



ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς  
μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας,  
ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)»

Του φοιτητή  
**ΣΤΑΥΡΟΥ ΑΝΔΡΕΑΔΗ**  
Αρ. Μητρώου: **511007**

Επιβλέπων  
Όνοματεπώνυμο: **Άγγελος Γιακουμής**  
Βαθμίδα: **Λέκτορας**

Ημερομηνία: **21/09/2021**

Τίτλος Δ.Ε.: Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

Κωδικός Δ.Ε.: 19124

Ονοματεπώνυμο φοιτητή: Σταύρος Ανδρεάδης

Ονοματεπώνυμο εισηγητή: Άγγελος Γιακουμής

Ημερομηνία ανάληψης Δ.Ε.: 15/11/2019

Ημερομηνία περάτωσης Δ.Ε.: 21/09/2021

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Σταύρου Ανδρεάδη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

## Πρόλογος

Η παρούσα εργασία εκπονήθηκε στα πλαίσια των προπτυχιακών σπουδών στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος. Η εργασία αυτή αποτελεί απαύγασμα προσωπικού ενδιαφέροντος προς τις κατασκευές που απαιτούν τη χρήση αισθητήρων και τη σύνδεση αυτών με το διαδίκτυο. Ο τομέας του Internet of Things (IoT) παρουσιάζει ιδιαίτερη θέση στα προσωπικά ενδιαφέροντα, με τη βλέψη της περαιτέρω ενασχόλησης με αυτό κατόπιν αποφοίτησης. Η εργασία αυτή, λοιπόν, αποτελεί ένα έναυσμα για τη μετέπειτα ενασχόληση αυτή. Μέσω του παρόντος εκπονήματος διαπιστώθηκε στην πράξη η κατασκευή ενός κυκλώματος, η προσωπική τριβή με το οποίο εμπλούτισε τις προσωπικές δεξιότητες και εμπειρία με το αντικείμενο.

## Περίληψη

Η παρούσα πτυχιακή εργασία πραγματεύεται την κατασκευή ενός κυκλώματος μετεωρολογικού σταθμού, το οποίο μέσω των αισθητήρων που διαθέτει στοχεύει στη συλλογή διάφορων μετεωρολογικών μετρήσεων και συγκεκριμένα της θερμοκρασίας, της υγρασίας, της ατμοσφαιρικής πίεσης και της ταχύτητας του ανέμου. Οι μετρήσεις των στοιχείων αυτών συλλέγονται και μεταφέρονται σε υπολογιστικό περιβάλλον. Ειδικότερα, στην εργασία αυτή γίνεται μία καταγραφή των εξαρτημάτων που χρησιμοποιήθηκαν στην κατασκευή του μετεωρολογικού κυκλώματος, στη συνέχεια παρατίθεται περιγραφή της διαδικασίας κατασκευής του κυκλώματος και έπειτα πραγματοποιείται δοκιμή του κυκλώματος με εμφάνιση των μετρήσεων στο περιβάλλον του υπολογιστή.

Από τα παραπάνω, λοιπόν, διαφαίνεται ο στόχος της εργασίας που είναι η παρουσίαση του τρόπου λειτουργίας ενός μετεωρολογικού οργάνου, μέσα από την κατασκευή και τη δοκιμή του. Από το εγχείρημα αυτό προκύπτει ότι με την απλή κατασκευή ενός τέτοιου κυκλώματος είναι δυνατή η ασφαλής μέτρηση δεδομένων που ανταποκρίνονται στις πραγματικές συνθήκες.

**Λέξεις – κλειδιά:** Κύκλωμα, μετεωρολογικό όργανο, θερμοκρασία, υγρασία, ατμοσφαιρική πίεση, ταχύτητα ανέμου.

# Design and construction of a circuit for the collection and transmission of meteorological measurements (temperature, humidity, atmospheric pressure, wind speed)

Stavros Andreadis

## Abstract

The present thesis discusses the construction of a meteorological station circuit, which through its sensors aims to collect various meteorological measurements, namely temperature, humidity, atmospheric pressure and wind speed. The measurements of these data are collected and transferred to a computer environment. In particular, in this work a record is made of the components used in the construction of the meteorological circuit, then a description of the construction process of the circuit is given and then a test of the circuit is performed by displaying the measurements in the computer environment.

From the above, then, the goal of the work is evident, which is the presentation of the operation of a meteorological instrument, through its construction and testing. From this project it follows that with the simple construction of such a circuit it is possible to safely measure data that correspond to the real conditions.

**Keywords:** Circuit, meteorological station, temperature, humidity, atmospheric pressure, wind speed.

## Πίνακας Περιεχομένων

<b>Πρόλογος</b> .....	3
<b>Περίληψη</b> .....	4
<b>Abstract</b> .....	5
<b>Ευχαριστίες</b> .....	7
<b>1. Εισαγωγή</b> .....	8
<b>2. Αισθητήρες</b> .....	9
2.1 Αισθητήρας DHT22.....	9
2.2 Αισθητήρας BMP280.....	12
2.2.1. Επαφές SPI.....	15
2.2.2. Επαφές I2C .....	15
2.3 Ανεμόμετρο .....	15
2.4 Μικροελεγκτής .....	16
2.5 Δομοστοιχείο Ethernet.....	18
<b>3. Κατασκευή και Αποτελέσματα</b> .....	21
3.1. Εισαγωγή στο Κύκλωμα.....	21
3.2. Κύκλωμα.....	21
3.3. Πλακέτα .....	29
3.4. Αποτελέσματα.....	31
<b>4. Λογισμικό</b> .....	33
4.1. Κώδικας Αισθητήρων .....	33
4.2. Κώδικας συλλογής δεδομένων .....	35
4.3. Ιστοσελίδα αποτελεσμάτων .....	41
<b>5. Συμπεράσματα</b> .....	42
<b>Βιβλιογραφικές αναφορές</b> .....	43
<b>Παράρτημα</b> .....	45
A. Κώδικας μικροελεγκτή (C++).....	45
B. Κώδικας SERVER (C#).....	49
B.i. Το στατικό κομμάτι .....	49
B.ii. Το δυναμικό κομμάτι.....	51
B.iii. Το στατικό κομμάτι .....	58
B.iv. Το δυναμικό κομμάτι .....	58

## Ευχαριστίες

Η παρούσα εργασία αποτελεί την πτυχιακή εργασία στα πλαίσια σπουδών μου στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος. Η εργασία αυτή είναι αποτέλεσμα προσωπικής μελέτης, για την οποία όμως μερίμνησαν πολλοί άνθρωποι, χωρίς τη βοήθεια των οποίων δεν θα μπορούσε να ολοκληρωθεί η εργασία και τους οποίους θα ήθελα να ευχαριστήσω ονομαστικά.

Για αρχή, οφείλω να ευχαριστήσω τον επιβλέποντα καθηγητή της εργασίας αυτής, κύριο Άγγελο Γιακουμή, για την άψογη συνεργασία και τη σαφή καθοδήγηση, η βοήθεια του οποίου ήταν ταχύτατη και ακριβέστατη και η συμβολή του στη διαμόρφωση της παρούσας εργασίας ανεκτίμητη.

Θα ήθελα να ευχαριστήσω τον καλό μου φίλο Νικόλαο Νικήσιανλη, που με βοήθησε με τις καταδείξεις του στην κατασκευή της πλακέτας με το πρόγραμμα EAGLE.

Επιπλέον, ευχαριστώ τον Χρήστο Σαμαρά για τη βοήθειά του στην κατανόηση της γλώσσας προγραμματισμού C Sharp (C#) και στην κατασκευή της ιστοσελίδας, αφού μου παρείχε τον ιδιωτικό του server για τις απαιτήσεις της εργασίας αυτής.

Τέλος, θα ήθελα να ευχαριστήσω τη σύντροφό μου, Ευαγγελία Καραγιαννίδου, για την αμέριστη συμπαράσταση που έδειξε αυτόν τον καιρό, κατά τη διάρκεια εκπόνησης της εργασίας αυτής και τη βοήθειά της στην επιμέλεια του κειμένου της παρούσας εργασίας.

## 1. Εισαγωγή

Η ανθρώπινη ζωή επηρεάζεται άμεσα από το φυσικό περιβάλλον. Οι δραστηριότητες του ανθρώπου σε πολλές περιπτώσεις εξαρτώνται από τις μεταβολές των καιρικών συνθηκών. Από τα προϊστορικά χρόνια ακόμα, αλλά και στους μετέπειτα αιώνες, οι άνθρωποι είχαν θεοποιήσει τα καιρικά φαινόμενα και στοιχεία (π.χ. κεραυνούς, ήλιο κλπ.), γεγονός που αποδεικνύει το αίσθημα κατωτερότητας που ένιωθε ο άνθρωπος απέναντι στη φύση, τον υψηλό βαθμό εξάρτησής του από τις μεταβολές της, καθώς επίσης και την τεράστια σημασία των φυσικών φαινομένων για την ανθρώπινη ζωή και δραστηριότητα. Στην προσπάθειά του να κατανοήσει τις μετεωρολογικές συνθήκες, από την αρχαιότητα ακόμα, ο άνθρωπος αναζητούσε τρόπους με τους οποίους θα μπορούσε να «διαβάσει» τις επικείμενες καιρικές μεταβολές, ώστε να προσαρμόσει και να προγραμματίσει τις δραστηριότητές του, οι οποίες ανέκαθεν συνδέονταν και με την οικονομία ενός τόπου.

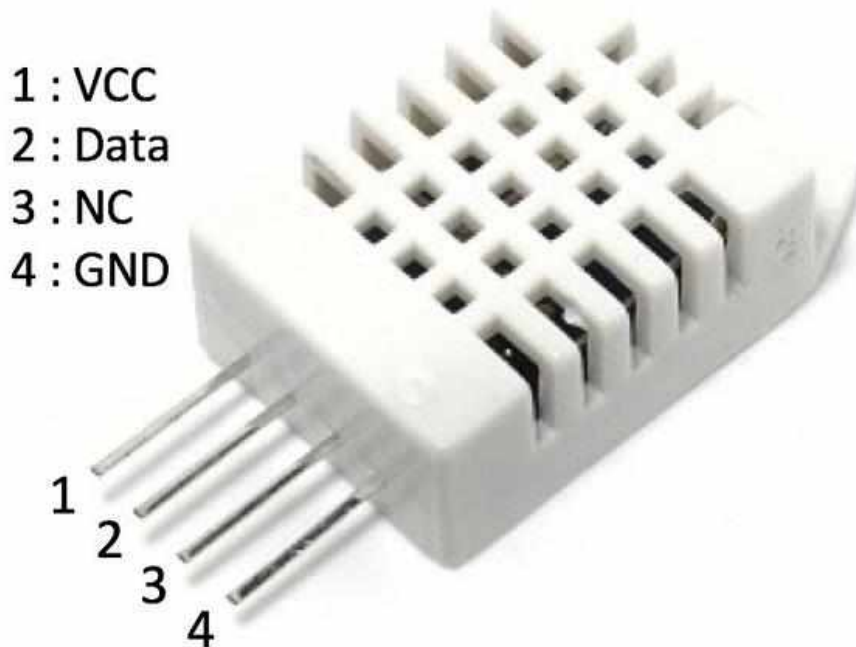
Η μελέτη των καιρικών μεταβολών ανά τους αιώνες συνέβαλε στην ανάπτυξη του κλάδου της Μετεωρολογίας, η οποία εξελίχθηκε από απλές παρατηρήσεις και λαϊκές γενικεύσεις σε επιστημονικό κλάδο. Αντίστοιχα εξελίχθηκαν και οι μέθοδοι παρατήρησης των μεταβολών των καιρικών συνθηκών. Από τα απλά εργαλεία έως και τα πλέον σύνθετα μηχανήματα που χρησιμοποιούνται στη σύγχρονη εποχή, ο άνθρωπος κατάφερε να κατανοήσει και να ερμηνεύσει τα φυσικά φαινόμενα, να καταγράψει δεδομένα για τις παροντικές καιρικές συνθήκες και να προβλέπει τις επικείμενες καιρικές μεταβολές (Πηγή 1).

Ένα τέτοιο όργανο είναι ο μετεωρολογικός σταθμός, η χρήση του οποίου είναι ιδιαίτερος διαδεδομένη. Ο μετεωρολογικός σταθμός μπορεί, μεταξύ άλλων, να δώσει στοιχεία για τη θερμοκρασία, την κατεύθυνση και την ισχύ του ανέμου, την ατμοσφαιρική πίεση, την υγρασία κ.ά., μέσω αισθητήρων που περισυλλέγουν τις αντίστοιχες πληροφορίες.

Η παρούσα εργασία, λοιπόν, πραγματεύεται την κατασκευή ενός τέτοιου μετεωρολογικού σταθμού, ενός οργάνου με κύκλωμα αισθητήρων για την καταγραφή δεδομένων που αφορούν τη θερμοκρασία, την υγρασία, την ατμοσφαιρική πίεση και την ταχύτητα του ανέμου. Συγκεκριμένα γίνεται καταγραφή των εξαρτημάτων που χρησιμοποιήθηκαν για τη σύνθεση του μετεωρολογικού οργάνου, περιγραφή της σύνθεσης (κατασκευής) του κυκλώματος και μεταφορά των μετρήσεων σε υπολογιστικό περιβάλλον.

## 2. Αισθητήρες

### 2.1 Αισθητήρας DHT22



Εικόνα 1: Αισθητήρας DHT22 με επεξήγηση των διεπαφών του (Πηγή 10).

Ο αισθητήρας DHT22 είναι ένας αισθητήρας μέτρησης υγρασίας και θερμοκρασίας. Χρησιμοποιείται για να συλλέγει τα συγκεκριμένα δεδομένα και να τα μεταδίδει με ένα κανάλι επικοινωνίας προς τον επεξεργαστή. Πρόκειται για έναν αισθητήρα ο οποίος έχει εύρος λειτουργίας από 0 μέχρι 100% για τη μέτρηση της υγρασίας και έχει τη δυνατότητα μέτρησης θερμοκρασίας από  $-40^{\circ}\text{C}$  μέχρι και  $+125^{\circ}\text{C}$ .

Επιπλέον, η επικοινωνία του γίνεται με έναν ψηφιακό τρόπο διαμέσου ενός μονού καναλιού που του επιτρέπει μεγαλύτερη ακρίβεια. Λειτουργεί με τάση τροφοδοσίας DC από 3,3V έως 5V. Ο αισθητήρας DHT22 απαιτεί 1,5mA ρεύματος για να είναι σε θέση να λειτουργήσει. Όταν βρίσκεται σε κατάσταση αναμονής, η κατανάλωση ρεύματος είναι 0,02mA και παρέχει ψηφιακές τιμές εξόδου. Ο αισθητήρας μπορεί να μετρήσει την υγρασία με ακρίβεια έως  $\pm 2\%$  RH. Η ακρίβεια θερμοκρασίας έως  $\leq \pm 0,5^{\circ}\text{C}$ . Αυτός ο αισθητήρας μετρά την υγρασία με ανάλυση 0,1%RH και τη θερμοκρασία με ανάλυση 0,1 $^{\circ}\text{C}$  έχοντας υστέρηση υγρασίας  $\pm 0,3\%$  RH και έχει μέση περίοδο ανίχνευσης 2 δευτερόλεπτα. Διατίθεται σε δύο μεγέθη: μικρό μέγεθος (14x18x5,5mm) και μεγάλο μέγεθος (22x28x5mm). Η υγρή συσκευή μέτρησης

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

θερμοκρασίας NTC χρησιμοποιείται στο DHT22 για τον υπολογισμό της θερμοκρασίας. Αυτός ο αισθητήρας μπορεί να μεταδώσει το σήμα έως και 2 μέτρα. Το DHT22 έχει σειριακή διασύνδεση ενός καλωδίου, το οποίο καθιστά την ενσωμάτωση με μικροεπεξεργαστές και μικροελεγκτές απλή και γρήγορη. Καθώς η σχετική υγρασία εξαρτάται από τη θερμοκρασία, αυτός ο αισθητήρας αντισταθμίζεται και βαθμονομείται σε ακριβείς θαλάμους βαθμονόμησης για ακριβείς μετρήσεις υγρασίας. Αυτός ο συντελεστής βαθμονόμησης αποθηκεύεται με τη μορφή μνήμης προγράμματος OTP. Τα σήματα ανίχνευσης εσωτερικών αισθητήρων μπορούν να ονομάσουν αυτόν τον συντελεστή βαθμονόμησης. Το DHT22 είναι εξαιρετικά ακριβές με εξαιρετική ποιότητα και γρήγορη απόκριση. Το DHT22 έχει εξαιρετική μακροπρόθεσμη σταθερότητα και δίνει μια ψηφιακή έξοδο η οποία είναι πολύ βαθμονομημένη (Πηγή 8).

Πρόκειται για έναν αισθητήρα που μπορεί να χρησιμοποιηθεί σε ένα μεγάλο φάσμα εφαρμογών, όπως την παρατήρηση θερμοκρασίας και υγρασίας σε εσωτερικούς ή εξωτερικούς χώρους, τη λήψη δεδομένων υγρασίας για μεγάλο χρονικό περιθώριο, τη σταθερή λειτουργία σε υψηλές θερμοκρασίες καθώς και την αντοχή σε δύσκολες περιβαλλοντικές συνθήκες, μιας και το εύρος του αισθητήρα καλύπτει τόσο υψηλές όσο και ψυχρές θερμοκρασίες. Είναι μία αξιόπιστη πηγή συλλογής δεδομένων η οποία μπορεί να αποτελέσει ακόμα και το κύριο όργανο σε κάποιο σύστημα συλλογής πληροφοριών για τις περιβαλλοντικές συνθήκες που επικρατούν σε κάποιο εσωτερικό ή εξωτερικό χώρο. (Πηγή 8).

- Τάση και Τροφοδοσία

Η τάση ισχύος πρέπει να είναι 3,3-5V DC. Ένας πυκνωτής 100nF μπορεί να προστεθεί μεταξύ VDD και GND για φιλτράρισμα κύματος.

- Επικοινωνία και σήμα

Τα δεδομένα του διαύλου που χρησιμοποιούνται για την επικοινωνία με τον DHT22 έχουν χρονική διάρκεια 5ms. Τα δεδομένα που συλλέγονται αποτελούνται από ακέραιο και δεκαδικό μέρος.

Η χρήση του αισθητήρα στην παρούσα εργασία βασίζεται στον συνδυασμό του αισθητήρα με ένα κύκλωμα παροχής σταθερής τάσης το οποίο αποτελείται από έναν διαιρέτη τάσης, έναν πυκνωτή για την σταθεροποίηση στην τάση καθώς και μία διεπαφή για σύνδεση σε έναν εκπομπό ESP-01. Η συγκεκριμένη διεπαφή δεν χρησιμοποιείται στην παρούσα εργασία (Πηγή 8).

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)



Εικόνα 2: Εμπρόσθια όψη DHT22 με ενσωματωμένες διεπαφές και κύκλωμα τροφοδοσίας.



Εικόνα 3: Οπίσθια όψη DHT22 με ενσωματωμένες διεπαφές και κύκλωμα τροφοδοσίας.

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

Παρακάτω παρατίθεται ένας πίνακας ο οποίος περιέχει όλα τα χαρακτηριστικά του DHT22:

Μοντέλο	DHT22
Τάση λειτουργίας	3,3-5V DC
Σήμα εξόδου	Ψηφιακό σήμα μέσο μονού διαύλου επικοινωνίας
Αισθητήριο στοιχείο	Πολυμερής πυκνωτής
Πεδίο μετρήσεων	Υγρασία 0-100%RH Θερμοκρασία -40~80°C
Accuracy	Υγρασία $\pm 2\%$ RH (Max $\pm 5\%$ RH) Θερμοκρασία $< \pm 0,5^\circ\text{C}$
Ευαισθησία	Υγρασία 0,1%RH Θερμοκρασία 0,1°C
Επαναληπτικότητα	Υγρασία $\pm 1\%$ RH Θερμοκρασία $\pm 0,2^\circ\text{C}$
Υστέρηση υγρασίας	$\pm 0,3\%$ RH
Σταθερότητα στην χρήση	$\pm 0,5\%$ RH/χρόνο
Διάρκεια εντοπισμού μετρήσεων	Μέσος όρος: 2s
Διαστάσεις	Μικρό μέγεθος: 14x18x5,5mm Μεγάλο μέγεθος: 22x28x5mm

Πίνακας 1: Χαρακτηριστικά DHT22 (Πηγή 9)

## 2.2 Αισθητήρας BMP280



Εικόνα 4: Αισθητήρας BMP280 (Πηγή 16).

Ο αισθητήρας BMP280 είναι ένας αισθητήρας μέτρησης θερμοκρασίας, υγρασίας και βαρομετρικής ατμοσφαιρικής πίεσης. Ο αισθητήρας αυτός συνδυάζει όλες αυτές τις

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

λειτουργίες σε ένα πολύ μικρό μέγεθος, προσφέροντας έτσι ένα μεγάλο πλεονέκτημα για εφαρμογές τόσο κινητής όσο και σταθερής εφαρμογής. Πρόκειται για ένα πολύ μικρό αισθητήριο το οποίο μπορεί να έχει μία αρκετά μεγάλη ακρίβεια στα δεδομένα, αντοχή στην εκτεταμένη χρήση καθώς και την ικανότητα παραμετροποίησής του, έτσι ώστε να μπορεί να συνδέεται και να συνεργάζεται με ένα μεγάλο εύρος επεξεργαστών και εφαρμογών τόσο σε επίπεδο hardware όσο και σε software. Καταναλώνει ελάχιστη ενέργεια, γεγονός που τον καθιστά ιδανικό για χρήση εφαρμογών που λειτουργούν και με μπαταρία.

Ο αισθητήρας αυτός βασίζεται σε μία αρχή λειτουργίας που ονομάζεται *πιεζοαντιστασιακή*. Άρα πρόκειται για έναν αισθητήρα πιεζοαντιστασιακής πίεσης. Το πιεζοαντιστασιακό φαινόμενο οφείλεται στην αλλαγή της ηλεκτρικής αντίστασης ενός ημιαγωγού όταν ένα μηχανικό φορτίο ασκείται πάνω του. Αυτό οδηγεί στην αλλαγή της ηλεκτρικής αντίστασης του ημιαγωγού, όχι όμως της διηλεκτρικής του ικανότητας (Πηγή 5).

Τεχνικές Προδιαγραφές	BMP280
Διαστάσεις	8-Pin LGA με μέταλλο 2,0 x 2,5 x 0,95 mm <sup>3</sup>
Εύρος λειτουργίας (πλήρης ακρίβειας)	Πίεση: 300...1100 hPa Θερμοκρασία: 0...+65°C
Τάση τροφοδοσίας V <sub>DDIO</sub>	1,2 ... 3,6V
Τάση τροφοδοσίας V <sub>DD</sub>	1,71 ... 3,6V
Διεπαφή	I <sup>2</sup> C and SPI
Μέση κατανάλωση ρεύματος (τυπική) (1Hz ρυθμός ανανέωσης)	2,74 μA (σε κατάσταση πολύ χαμηλής κατανάλωσης)
Μέσος όρος κατανάλωσης ρεύματος σε κατάσταση ύπνου	0,1 μA
Μέσος όρος λήψης μέτρησης	5,5msec (σε κατάσταση υπερβολικά μικρής κατανάλωσης)
Εύρος ανάλυσης δεδομένων	Πίεση: 0,18 Pa (ίσο με <10 cm) Θερμοκρασία: 0,01 K

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

Απόλυτη ακρίβεια $p = 950 \dots 1100\text{hPa}$ ( $T=0 \dots +65^\circ\text{C}$ )	$\sim \pm 1 \text{ hPa}$
Σχετική ακρίβεια πίεσης (τυπική) $p = 950 \dots 1050\text{hPa}$ ( $+25^\circ\text{C}$ )	$\pm 0,12 \text{ hPa}$ (ίσο με $\pm 1 \text{ m}$ )
Αντιστάθμιση συντελεστή θερμοκρασίας ( $+25^\circ \dots +40^\circ\text{C}$ @ $900\text{hPa}$ )	$1,5 \text{ Pa/K}$ (ίσο με $12,6 \text{ cm/K}$ )

Πίνακας 2: Χαρακτηριστικά BMP280 (Πηγή 3).

Η τάση τροφοδοσίας του αισθητήρα βρίσκεται ανάμεσα στην περιοχή τιμών τάσεων από 3V μέχρι και 5V. Η επικοινωνία του με άλλη συσκευή μπορεί να γίνει μέσω του πρωτοκόλλου I2C καθώς και με SPI. Ο αισθητήρας έχει ακρίβεια στην ατμοσφαιρική πίεση  $\pm 1\text{hPa}$ , ενώ η ακρίβεια του στην θερμοκρασία είναι στον  $\pm 1,0^\circ\text{C}$ .

Πρόκειται για έναν μικρό αισθητήρα που έχει διαστάσεις  $19,2 \times 17,9 \times 2,9\text{mm}$  /  $0,8'' \times 0,7'' \times 0,1''$  και βάρος 1,3g. Η πλακέτα διαθέτει ενσωματωμένο ρυθμιστή τάσης που λαμβάνει 3-5V DC και τις μετατρέπει σε 3V. Αυτή είναι η έξοδος από τον ρυθμιστή 3,3V μπορεί να παρέχει έως και 100mA. (Πηγή 3).



Εικόνα 5: Αισθητήρας BMP280, εμπρόσθια όψη.



Εικόνα 6: Αισθητήρας BMP280, οπίσθια όψη.

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

### 2.2.1. Επαφές SPI

Ο αισθητήρας διαθέτει 4 επαφές για τη χρήση με SPI (Πηγή 17):

- SCK - SPI Clock, είσοδος.
- SDO - Serial Data Out / Master In Slave Out pin, για να στέλνει προς τον επεξεργαστή.
- SDI - Serial Data In / Master Out Slave In pin, για να δέχεται δεδομένα από τον επεξεργαστή.
- CS - Chip Select, τύπος εισόδου.

### 2.2.2. Επαφές I2C

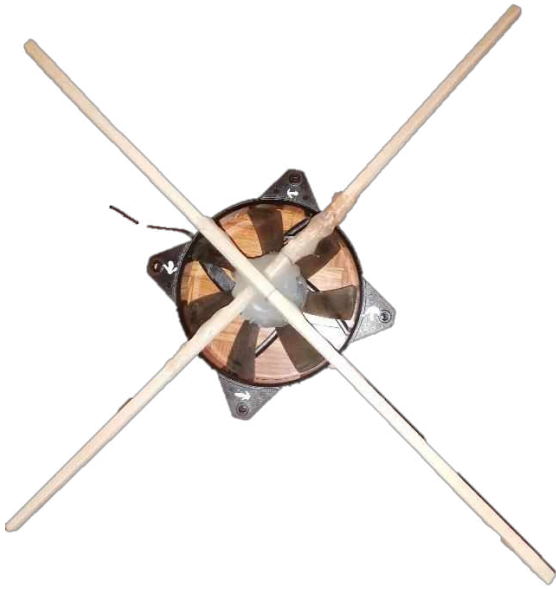
- SCK - είναι επίσης ο ακροδέκτης ρολογιού I2C, η επαφή συνδέεται με το ρολόι μικροεπεξεργαστή.
- SDI - είναι επίσης ακροδέκτης δεδομένων χρόνου I2C συνδέεται με τον μικροεπεξεργαστή

(Πηγή 17).

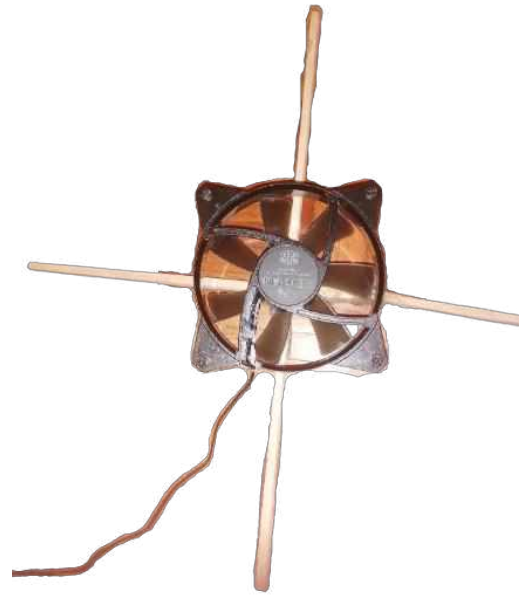
## 2.3 Ανεμόμετρο

Ο αισθητήρας αυτός είναι ένας αυτοσχέδιος αισθητήρας. Βασίζεται σε έναν ανεμιστήρα υπολογιστή ο οποίος έχει χρησιμοποιηθεί, με τη χρήση θερμής σιλικόνης, για να φιλοξενεί πάνω του δύο ξύλα, στις άκρες των οποίων έχουν δημιουργηθεί υποδοχές για να προσαρτηθούν πάνω τους τέσσερα κύπελλα. Τα κύπελλα είναι τοποθετημένα προς μία κοινή κατεύθυνση, έτσι ώστε να υπάρχει γραμμική κίνηση χωρίς αντιστάσεις. Πρόκειται για μία απλή κατασκευή που εκμεταλλεύεται τη δυνατότητα του ανεμιστήρα του υπολογιστή που λειτουργεί με PWM για να γνωρίζει σε τι ταχύτητα περιστρέφεται ανά πάσα στιγμή. Περαιτέρω αναφορά στην κατασκευή καθώς και τη λειτουργία περιγράφεται στο Κεφάλαιο 3, παράρτημα 3.3 (Πηγή 4).

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)



Εικόνα 7: Εμπρόσθιο τμήμα ανεμόμετρου.



Εικόνα 8: Οπίσθιο τμήμα ανεμόμετρου.

## 2.4 Μικροελεγκτής

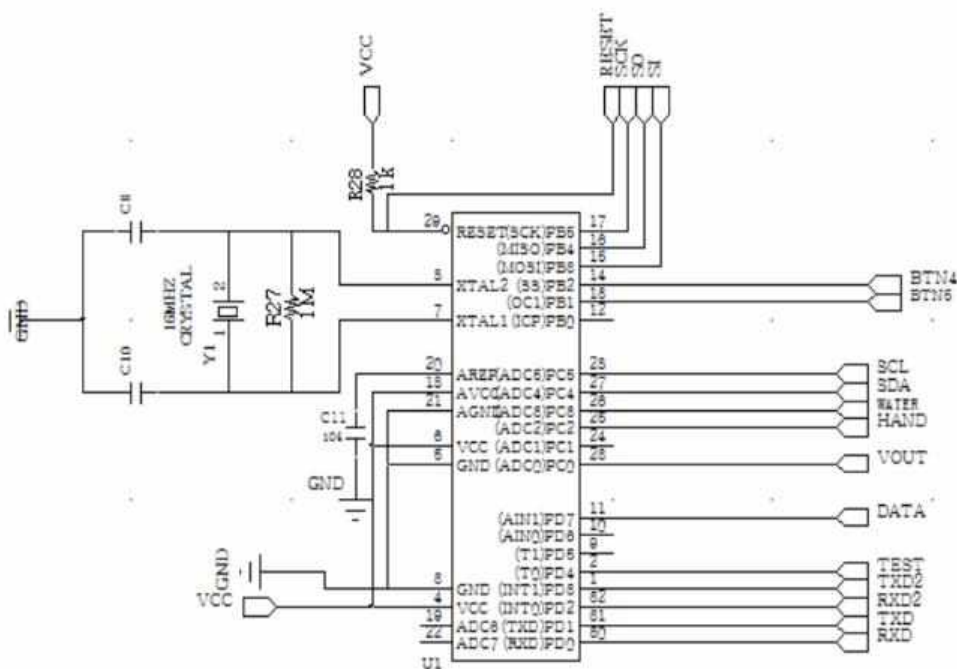


Εικόνα 9: Μικροελεγκτής Atmel Atmega 328p.

Το κύριο μέρος της εργασίας βασίζεται πάνω στον μικροελεγκτή. Αυτός είναι το κέντρο ελέγχου για τους αισθητήρες καθώς και για την επικοινωνία και τη συλλογή των δεδομένων. Πρόκειται για τον μικροελεγκτή της Atmel Atmega 328P. Ο μικροελεγκτής αυτός βασίζεται σε μια 8-bit RISC αρχιτεκτονική που εφαρμόζεται από την Atmel από το 1996 σε μία μεγάλη οικογένεια μικροελεγκτών. Αυτού του τύπου οι μικροελεγκτές μπορούν να χρησιμοποιηθούν

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

σε πολλούς τύπους εφαρμογών, όπως σε μικρά κυκλώματα ρομποτικής ή σε εφαρμογές εκπαιδευτικού χαρακτήρα. Η κύρια χρήση τους είναι να βοηθούν τον χρήστη να υλοποιήσει ένα ενσωματωμένο σύστημα με απλό και γρήγορο τρόπο. Ο επεξεργαστής αυτός διαθέτει 32KB ISP μνήμης flash που διαθέτει ικανότητα read-while-write, δημιουργώντας έτσι μία απόλυτα διαμορφώσιμη ικανότητα προγραμματισμού και επικοινωνίας του επεξεργαστή με οποιοδήποτε λογισμικό. Διαθέτει 2KB μνήμης SRAM (Static Random Access Memory), δηλαδή μία στατική RAM που χρησιμοποιεί ένα κύκλωμα flip-flop για να αποθηκεύει στις δύο σταθερές του καταστάσεις πληροφορίες μέχρι να ανοίξει το κύκλωμα και να αφαιρεθεί η τροφοδοσία από τον μικροελεγκτή. Επιπλέον διαθέτει 1KB μνήμης EEPROM (Electrically Erasable Programmable Read-Only Memory), για να αποθηκεύει δεδομένα και αφού έχει αφαιρεθεί η παροχή ρεύματος. Ο μικροελεγκτής περιλαμβάνει 23 διαύλους γενικής επικοινωνίας για τις εισόδους/εξόδους, 3 χρονομετρητές με ικανότητα σύγκρισης, 32 γενικής χρήσης καταχωριστές, μια θύρα SPI, εσωτερικούς και εξωτερικούς διακόπτες, σειριακά προγραμματιζόμενο USART (Universal Synchronous and Asynchronous Receiver-Transmitter), έναν 6-κάναλο 10-bit A/D (analog-to-digital) μετατροπέα, έναν προγραμματιζόμενο χρονιστή με εσωτερικό ταλαντωτή και πέντε επίπεδα διακύμανσης της κατανάλωσης ενέργειας. Η παροχή ενέργειας ορίζεται από τα 1,8V μέχρι τα 5,5V (Πηγή 14).



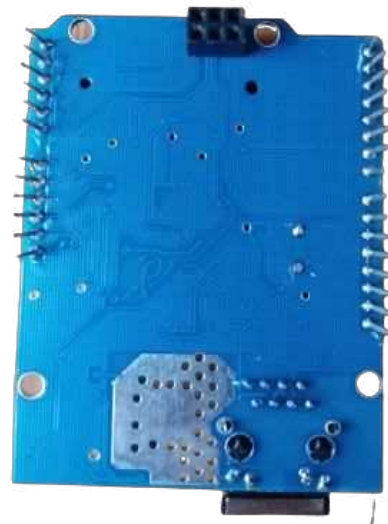
Εικόνα 10: Σχηματικό μικροελεγκτή Atmega 328P (Πηγή 18).

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

## 2.5 Δομοστοιχείο Ethernet



Εικόνα 11: Δομοστοιχείο (module) Ethernet, εμπρός όψη.



Εικόνα 12: Δομοστοιχείο (module) Ethernet, οπίσθια όψη.

Το δομοστοιχείο (module) ή μονάδα Ethernet αποτελεί το τελικό στοιχείο της εργασίας. Πρόκειται για μία μονάδα διασύνδεσης η οποία επιτυγχάνει την σύνδεση των αισθητήρων και του μικροελεγκτή με το internet και επιτυγχάνει την αμφίδρομη επικοινωνία τους με την ιστοσελίδα που έχει δημιουργηθεί, όπου αποθηκεύονται οι τιμές και τα αποτελέσματα των αισθητήρων.

Πρόκειται για το δομοστοιχείο Ethernet Shield W5100. Ο επεξεργαστής του δομοστοιχείου είναι ο Wiznet 5100, από τον οποίο παίρνει και το όνομά του. Πρόκειται για ένα module το οποίο έχει δυνατότητα ταυτόχρονης σύνδεσης μέχρι και τεσσάρων συσκευών. Έχει την ικανότητα να παρέχει διαδικτυακά πρωτόκολλα τόσο σε μορφή TCP (Transmission Control Protocol – Πρωτόκολλο ελέγχου εκπομπής) όσο και σε μορφή UDP (User Datagram Protocol - Πρωτόκολλο δεδομένων χρήστη). Πρόκειται για τα κυριότερα πρωτόκολλα που χρησιμοποιούνται στην ανταλλαγή δεδομένων μεταξύ 2 ή και περισσότερων υπολογιστών. Αυτά τα δύο πρωτόκολλα έχουν βασιστεί στο κύριο πρωτόκολλο της IP και πραγματοποιούν έλεγχο λαθών πριν αποστείλουν τα δεδομένα τους. Επιπλέον στοιχεία που περιλαμβάνονται

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

στα απεσταλμένα δεδομένα περιλαμβάνουν τη θύρα αποστολής του αποστολέα και τη θύρα του παραλήπτη. Όλα αυτά τοποθετούνται σε μία κεφαλίδα στην οποία περιλαμβάνονται και τα χαρακτηριστικά του πακέτου (Πηγή 11).

Η σύνδεση με το module αυτό επιτυγχάνεται με την χρήση wire-wrap επαφών. Οι επαφές αυτές έχουν μεγαλύτερο μήκος οι οποίες εκτείνονται διαμέσου του module. Αυτό ευνοεί τη σύνδεση τόσο από την πάνω όσο και από την κάτω μεριά περιφερειακών εξαρτημάτων.

Υπάρχει επίσης και η υποδοχή μιας micro SD κάρτας, η οποία μπορεί να χρησιμεύσει στην τοπική αποθήκευση δεδομένων καθώς και στην επεξεργασία αυτών των δεδομένων από το διαδίκτυο.

Το module αυτό περιλαμβάνει και ενδείκτες LED για να πληροφορήσει τον χρήστη για την κατάσταση στην οποία βρίσκεται. Τα ενδεικτικά LED έχουν τις εξής ενδείξεις (Πηγή 15):

- PWR - ένδειξη ότι η πλακέτα τροφοδοτείται.
- LINK - ένδειξη της παρουσίας ενός συνδέσμου δικτύου και αναβοσβήνει όταν το δομοστοιχείο μεταδίδει ή λαμβάνει δεδομένα.
- 100M - ένδειξη παρουσίας δικτύου 100Mb/s.
- COLL - ένδειξη όταν ανιχνευτούν συγκρούσεις δικτύου.
- TX - ένδειξη αποστολής δεδομένων από το δομοστοιχείο.
- RX - ένδειξη λήψης δεδομένων από το δομοστοιχείο.
- FULLD - ένδειξη λειτουργίας δικτύου σε επίπεδο αποστολής πλήρους διπλής διαδρομής, ικανότητα ταυτόχρονης αποστολής πληροφοριών μεταξύ του switch και του τελικού παραλήπτη.



### **3. Κατασκευή και Αποτελέσματα**

Σε αυτό το κεφάλαιο θα γίνει αναφορά στα ηλεκτρονικά εξαρτήματα και μέρη του κυκλώματος τα οποία χρησιμοποιήθηκαν για την παρούσα εργασία, καθώς επίσης και στον τρόπο σχεδίασης του κυκλώματος. Επίσης θα γίνει μνεία και για τα προβλήματα τα οποία αντιμετωπίστηκαν καθώς και στην ίδια την κατασκευή.

#### *3.1. Εισαγωγή στο Κύκλωμα*

Το κύκλωμα που κατασκευάστηκε για την παρούσα εργασία θα αναλυθεί διεξοδικά σε αυτό το κεφάλαιο. Ο μετεωρολογικός σταθμός απαιτεί την συλλογή πληροφοριών για τη θερμοκρασία, την υγρασία, την ατμοσφαιρική πίεση, το υψόμετρο και την ταχύτητα του ανέμου. Όλα τα παραπάνω είναι τιμές οι οποίες θα συλλεχθούν και θα αποσταλούν σε μία ιστοσελίδα στον φυλλομετρητή (browser), η οποία θα παρουσιάσει τα τελικά αποτελέσματα των τιμών κατά τη στιγμή που θα πραγματοποιηθεί είσοδος σε αυτή. Η σελίδα αυτή έχει τη δυνατότητα ανανέωσης του περιεχομένου της με το πάτημα ενός κουμπιού, ενσωματωμένου στη σελίδα προβολής των αποτελεσμάτων ή μπορεί να ανανεωθεί και με τη χρήση του κουμπιού F5 ή την επιλογή ανανέωσης σελίδας του φυλλομετρητή.

Για αυτόν τον σκοπό, χρησιμοποιήθηκαν οι αισθητήρες που αναφέρθηκαν στο κεφάλαιο 1, καθώς και κώδικες οι οποίοι βρίσκονται στο παράρτημα της εργασίας.

#### *3.2. Κύκλωμα*

Το κύκλωμα αποτελείται από τα παραπάνω στοιχεία και εξαρτήματα που αναφέρθηκαν. Έχει δημιουργηθεί ένα κύκλωμα τροφοδοσίας. Για τη λειτουργία του χρησιμοποιείται ένας μετασχηματιστής πρίζας AC-DC ο οποίος μετατρέπει το AC ρεύμα 220V και 0.6A σε DC ρεύμα 12V και 2000mA.

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)



Εικόνα 14: Μετασχηματιστής 220V σε 12V.

Πραγματοποιείται σύνδεση του μετασχηματιστή αυτού σε μία θύρα τροφοδοσίας η οποία είναι μία απλή κυλινδρική βάση με τις επαφές εσωτερικά της (βαρελάκι).



Εικόνα 15: Θύρα τροφοδοσίας.

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

Από εκεί δημιουργούνται μία γείωση και ένας αγωγός ρεύματος ο οποίος καταλήγει στον πρώτο ρυθμιστή τάσης (voltage regulator – L78505CV) ο οποίος μετατρέπει τα 12V σε 5V.



*Εικόνα 16: Voltage Regulator L78505CV 5V.*

Από τα 5V τροφοδοτούνται ο αισθητήρας DHT22 καθώς και ο μικροελεγκτής. Παρατηρείται ότι στο κύκλωμα έχει τοποθετηθεί και ένας πυκνωτής ο οποίος συμβάλλει στην εξισορρόπηση του ρεύματος. Επίσης, έχει προστεθεί μία αντίσταση στο «ποδαράκι» του RESET και έχει απομονωθεί έτσι ώστε να μην ενεργοποιηθεί κατά λάθος από κάποια ψηφιακή εναλλαγή κατάστασης του μικροελεγκτή.

Μετά από τον ρυθμιστή τάσης των 5V, υπάρχει ένας ακόμα ρυθμιστής τάσης στα 3,3V (LM3940IT), στον οποίο έχει επίσης προστεθεί ένας πυκνωτής (decoupling) για σταθεροποίηση της τάσης.



*Εικόνα 17: Voltage Regulator LM3940IT 3,3V.*

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

Ο πυκνωτής decoupling χρησιμοποιείται για να αποφευχθούν τυχόν παρεμβολές από κοντινά κυκλώματα, έτσι ώστε να μην υπάρχει κάποια παρασιτική μεταβολή τάσης (spike). Τέτοιου είδους παρεμβολές μπορεί να προκαλέσουν αστάθεια στο σύστημα ή και βραχυκύκλωμα. Έπειτα από τον πυκνωτή, τροφοδοτούνται τόσο ο αισθητήρας BMP280 όσο και το αυτοσχέδιο ανεμόμετρο με 3,3V.

Από το ανεμόμετρο γίνεται φανερό ότι υπάρχει μία επιπλέον σύνδεση πριν να καταλήξει στον μικροελεγκτή. Είναι μία αντίσταση 10K Ohm. Για να εξηγηθεί η ύπαρξή της, πρέπει πρώτα να διαπιστωθεί ο τρόπος με τον οποίο λειτουργεί ο ανεμιστήρας του υπολογιστή.



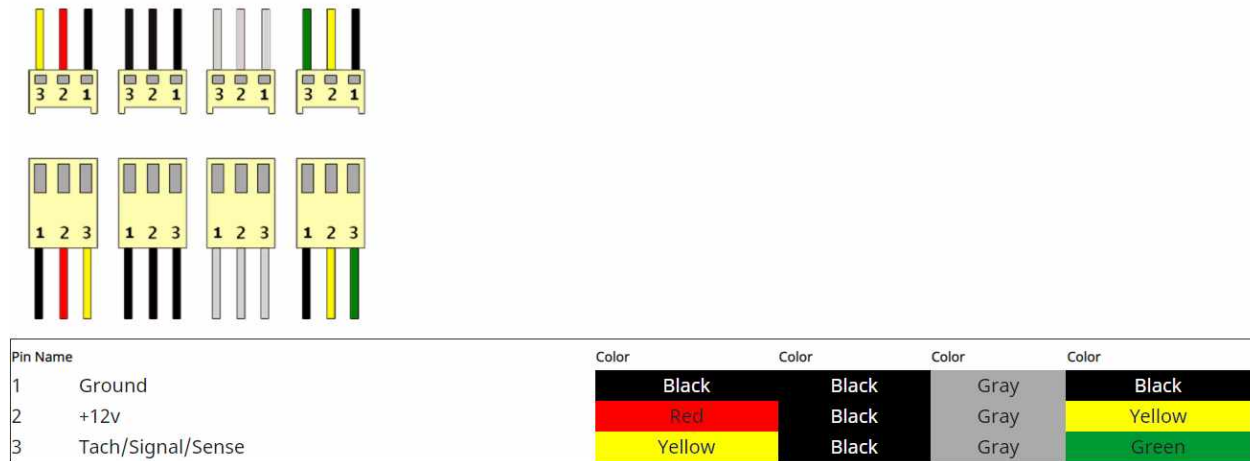
Εικόνα 18: Είδη ανεμιστήρων (από αριστερά προς τα δεξιά: 2pin, 3pin, 4pin) (Πηγή 13).

Υπάρχουν πολλά είδη ανεμιστήρων υπολογιστή. Υπάρχουν με δύο, τρία ή τέσσερα καλώδια. Ο πιο απλός ανεμιστήρας αποτελείται από δύο καλώδια, τα οποία καθορίζουν μόνο αν είναι σε λειτουργία (on) ή σε αδράνεια (off). Οι ανεμιστήρες με τρία καλώδια μπορούν να λάβουν έναν παλμό, έτσι ώστε να καθορίζεται από τον παλμό η κατάσταση λειτουργίας τους.

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

### 3 pin Fan Connections

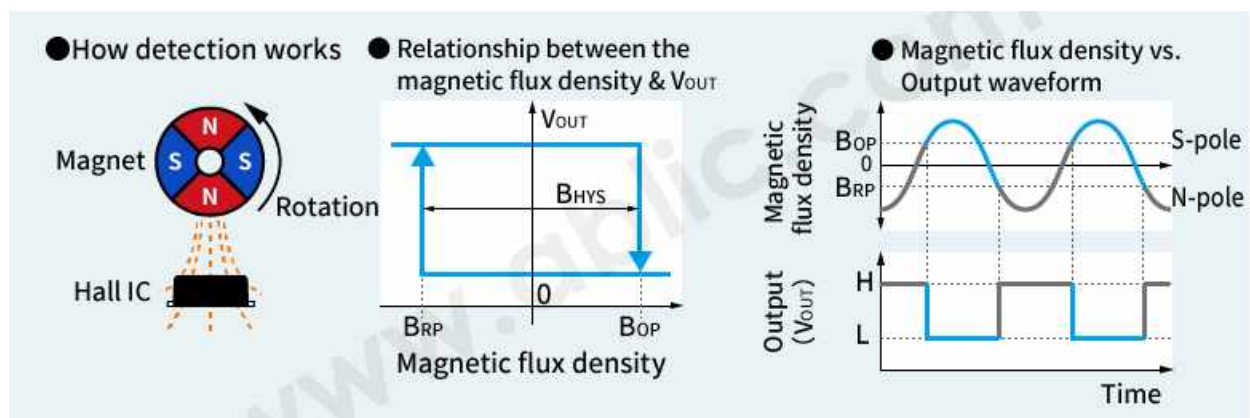
\*cable coloring varies from fan to fan



Εικόνα 19: Πιθανές επιλογές χρωμάτων και συνδέσεων σε ανεμιστήρα τριών pin (Πηγή 12).

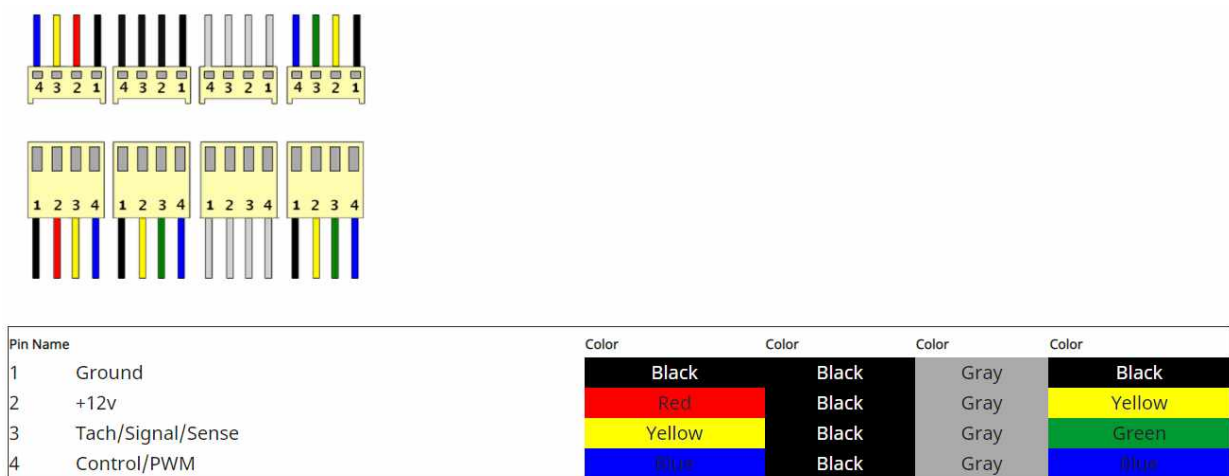
Τέτοιου είδους ανεμιστήρες χρησιμοποιούν έναν αισθητήρα που βασίζεται στο φαινόμενο του Hall. Το φαινόμενο του Hall αποτελείται από την παραγωγή διαφοράς τάσης σε έναν αγωγό που βρίσκεται σε εγκάρσια θέση σε σχέση με το ηλεκτρικό ρεύμα που τον διαρρέει, καθώς και σε ένα μαγνητικό πεδίο που είναι κάθετο στο ρεύμα (Πηγή 2).

Σύμφωνα με το φαινόμενο του Hall, είναι δυνατό να πραγματοποιηθούν μετρήσεις με έναν απλό αισθητήρα που βασίζεται σε αυτό το φαινόμενο, την περιστροφική ταχύτητα του ανεμιστήρα.



Εικόνα 20: Φαινόμενο του Hall σε περιστρεφόμενο δίσκο (Πηγή 2).

Τέλος, υπάρχει και ο ανεμιστήρας με τα 4 καλώδια, ο οποίος βασίζεται στο ένα επιπλέον καλώδιο που διαθέτει για να χρησιμοποιήσει μετρήσεις μέσω PWM (Pulse Width Modulation – Διαμόρφωση πλάτους παλμού). Το PWM λειτουργεί με το να μετατρέπει ένα αναλογικό σήμα σε παλμούς (ψηφιακό σήμα). Χρησιμεύει στο να τροφοδοτεί με ρεύμα τον ανεμιστήρα σε παλμούς οι οποίοι διαφέρουν στο πλάτος. Όσο μεγαλύτερο το πλάτος του παλμού, τόσο περισσότερο ρεύμα τροφοδοτεί τον ανεμιστήρα, άρα λειτουργεί και σε μεγαλύτερη ταχύτητα.



Εικόνα 21: Πιθανές επιλογές χρωμάτων και συνδέσεων σε ανεμιστήρα τεσσάρων pin (Πηγή 12).

Στην παρούσα περίπτωση, ένας ανεμιστήρας με 3 καλώδια είναι ιδανικός, γιατί προσφέρει ικανοποιητικό βαθμό λειτουργιών για τη λειτουργία του κυκλώματος. Καθώς ο ανεμιστήρας περιστρέφεται, με τη χρήση του αισθητήρα Hall, είναι δυνατό να διαβαστούν οι παλμοί που καταγράφει από την περιστροφή των μαγνητών που διαθέτει.

Ο ανεμιστήρας συνδέεται στη γείωση, το ένα πόδι στην τροφοδοσία των 3,3V και το τελικό του pin στη μία πλευρά της αντίστασης. Για να λειτουργήσει ο ανεμιστήρας, πρέπει να του παρέχεται τάση 5V. Ο λόγος για τον οποίο δεν του παρέχεται τάση 5V αλλά 3,3V είναι επειδή ο ανεμιστήρας θα ξεκινούσε να δουλεύει κανονικά σαν ανεμιστήρας αν τον τροφοδοτούσαν με την τάση των 5V. Επειδή η χρήση του αισθητήρα Hall γίνεται μόνο με την τροφοδοσία, άρα και η λήψη των μετρήσεων γίνεται μόνο αν τροφοδοτείται ο ανεμιστήρας, ορίζεται μία

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

μικρότερη τάση των 5V, συγκεκριμένα των 3,3V, έτσι ώστε και να μπορεί ο αισθητήρας Hall να παρέχει μια αξιόπιστη μέτρηση αλλά και να μην πραγματοποιηθεί η ανεπιθύμητη περιστροφή του ανεμιστήρα.

Η αντίσταση είναι των 10K και λειτουργεί σαν pull-up αντίσταση. Χρησιμοποιείται για την ενίσχυση του σήματος, έτσι ώστε να ληφθεί μία σωστή μέτρηση από τον ανεμιστήρα και την κατάσταση στην οποία βρίσκεται. Δίχως την αντίσταση pull-up, το σήμα σε μία ασθενή περιστροφή του ανεμιστήρα δεν θα μπορούσε να καταγραφεί και δεν θα λαμβάναμε δεδομένα σε μικρές ταχύτητες ανέμου, π.χ. κάτω από 2km/h. Η αντίσταση αυτή έπειτα συνδέεται με τον μικροελεγκτή και την τροφοδοσία των 3,3V.



Εικόνα 22: Κρύσταλλος 16MHz με 2 πυκνωτές 22pF.

Στο κύκλωμα περιλαμβάνεται και ένας κρύσταλλος των 16MHz, με δύο πυκνωτές. Ο ένας βρίσκεται στην είσοδο και ο άλλος στην έξοδο του κρυστάλλου. Ο μικροελεγκτής περιλαμβάνει έναν εσωτερικό κρύσταλλο, τιμής 8MHz. Δεν θα χρησιμοποιηθεί, όμως, ο εσωτερικός αυτός κρύσταλλος, γιατί δεν γίνει την επιθυμητή σταθερότητα στον συγχρονισμό των σημάτων, τόσο των αισθητήρων όσο και της αποστολής των δεδομένων στο διαδίκτυο. Ο εξωτερικός κρύσταλλος προσδίδει μεγαλύτερη ακρίβεια στους χρονισμούς (clock) του μικροελεγκτή, κρατώντας μία σταθερή χρονική κατάσταση για τις τιμές που λαμβάνονται.

Το τελικό κομμάτι του κυκλώματος είναι η σύνδεση της πλακέτας που κατασκευάστηκε με το δομοστοιχείο (module) του Ethernet από το οποίο ολοκληρώνεται η σύνδεση και η μεταφορά

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

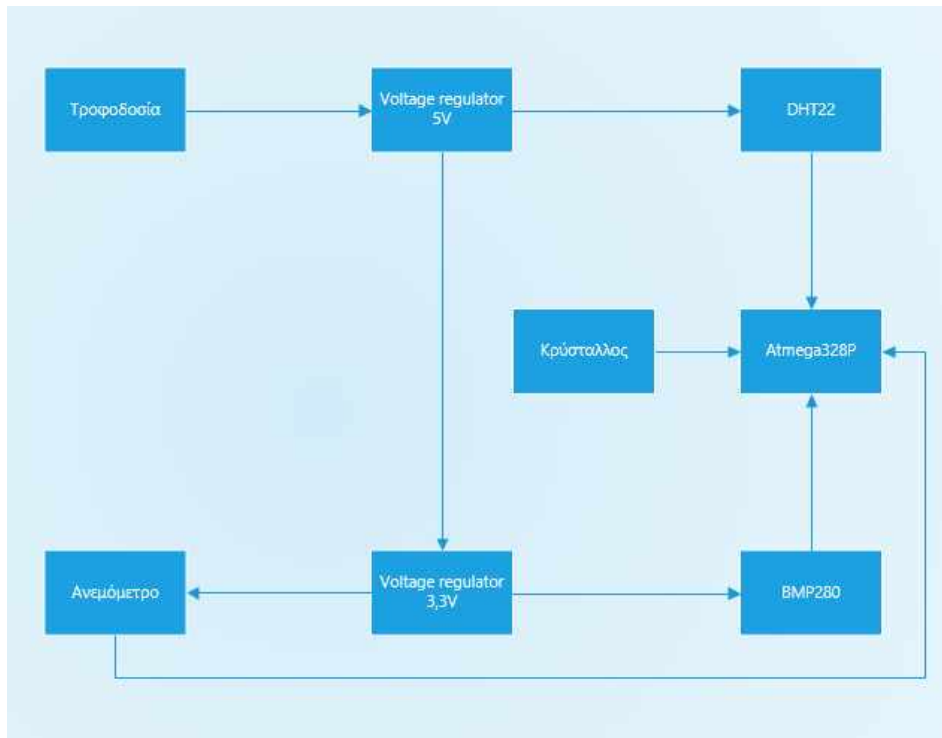
δεδομένων στο διαδίκτυο. Η σύνδεση πραγματοποιείται με τη χρήση pins, τα οποία έχουν προσαρμοστεί στο μέγεθος του δομοστοιχείου στο οποίο θα ενσωματωθεί η πλακέτα.



Εικόνα 23: Pins που χρησιμοποιήθηκαν στην πλακέτα.

Διαθέτει 28 pins από τα οποία θα χρησιμοποιηθούν μόνο 5 για την επικοινωνία των αισθητήρων και τη σύνδεσή τους με το διαδίκτυο. Αυτά είναι τα πόδια 10, 11, 12, 13 από το δομοστοιχείο του ethernet, τα οποία αντικατοπτρίζονται στα πόδια 16, 17, 18, 19 του μικροελεγκτή. Από αυτά τα πόδια τα οποία χρησιμοποιήθηκαν στη σύνδεση, το 10 και το 11 έχουν τη δυνατότητα να λαμβάνουν σήματα PWM. Επίσης έχει συνδεθεί το πόδι 9 του δομοστοιχείου στο πόδι 1 του μικροελεγκτή με μία αντίσταση. Αυτό είναι το πόδι που ελέγχει την κατάσταση επαναφοράς (RESET) του μικροελεγκτή. Προστέθηκε η αντίσταση με σκοπό την αποτροπή τυχαίας ενεργοποίησης της κατάστασης επαναφοράς του επεξεργαστή, ειδάλλως υπάρχει ο κίνδυνος να διαγραφεί ο κώδικας που «τρέχει» ο μικροελεγκτής, ωθώντας στον επαναπρογραμματισμό του.

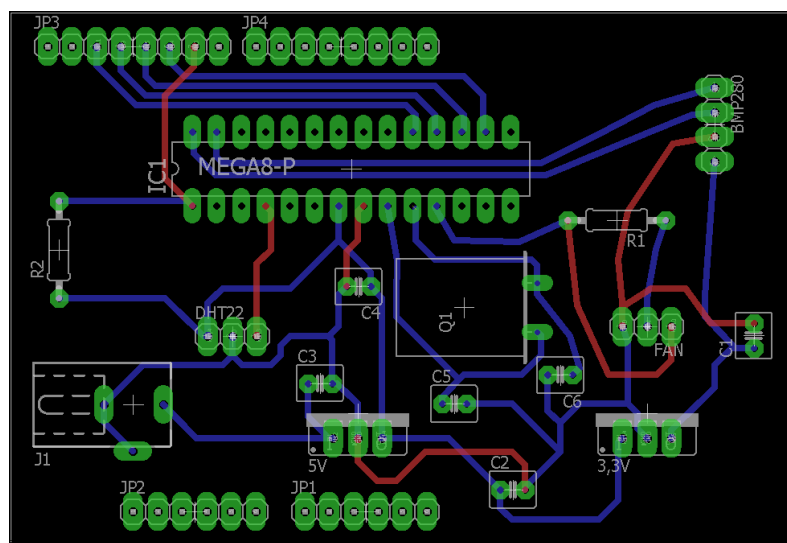
Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)



Μπλοκ διάγραμμα 1: Λειτουργία κυκλώματος.

### 3.3. Πλακέτα

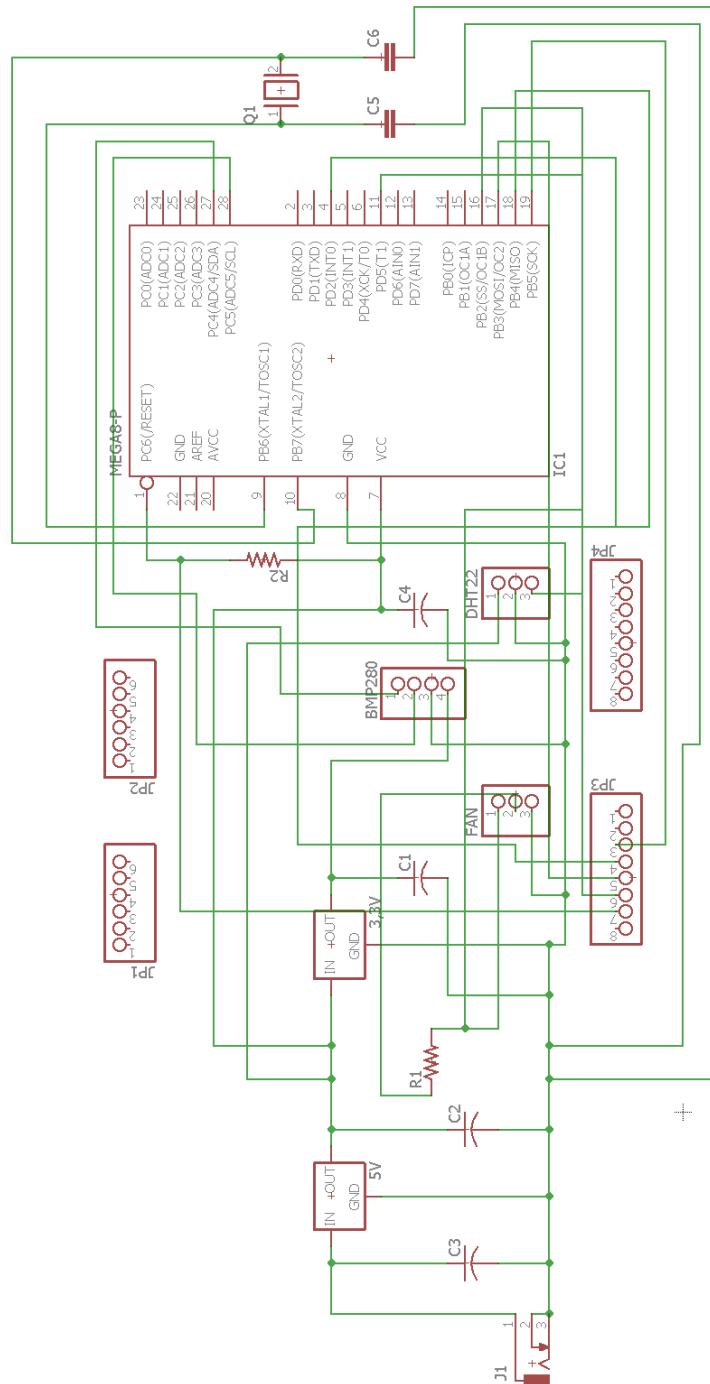
Η πλακέτα που κατασκευάστηκε για να συνδέσει τους αισθητήρες με τον μικροελεγκτή έχει μήκος 7,7cm πλάτος 5,3cm. Είναι σχεδιασμένη έτσι ώστε να ενωθεί με το δομοστοιχείο Ethernet από τις άνω του επαφές.



Εικόνα 24: Η πλακέτα με απεικόνιση των θέσεων των εξαρτημάτων και των αγωγών που τη διαρρέουν.

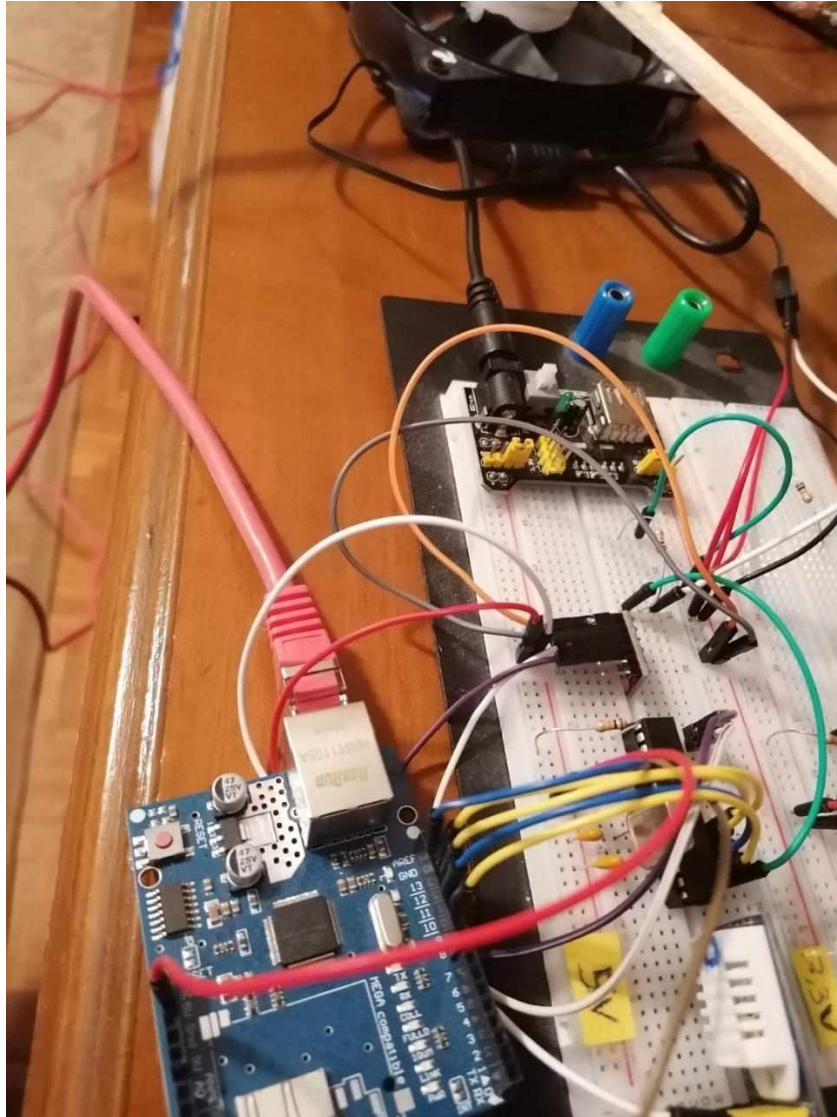
Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

Υπάρχει η δυνατότητα να διακριθούν όλες τις επαφές των αισθητήρων με τα αντίστοιχά τους ονόματα, τις αντιστάσεις και τους πυκνωτές, τον κρύσταλλο, τους voltage regulators, καθώς και τις σειρές επαφών (pins) JP1, JP2, JP3, JP4, τα οποία θα χρησιμοποιηθούν στην ένωση της πλακέτας με το δομοστοιχείο Ethernet.



Εικόνα 25: Σχηματικό πλακέτας προς κατασκευή.

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)



Εικόνα 26: Κύκλωμα εργασίας στο ράστερ.

### 3.4. Αποτελέσματα

Στην παρακάτω εικόνα παρατίθενται τα αποτελέσματα της μέτρησης που πραγματοποιήθηκε στις 12/09/2021, ημέρα Κυριακή.

Ειδικότερα: η θερμοκρασία που σημειώθηκε είναι 25°C, η ατμοσφαιρική πίεση 0,99 bar, το υψόμετρο 193 m., η υγρασία 63%, η ταχύτητα του ανέμου 2 km/h, ενώ αναγράφονται και η ημερομηνία (12/09/2021) και η ώρα (09.08'10" am) της μέτρησης. Το κουμπί «Refresh» δίνει τη δυνατότητα εκ νέου μέτρησης στα παροντικά δεδομένα.

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

Θερμοκρασία (oC):	25
Ατμοσφαιρική πίεση (bar):	0.99
Υψόμετρο (m):	193
Υγρασία (%):	63
Ταχύτητα ανέμου (km/h):	2
Date-Time:	9/12/2021 9:08:10 AM
<input type="button" value="Refresh"/>	

Εικόνα 27: Αποτελέσματα μετρήσεων μετεωρολογικού σταθμού στις 12/09/2021.

Θερμοκρασία (oC):	36
Ατμοσφαιρική πίεση (bar):	0.99
Υψόμετρο (m):	193
Υγρασία (%):	59
Ταχύτητα ανέμου (km/h):	0
Date-Time:	8/9/2021 19:43:49 PM
<input type="button" value="Refresh"/>	

Εικόνα 28: Αποτελέσματα μετρήσεων μετεωρολογικού σταθμού στις 09/08/2021.

Θερμοκρασία (oC):	35
Ατμοσφαιρική πίεση (bar):	0,99
Υψόμετρο (m):	193
Υγρασία (%):	61
Ταχύτητα ανέμου (km/h):	1
Date-Time:	8/22/2021 17:04:20 PM
<input type="button" value="Refresh"/>	

Εικόνα 29: Αποτελέσματα μετρήσεων μετεωρολογικού σταθμού στις 22/08/2021.

Θερμοκρασία (oC):	39
Ατμοσφαιρική πίεση (bar):	0,99
Υψόμετρο (m):	193
Υγρασία (%):	61
Ταχύτητα ανέμου (km/h):	3
Date-Time:	8/5/2021 14:24:16 PM
<input type="button" value="Refresh"/>	

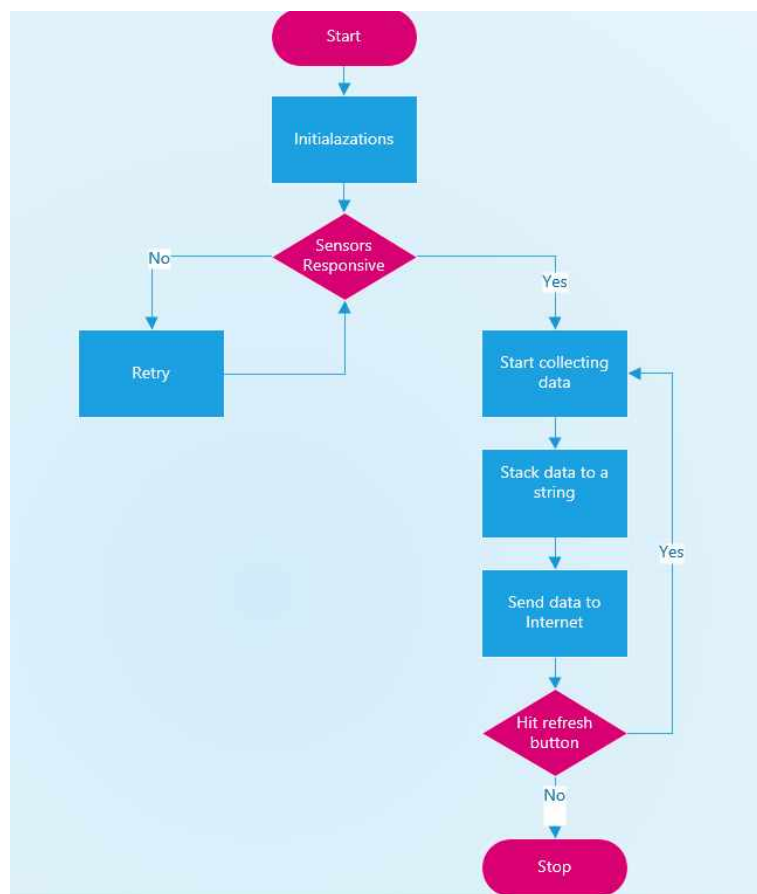
Εικόνα 30: Αποτελέσματα μετρήσεων μετεωρολογικού σταθμού στις 05/08/2021

## 4. Λογισμικό

Στο κεφάλαιο αυτό αναλύεται το λογισμικό που χρησιμοποιήθηκε στην εργασία. Πρόκειται για δύο ξεχωριστά τμήματα κώδικα. Το κάθε τμήμα είναι υπεύθυνο για μία συγκεκριμένη λειτουργία του κυκλώματος καθώς και της ιστοσελίδας όπου αναρτώνται τα αποτελέσματα.

### 4.1. Κώδικας Αισθητήρων

Στον κώδικα αυτόν θα αναλυθεί η διαδικασία που χρησιμοποιήθηκε για να ληφθούν τα δεδομένα από τους αισθητήρες. Ο κώδικας αυτός είναι το πρώτο κύριο μέρος από τα δύο που καθορίζουν την εργασία αυτή. Ο κώδικας αυτός είναι γραμμένος στην γλώσσα προγραμματισμού C++.



Διάγραμμα ροής 1: Λειτουργία κώδικα αισθητήρων.

Ο κώδικας αυτός ξεκινά με την αρχικοποίηση των αισθητήρων. Έπειτα γίνεται η δήλωση των ποδιών του μικροελεγκτή, στα οποία είναι συνδεδεμένοι οι αισθητήρες. Συνεχίζοντας, γίνεται ορατή η διαδικασία επικοινωνίας του μικροελεγκτή με τον server ο οποίος έχει δημιουργηθεί και λειτουργεί με ένα πρωτόκολλο DNS. Αφού δηλωθεί η IP από την οποία θα συνδεθεί ο μικροελεγκτής με το διαδίκτυο (σ.σ.: πρέπει να προγραμματίζεται ξανά σε περίπτωση που η κατασκευή μετακινηθεί σε άλλο δίκτυο ή χώρο) και έχει τεθεί θύρα του δικτύου (πόρτα δικτύου - port) στην τιμή 80 (συνήθως είναι η προκαθορισμένη πόρτα σε ένα δίκτυο), τότε ξεκινά η επικοινωνία του πελάτη, στην συγκεκριμένη περίπτωση το δομοστοιχείο ethernet.

Αφού επιτευχθεί η επικοινωνία, ξεκινούν οι ρουτίνες των αισθητήρων οι οποίες θα ξεκινήσουν την καταγραφή των δεδομένων που θα ληφθούν. Έπειτα, και αφού είναι βέβαιο ότι έχει πραγματοποιηθεί σύνδεση σε μία σωστή MAC Address μέσω του DHCP, τότε ξεκινά το κύριο κομμάτι του κώδικα αυτού που είναι η συλλογή δεδομένων από τους αισθητήρες. Πρώτος είναι ο αισθητήρας του ανεμόμετρου. Ακολουθεί η πρώτη μέτρηση θερμοκρασίας, η οποία λαμβάνεται από τον αισθητήρα BMP280.

Στη συνέχεια, πραγματοποιούνται οι μετρήσεις ατμοσφαιρικής πίεσης και υψόμετρου από τον ίδιο αισθητήρα (BMP280). Επειδή ο αισθητήρας πίεσης του BMP280 μετρά την πίεση σε hPa (hectopascal), πρέπει να γίνει μετατροπή σε Bar (ατμόσφαιρες), το οποίο γίνεται με μία διαίρεση με το 1000. Μετά πραγματοποιείται μέτρηση του υψόμετρου, στην οποία πρέπει να προστεθεί +10, γιατί ο συγκεκριμένος αισθητήρας είναι μη βαθμονομημένος και παρουσιάζει απόκλιση 10 μονάδων από τη σωστή μέτρηση (σε περίπτωση βαθμονομημένου αισθητήρα αφαιρείται το +10.) Η λανθασμένη βαθμονόμηση δεν επηρεάζει τις υπόλοιπες μετρήσεις και τμήματα του αισθητήρα, καθώς τα συγκεκριμένα είναι σωστά βαθμονομημένα και το πρόβλημα βρίσκεται στο λογισμικό του αισθητήρα μέτρησης υψομέτρου.

Έπειτα εισάγονται τιμές από τον αισθητήρα DHT22. Πρώτη μέτρηση είναι αυτή της υγρασίας. Η επόμενη μέτρηση είναι αυτή της δεύτερης θερμοκρασίας. Ακολουθεί ο μέσος όρος θερμοκρασίας από τις μετρήσεις ένα (1) και δύο (2). Χρησιμοποιήθηκαν δύο μετρήσεις θερμοκρασίας, από τις οποίες εξάγεται ο μέσος όρος, έτσι ώστε να εμφανίζεται μία πιο σταθερή και αξιόπιστη μέτρηση σε περίπτωση τοπικής αύξησης θερμοκρασίας σε μόνο έναν από τους δύο αισθητήρες. Η τελευταία μέτρηση που πραγματοποιείται είναι η μέτρηση της ταχύτητας περιστροφής του ανεμόμετρου. Για την επίτευξη της μέτρησης αυτής, χρησιμοποιήθηκε ο εξής

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

τύπος.

$$K = D \times r \times 0.001885$$

Όπου:

- K: χιλιόμετρα ανά ώρα (km/h)
- D: διάμετρος του ανεμιστήρα (cm)
- r: στροφές ανά λεπτό (rpm)
- 0.001885: σταθερά

Αυτός ο τύπος μετατρέπει την ταχύτητα περιστροφής ενός ανεμιστήρα στην ταχύτητα διέλευσης του αέρα. Αποτελείται από έναν πολλαπλασιασμό της διαμέτρου του ανεμιστήρα σε mm (120 mm), τις στροφές ανά λεπτό (rpm) του ανεμιστήρα και μίας σταθεράς (0.001885). Το αποτέλεσμα δίνει την ταχύτητα του αέρα που διαρρέει από τον ανεμιστήρα και έτσι γίνεται γνωστή η ταχύτητά του (Πηγή 7).

Τέλος όλες μαζί οι μετρήσεις μεταφέρονται σε ένα string και από εκεί μεταδίδονται στο διαδίκτυο.

#### 4.2. Κώδικας συλλογής δεδομένων

Ο κώδικας συλλογής δεδομένων είναι ο κώδικας που δημιουργεί μία βάση δεδομένων η οποία βρίσκεται σε έναν διαδικτυακό server, έτσι ώστε να είναι δυνατή η μεταφορά στο διαδίκτυο των δεδομένων που συλλέχθηκαν από τους αισθητήρες του κυκλώματος στην πλακέτα κατασκευής. Ο κώδικας είναι γραμμένος στην γλώσσα προγραμματισμού C# (C SHARP).

Για τη συλλογή δεδομένων χρησιμοποιείται ένας server ο οποίος παρέχεται από το autosoft. Μέσω του autosoft είναι δυνατή η δημιουργία ιστοσελίδων σε μορφές html, php και asp. Η σελίδα αυτή είναι βασισμένη στη δομή της asp. Γίνεται χρήση μίας βάσης δεδομένων για τη συλλογή των δεδομένων από τον μικροελεγκτή. Το autosoft είναι ένα πρόγραμμα που επιτρέπει τη δημιουργία ιστοσελίδων με τη χρήση του Microsoft Visual Studio.

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

Το Microsoft Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης, μέσα από το οποίο είναι δυνατόν να αναπτυχθούν μία ή περισσότερες ιστοσελίδες, εφαρμογές κινητής τηλεφωνίας καθώς και διαδικτυακές εφαρμογές. Το Microsoft Visual Studio επιτρέπει τη χρήση πάνω από 30 γλωσσών προγραμματισμού, μερικές από τις σημαντικότερες είναι η C++, η C#, η JavaScript, η Visual Basic και HTML και πολλά από τα στοιχεία του είναι προσβάσιμα και με δωρεάν εκδόσεις του προγράμματος (Πηγή 19).

Ο κώδικας αυτός είναι βασισμένος στο πρόγραμμα ανοιχτού κώδικα ASP.NET (Active Server Pages) της Microsoft. Η χρήση του ASP βρίσκεται στη δημιουργία ιστοσελίδων οι οποίες είναι δυναμικές και επιτρέπουν τον δημιουργό τους να κατασκευάσει μια ιστοσελίδα η οποία συνέχεια θα μεταβάλλεται ανάλογα με τις παραμέτρους που έχει θέσει αυτός (Πηγή 6). Οι αλλαγές πραγματοποιούνται με τη χρήση scripts τα οποία ανήκουν σε έναν διαδικτυακό server και μορφοποιούν τη σελίδα ανάλογα με τις ανάγκες του χρήστη, για παράδειγμα όταν επιθυμεί ο χρήστης να δει τα αποτελέσματα μιας μέτρησης.

Ο κώδικας συλλογής αποτελεσμάτων αποτελείται από δύο μέρη: το στατικό μέρος και το δυναμικό μέρος. Συγκεκριμένα:

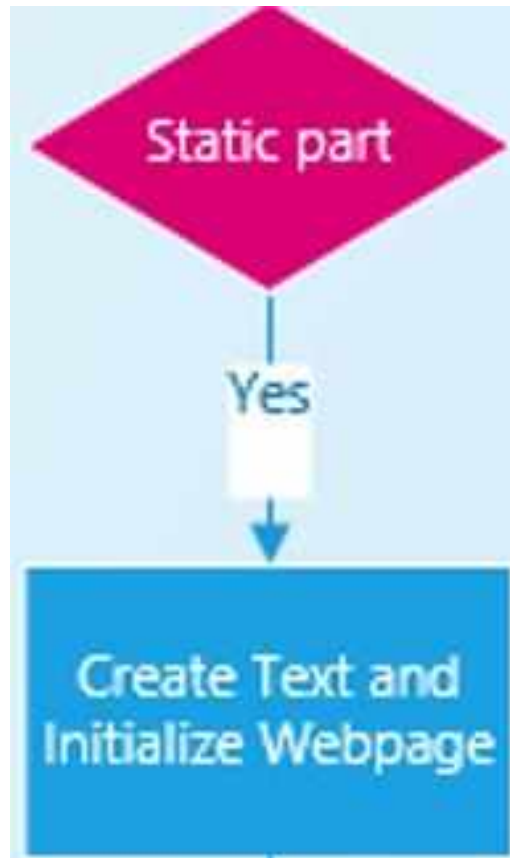
- Το στατικό μέρος του είναι το μέρος του κώδικα το οποίο δεν αλλάζει μορφή κατά τη διάρκεια εκτέλεσης του κώδικα. Η ύπαρξή του στοχεύει στη δημιουργία του περιβάλλοντος στο οποίο θα μπορεί ο χρήστης να παρακολουθήσει τις τιμές καθώς και το περιβάλλον διεπαφής μεταξύ χρήστη και ιστοσελίδας.
- Το δυναμικό μέρος του κώδικα είναι το μέρος του κώδικα το οποίο είναι υπεύθυνο να μεταβιβάζει τις τιμές από τον μικροελεγκτή διαμέσου του Ethernet και τελικά στον server που λειτουργεί στο autosoft. Από εκεί, οι τιμές ανεβαίνουν στη σελίδα που έχει δημιουργήσει το στατικό κομμάτι και εισέρχονται στα πεδία των εκάστοτε μεταβλητών. Τέλος, το δυναμικό κομμάτι είναι υπεύθυνο και για την ανανέωση του περιεχομένου όταν πατηθεί το κουμπί Refresh από τον χρήστη.

Το στατικό κομμάτι ξεκινά πραγματοποιώντας σύνδεση στην ιστοσελίδα που έχει οριστεί. Κάνει την αρχικοποίησή της και μετά δημιουργεί τον δίαυλο επικοινωνίας μεταξύ σελίδας και server.

Αφού πραγματοποιηθούν αυτά, προχωράει στο κομμάτι κατασκευής και σχεδίασης του οπτικού περιβάλλοντος στην ιστοσελίδα. Προσθέτει τους τίτλους, τις μεταβλητές, την ώρα και

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

την ημερομηνία (στον αμερικανικό τρόπο παρουσίασης τους), τη γραμματοσειρά και το μέγεθος των γραμμάτων και τέλος δημιουργεί και το κουμπί του Refresh (όχι την ικανότητά του να πατηθεί, αλλά μόνο το περίβλημά του).



Διάγραμμα ροής 2: Στατικό μέρος κώδικα.

Το δυναμικό κομμάτι είναι αυτό που έχει την κύρια λειτουργία του κώδικα. Το δυναμικό κομμάτι αποτελείται από δύο συναρτήσεις: τη συνάρτηση Select και τη συνάρτηση Create.

Η μεταβλητή Create χρησιμοποιείται μόνο την πρώτη φορά εκκίνησης της βάσης δεδομένων. Η λειτουργία της είναι να δημιουργήσει έναν πίνακα στη βάση δεδομένων, τον πίνακα στον οποίο θα εισαχθούν οι τιμές των αισθητήρων. Έπειτα η συνάρτηση αυτή ξανά καλείται μόνο σε περίπτωση επανεκκίνησης της βάσης δεδομένων ή του κυκλώματος.

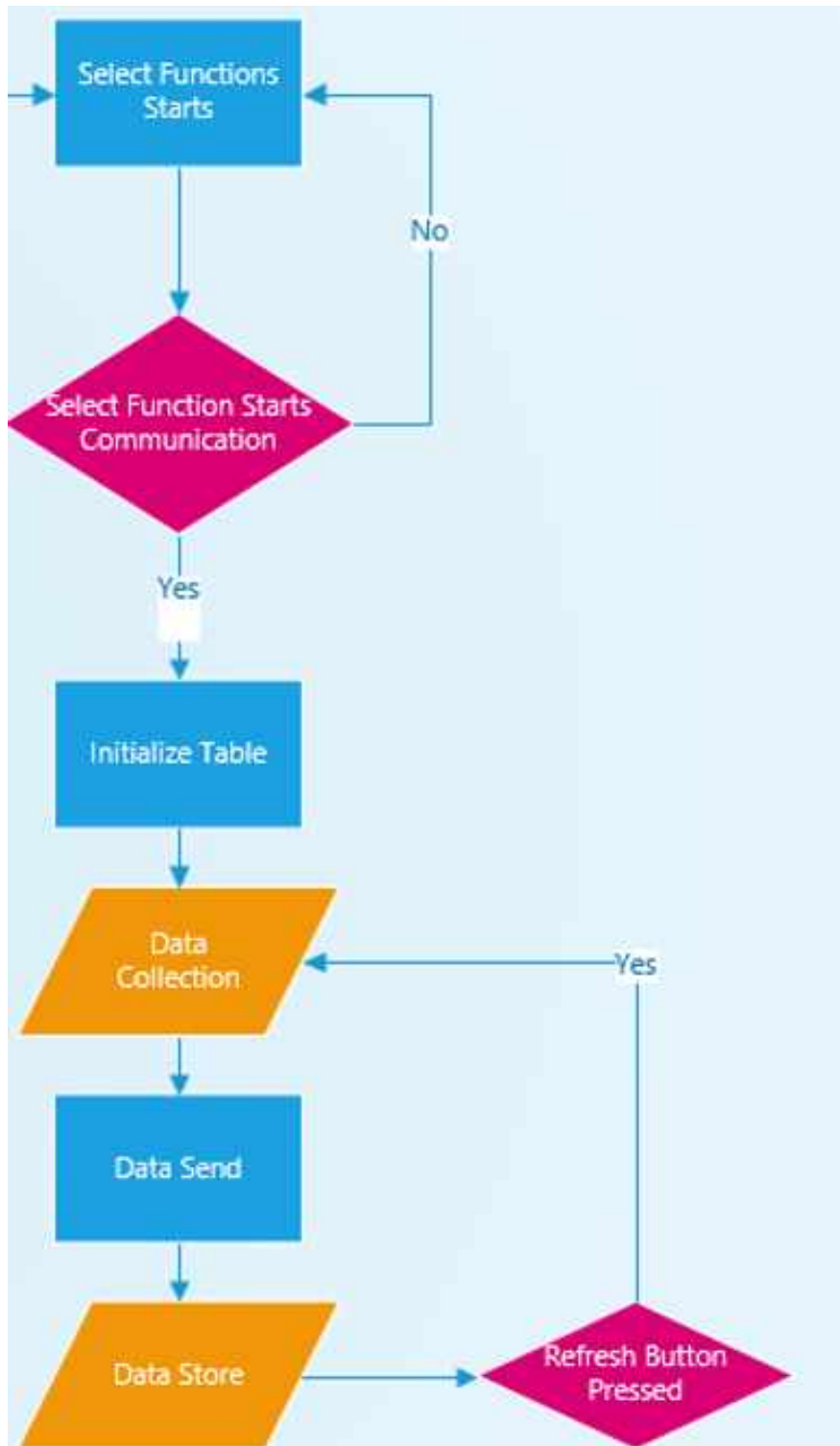
Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)



Διάγραμμα ροής 3: Δυναμικό μέρος κώδικα με την συνάρτηση Create.

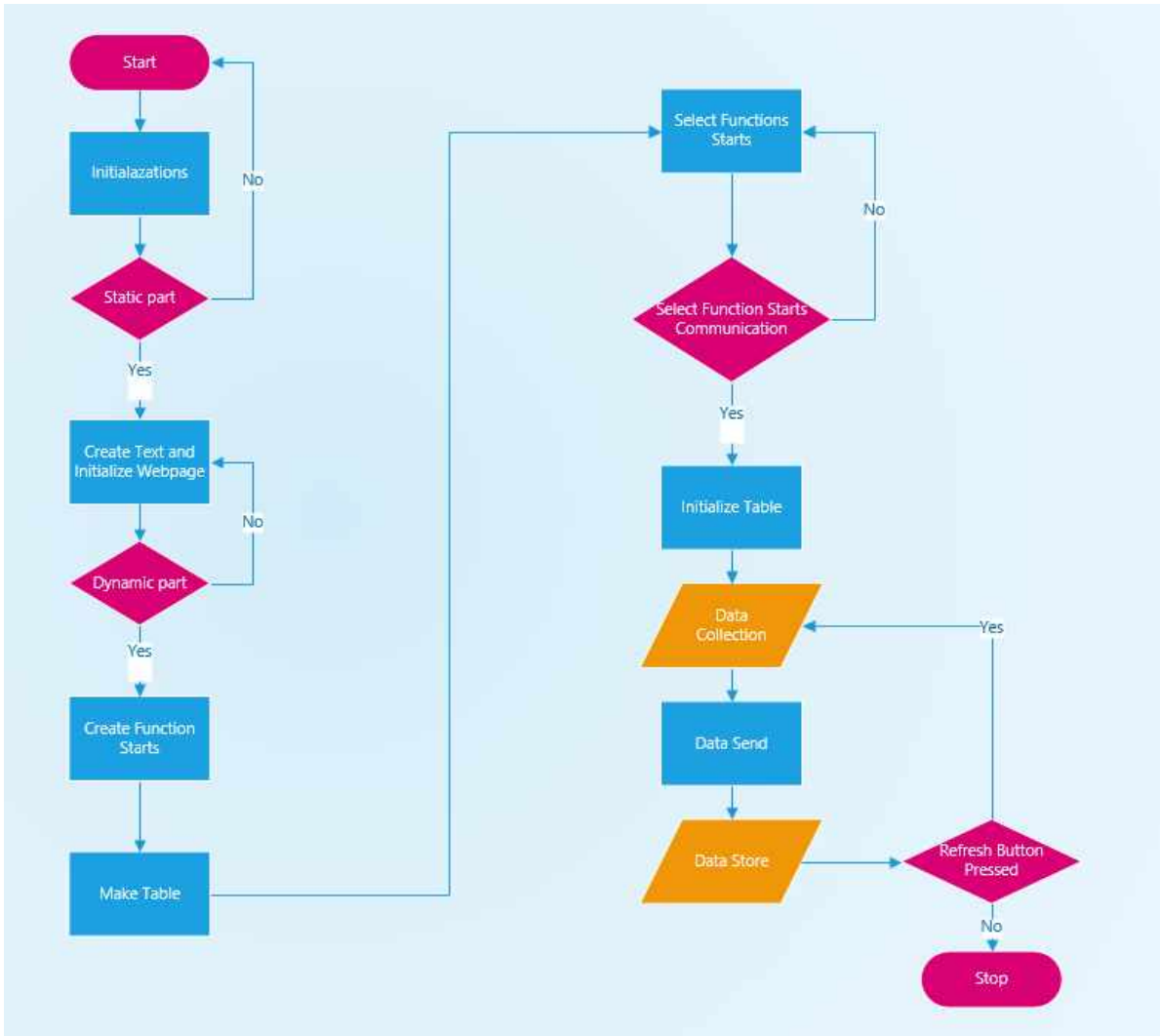
Η συνάρτηση Select είναι αυτή που τοποθετεί τις τιμές των αισθητήρων στη βάση δεδομένων. Είναι αυτή που συμπληρώνει τον πίνακα που δημιουργήθηκε από τη συνάρτηση Create με τις μετρήσεις των αισθητήρων. Είναι επίσης αυτή η οποία επιτρέπει την ενεργοποίηση του κουμπιού Refresh που υπάρχει στην ιστοσελίδα, για να ανανεώσει το περιεχόμενό της. Επίσης η συνάρτηση Select είναι αυτή που κρατάει τις τελευταίες τιμές που έχουν ληφθεί από το κύκλωμα σε περίπτωση κάποιου σφάλματος ή απενεργοποίησης του κυκλώματος.

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

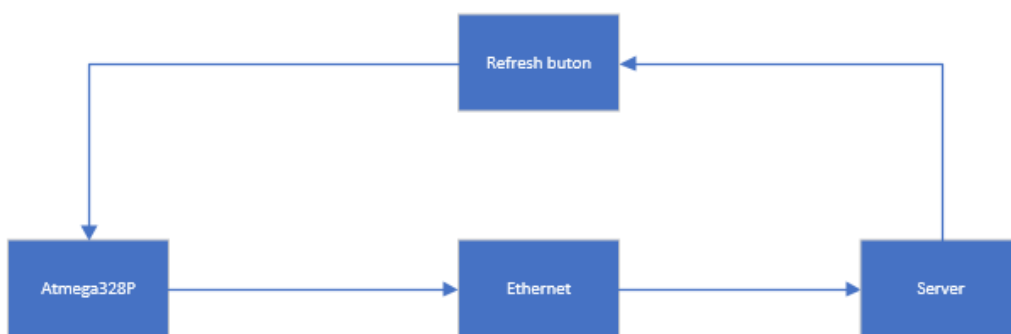


Διάγραμμα ροής 4: Δυναμικό μέρος κώδικα με την συνάρτηση *Select*.

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)



Διάγραμμα ροής 5: Πλήρης κώδικας συλλογής δεδομένων.



Μπλοκ διάγραμμα 2: Πλήρης λειτουργία κώδικα.

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

#### *4.3. Ιστοσελίδα αποτελεσμάτων*

Η ιστοσελίδα στην οποία ανεβαίνουν τα αποτελέσματα είναι μία απλή σελίδα στον φυλλομετρητή (browser) η οποία είναι προσβάσιμη από τον εξής σύνδεσμο (link):

[http://autosoft.gr/arduino\\_var/?fbclid=IwAR16Xnd4sMHRz1spMtFLBmvuTdPFcBIm3tkm03a1ICZGrvOF5uO0mLsAx\\_c](http://autosoft.gr/arduino_var/?fbclid=IwAR16Xnd4sMHRz1spMtFLBmvuTdPFcBIm3tkm03a1ICZGrvOF5uO0mLsAx_c)

## 5. Συμπεράσματα

Από την εργασία αυτή μπορεί να εξαχθεί το συμπέρασμα πως η κατασκευή μπορεί να είναι αποδοτική στη λήψη περιβαλλοντικών μετρήσεων οι οποίες μπορούν να χρησιμοποιηθούν για την κατανόηση των καιρικών συνθηκών που επικρατούν σε μία περιοχή.

Το επίπεδο λειτουργίας της είναι ικανοποιητικό για τις απλές αυτές μετρήσεις που λαμβάνει. Σε μεταγενέστερη κατάσταση, για βελτίωση των υπαρχόντων μετρήσεων, μπορεί να προστεθεί ένα ανεμόμετρο, με παρόμοια κατασκευή με το τωρινό, με τη χρήση ανεμιστήρα 4 επαφών. Θα δώσει ακριβέστερες μετρήσεις. Επίσης, βελτιώσεις μπορούν να γίνουν και στην ιστοσελίδα όπου αναρτώνται τα αποτελέσματα των μετρήσεων, τροποποιώντας την ώστε να δείχνει μία σειρά από μετρήσεις αυτόματα και όχι μόνο μίας συγκεκριμένης χρονικής στιγμής και αυτό με μη αυτοματοποιημένο τρόπο. Η μετατροπή αυτή θα ήταν σχετικά εύκολο να γίνει, μιας και οι τιμές των αισθητήρων που συλλέγονται ήδη προστίθενται σε μία βάση δεδομένων η οποία αποτελείται από έναν πίνακα τιμών.

Εν κατακλείδι, η εργασία αυτή δημιουργήθηκε για να μετράει αποδοτικά το περιβάλλον στο οποίο βρίσκεται. Οι μετρήσεις που λαμβάνει κρίνονται επαρκείς σε ακρίβεια καθώς και τα χρονικά διαστήματα που μεσολαβούν μεταξύ των μετρήσεων. Έχει σκοπό την ενημέρωση και την κατανόηση του περιβάλλοντος καθώς και την αποφυγή όσο το δυνατόν περισσότερων λαθών κατά τη χρήση της.

## Βιβλιογραφικές αναφορές

1. Βίγκλας, Π. (επιμ.) (2007). *Εισαγωγή στη Μετεωρολογία: Μία Εκπαιδευτική Προσέγγιση. Μέρος Α': Βασικές Έννοιες*. Μακρινίτσα Πηλίου: Υπουργείο Εθνικής Παιδείας και Θρησκευμάτων – Κέντρο Περιβαλλοντικής Εκπαίδευσης Μακρινίτσας. Ανακτήθηκε στις 18/09/2021 από: <https://meteovyronas.gr/pdf/meteo.pdf>
2. <https://www.ablic.com/en/semicon/products/sensor/magnetism-sensor-ic/intro/> (Ανακτήθηκε στις 18/09/2021).
3. <https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp280/> (Ανακτήθηκε στις 18/09/2021).
4. <https://www.codrey.com/electronic-circuits/3-wire-pc-fan-tips-tricks/> (Ανακτήθηκε στις 18/09/2021).
5. <https://www.comsol.com/multiphysics/piezoresistive-effect> (Ανακτήθηκε στις 18/09/2021).
6. <https://dotnet.microsoft.com/apps/aspnet> (Ανακτήθηκε στις 18/09/2021).
7. <https://www.easycalculation.com/unit-conversion/rpm-conversion.php> (Ανακτήθηκε στις 18/09/2021).
8. <https://www.elprocus.com/dht22-pin-diagram-circuit-and-its-applications/> (Ανακτήθηκε στις 18/09/2021).
9. <https://www.espruino.com/datasheets/DHT22.pdf> (Ανακτήθηκε στις 18/09/2021).
10. <https://www.espruino.com/DHT22> (Ανακτήθηκε στις 18/09/2021).
11. <https://www.ibm.com/docs/en/zos/2.2.0?topic=internets-tcpip-tcp-udp-ip-protocols> (Ανακτήθηκε στις 18/09/2021).
12. <https://landing.coolermaster.com/faq/3-pin-and-4-pin-fan-wire-diagrams/> (Ανακτήθηκε στις 18/09/2021).
13. <https://laptop.gemwon.com/blog/Difference-in-2-Pin-Fan-and-3-or-4-Pin-Fan> (Ανακτήθηκε στις 18/09/2021).
14. <https://www.microchip.com/en-us/product/ATmega328P#document-table> (Ανακτήθηκε στις 18/09/2021).

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

15. [https://www.mouser.com/catalog/specsheets/a000056\\_datasheet.pdf](https://www.mouser.com/catalog/specsheets/a000056_datasheet.pdf) (Ανακτήθηκε στις 18/09/2021).
16. <https://gr.mouser.com/ProductDetail/Bosch-Sensortec/BMP280?qs=QhAb4EtQfbUP9Z%252BCHM3Wyg%3D%3D> (Ανακτήθηκε στις 18/09/2021).
17. <https://www.openimpulse.com/blog/products-page/product-category/bmp280-barometric-pressure-sensor-module-2/> (Ανακτήθηκε στις 18/09/2021).
18. [https://www.researchgate.net/publication/334329108\\_Intelligent\\_Opening\\_and\\_Closing\\_System\\_of\\_Doors\\_and\\_Windows#pf3](https://www.researchgate.net/publication/334329108_Intelligent_Opening_and_Closing_System_of_Doors_and_Windows#pf3) (Ανακτήθηκε στις 18/09/2021).
19. <https://visualstudio.microsoft.com/> (Ανακτήθηκε στις 18/09/2021).

## Παράρτημα

### A. Κώδικας μικροελεγκτή (C++)

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

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

```
#include <math.h>
```

```
#include <Wire.h>
```

```
#include <Adafruit_Sensor.h>
```

```
#include <Adafruit_BMP280.h>
```

```
#include <DHT.h>
```

```
#include <Arduino.h>
```

```
#include "FanMonitor.h" //Αρχικοποιήσεις
```

```
#define BMP280_I2C_ADDRESS 0x76
```

```
#define DHTPIN 2 // what pin we're connected to
```

```
#define DHTTYPE DHT22 // DHT 22 (AM2302)
```

```
#define FAN_MONITOR_PIN 5 // Δήλωση ποδιών αισθητήρων στον μικροελεγκτή
```

```
Adafruit_BMP280 bmp280;
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
FanMonitor _fanMonitor = FanMonitor(FAN_MONITOR_PIN, FAN_TYPE_BIPOLE); //  
ενεργοποίηση αισθητήρων
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
String str; // Δημιουργία String
```

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // { 0x00, 0x26, 0x55, 0x39, 0x04, 0x4B }; // Αρχικοποίηση Ethernet
```

```
char server[] = "www.autosoft.gr"; // Διεύθυνση για Google (χρησιμοποιώντας DNS)
```

```
// Θέτω στατική IP σε περίπτωση που δεν την πάρει αυτόματα.
```

```
IPAddress ip(79,103,168,224); // ip(10,11,15,192); // η IP μου
```

```
// Αρχικοποίηση βιβλιοθήκης Ethernet
```

```
// Με την πόρτα και την IP του Servers
```

```
// που θέλουμε να χρησιμοποιήσουμε (η πόρτα 80 είναι η προκαθορισμένη)
```

```
EthernetClient client;
```

```
void setup() {
```

```
    // Αρχή σειριακής επικοινωνίας και αναμονή ανταπόκρισης:
```

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

```
    dht.begin();
```

```
    _fanMonitor.begin();
```

```
while (!Serial) {  
  
  ; // Αναμονή  
  
}  
  
if (Ethernet.begin(mac) == 0) {  
  
  Serial.println("Failed to configure Ethernet using DHCP");  
  
  // Λήξη λειτουργίας σε περίπτωση λάθους  
  
  Ethernet.begin(mac, ip);  
  
}  
  
}  
  
void loop()  
  
{  
  
  uint16_t rpm = _fanMonitor.getSpeed();  
  
  float Temp1 = bmp280.readTemperature(); //  
  
  float Pressur = ((bmp280.readPressure()/100)/1000);  
  
  float Altitude = ((bmp280.readAltitude()+10);  
  
  float H = dht.readHumidity();
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
float Temp2 = dht.readTemperature();
```

```
float TempF = ((Temp1 + Temp2)/2);
```

```
float Speed = (120*rpm*0.001885); //συλλογή δεδομένων
```

```
// αναμονή για το Ethernet:
```

```
delay(1000);
```

```
//Serial.println("connecting...");
```

```
//
```

```
// if you get a connection, report back via serial:
```

```
if (client.connect(server, 80)) {
```

```
while (client.connected()) {
```

```
    client.println("GET
```

```
/var/ArduinoToSQL.aspx?var1=TempF&var2=Pressur&var3=Altitude&var4=H&var5=Speed  
= HTTP/1.1"); // Μεταφορά μεταβλητών στο string
```

```
    //client.println("GET
```

```
/var/ArduinoToSQL.aspx?var1=0909&var2=&var3=0909&var4=4578&var5="+stringtemper  
atureC+=" HTTP/1.1");
```

```
    client.println("Host: www.autosoft.gr");
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
//client.println("Connection: close");

client.println();

delay(5000);

client.stop();

}

}

else {

// kf you didn't get a connection to the server:

//Serial.println("connection failed");

}

}
```

*B. Κώδικας SERVER (C#)*

*B.i. Το στατικό κομμάτι*

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %> //αρχικοποίηση ιστοσελίδας
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml" > //σύνδεση σε html σελίδα
```

```
<head runat="server">
```

```
<title></title>
```

```
</head>
```

```
<body>
```

```
<form id="form1" runat="server"> //Δημιουργία διαύλου επικοινωνίας
```

```
<div>
```

```
<asp:Label ID="Label1" runat="server" Text="Θερμοκρασία  
(οC):"></asp:Label><asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br />
```

```
<asp:Label ID="Label2" runat="server" Text="Ατμοσφαιρική πίεση  
(bar):"></asp:Label><asp:TextBox ID="TextBox2" runat="server"></asp:TextBox><br />
```

```
<asp:Label ID="Label3" runat="server" Text="Υψόμετρο  
(m):"></asp:Label><asp:TextBox ID="TextBox3" runat="server"></asp:TextBox><br />
```

```
<asp:Label ID="Label4" runat="server" Text="Υγρασία  
(%):"></asp:Label><asp:TextBox ID="TextBox4" runat="server"></asp:TextBox><br />
```

```
<asp:Label ID="Label5" runat="server" Text="Ταχύτητα ανέμου  
(km/h):"></asp:Label><asp:TextBox ID="TextBox5" runat="server"></asp:TextBox><br />
```

```
<asp:Label ID="Label6" runat="server" Text="Date-  
Time:"></asp:Label><asp:TextBox ID="TextBox6" runat="server"></asp:TextBox><br />
```

```
// δημιουργία ιστοσελίδας με πίνακα
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
<asp:Button ID="Button1" runat="server" Text="Refresh" OnClick="Button1_Click"
/>

</div> // δημιουργία κουμπιού Refresh

</form>

</body>

</html>
```

*B.ii. Το δυναμικό κομμάτι*

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Configuration;

using System.Web.Security;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls.WebParts;

using System.Data.OleDb;

using System.IO;
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
using System.Net;

using System.Net.Mail;

using System.Threading;

using System.Net.Mime;

using System.Data.SqlClient;

using System.Data;                                     //βιβλιοθήκες

public partial class _Default : System.Web.UI.Page

{

    private SQL newSQL = new SQL();                     //κλάση

    private string Exception_ = "";

    protected void Page_Load(object sender, EventArgs e)

    {

        newSQL.select_(ref Exception_);

        this.TextBox1.Text = newSQL.Sensor1_;

        this.TextBox2.Text = newSQL.Sensor2_;

        this.TextBox3.Text = newSQL.Sensor3_;

        this.TextBox4.Text = newSQL.Sensor4_;

        this.TextBox5.Text = newSQL.Sensor5_;
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
        this.TextBox6.Text = newSQL.Time_;
    }

protected void Button1_Click(object sender, EventArgs e)
{
    newSQL.select_(ref Exception_);    // Η συνάρτηση select τοποθετεί τις τιμές

    this.TextBox1.Text = newSQL.Sensor1_;

    this.TextBox2.Text = newSQL.Sensor2_;

    this.TextBox3.Text = newSQL.Sensor3_;

    this.TextBox4.Text = newSQL.Sensor4_;

    this.TextBox5.Text = newSQL.Sensor5_;

    this.TextBox6.Text = newSQL.Time_;
}

//*****
//*****
//*****

public partial class SQL
{
    string    connectionString_    =    "Data    Source=5.172.201.22,1524;Initial
Catalog=autosoftdata;User    ID=samaras;Password="XXXXXXXXX";    //πρόσβαση σε
```

ιδιωτικό server

```
public Int32 ID_ = 0;

public string Sensor1_ = "";

public string Sensor2_ = "";

public string Sensor3_ = "";

public string Sensor4_ = "";

public string Sensor5_ = "";

public string Time_ = "";

public DateTime DateTime_ = DateTime.Now;    //Ενημέρωση ώρας και ημερομηνίας
στο Αμερικάνικο πρότυπο προβολής

public SQL()

{ }

public void select_(ref string Exception_)

{

    using (SqlConnection connection = new SqlConnection(connectionString_)
//SqlDataSource1.ConnectionString))    // Δημιουργία σύνδεσης

    {

        SqlCommand command = connection.CreateCommand();
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
try
{
    connection.Open();

    string queryString = "Select Sensor1,Sensor2,Sensor3,Sensor4,Sensor5,Time
FROM Stauros WHERE ID=(SELECT max(ID) FROM Stauros);" // Είσοδος τιμών στην
βάση δεδομένων με το όνομα Stauros

    command.CommandText = queryString;

    SqlDataReader reader = command.ExecuteReader();

    if (reader.Read())
    {
        Sensor1_ = reader.GetValue(0).ToString();

        Sensor2_ = reader.GetValue(1).ToString();

        Sensor3_ = reader.GetValue(2).ToString();

        Sensor4_ = reader.GetValue(3).ToString();

        Sensor5_ = reader.GetValue(4).ToString();

        Time_ = reader.GetValue(5).ToString(); //Συλλογή δεδομένων από
αισθητήρες
    }

    reader.Close();

    connection.Close();
}
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
    }

    catch (Exception ex)

    {

        Exception_ = ex.ToString();

    }

}

}

}

public void Create_(ref string Exception_)

{

    using (SqlConnection connection = new SqlConnection(connectionString_))
//SqlDataSource1.ConnectionString))

    {

        SqlCommand command = connection.CreateCommand();

        try

        {

            connection.Open();

            string queryString = "CREATE TABLE Stauros (" +

                "ID int, " +
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
"Sensor1 varchar(255)," +  
  
"Sensor2 varchar(255)," +  
  
"Sensor3 varchar(255)," +  
  
"Sensor4 varchar(255)," +  
  
"Sensor5 varchar(255)," +  
  
"Time Datetime)"; //δημιουργία πίνακα δεδομένων  
  
command.CommandText = queryString;  
  
SqlDataReader reader = command.ExecuteReader();  
  
reader.Close();  
  
connection.Close();  
  
}  
  
catch (Exception ex)  
  
{  
  
    Exception_ = ex.ToString();  
  
}  
  
}  
  
}  
  
}
```

*B.iii. Το στατικό κομμάτι*

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ArduinoToSQL.aspx.cs"
Inherits="ArduinoToSQL" %> //κενό κομμάτι
```

*B.iv. Το δυναμικό κομμάτι*

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Configuration;

using System.Web.Security;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls.WebParts;

using System.Data.OleDb;

using System.IO;

using System.Net;

using System.Net.Mail;

using System.Threading;
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
using System.Net.Mime;
```

```
using System.Data.SqlClient;
```

```
using System.Data;
```

```
public partial class ArduinoToSQL : System.Web.UI.Page
```

```
{
```

```
    private string Rfid = "";
```

```
    private string var1 = "n";
```

```
    private string var2 = "n";
```

```
    private string var3 = "n";
```

```
    private string var4 = "n";
```

```
    private string var5 = "n";
```

```
    private SQL newSQL = new SQL();
```

```
    private string Exception_ = "";
```

```
protected void Page_Load(object sender, EventArgs e)
```

```
{
```

```
    //asp.net get variables from url
```

```
    if (Request.QueryString["var1"] != null) var1 = Request.QueryString["var1"];
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
if (Request.QueryString["var2"] != null) var2 = Request.QueryString["var2"];
```

```
if (Request.QueryString["var3"] != null) var3 = Request.QueryString["var3"];
```

```
if (Request.QueryString["var4"] != null) var4 = Request.QueryString["var4"];
```

```
if (Request.QueryString["var5"] != null) var5 = Request.QueryString["var5"];
```

```
//newSQL.Create_(ref Exception_);
```

```
newSQL.insert_(var1, var2, var3, var4, var5, ref Exception_);
```

```
//newSQL.delete_(ref Exception_);
```

```
}
```

```
/**  
**  
**
```

```
public partial class SQL
```

```
{
```

```
    string    connectionString_    =    "Data    Source=5.172.201.22,1524;Initial  
Catalog=autosoftdata;User    ID=samaras;Password=XXXXXXX";    //επικοινωνία με τον  
προσωπικό server
```

```
    public Int32 ID_ = 0;
```

```
    public string Sensor1_ = "";
```

```
    public string Sensor2_ = "";
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
public string Sensor3_ = "";
```

```
public string Sensor4_ = "";
```

```
public string Sensor5_ = "";
```

```
public DateTime DateTime_ = DateTime.Now;
```

```
public SQL()
```

```
{ }
```

```
public void insert_(string Sensor1__, string Sensor2__, string Sensor3__, string  
Sensor4__, string Sensor5__, ref string Exception_)
```

```
{
```

```
using (SqlConnection connection = new SqlConnection(connectionString_))
```

```
{
```

```
SqlCommand command = connection.CreateCommand();
```

```
try
```

```
{
```

```
connection.Open();
```

```
string queryString = "Select MAX(Id) FROM Stauros";
```

```
command.CommandText = queryString;
```

```
SqlDataReader reader = command.ExecuteReader();
```

```
while (reader.Read())
```

```
{
```

```
    if (!reader.IsDBNull(0))
```

```
        ID_ = reader.GetInt32(0) + 1;
```

```
    else ID_ = 1;
```

```
}
```

```
reader.Close();
```

```
queryString = "INSERT INTO Stauros VALUES  
(@pID_,@pSensor1_,@pSensor2_,@pSensor3_,@pSensor4_,@pSensor5_,@pTime)";
```

```
SqlParameter param0 = new SqlParameter("@pID_", SqlDbType.Int);
```

```
command.Parameters.Add(param0);
```

```
command.Parameters[0].Value = ID_;
```

```
SqlParameter param1 = new SqlParameter("@pSensor1_", SqlDbType.VarChar);
```

```
command.Parameters.Add(param1);
```

```
command.Parameters[1].Value = Sensor1__;
```

```
SqlParameter param2 = new SqlParameter("@pSensor2_", SqlDbType.VarChar);
```

```
command.Parameters.Add(param2);

command.Parameters[2].Value = Sensor2__;

SqlParameter param3 = new SqlParameter("@pSensor3_", SqlDbType.VarChar);

command.Parameters.Add(param3);

command.Parameters[3].Value = Sensor3__;

SqlParameter param4 = new SqlParameter("@pSensor4_", SqlDbType.VarChar);

command.Parameters.Add(param4);

command.Parameters[4].Value = Sensor4__;

SqlParameter param5 = new SqlParameter("@pSensor5_", SqlDbType.VarChar);

command.Parameters.Add(param5);

command.Parameters[5].Value = Sensor5__;

SqlParameter param6= new SqlParameter("@pTime", SqlDbType.DateTime);

command.Parameters.Add(param6);

command.Parameters[6].Value = DateTime.Now; //Δημιουργία μεταβλητών στην
βάση δεδομένων

command.CommandText = queryString;

reader = command.ExecuteReader();

reader.Close();

connection.Close();
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
    }  
  
    catch (Exception ex)  
  
    {  
  
        Exception_ = ex.ToString();  
  
    }  
  
    }  
  
    }
```

`public void Create_(ref string Exception_)` // συνάρτηση create τρέχει μόνο μια φορά στην αρχή και ποτέ ξανά. Δημιουργεί τους πίνακες.

```
{  
  
    using (SqlConnection connection = new SqlConnection(connectionString))  
  
    {  
  
        SqlCommand command = connection.CreateCommand();  
  
  
  
        try  
  
        {  
  
            connection.Open();  
  
  
  
            string queryString = "CREATE TABLE Stauros (" +  
  
                "ID int, " +
```

Μελέτη και κατασκευή κυκλώματος συλλογής και μεταφοράς μετεωρολογικών μετρήσεων (θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας ανέμου)

```
"Sensor1 varchar(255)," +  
  
"Sensor2 varchar(255)," +  
  
"Sensor3 varchar(255)," +  
  
"Sensor4 varchar(255)," +  
  
"Sensor5 varchar(255)," +  
  
"Time Datetime)";  
  
command.CommandText = queryString;  
  
SqlDataReader reader = command.ExecuteReader();  
  
reader.Close();  
  
connection.Close();  
  
}  
  
catch (Exception ex)  
  
{  
  
    Exception_ = ex.ToString();  
  
}  
  
}  
  
}  
  
}
```