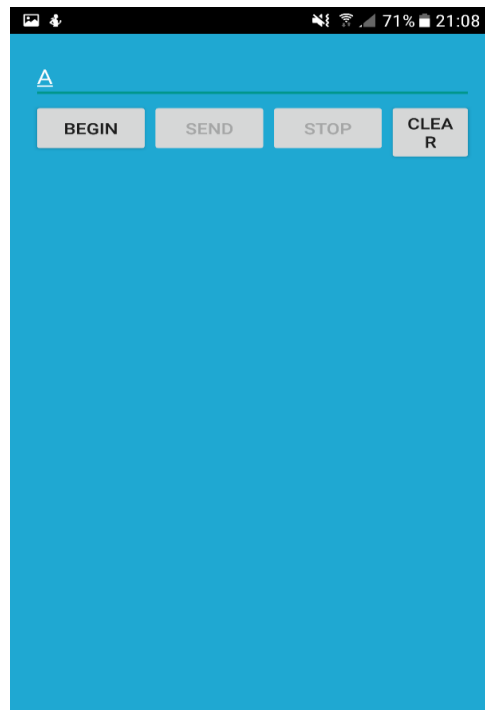
Εικόνα 5.5: Πατώντας το κουμπί Begin ανοίγει η σύνδεση

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android



Εικόνα 5.6: Η λυχνία κατάστασης όταν ανοίγει η σύνδεση

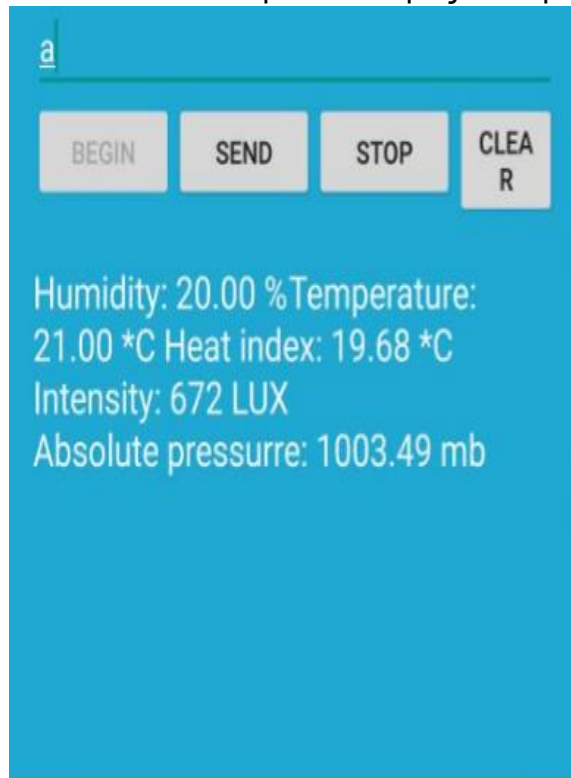
Όταν το "a" στέλνεται στο Arduino, αυτό μας απαντά στέλνοντας τις μετρήσεις της θερμοκρασίας, της υγρασίας, της ατμοσφαιρικής πίεσης, του δείκτη θερμότητας και της φωτεινότητας.



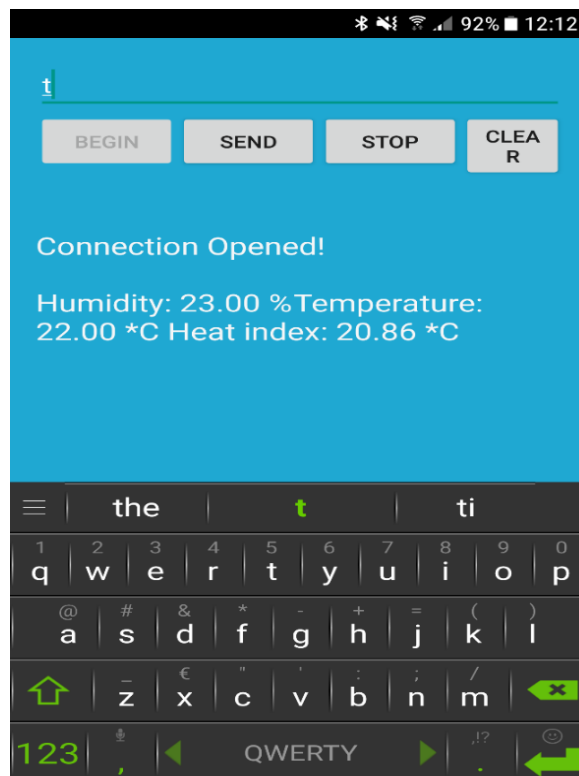
Εικόνα 5.7: Πατώντας το γράμμα A

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

Και εδώ τα αποτελέσματα που μας επιστρέφει:



Εικόνα 5.8: Τα αποτελέσματα όλων των αισθητήρων



Εικόνα 5.9: Τα αποτελέσματα μέτρησης του αισθητήρα DHT-11

ΚΕΦΑΛΑΙΟ 6

6.1 Κινητός Μετεωρολογικός Σταθμός

Σκοπός της εργασίας είναι ο μετεωρολογικός σταθμός που δημιουργήσαμε να μπορεί να μετακινηθεί ανά πάσα στιγμή ώστε να λαμβάνουμε δεδομένα και μετρήσεις από οποιοδήποτε σημείο βρισκόμαστε. Δηλαδή ο μετεωρολογικός μας σταθμός πρέπει να είναι κινητός. Αυτό επιτεύχθηκε με τη σύνδεση ενός αντάπτορα μπαταρίας 9V και μιας θήκης για την μπαταρία αυτή.



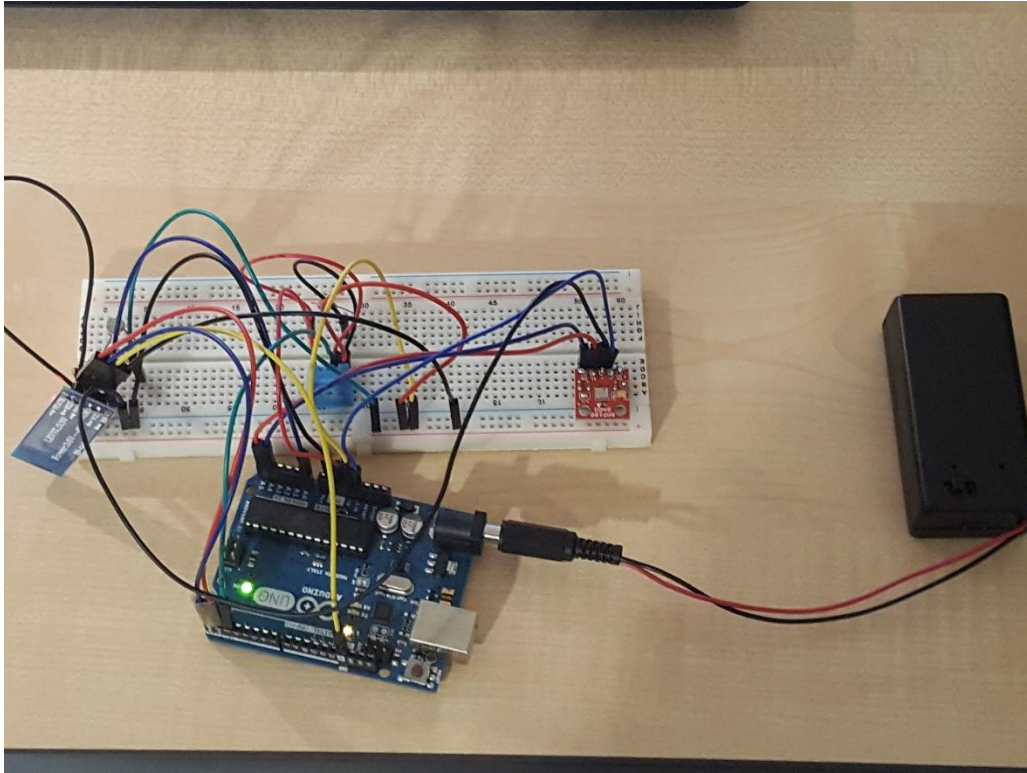
Εικόνα 6.1: Κουτί για μπαταρία 9V



Εικόνα 6.2: Αντάπτορας σύνδεσης Arduino με εξωτερική τροφοδοσία

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

Στη συνέχεια το συνδέσαμε στην θύρα τροφοδοσίας του Arduino Uno και έτσι ο ολοκληρωμένος μετεωρολογικός σταθμός μας λειτουργεί πια χωρίς την αναγκαιότητα σύνδεσης με κάποιο ηλεκτρονικό υπολογιστή. Τροφοδοτείται από την 9V μπαταρία και μας δίνει μετρήσεις από τους αισθητήρες στέλνοντας τις τιμές τους μέσω του Bluetooth στην Android εφαρμογή του κινητού μας.



Εικόνα 6.3: Ολόκληρο το κύκλωμά μας με εξωτερική τροφοδοσία

ΚΕΦΑΛΑΙΟ 7

7.1 Συμπεράσματα και Προοπτικές

7.1.1 Σύνοψη της διπλωματικής εργασίας

Ο στόχος της παρούσης Διπλωματικής Εργασίας ήταν η υλοποίηση ενός αυτόνομου μετεωρολογικού σταθμού για την παρακολούθηση και καταγραφή μετεωρολογικών φαινομένων καθώς και η ανάπτυξη μιας εφαρμογής σε περιβάλλον Android για αποστολή των μετρήσεων των αισθητήρων σε μια φορητή συσκευή. Το επίπεδο ευαισθησίας των αισθητήρων είναι σε ικανοποιητικό βαθμό, όπως και οι αλγόριθμοι καταγραφής και αποθήκευσης των δεδομένων. Παρατηρούμε ελάχιστες αποκλίσεις σε σχέση με τους επαγγελματικού τύπου αισθητήρες.

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

- α) Γλώσσα προγραμματισμού Wiring για την κατασκευή του κώδικα λειτουργίας της πλακέτας Arduino.
- β) Γλώσσα προγραμματισμού Android για την δημιουργία μιας εφαρμογής σε κάποια φορητή συσκευή ώστε να γίνεται απεικόνιση των μετρήσεων των αισθητήρων ανά πάσα χρονική στιγμή.

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

7.1.2 Προοπτικές

Η δημιουργία ενός αυτόνομου μετεωρολογικού σταθμού και η παρουσίαση των μετρήσεων των αισθητήρων σε μια φορητή συσκευή είναι μόνο η κορυφή του παγόβουνου καθώς οι δυνατότητες εξελίσσονται με τέτοιο τρόπο, που το μόνο πλέον που μπορεί να το περιορίσει είναι η ανθρώπινη φαντασία.

Πηγαίνοντας ένα βήμα πιο μπροστά, μια εξέλιξη της παρούσας διπλωματικής θα μπορούσε να είναι η αποτύπωση των τιμών των μετρήσεων σε χάρτη (Google Maps) ανάλογα με τη θέση που βρισκόμαστε. Δηλαδή θα μπορούσε να γίνει εντοπισμός της θέσης του χρήστη μέσω του gps του κινητού ή του tablet και καταχώρηση των μετρήσεων, ανάλογα με το που βρίσκεται, σε κάποιο χάρτη της εφαρμογής που χρησιμοποιεί η συσκευή(Google).

ΚΕΦΑΛΑΙΟ 8

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Wikipedia, "Weather Focasting", http://en.wikipedia.org/wiki/Weather_forecasting , (Ημερομηνία. Προσπέλασης 10/10/2017)
- [2] Wikipedia, "Μηχανές του καιρού", <https://el.wikipedia.org/wiki/> , (Ημερομηνία Προσπέλασης 10/10/2017)
- [3] Wikipedia, "Θερμοκρασία Ατμόσφαιρας", <http://el.wikipedia.org/wiki/> , (Ημερομηνία Προσπέλασης 12/10/2017)
- [4] Υπουργείο Εθνικής Παιδείας και Θρησκευμάτων Κέντρο Περιβαλλοντικής Εκπαίδευσης Μακρινίτσας, Εισαγωγή στη Μετεωρολογία, Μια εκπαιδευτική προσέγγιση Μέρος Α΄: Βασικές Έννοιες, σελ. 16, Δεκέμβριος 2007
- [5] Wikipedia, "Σημείο Δρόσου", <https://el.wikipedia.org/wiki/> , (Ημερομηνία Προσπέλασης 15/10/2017)
- [6] Wikipedia, "Heat Index", http://en.wikipedia.org/wiki/Heat_index/ , (Ημερομηνία Προσπέλασης 14/10/2017)
- [7] Wikipedia, "Το κλίμα της Ελλάδος", <https://el.wikipedia.org/wiki/> , (Ημερομηνία Προσπέλασης 16/10/2017)
- [8] Wikipedia, "Μετεωρολογικός σταθμός", <https://el.wikipedia.org/wiki/> , (Ημερομηνία Προσπέλασης 17/10/2017)
- [9] Northmeteo, <http://www.northmeteo.gr/bibliothiki/meteorologia/meteorologikoi-stathmoi>, (Ημερομηνία Προσπέλασης 18/10/2017)
- [10] World Meteorological Organization, "Guide to Meteorological Instruments and Methods of Observation", 7th edition, p.385, 2008
- [11] Arduino Επίσημη Ιστοσελίδα, www.arduino.cc (Ημερομηνία προσπέλασης 18/10/2017)
- [12] Ardupilot, <http://www.diydrones.com/notes/ArduPilot> , (Ημερομηνία Προσπέλασης 18/10/2017)
- [13] DHT11 Humidity & Temperature Sensor, D-Robotics UK, <http://www.micropik.com/PDF/dht11.pdf>, (Ημερομηνία προσπέλασης 18/10/2017)
- [14] <http://www.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf> , (Ημερομηνία προσπέλασης 22/10/2017)

Ανάπτυξη φορητού μετεωρολογικού σταθμού με Arduino και αποστολή των μετρήσεων σε συσκευή Android

[15] Wikipedia, "Photoresistor", <http://en.wikipedia.org/wiki/Photoresistor>, (Ημερομηνία προσπέλασης 22/10/2017)

[16] Γεώργιος Εμμανουήλ Καλλέργης, "Ανάπτυξη Εφαρμογών σε Περιβάλλον Android", Πανεπιστήμιο Πατρών Πολυτεχνική Σχολή, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Ιούνιος 2013

[17] Control an Arduino with Bluetooth, <https://www.allaboutcircuits.com/projects/control-an-arduino-using-your-phone/>, (Ημερομηνία προσπέλασης 9/9/2017)