



# **ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

## **ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

### **ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

#### **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

#### **IoT Μετεωρολογικός Σταθμός**

**Νικολιδάκης Στυλιανός**

**Βαλσαμάκης Κωνσταντίνος**

**Εισηγητής: Δρ Ιωάννης Έλληνας, Καθηγητής**



## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

### **ΑΥΤΟΝΟΜΟΣ ΙΟΤ ΜΕΤΕΩΡΟΛΟΓΙΚΟΣ ΣΤΑΘΜΟΣ**

**Κωνσταντίνος Βαλσαμάκης**

**A.M. 43091**

**Στυλιανός Νικολιδάκης**

**A.M. 42071**

**Εισηγητής: Δρ. Ιωάννης Έλληνας, Καθηγητής**

**Εξεταστική επιτροπή:**

**Ημερομηνία εξέτασης**



## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/Η κάτωθι υπογεγραμμένος/η .....Βαλσαμάκης Κωνσταντίνος....., του .....Γερασίμου....., με αριθμό μητρώου .....43091..... 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 δμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»



## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/Η κάτωθι υπογεγραμμένος/η .....Νικολιδάκης Στυλιανός....., του .....Γεωργίου....., με αριθμό μητρώου .....42071..... 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 δμήνου 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 ήρθε εις πέρας μετά από πολλές δυσκολίες και επίπονες προσπάθειες καθώς η πρακτική άσκηση και ο Ελληνικός Στρατός κατέστησαν το έργο μας ακόμη πιο δύσκολο. Το συγκεκριμένο γνωστικό πεδίο, αυτό των IoT εφαρμογών με μικροελεγκτές, κίνησε εξ αρχής το ενδιαφέρον και των 2 μας καθώς είναι ταχύτατα αναπτυσσόμενο και πρόκειται ν' αλλάξει άρδην την καθημερινότητα του ανθρώπου.

Κομβικό ρόλο στη επιλογή του αντικειμένου έπαιξαν οι καθηγητές μας κ. Έλληνας, κ. Βουτσινάς και κ. Μαστροκώστας που μας μύησαν στους μικροελεγκτές και τη μηχανική μέσα από τις διαλέξεις και τις εργασίες που μας ανέθεσαν. Ευχαριστούμε θερμά τις οικογένειές μας για τη συνεχή στήριξη όσο υπομέναμε τις αντίξοες συνθήκες του Ελληνικού εκπαιδευτικού συστήματος.

## ΠΕΡΙΛΗΨΗ

Σκοπός της συγκεκριμένης εργασίας είναι η μέτρηση και η καταγραφή των καιρικών συνθηκών καθώς και η πρόβλεψη των καιρικών φαινομένων ενός μικροκλίματος. Αρχικά με τη χρήση αισθητήρων μετράται υγρασία, θερμοκρασία, ατμοσφαιρική πίεση, κατεύθυνση και ταχύτητα ανέμου. Στη συνέχεια μέσω της παρατήρησης της μεταβολής της βαρομετρικής πίεσης σε συνάρτηση με την κατεύθυνση του ανέμου πραγματοποιείται πρόβλεψη καιρού. Τα μετρούμενα μεγέθη και η πρόβλεψη αποστέλλονται μέσω WiFi στο smartphone του χρήστη και σε μια IoT υπηρεσία επεξεργασίας δεδομένων. Το πλεονέκτημα και η ιδιαιτερότητα του συστήματος είναι ότι διαθέτει πλήρη αυτονομία χάρη στο συνδυασμό ενός ηλιακού panel και μίας Li-Ion μπαταρίας καθώς και τη δυνατότητα αναστολής λειτουργίας για εξοικονόμηση ενέργειας.

**Λέξεις Κλειδιά:** Μετεωρολογικός σταθμός, IoT, Πρόβλεψη καιρού, μέτρηση μικροκλίματος, esp8266, bme280, wemos d1 mini, blynk, Thingspeak.

## ABSTRACT

The purpose of this project is the measurement and registry of weather conditions as well as the prediction of the weather in a microclimate. Firstly using sensors, humidity, temperature, atmospheric pressure, wind direction and speed are measured. By observing the change in barometric pressure in relation to the wind direction the weather can be predicted. The measured data and the prediction are sent through WiFi to the user's smarphone and to an IoT data processing service. The project's advantage is that it is fully autonomous thanks to a combination of a solar panel and a LiPo battery as well as the ability to hibernate to save power.

**Key Words:** Weather station, IoT, Weather forecasting, microclimate, esp8266,bme280, wemos d1 mini, blynk, Thingspeak.



## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ .....	10
1.ΕΙΣΑΓΩΓΗ.....	17
1.1.Η πρόβλεψη σήμερα με τη χρήση μικροελεγκτών .....	17
1.2.Οργάνωση της πτυχιακής εργασίας .....	17
2. Το υλικό μέρος του συστήματος .....	18
2.1. Το Wemos D1 Mini .....	18
2.2. Ο αισθητήρας BME280 .....	19
2.3. Ο φορτιστής TP 4056 .....	19
2.4. Το Ηλιακό Πάνελ.....	20
2.5. Ο πολυπλέκτης CD4051BE .....	20
2.6. Η μπαταρία ιόντων λιθίου 18650.....	20
2.7. Το ανεμόμετρο.....	21
2.8.Ο μικροελεγκτής ESP-8266 .....	21
2.9. Το Τελικό Κύκλωμα .....	21
3.Internet of Things .....	23
3.1.Σύνδεση Esp-8266 μέσω Wi-fi .....	23
3.3.Η πλατφόρμα ThingSpeak™ .....	24
3.4.Η πλατφόρμα Blynk .....	25
4.Πρόβλεψη Καιρού .....	27
4.1. Η Δική μας Υλοποίηση.....	27
5.Περιβάλλον και ανανεώσιμες πηγές ενέργειας.....	28
5.1Ηλιακή ενέργεια .....	28
5.2Φωτοβολταϊκά ηλιακά συστήματα.....	28
5.3.Εξοικονόμηση ενέργειας μέσω προγραμματισμού .....	28
6.Λογισμικό μέρος του συστήματος.....	29
6.1.Η συνάρτηση sleepMeasurements( ) .....	29
6.2.Η συνάρτηση liveMeasurements ( ) .....	32

6.3.Η συνάρτηση prediction ( ).....	33
6.4.Η συνάρτηση postData ( ).....	35
6.5.Οι συναρτήσεις BLYNK_WRITE ( ) και BLYNK_CONNECTED ( ) .....	36
6.6.Η συνάρτηση changeMux ( ).....	36
6.7.Η συνάρτηση isr_rotation ( ) .....	37
6.8.Η συνάρτηση setup ( ) .....	37
6.9.Βιβλιοθήκες, αντικείμενα και σταθερές .....	38
7.Κατασκευαστικό μέρος του συστήματος.....	39
7.1.Η Βάση .....	39
7.2.Το κουτί .....	39
7.3.Η πλακέτα.....	40
8.Προβλήματα.....	41
8.1.Πρόβλημα διατήρησης δεδομένων μετά το deepSleep .....	41
8.2.Ο προβληματικός αισθητήρας MH-Z19.....	41
8.3.Έλλειψη εγχειριδίου ανεμόμετρου .....	41
9.Σύγκριση με σταθμό εμπορίου .....	42
10.ΒΙΒΛΙΟΓΡΑΦΙΑ.....	45

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Wemos D1 mini .....	18
Εικόνα 2: BME280 χωρίς module .....	19
Εικόνα 3: BME280 εντός module.....	19
Εικόνα 4: TP-4056 charger module .....	19
Εικόνα 5: Mini solar panel .....	20
Εικόνα 6: CD4051BE multiplexer .....	20
Εικόνα 7: 18650 Li-Ion battery.....	20
Εικόνα 8: Ανεμόμετρο .....	21
Εικόνα 9: ESP-8266.....	21
Εικόνα 10: Το τελικό κύκλωμα σε breadboard.....	22
Εικόνα 11:Ο esp8266 σε λειτουργία station.....	23
Εικόνα 12:Ping request στην ip διεύθυνση του esp8266 .....	24
Εικόνα 13:Καρτέλα επίβλεψης του συστήματος.....	26
Εικόνα 14:Η κύρια καρτέλα της εφαρμογής blynk.....	26
Εικόνα 15:Παραδείγματα παραμετροποίησης εικονικών πορτών .....	26
Εικόνα 16: Κατανάλωση καταστάσεων λειτουργίας ESP8266 .....	28
Εικόνα 17: Η βάση με το σωλήνα βιδωμένο .....	39
Εικόνα 18: Το κουτί της κατασκευής πριν και μετά το βάψιμο.....	40
Εικόνα 19: Η διαδικασία βαθμονόμησης του ανεμοδουριού.....	42
Εικόνα 20: Ο σταθμός μας δίπλα στον εμπορικό.....	43
Εικόνα 21: Αντιπαραβολή εμπρόσθιων τμημάτων και μετρήσεων με τον εμπορικό μετεωρολογικό σταθμό .....	44

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1:Μπλόκ διάγραμμα κυκλώματος .....	22
Σχήμα 2:Διάγραμμα υγρασίας σε συνάρτηση με το χρόνο.....	24
Σχήμα 3: Διάγραμμα θερμοκρασίας σε συνάρτηση με το χρόνο .....	25
Σχήμα 4: Διάγραμμα ατμοσφαιρικής πίεσης σε συνάρτηση με το χρόνο.....	25

## **ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ**

**IoT** Internet of Things

**WiFi** Radio wireless local area networking

**Li-Ion** lithium–ion batteries

**CPU** Central Process Unit

**RAM** Random Access Memory

**ROM** Read Only Memory

**I/O** Input/Output

**ADC** Analog to Digital Converter

**UART** Universal Asynchronous Receiver Transmitter

**MSSP** Master Synchronous Serial Port

**SPI** Serial Peripheral Interface

**I2C** Inter-integrated Circuit

**PWM** Pulse Width Modulation

**GPS** Global Positioning System

**RISC** Reduced Instruction Set Computer

**SSID** Service Set Identifier

**PCB** Printed Circuit Board

**SD** Secure Digital

**RTC** Real Time Clock



## 1.ΕΙΣΑΓΩΓΗ

Η ανάγκη για την μέτρηση αλλά και την πρόβλεψη των καιρικών συνθηκών κατέστη αναγκαία από τα πρώιμα χρόνια του ανθρώπινου είδους, λόγω της άρρηκτης σχέσης τους με την καθημερινότητα αλλά και την επιβίωσή του ανθρώπου.

Ο άνθρωπος δεν διαθέτει τις αισθήσεις να ανιχνεύσει τις καιρικές συνθήκες, γι' αυτό και χρειάστηκε να εφεύρει όργανα που να του το επιτρέπουν, με χαρακτηριστικά παραδείγματα τον Galileo Galilei και το πρώτο στοιχειώδες θερμόμετρο νερού το 1593, που είναι και το πρώτο όργανο που επέτρεψε την ανίχνευση μεταβολών της θερμοκρασίας, καθώς και τον Evangelista Torricelli με την εφεύρεση του πρώτου βαρόμετρου το 1647.

Μετά την μέτρηση των καιρικών συνθηκών παρουσιάστηκε η ανάγκη για την πρόβλεψη των καιρικών φαινομένων που κατέστη δυνατή μέσω της κατανόησης της ατμοσφαιρικής φυσικής. Πιο συγκεκριμένα, το 1922 ο Lewis Fry Richardson δημοσιεύει το βιβλίο του " Weather Prediction By Numerical Process".

### 1.1.Η πρόβλεψη σήμερα με τη χρήση μικροελεγκτών

Στην παρούσα πτυχιακή επιτυγχάνεται πρόβλεψη καιρού μέσω της χρήσης μικροελεγκτών και αισθητήρων και εξετάζεται το κατά πόσο ένα σχετικά οικονομικό σύστημα το οποίο ο καθένας μπορεί να υλοποιήσει αν έχει τη δυνατότητα να δώσει πρόβλεψη ακριβείας.

### 1.2.Οργάνωση της πτυχιακής εργασίας

Στα πλαίσια της πτυχιακής εργασίας περιγράφεται αναλυτικά σε κάθε κεφάλαιο οι ανάγκες και με ποιο τρόπο αυτές επιλύθηκαν αλλά και ο τρόπος λειτουργίας και επικοινωνίας όλων των εξαρτημάτων καθώς και η διαδικασία της κατασκευής.

## 2. Το υλικό μέρος του συστήματος

### 2.1. Το Wemos D1 Mini

Ένας μικροελεγκτής (microcontroller) είναι ουσιαστικά ένας μικροσκοπικός υπολογιστής σε ολοκληρωμένο κύκλωμα. Έχει κεντρική μονάδα επεξεργασίας (CPU), μνήμη τυχαίας προσπέλασης (RAM), μνήμη μόνο για ανάγνωση (ROM), πολλαπλές I/O (Input/Output) πόρτες και μπορεί να διαθέτει πλήθος περιφερειακών, όπως για παράδειγμα μετατροπέα αναλογικού σε ψηφιακό (ADC), σειριακό UART, ένα ή και παραπάνω χρονόμετρα, MSSP για SPI ή I2C επικοινωνία κ.α.



Εικόνα 1: Wemos D1 mini

Εμείς επιλέξαμε τον μικροελεγκτή Wemos D1 Mini που είναι βασισμένος στον ESP-8266EX, με flash μνήμη 4 MB, ταχύτητα ρολογιού 80MHz/160MHz, 9 ψηφιακές PWM I/O πόρτες (pins), 1 αναλογική I/O πόρτα, 2 πόρτες τάσης (5V και 3,3V αντίστοιχα), 1 πόρτα γείωσης και 1 πόρτα reset. Ακόμα, είναι συμβατός με την πλατφόρμα ανάπτυξης ανοιχτού λογισμικού Arduino, που διαθέτει μια wire-based γλώσσα προγραμματισμού που επιτρέπει την δημιουργία πολύπλοκων project σε μικρό χρονικό διάστημα. Επιπλέον η γλώσσα μπορεί να επεκταθεί με βιβλιοθήκες της C++ για περισσότερες λειτουργίες.

Ένα από τα πλεονεκτήματα του wemos σε σχέση με άλλες γνωστές αντίστοιχες πλακέτες ανάπτυξης πρωτοτύπων είναι το εξαιρετικά μικρό μέγεθός του (32.2mm x 25.6mm), που το κατέστησε τον τέλειο μικροελεγκτή για το project μας, μιας και το κύκλωμα έπρεπε να καταλαμβάνει όσο το δυνατόν λιγότερο χώρο. Επίσης το wemos έχει την ιδιαιτερότητα ότι περιλαμβάνει το μικροσίπ esp8266 που του δίνει δύο εξαιρετικά σημαντικές για το project μας λειτουργίες, την σύνδεση σε δίκτυο μέσω της TCP/IP στοίβας (stack) του αλλά και την δυνατότητα DeepSleep που βάζει όλο το σύστημα σε μια κατάσταση αδράνειας για ορισμένο χρονικό διάστημα.

## 2.2. Ο αισθητήρας BME280



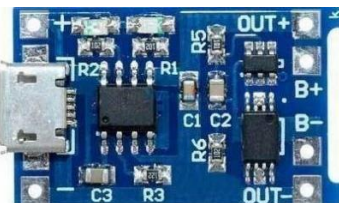
Εικόνα 2: BME280 χωρίς module

Εικόνα 3: BME280 εντός module

Ο BME280 είναι ένας χαμηλού κόστους αισθητήρας υγρασίας, θερμοκρασίας και ατμοσφαιρικής πίεσης της εταιρίας Bosch Sensortec. Είναι αισθητήρας ακριβείας με δυνατότητα μέτρησης υγρασίας με ακρίβεια  $\pm 3\%$ , βαρομετρικής πίεσης με  $\pm 1$  hPa και θερμοκρασίας με  $\pm 1.0^{\circ}\text{C}$ . Επίσης σε αντίθεση με παλιότερους αισθητήρες της εταιρίας ο συγκεκριμένος έχει δυνατότητα σύνδεσης τόσο I2C αλλά και SPI.

Όπως για το wemos έτσι και για τον BME ένα από τα μεγάλα του πλεονεκτήματα είναι το εξαιρετικά μικρό μέγεθός του, με διαστάσεις 2.5mm x 2.5mm x 0.93mm στην γυμνή του μορφή και 1.3mm x 0.9mm σε μορφή module. Επιπλέον έχει εξαιρετικά χαμηλή κατανάλωση στα 3.6mA στον 1Hz data refresh rate.

## 2.3. Ο φορτιστής TP 4056



Εικόνα 4: TP-4056 charger module

Ο TP4056 είναι ένας ολοκληρωμένος γραμμικός φορτιστής συνεχούς ρεύματος/συνεχούς τάσης για μπαταρίες ιόντων λιθίου. Χρησιμοποιείται ευρέως σε φορητές κατασκευές χάρη στο μικρό του μέγεθος και εκτός από φορτιστής λειτουργεί ως ασφάλεια για το κύκλωμα.

## 2.4. Το Ηλιακό Πάνελ



Εικόνα 5: Mini solar panel

Για την φόρτιση της μπαταρίας πέρα από τον TP 4056 χρησιμοποιούμε ένα μικρό ηλιακό πάνελ 6V-583mA, διαστάσεων 16.5cm x 13.5cm.

## 2.5. Ο πολυπλέκτης CD4051BE



Εικόνα 6: CD4051BE multiplexer

Ο CD4051 είναι ένας κλασικός multiplexer που μας επέτρεψε να πολυπλέξουμε την μοναδική αναλογική πόρτα του Wemos παρακάμπτοντας έτσι ένα πολύ μεγάλο μειονέκτημά της συγκεκριμένης πλακέτας σε σχέση με άλλες του εμπορίου, την αδυναμία διαχείρισης πολλαπλών αναλογικών εισόδων.

Για τις ανάγκες του project χρειαζόμασταν μια μόνο παραπάνω αναλογική είσοδο, οπότε όπως φαίνεται και στο Σχήμα 1 γειώσαμε 2 από τα ποδαράκια ορτίση του πολυπλέκτη και συνδέσαμε το 3<sup>ο</sup> σε ψηφιακή είσοδο, με αποτέλεσμα να μπορούμε να εναλλάσσουμε την είσοδο στην αναλογική πόρτα του wemos.

## 2.6. Η μπαταρία ιόντων λιθίου 18650



Εικόνα 7: 18650 Li-Ion battery

Οι μπαταρίες 18650 είναι επαναφορτιζόμενες και χρησιμοποιούνται ευρέως σε φορητά ηλεκτρονικά συστήματα χάρη στην υψηλή πυκνότητα ενέργειας, το πολύ

μικρό battery effect, και χαμηλή αυτοαποφόρτιση. Στην κατασκευή μας χρησιμοποιήσαμε την έκδοση των 3.6V, 3120 mAh, 30A.

## 2.7. Το ανεμόμετρο



Εικόνα 8: Ανεμόμετρο

Τα ψηφιακά ανεμόμετρα αποτελούνται από 2 μέρη, το μέρος που υπολογίζει την ταχύτητα του ανέμου και το μέρος που υπολογίζει την κατεύθυνση του ανέμου. Το 1<sup>ο</sup> μέρος αποτελείται από ένα περιστρεφόμενο «καπάκι» που σε κάθε περιστροφή ανοιγοκλείνει έναν μαγνητικό διακόπτη reed. Το 2<sup>ο</sup> αποτελείται επίσης από ένα περιστρεφόμενο «καπάκι» κουμπωμένο σε ένα ποτενσιόμετρο, το οποίο αντιπροσωπεύει την κατεύθυνση του αέρα. Εμείς χρησιμοποιήσαμε ένα μεταχειρισμένο ανεμόμετρο παλιάς γενιάς με ποτενσιόμετρο των 21kΩ.

## 2.8.Ο μικροελεγκτής ESP-8266

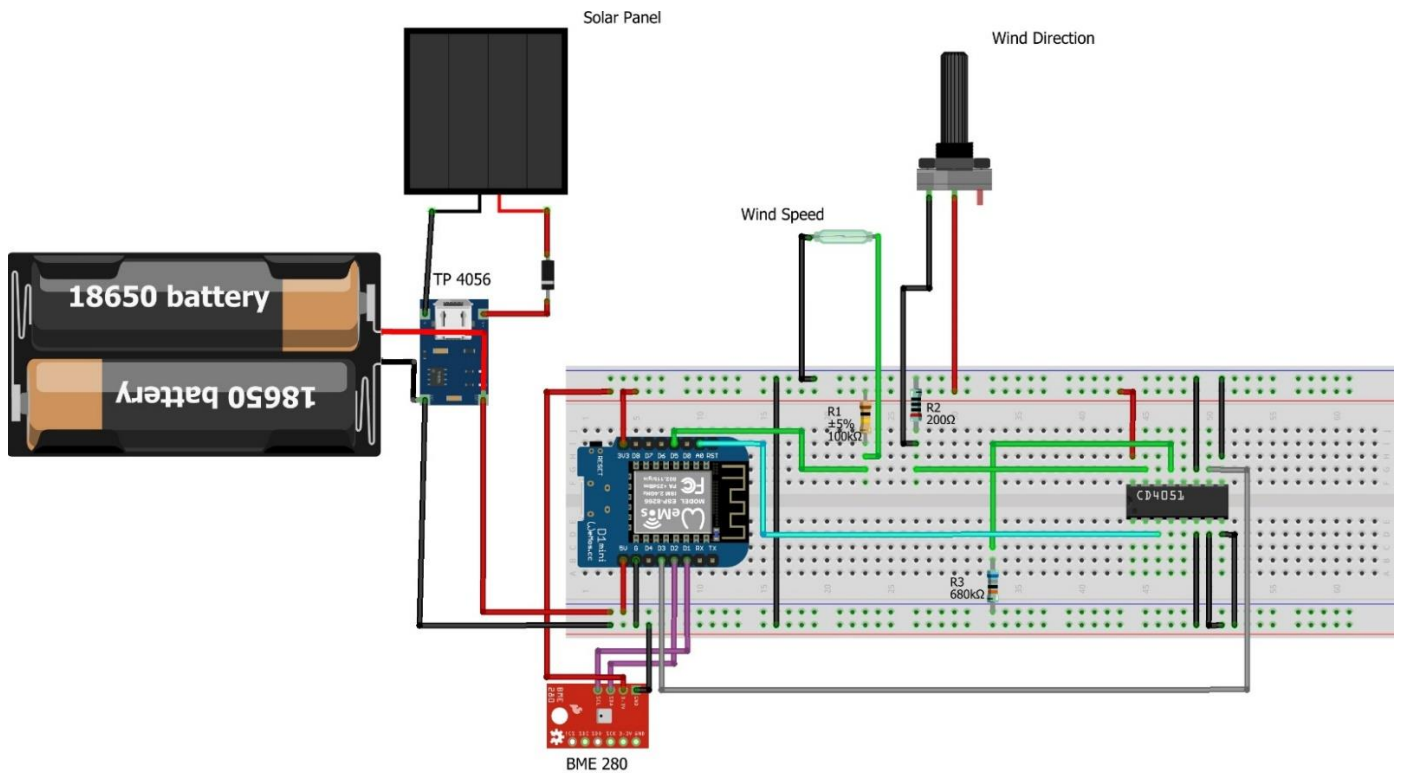


Εικόνα 9: ESP-8266

Το esp8266 είναι ένα μικροσίπ χαμηλού κόστους με πλήρη TCP/IP στοίβα και δυνατότητες μικροελεγκτή (microcontroller) που παράγεται από την Κινέζικη εταιρία Espressif Systems. Έχει ενσωματωμένο τον 32-bit επεξεργαστή τύπου RISC, Tensilica L106 ο οποίος έχει τη δυνατότητα να δουλεύει με ελάχιστη κατανάλωση ρεύματος και φτάνει τα 160 MHz ταχύτητα ρολογιού. Ο esp χάρη στις πλήρεις και αυτοδύναμες δυνατότητες WiFi δικτύωσής του μπορεί να δουλέψει είτε σαν αυτόνομη εφαρμογή ή σαν slave σε κάποιο μικροελεγκτή-οικοδεσπότη. Όπως αναφέρεται στο [κεφάλαιο 2.1](#), εμείς επιλέξαμε τον 2<sup>ο</sup> τρόπο λειτουργίας για την υλοποίηση του συστήματος.

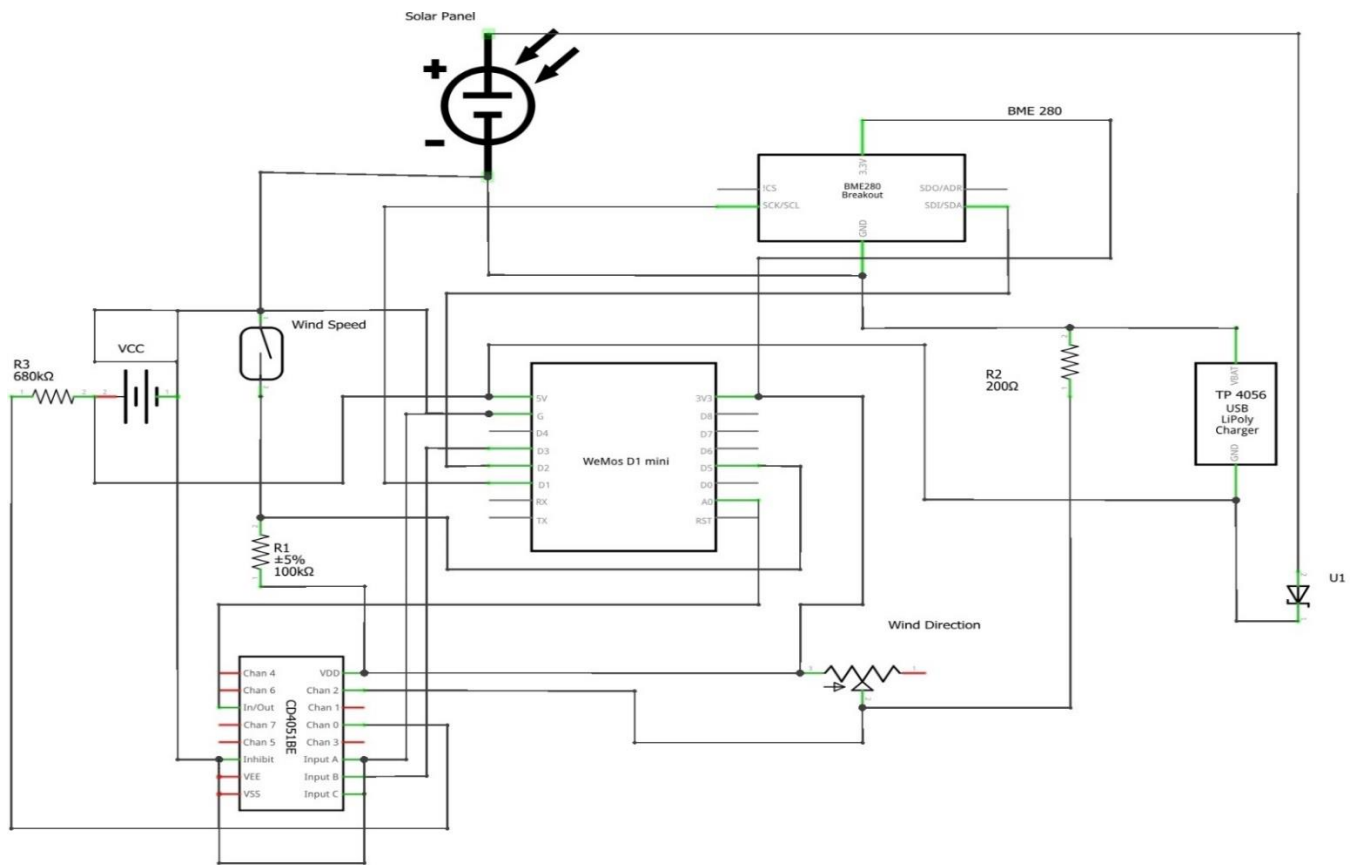
## 2.9. Το Τελικό Κύκλωμα

Συνδυάζοντας όλα τα παραπάνω συνθέσαμε το τελικό κύκλωμα της εργασίας μας το οποίο έχουμε αναλύσει και σε μπλοκ διάγραμμα χρησιμοποιώντας την πλατφόρμα fritzing.



fritzing

Εικόνα 10: Το τελικό κύκλωμα σε breadboard



fritzing

Σχήμα 1: Μπλόκ διάγραμμα κυκλώματος

### 3. Internet of Things

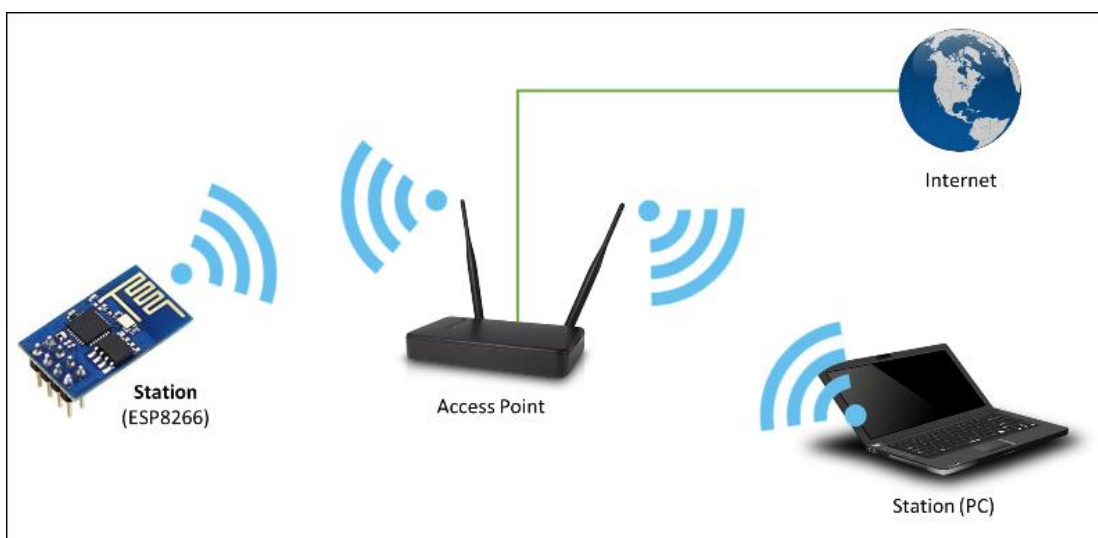
Ο όρος «Διαδίκτυο των Πραγμάτων» χρησιμοποιήθηκε για πρώτη φορά το 1999 από τον βρετανό Kevin Ashton, έναν πρωτοπόρο της τεχνολογίας, για να περιγράψει ένα σύστημα στο οποίο πληθώρα των αντικείμενων του φυσικού κόσμου θα μπορούσαν να συνδεθούν είτε τοπικά, είτε στο Internet ώστε να επιτρέπεται η ανταλλαγή δεδομένων. Είναι κάτι σαν τη 2η γενιά του internet. Στην πρώτη, την πληροφορία την δημιουργούσε ο άνθρωπος με εισαγωγή δεδομένων ενώ στη δεύτερη θα δημιουργείται και θα αποστέλλεται από συσκευές και αισθητήρες.

Το πρώτο «αντικείμενο» (“thing”) το οποίο τροποποιήθηκε ώστε να έχει τη δυνατότητα να συνδέεται στο διαδίκτυο και να δίνει πληροφορίες ήταν ένας αυτόματος πωλητής αναψυκτικών, τον οποίο δημιούργησαν μεταπτυχιακοί φοιτητές στο Πανεπιστήμιο του Carnegie-Mellon.

Με μια πρώτη ματιά δεν αντιλαμβάνεται κανείς τις δυνατότητες και την πραγματική χρήση του IoT, η εφαρμογή του όμως υπόσχεται να μετατρέψει πολλές πτυχές του σύγχρονου τρόπου ζωής. Ήδη κάποια απτά παραδείγματα που έχουν προσφέρει τεράστιες ευκολίες στον καθημερινότητα του ανθρώπου αποτελούν η τοποθέτηση IoT GPS στα λεωφορεία, η απομακρυσμένη παρακολούθηση της αγροτικής παραγωγής, η συνεχής παρακολούθηση της υγείας ασθενών μέσω αισθητήρων και πολλά άλλα.

#### 3.1. Σύνδεση ESP-8266 μέσω Wi-fi

Ο τρόπος διασύνδεσης του μικροελεγκτή στο διαδίκτυο επιτυγχάνεται με 2 τρόπους. Ο πρώτος τρόπος είναι να συνδεθεί κατευθείαν μέσω Wi-Fi στο router όπου και θα αποτελεί ένα ασύρματο Station, ένα κομμάτι του τοπικού δικτύου (LAN). Ο δεύτερος τρόπος είναι να υλοποιηθεί σαν σημείο πρόσβασης (Access Point) στο οποίο μπορούν να συνδεθούν περισσότερες από μία συσκευές, κάτι σαν το Δρομολογητή (Router), κάτι που δεν θα αγγίξουμε στην παρούσα πτυχιακή.



Εικόνα 11: Ο esp8266 σε λειτουργία station

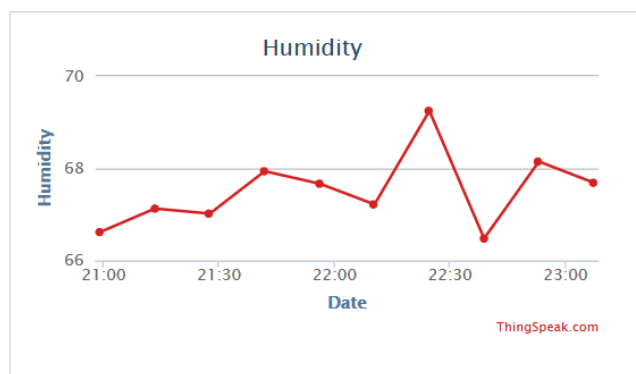
Η σύνδεση πραγματοποιείται με τη συνάρτηση WiFi.begin(), όπου δίνουμε 2 παραμέτρους, το όνομα (SSID) καθώς και τον κωδικό του δικτύου. Μόλις γίνει η σύνδεση και δοθεί IP διεύθυνση στον ESP τότε πρακτικά έχουμε και επικοινωνία μεταξύ των συσκευών του Τοπικού Δικτύου κάτι το οποίο επιβεβαιώνεται και με ένα απλό Ping Request από τον υπολογιστή μας.

```
user@computername:~$ ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
 64 bytes from 192.168.1.3: icmp_seq=1 ttl=128 time=6.38 ms
 64 bytes from 192.168.1.3: icmp_seq=2 ttl=128 time=45.2 ms
 64 bytes from 192.168.1.3: icmp_seq=3 ttl=128 time=69.1 ms
 64 bytes from 192.168.1.3: icmp_seq=4 ttl=128 time=94.0 ms
 64 bytes from 192.168.1.3: icmp_seq=5 ttl=128 time=20.5 ms
 64 bytes from 192.168.1.3: icmp_seq=6 ttl=128 time=7.37 ms
^C
--- 192.168.1.3 ping statistics ---
 6 packets transmitted, 6 received, 0% packet loss, time 5003ms
 rtt min/avg/max/mdev = 6.384/40.463/94.047/32.588 ms
```

Εικόνα 12: Ping request στην ip διεύθυνση του esp8266

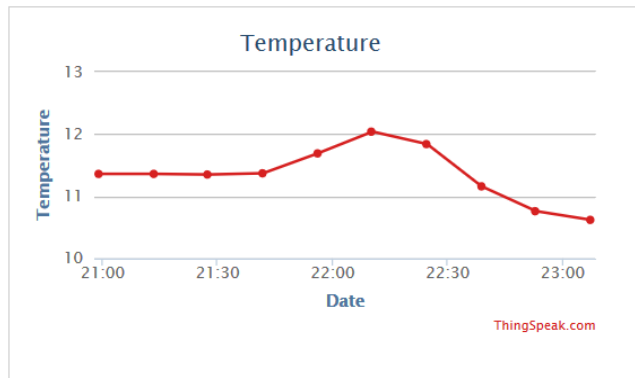
### 3.3.Η πλατφόρμα ThingSpeak™

Το ThingSpeak είναι μια δωρεάν διαδικτυακή πλατφόρμα, που επιτρέπει την συλλογή και την αποθήκευση δεδομένων στο σύννεφο (cloud) αλλά και την ανάπτυξη IoT εφαρμογών. Μια από τις βασικές λειτουργίες του είναι η άμμεση απεικόνιση των δεδομένων που δέχεται από το σύστημα σε γραφήματα που παράγονται μέσω MATLAB, ώστε μετέπειτα να δίνεται η δυνατότητα για online ανάλυση και επεξεργασία. Οι συσκευές επικοινωνούν με το ThingSpeak μέσω ενός RESTful API το οποίο χρησιμοποιεί το πρωτόκολλο μεταφοράς υπερκειμένου (HTTP) για αιτήσεις τύπου GET, PUT, POST, DELETE για την επεξεργασία των δεδομένων. Η ιδιαιτερότητα της συγκεκριμένης κατηγορίας API είναι το χαμηλό bandwidth που απαιτούν η οποία τα καθιστά ιδανικά για χρήση σε ένα δίκτυο, χωρίς να προκαλούν συμφόρηση σε αυτό.

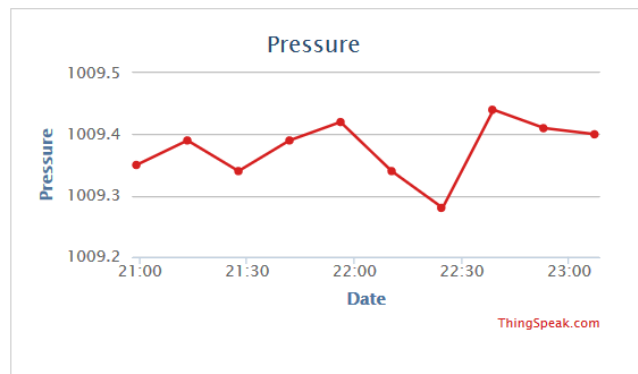


Σχήμα 2: Διάγραμμα υγρασίας σε συνάρτηση με το χρόνο





Σχήμα 3: Διάγραμμα θερμοκρασίας σε συνάρτηση με το χρόνο



Σχήμα 4: Διάγραμμα ατμοσφαιρικής πίεσης σε συνάρτηση με το χρόνο

### 3.4.Η πλατφόρμα Blynk

Η πλατφόρμα blynk σχεδιάστηκε για το IoT και έχει τη δυνατότητα να ελέγξει υλικό απομακρυσμένα, να απεικονίσει δεδομένα αισθητήρων, να αποθηκεύσει δεδομένα αλλά και να τα οπτικοποιήσει. Η πλατφόρμα αποτελείται από 3 μέρη:

- τον blynk server που είναι υπεύθυνος για την επικοινωνία του smartphone με το hardware
- τις βιβλιοθήκες blynk που παρέχουν την επικοινωνία με τον server και επεξεργάζονται όλες τις εισερχόμενες και εξερχόμενες εντολές
- την εφαρμογή blynk που επιτρέπει την δημιουργία διεπαφών με το εκάστοτε project με τη χρήση διαφόρων widgets που παρέχουν οι δημιουργοί του.

Χρησιμοποιώντας την εφαρμογή blynk συνθέσαμε το τελικό εμπρόσθιο μέρος (Front-End) της πτυχιακής μας εργασίας

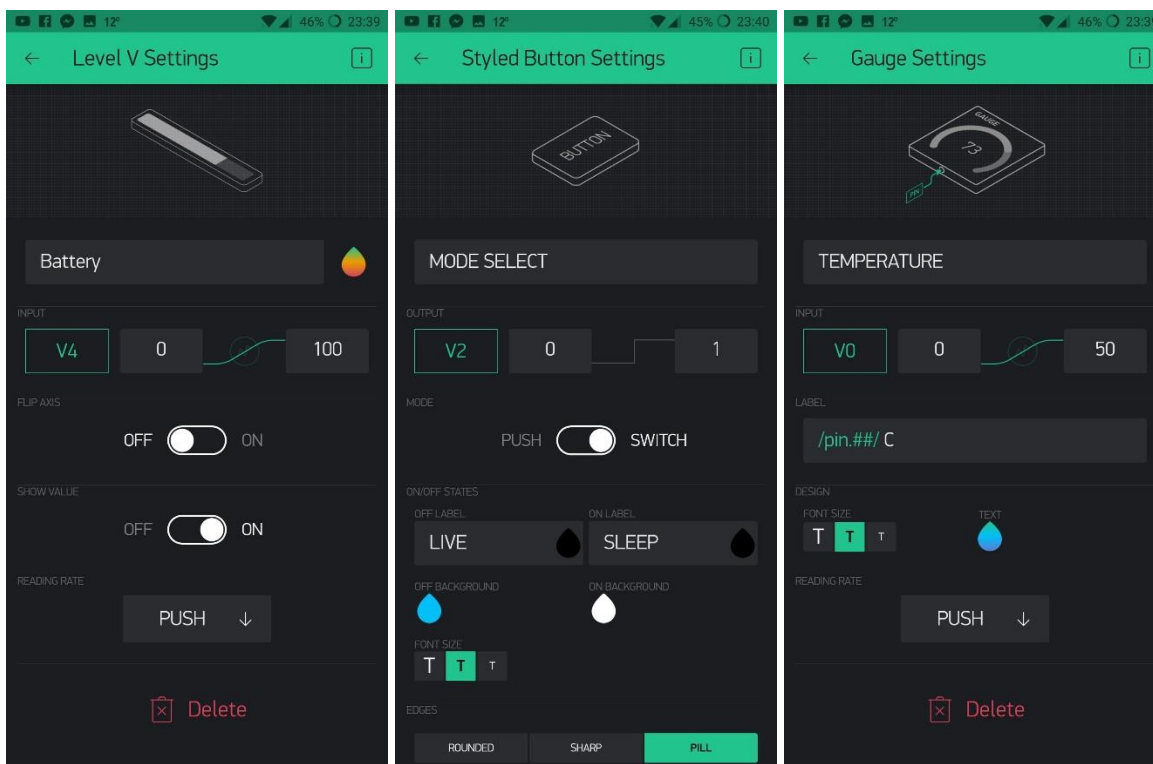


Εικόνα 13:Καρτέλα επίβλεψης του συστήματος



Εικόνα 14:Η κύρια καρτέλα της εφαρμογής blynk

Βασίζεται σε ένα γραφικό προγραμματιστικό περιβάλλον, που χρησιμοποιεί εικονικές πόρτες εισόδου/εξόδου στις οποίες μπορούν να ανατεθούν διάφορες λειτουργίες.



Εικόνα 15:Παραδείγματα παραμετροποίησης εικονικών πορτών

## 4.Πρόβλεψη Καιρού

Η τέχνη της πρόβλεψης του καιρού ξεκίνησε από τους αρχαίους λαούς που χρησιμοποιούσαν επαναλαμβανόμενα αστρονομικά και μετεωρολογικά φαινόμενα για να παρατηρήσουν εποχιακές αλλαγές στον καιρό. Αρχικά οι Βαβυλώνιοι το 650 π.Χ. και αργότερα οι Κινέζοι το 300 π.Χ. προσπάθησαν να αποκρυπτογραφήσουν και να κατηγοριοποιήσουν τα καιρικά φαινόμενα. Περί τα 340 π.Χ. ο Αριστοτέλης γράφει το «Μετεωρολογικά», μια φιλοσοφική πραγματεία που περιείχε θεωρίες για το σχηματισμό διαφόρων καιρικών φαινομένων. Για 2000 χρόνια οι θεωρίες του Αριστοτέλη αποτελούσαν αρχές της θεωρίας του καιρού.

Με την πάροδο των χρόνων και πιο συγκεκριμένα προς το τέλος της Αναγέννησης ήταν ξεκάθαρο πλέον ότι οι φιλοσοφικές μέθοδοι πρόβλεψης των προηγούμενων χιλιετιών ήταν ανεπαρκείς και ήταν απαραίτητη πλέον η χρήση επιστημονικών μεθόδων και εργαλείων. Τα πρώτα βήματα έγιναν, όπως αναφέραμε και στην εισαγωγή από τον Galileo Galilei καθώς και τον Evangelista Torricelli με την εφεύρεση του πρώτου βαρόμετρου το 1647.

Μετά την μέτρηση των καιρικών συνθηκών παρουσιάστηκε η ανάγκη για την πρόβλεψη των καιρικών φαινομένων που κατέστη δυνατή μέσω της κατανόησης της ατμοσφαιρικής φυσικής. Πιο συγκεκριμένα, το 1922 ο Lewis Fry Richardson δημοσιεύει το βιβλίο του «Weather Prediction By Numerical Process», που ουσιαστικά χρησιμοποιεί μαθηματικά μοντέλα και εξισώσεις για να επιτύχει πρόβλεψη.

### 4.1. Η Δική μας Υλοποίηση

Η δική μας υλοποίηση είναι μια απλοποιημένη μέθοδος που βασίζεται στα ευρήματα των Φινλανδών Rinne, Juhani - Jarmo Koistinen, Elena Saltikoff από τα οποία προέκυψαν 2 πίνακες συσχέτισης κατεύθυνσης ανέμου με μεταβολή ατμοσφαιρικής πίεσης.

Άνοδος Ατμοσφαιρικής Πίεσης		Πτώση Ατμοσφαιρικής Πίεσης	
N	Dry & cold	N	Probability of rain or snow
NE	Cold & cloudy	NE	Probability of rain
E	Cold & cloudy with probability of rain	E	Probability of rain
SE	Dry & hot	SE	High probability of rain or thunderstorms
S	Hot	S	High probability of rain or thunderstorms
SW	Humid & hot	SW	High probability of rain or thunderstorms
W	Humid & hot	W	Probability of rain
NW	Humid & cool	NW	High probability of rain or thunderstorms

Πίνακας 1: Πίνακας αντιστοίχισης κατεύθυνσης ανέμου με πρόβλεψη καιρού, ανάλογα με την τάση της πίεσης

Αρχικά παρατηρούμε την μεταβολή της ατμοσφαιρικής πίεσης ανά συγκεκριμένα χρονικά διαστήματα. Στη συνέχεια προσδιορίζουμε εάν η ατμοσφαιρική πίεση είναι σε πτώση, άνοδο ή παραμένει σταθερή. Τέλος ανάλογα με την κατεύθυνση του ανέμου και την «τάση» της πίεσης καταλήγουμε σε πρόβλεψη. Περαιτέρω ανάλυση στο [κεφάλαιο 6.3](#).

## 5.Περιβάλλον και ανανεώσιμες πηγές ενέργειας

Η ενέργεια αποτελεί την κινητήριο δύναμη στις καθημερινές ανθρώπινες δραστηριότητες, στην οικονομία αλλά και την τεχνολογία. Καθημερινά καλύπτουμε αυτές τις ενεργειακές ανάγκες, σχεδόν αποκλειστικά από συμβατικές πηγές ενέργειας όπως είναι η βενζίνη το πετρέλαιο και ο άνθρακας. Ο ηλεκτρισμός που καταναλώνουμε παράγεται χρησιμοποιώντας τις πηγές αυτές, των οποίων η διαθεσιμότητα και η επάρκεια, πλέον δεν είναι δεδομένες. Επίσης είναι ορατές και οι επιπτώσεις που έχει στο περιβάλλον αυτή η ανεξέλεγκτη κατανάλωση και είναι ξεκάθαρο ότι η κλιματική αλλαγή είναι γεγονός και ο ανθρώπινος παράγοντας έχει παρήμβει σε μεγάλο βαθμό σε αυτή.

### 5.1 Ηλιακή ενέργεια

Με τον όρο ηλιακή ενέργεια χαρακτηρίζεται το σύνολο των διαφόρων μορφών ενέργειας που προέρχονται από τον ήλιο. Αυτές είναι η φωτεινή ενέργεια, η θερμότητα αλλά και οι διάφορες ακτινοβολίες. Η ηλιακή ενέργεια είναι πρακτικά ανεξάντλητη οπότε δεν υπάρχουν περιορισμοί χώρου και χρόνου όσων αφορά την εκμετάλλευσή της. Η εκμετάλλευσή της χωρίζεται σε 3 κατηγορίες εφαρμογών: τα παθητικά ηλιακά συστήματα, τα ενεργητικά καθώς και τα φωτοβολταϊκά συστήματα τα οποία και θα εξετάσουμε στο παρακάτω κεφάλαιο.

### 5.2 Φωτοβολταϊκά ηλιακά συστήματα

Η λειτουργία των φωτοβολταϊκών ηλιακών στηρίζεται στο φωτοβολταϊκό φαινόμενο, δηλαδή την άμεση μετατροπή της ηλεκτρομαγνητικής ακτινοβολίας σε ηλεκτρικό ρεύμα. Μερικά υλικά όπως το πυρίτιο έχουν τη δυνατότητα να δημιουργούν διαφορά δυναμικού όταν φωτίζονται. Συνδέοντας μεταξύ τους πολλά κομμάτια πυριτίου (φωτοβολταϊκές κυψέλες ή στοιχεία) πάνω σε μία επίπεδη επιφάνεια στραμμένη προς τον ήλιο (στην περίπτωση μας υπό γωνία 45°) επιτυγχάνεται παραγωγή ηλεκτρικού ρεύματος. Αυτό το ρεύμα μπορεί να αποθηκευτεί μετά σε ηλεκτρικούς συσσωρευτές (μπαταρίες). Όλη αυτή η διαδικασία επιφέρει το ζωτικής σημασίας αποτέλεσμα να υπάρχει πλήρως ανανεώσιμη, φθηνή και κυρίως «καθαρή» ενέργεια.

### 5.3. Εξοικονόμηση ενέργειας μέσω προγραμματισμού

Η παρούσα πτυχιακή, πέρα από την οικολογική οπτική που έχει λόγω του φωτοβολταϊκού που διαθέτει σε συνδυασμό με την μπαταρία, έχει άλλο ένα χαρακτηριστικό που προσδίδει στο project την ιδιότητα του έξυπνου συστήματος. Έχει προγραμματιστεί έχοντας τη δυνατότητα να πέφτει σε χειμερία νάρκη ανά τακτά προκαθορισμένα χρονικά διαστήματα (15 λεπτά). Πιο συγκεκριμένα από το datasheet του ESP διαπιστώνουμε ότι σε κατάσταση deepSleep όλο το τσιπ είναι ανενεργό εκτός του RTC και η κατανάλωση ρεύματος είναι στα 20  $\mu$ A.

Light-sleep <sup>2)</sup>	-	0.9 mA
Deep-sleep <sup>3)</sup>	Only RTC is working	20 $\mu$ A
Shut down	-	0.5 $\mu$ A

Εικόνα 16: Κατανάλωση καταστάσεων λειτουργίας ESP8266

## 6.Λογισμικό μέρος του συστήματος

Ο πηγαίος κώδικας του software αποτελείται από 2 καταστάσεις λειτουργίας: Sleep και Live. Κατά την εκκίνηση του προγράμματος ανάλογα με την επιλογή που έχει κάνει ο χρήστης από το application καθορίζεται ποιά συνάρτηση θα κληθεί.

```
void loop()
{
  Blynk.syncAll();
  Blynk.run();
  terminal.print ("STATE IS: ");
  terminal.println(state);

  if (state == 1) {
    terminal.println ("-----SLEEP VERSION-----");
    api_key = sleepkey;
    sleepMeasurements();
    terminal.println();
    ESP.deepSleep(15 * 60 * 1000000);

    delay(200);
  } else if (state == 0) {
    terminal.println ("-----LIVE VERSION-----");
    api_key = livekey;
    liveMeasurements(StartTime, p1);
  }
}
```

Εδώ καλείται η συνάρτηση deepSleep που είναι υπεύθυνη για το hibernation (κοίμηση) του συστήματος και παίρνει σαν όρισμα χρόνο σε msec.

```
    delay(200);
  } else if (state == 0) {
    terminal.println ("-----LIVE VERSION-----");
    api_key = livekey;
    liveMeasurements(StartTime, p1);
  }
}
```

Η παρούσα πτυχιακή αναπτύχθηκε με σκοπό η βασική κατάσταση λειτουργίας του συστήματος να είναι το Sleep Version,εξ' ου και η περαιτέρω ανάλυση του, αλλά στην πορεία της ανάπτυξης του project προέκυψαν προβλήματα που αναφέρονται στον κεφάλαιο 9, τα οποία κατέστησαν αναγκαία την ανάπτυξη και του live version.

### 6.1.Η συνάρτηση sleepMeasurements( )

Η συνάρτηση sleepMeasurements είναι τύπου Void και αποτελεί τη ραχοκοκαλιά του κώδικά μας όντας υπεύθυνη για όλες τις μετρήσεις, την πρόβλεψη και την ενημέρωση των IoT υπηρεσιών.

```

void sleepMeasurements(void) {

int windir, i = 0;
String wd;
char buffer[80], trend;

//WEATHER VALUES
temperature = bme.readTemperature();
humidity    = bme.readHumidity();
pressure    = bme.readPressure();
pressure /= 100; //division needed for hPa conversion

```

Χρησιμοποιώντας τις συναρτήσεις readTemperature, readHumidity και readPressure του library Adafruit\_BME280.h παίρνουμε τις αντίστοιχες μετρήσεις και τις αποθηκεύουμε στις ανάλογες μεταβλητές.

```

//BATTERY LEVEL
changeMux(LOW);

for (int i = 0; i < 1000; i++) {
  raw += analogRead(0);
}
raw /= 1000;
int level = map(raw, 335, 415, 0, 100);
if ( level < 0 ) {
  level = 0;
}
if ( level > 100 ) {
  level = 100;
}

```

Εδώ γίνεται η μέτρηση της φόρτισης της μπαταρίας μετρώντας την αναλογική έξοδο η οποία κανονικοποιείται με τη χρήση της map

```

Blynk.virtualWrite(4, level);

```

```

//WRITE WEATHER MEASUREMENTS ON BLYNK APP
Blynk.virtualWrite(0, temperature);
Blynk.virtualWrite(3, humidity);
Blynk.virtualWrite(1, pressure);

```

Στη συνέχεια με τη χρήση της virtualWrite αποστέλλεται η κάθε μέτρηση στο Blynk application

```

//WIND DIRECTION RAW MEASUREMENT
changeMux(HIGH);
windir = analogRead(0);

```

Εδώ διαβάζουμε την αναλογική έξοδο του ποτενσιόμετρου από το ανεμοδούρι και εκχωρούμε την τιμή σε μία μεταβλητή.

```
//DETERMINE WIND DIRECTION
if (windir <= 940 && windir >= 922) {
  wd = "N";
} else if (windir <= 921 && windir >= 905) {
  wd = "NE";
} else if (windir <= 904 && windir >= 870) {
  wd = "E";
} else if (windir <= 869 && windir >= 814) {
  wd = "SE";
} else if (windir <= 813 && windir >= 725) {
  wd = "S";
} else if (windir <= 724 && windir >= 470) {
  wd = "SW";
} else if (windir <= 469 || windir >= 953) {
  wd = "W";
} else if (windir <= 952 && windir >= 941){
  wd = "NW";
}
}
```

Στη συνέχεια με βάση το calibration που περιγράφεται στο [κεφάλαιο 8.3](#), καθορίζεται η κατεύθυνση του ανέμου και εκχωρείται σε μεταβλητή.

```
Blynk.virtualWrite(9, wd);
```

```
//EEPROM ADDRESS
addr = EEPROM.read(0);
EEPROM.put(addr, pressure);
addr++;
```

Διαβάζουμε από την μνήμη flash το περιεχόμενο της θέσης 0 ώστε να πάρουμε τη θέση μνήμης στην οποία καταχωρείται η πίεση.

```
//1 Hour Forecast
if (addr == 6) {
  EEPROM.get(1, p1); //Read data on address 1 and write in p1
  EEPROM.get(4, p2); //Read data on address 4 and write in p2

  //DETERMINE THE PRESSURE TREND
  if ((p1 - p2) >= 5 ) {
    trend = 'f'; //falling
  } else if ((p1 - p2) <= -5) {
    trend = 'r'; //rising
  } else {
    trend = 's'; //stable
  }
  forecast = prediction(wd, trend, 13);
  EEPROM.put(13, forecast);
  EEPROM.commit();
}
```

```

//3 hour Forecast
} else if (addr == 14) {
  EEPROM.get(1, p1);
  EEPROM.get(12, p2);

  //DETERMINE THE PRESSURE TREND
  if ((p1 - p2) >= 5) {
    trend = 'f'; //falling
  } else if ((p1 - p2) <= -5) {
    trend = 'r'; //rising
  } else {
    trend = 's'; //stable
  }
  addr = 1;
  forecast = prediction(wd, trend, 14);
  EEPROM.put(14, forecast);
  EEPROM.commit();
}

```

Εδώ ανάλογα με το αν έχει περάσει 1 ώρα ή 3 καθορίζεται η τάση της ατμοσφαιρικής πίεσης, καλείται η συνάρτηση prediction με τα ανάλογα ορίσματα και τελικά εκχωρείται η πρόβλεψη στην ανάλογη θέση μνήμης.

```

currentTime = String(hour()) + ":" + minute() + ":" + second();
currentDate = String(day()) + "/" + month() + "/" + year();
formattedDNT = currentDate + "," + currentTime;

```

Χρησιμοποιούμε τις συναρτήσεις της βιβλιοθήκης WidgetRTC.h για να καθοριστεί ο χρόνος και η ημερομηνία της πρόβλεψης.

```

Blynk.virtualWrite(8, formattedDNT);
Blynk.virtualWrite(7, forecast);
EEPROM.write(0, addr); //pointer reset
EEPROM.commit();

```

```

// Post data to ThingSpeak
postData(temperature, humidity, pressure, level, raw);

```

Τέλος τα δεδομένα αποστέλλονται στην IoT πλατφόρμα ThingSpeak μέσω της συνάρτησης postData.

```

}

```

## 6.2.Η συνάρτηση liveMeasurements ( )

Η συγκεκριμένη συνάρτηση είναι η 2<sup>η</sup> κατάσταση λειτουργίας του συστήματος και παρουσιάζει πολλές ομοιότητες με την sleepMeasurements γι'αυτό και θα αναφερθούν μόνο οι διαφορές τους:



- Η liveMeasurements δέχεται ορίσματα τη χρονική στιγμή εκκίνησης και τη μέτρηση ατμοσφαιρικής πίεσης εκείνη τη στιγμή.
- Η απώλεια deepSleep μας επιτρέπει να χρησιμοποιήσουμε Interrupts και έτσι μπορούμε να υπολογίσουμε την ταχύτητα του ανέμου κατά το live mode με τη βοήθεια της isr\_rotation αλλά και να μην χρειάζεται να αποθηκεύσουμε δεδομένα στην EEPROM.

```
Rotations = 0; // Set Rotations count to 0
sei(); // Enables interrupts
delay (3000); // Wait 3 seconds to average
cli(); // Disable interrupts
WindSpeed = Rotations * 0.75;
```

Για τον υπολογισμό της ταχύτητας του ανέμου σε km/h χρησιμοποιούμε τον τύπο  $V = P * 0.75$  όπου V η ταχύτητα και P ο αριθμός περιστροφών του ανεμόμετρου.

- Ο καθορισμός του πέρατος 1 ή 3 ωρών γίνεται με τη χρήση της συνάρτησης millis που έχει σαν αποτέλεσμα να μην χρησιμοποιείται καθόλου η μνήμη EEPROM.

```
//1h forecast
if (ElapsedTime >= 3600000) {
    .
    .
    .
    .

//3h forecast
} else if (ElapsedTime >= 10800000) {
    .
    .
    .
    .
}
```

### 6.3.Η συνάρτηση prediction ( )

Η συνάρτηση prediction καλείται από τις συναρτήσεις sleepMeasurements και liveMeasurements και είναι αυτή η οποία καθορίζει την πρόβλεψη. Είναι μια συνάρτηση τύπου String και παίρνει 3 ορίσματα.

Από τον Πίνακα 1 προκύπτουν οι παρακάτω σταθερές τύπου String που αντιπροσωπεύουν την πρόβλεψη.

```

//Forecast constants
String ffc1 = "Probability of rain";
String ffc2 = "Probability of rain/snow";
String ffc3 = "High probability of rain/thunderstorms";
String rfc1 = "Dry & Cool";
String rfc2 = "Cloudy & Cold";
String rfc3 = "Cloudy & Cold with probability of rain";
String rfc4 = "Dry & Hot";
String rfc5 = "Hot";
String rfc6 = "Humid & Hot";
String rfc7 = "Humid & Cool";

```

```
String prediction(String wd, char trend, int j) {
```

Τα ορίσματα είναι κατά σειρά κατεύθυνση ανέμου, τάση ατμοσφαιρικής πίεσης και ένας ακέραιος που αντιπροσωπεύει τη θέση μνήμης που πρέπει να αποθηκευτεί η πρόβλεψη.

```

//MAKE FORECAST
if (trend == 'r') {
    if (wd == "N") {
        forecast = rfc1;
    } else if (wd == "NE") {
        forecast = rfc2;
    } else if (wd == "E") {
        forecast = rfc3;
    } else if (wd == "SE") {
        forecast = rfc4;
    } else if (wd == "S") {
        forecast = rfc5;
    } else if (wd == "SW") {
        forecast = rfc6;
    } else if (wd == "W") {
        forecast = rfc6;
    } else if (wd == "NW") {
        forecast = rfc7;
    }
} else if ( trend == 'f') {
    if (wd == "N") {
        forecast = ffc2;
    } else if (wd == "NE") {
        forecast = ffc1;
    } else if (wd == "E") {
        forecast = ffc1;
    } else if (wd == "SE") {
        forecast = ffc3;
    } else if (wd == "S") {
        forecast = ffc3;
    } else if (wd == "SW") {
        forecast = ffc3;
    }
}

```

```

} else if (wd == "W") {
  forecast = ffc1;
} else if (wd == "NW") {
  forecast = ffc3;
}
} else {
  terminal.println("steady");
  EEPROM.get(j, forecast);
}

```

Στην περίπτωση που έχουμε κατάσταση steady, αυτό πρακτικά σημαίνει ότι έχουμε αμελητέα μεταβολή της ατμοσφαιρικής πίεσης οπότε η πρόβλεψη παραμένει σταθερή.

```

}
return forecast;
}

```

#### 6.4.Η συνάρτηση postData ( )

Η συνάρτηση postData είναι υπεύθυνη για τη μεταφορά δεδομένων στην πλατφόρμα του ThingSpeak. Είναι μια συνάρτηση τύπου void και παίρνει 5 ορίσματα.

```

const char* server = "api.thingspeak.com";
const char* api_key = "API_KEY";

```

```

void postData(float temperature, float humidity, float pressure, int level) {

```

Τα ορίσματα κατά σειρά είναι θερμοκρασία, υγρασία, ατμοσφαιρική πίεση και φόρτιση μπαταρίας.

```

// Send data to ThingSpeak
if (client.connect(server, 80)) {

```

Εδώ γίνεται η σύνδεση του wemos με το web server του ThingSpeak στην πόρτα 80.

```

terminal.println("Connect to ThingSpeak - OK");

```

```

String dataToThingSpeak = "";
dataToThingSpeak += "GET /update?api_key=";
dataToThingSpeak += api_key;

```

```

dataToThingSpeak += "&field1=";
dataToThingSpeak += String(temperature);

```

```

dataToThingSpeak += "&field2=";
dataToThingSpeak += String(humidity);

```

```

dataToThingSpeak += "&field3=";
dataToThingSpeak += String(pressure);

dataToThingSpeak += "&field4=";
dataToThingSpeak += String(level);

dataToThingSpeak += " HTTP/1.1\r\nHost: a.c.d\r\nConnection: close\r\n\r\n";
dataToThingSpeak += "";
client.print(dataToThingSpeak);

int timeout = millis() + 5000;
while (client.available() == 0) {
  if (timeout - millis() < 0) {
    terminal.println("Error: Client Timeout!");
    client.stop();
    return;
  }
}
}
}
}

```

## 6.5.Οι συναρτήσεις BLYNK\_WRITE ( ) και BLYNK\_CONNECTED ( )

Η BLYNK\_WRITE ελέγχει την κατάσταση του κουμπιού επιλογής mode λειτουργίας στο blynk application και εκχωρεί την κατάλληλη τιμή στη μεταβλητή ελέγχου state.

```

BLYNK_WRITE(V2) {
  state = param.asInt(); // assigning incoming value from pin V1 to a variable
}

```

Η BLYNK\_CONNECTED εκτελείται κάθε φορά που γίνεται σύνδεση με το blynk application και είναι υπεύθυνη για την εκκίνηση του widget ρολογιού και την ενημέρωσή του.

```

BLYNK_CONNECTED(){
  rtc.begin();
  if (isFirstConnect)
  {
    // Request server to re-send latest values for all pins
    Blynk.syncAll();
    isFirstConnect = false;
  }
}

```

## 6.6.Η συνάρτηση changeMux ( )

Η changeMux είναι υπεύθυνη για την εναλλαγή ανάμεσα σε μπαταρία και ανεμοδούρι, στην αναλογική είσοδο του wemos.

```
void changeMux(int b) {
  digitalWrite(MUX_B, b);
}
```

Υπό κανονικές συνθήκες η συνάρτηση θα είχε 3 ορίσματα a,b,c που αντιστοιχούν στα option pins του CD4051. Όπως αναφέραμε και στο [κεφάλαιο 2.5](#), τα options a και b είναι γειωμένα, συνεπώς η συνάρτησή μας χρειάζεται ένα μόνο όρισμα.

## 6.7.Η συνάρτηση isr\_rotation ( )

Η συνάρτηση isr\_rotation είναι υπεύθυνη για την καταμέτρηση των περιστροφών του ανεμόμετρου.

```
void isr_rotation () {
  if ((millis() - ContactBounceTime) > 15 ) { // debounce the switch contact.
    Rotations++;
    ContactBounceTime = millis();
  }
}
```

Αποτελεί ένα Interrupt Service Routine και καλείται από την liveMeasurements μέσω της ενεργοποίησης των Interrupt από την συνάρτηση sei().

## 6.8.Η συνάρτηση setup ( )

```
void setup() {

  StartTime = millis();
  EEPROM.begin(512); // EEPROM initialization for 512 bytes
  Wire.begin(); //I2C initialization
  pinMode(A0, INPUT);
  pinMode(MUX_B, OUTPUT);

  //CONNECT TO BLYNK APP
  Blynk.begin(auth, ssid, password);
  Serial.begin(115200);

  //CONNECT TO WIFI
  WiFi.begin(ssid, password);

  //INITIALIZE BME I2C
  if (!bme.begin(BME280_I2C_ADDR)) {
    terminal.println("Could not find a valid BME280 sensor, check wiring!");
    while (1);
  }
  p1 = bme.readPressure() / 100;
```

```
pinMode(WindSensorPin, INPUT);
attachInterrupt(digitalPinToInterrupt(WindSensorPin), isr_rotation, FALLING);

}
```

## 6.9.Βιβλιοθήκες, αντικείμενα και σταθερές

```
#include <math.h>
#include <Wire.h>
#include <BlynkSimpleEsp8266.h>
#include <SPI.h>
#include <Adafruit_BME280.h>
#include <ESP8266WiFi.h>
#include <EEPROM.h>
#include <TimeLib.h>
#include <WidgetRTC.h>

char auth[] = "bbe0e18ed2024c0da5eab3ff97ec8587";
```

Η μεταβλητή auth αποτελεί το διαπιστευτήριο για την εφαρμογή blynk.

```
Adafruit_BME280 bme;
#define BME280_I2C_ADDR 0x76 //BME I2C ADDRESS
#define MUX_B D3
#define WindSensorPin D5

WiFiClient client;

WidgetTerminal terminal(V6); //terminal object declaration
WidgetRTC rtc; //real time clock object declaration
```

## 7.Κατασκευαστικό μέρος του συστήματος

Για το κατασκευαστικό κομμάτι της πτυχιακής μας εργασίας αποφασίσαμε να έχουμε όσο το δυνατόν χαμηλότερο κόστος και να εκμεταλλευτούμε παλιά «μεταχειρισμένα» υλικά.

### 7.1.Η Βάση

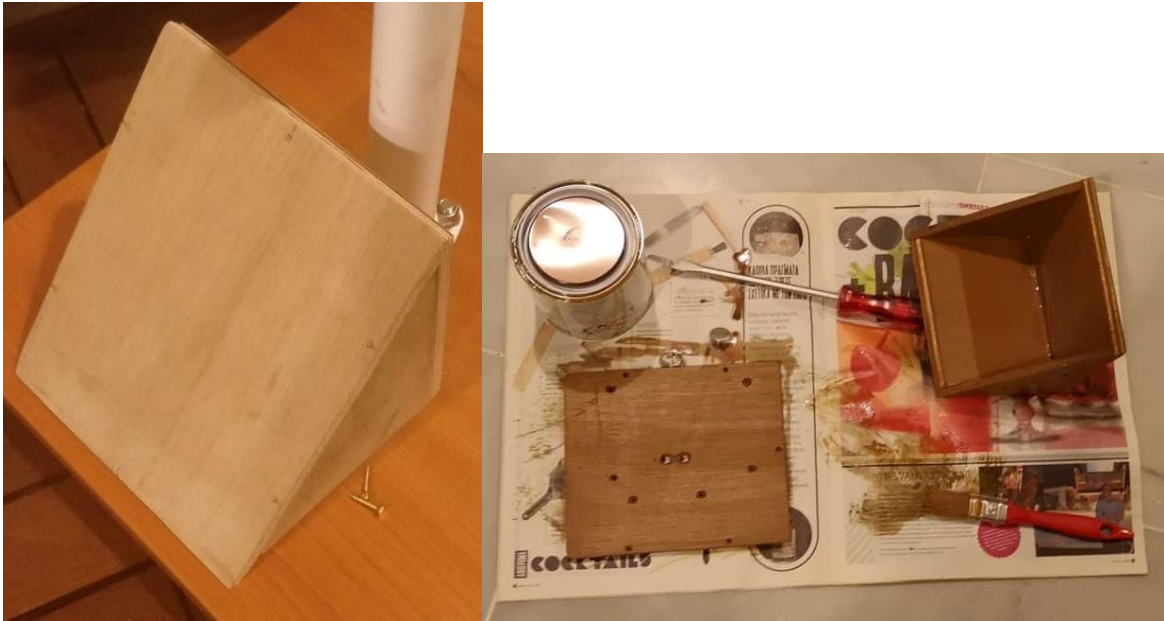
Για τη βάση του συστήματος χρησιμοποιήσαμε μελαμίνη 2,5 εκατοστών λόγω του βάρους του συγκεκριμένου ξύλου μιας και παρέχει την απαραίτητη στήριξη για το σύστημα. Το κομμάτι προήλθε από παλιό γραφείο και κόπηκε σε διαστάσεις 35,4cm x 32,5cm. Πάνω στο ξύλο βιδώσαμε ένα σωλήνα αλουμινίου διαμέτρου 33mm και ύψους 24,5cm, που αποτελεί τη βάση του ανεμόμετρου και προήλθε από σκελετό παλιάς τέντας.



Εικόνα 17: Η βάση με το σωλήνα βιδωμένο

### 7.2.Το κουτί

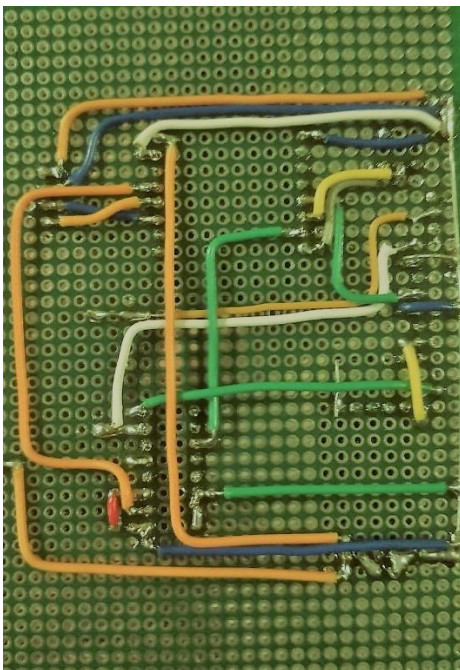
Το κουτί που περιέχει την πλακέτα του συστήματος κατασκευάστηκε εξολοκλήρου απ' το μηδέν από σημίδα κόντρα πλακέ θαλάσσης, πάχους 0,75cm. Επιλέξαμε αυτό το ξύλο λόγω του υψηλού δείκτη ανθεκτικότητάς του στο νερό μιας και είναι ένα από τα πιο διαδεδομένα είδη ξύλου για κατασκευή βαρκών. Το συγκεκριμένο κομμάτι προήλθε από εγκαταλελειμμένο ιστιοπλοϊκό σκάφος.



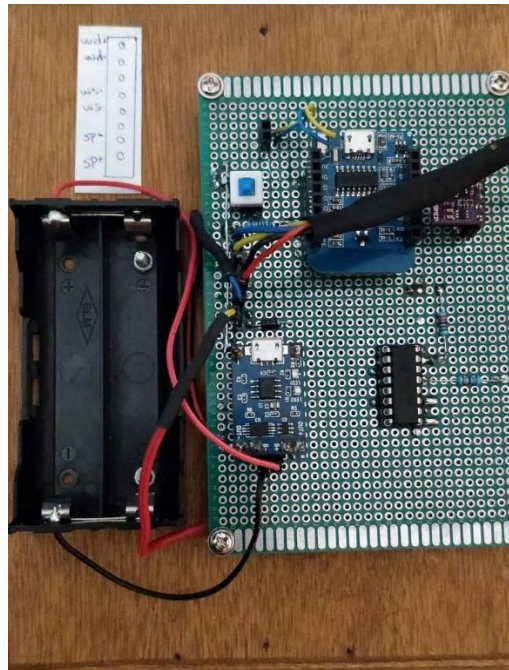
Εικόνα 18: Το κουτί της κατασκευής πριν και μετά το βάψιμο

### 7.3.Η πλακέτα

Η πλακέτα του συστήματος φτιάχτηκε από εμάς, με βάση το κύκλωμα της Εικόνας 9. με την προσθήκη προέκτασης του διακόπτη reset σε εξωτερικό κουμπί, και ενός διακόπτη για το βραχυκύκλωμα της πόρτας reset με την πόρτα D0 του wemos.



Εικόνα 18: Οι κολλήσεις του PCB



Εικόνα 19: Η κάτοψη του ηλεκτρονικού μέρους του συστήματος



## 8. Προβλήματα

Στο παρών κεφάλαιο θα γίνει αναφορά στα προβλήματα που προέκυψαν κατά το σχεδιασμό αλλά και την υλοποίηση του project καθώς και οι τρόποι με τους οποίους αντιμετωπίσαμε καθένα απ' αυτά. Τα προβλήματα αυτά είχαν σαν αποτέλεσμα την αύξηση του κόστους κατασκευής, αλλά και την καθυστέρηση ολόκληρου του project.

### 8.1. Πρόβλημα διατήρησης δεδομένων μετά το deepSleep

Το πρώτο και κυριότερο πρόβλημα που προέκυψε κατά τον σχεδιασμό του έργου, ήταν ο τρόπος με τον οποίο θα διατηρούνταν τα δεδομένα μετά το πέρας του deepSleep, όσο δηλαδή το wemos ήταν ανενεργό. Πρώτη μας σκέψη ήταν η χρήση μιας SD κάρτας για την αποθήκευση αυτών ώστε μόλις το wemos «ξυπνήσει» να ανακτήσει τα δεδομένα που χρειάζεται. Αρχικά το σύστημα «κρεμούσε» όταν συνδυάζαμε το SD shield με τον αισθητήρα BME, που μετά από μελέτη στο διαδίκτυο διαπιστώσαμε ότι συμβαίνει διότι το πρωτόκολλο επικοινωνίας SPI που χρησιμοποιεί το shield δεν συγχρονιζόταν με το πρωτόκολλο I2C που χρησιμοποιεί ο BME με αποτέλεσμα να μην αποθηκεύεται τίποτα τελικά στην SD κάρτα. Αυτό μας ανάγκασε να αγοράσουμε την έκδοση του BME με δυνατότητα επικοινωνίας SPI που όμως τελικά κατέληξε επίσης σε αδιέξοδο μιας και δεν έλυσε το πρόβλημα και το σύστημα συνέχισε να «κρεμάει». Εν τέλει μετά από εξονυχιστική έρευνα στο διαδίκτυο, αισίως ανακαλύψαμε την ύπαρξη της μνήμης flash του μικροελεγκτή η οποία μας έδωσε τη δυνατότητα να αντικαταστήσουμε την κάρτα SD. Αναφορικά η «περιπέτεια» με την SD μας κόστισε 2 καμένους μικροελεγκτές, 1 SD shield και 2 κάρτες SD.

### 8.2. Ο προβληματικός αισθητήρας MH-Z19

Κατά την αρχική σύλληψη του project, υπήρχε η ιδέα να εντάξουμε αισθητήρα ποιότητας αέρα. Η πιο γνωστή οικογένεια αισθητήρων για αυτή τη δουλειά είναι οι αισθητήρες MQ. Λόγω εμπειρίας με τους συγκεκριμένους αισθητήρες από προηγούμενα project, ξέραμε ότι χρειάζονται ένα μεγάλο χρονικό διάστημα προθέρμανσης πράγμα αδύνατο λόγω του deepSleep. Μετά από έρευνα αγοράς στο διαδίκτυο βρήκαμε τον αισθητήρα MH-Z19 της Κινέζικης εταιρίας Winsen που χρειάζεται πολύ μικρότερο χρόνο προθέρμανσης. Ο συγκεκριμένος αισθητήρας είναι 10 φορές ακριβότερος από έναν MQ και παρόλα αυτά έφτασε στα χέρια μας μη λειτουργικός και με 3 μήνες καθυστέρηση. Δυστυχώς, αναγκαστήκαμε να απορρίψουμε την ιδέα.

### 8.3. Έλλειψη εγχειριδίου ανεμόμετρου

Το συγκεκριμένο ανεμόμετρο διασώθηκε από παλιό μετεωρολογικό σταθμό και συνεπώς δεν γνωρίζαμε καθόλου τη διάταξη των καλωδίων και χρειάστηκε η βοήθεια πολύμετρου για να μπορέσουμε να καθορίσουμε ποια καλώδια αντιστοιχούν στο κάθε εξάρτημα. Επίσης χρειάστηκε να κάνουμε την βαθμονόμηση του ποτενσιόμετρου μεταφράζοντας την γωνία που σχηματίζει το ανεμοδούρι με το σκελετό του σε αναλογική μέτρηση στο λογισμικό της εργασίας.



Εικόνα 19: Η διαδικασία βαθμονόμησης του ανεμοδουρίου

## 9. Σύγκριση με σταθμό εμπορίου

Επειδή ο εχθρός του καλού είναι το καλύτερο αποφασίσαμε να συγκρίνουμε τον μετεωρολογικό σταθμό μας με έναν βιομηχανοποιημένο.



*Εικόνα 20: Ο σταθμός μας δίπλα στον εμπορικό*

Τοποθετήσαμε τους 2 σταθμούς στο ίδιο περιβάλλον λειτουργίας για 36 ώρες και παρατηρήσαμε τις διαφορές.



Εικόνα 21: Αντιπαραβολή εμπρόσθιων τμημάτων και μετρήσεων με τον εμπορικό μετεωρολογικό σταθμό

Ομολογουμένως υπάρχουν αποκλίσεις ανάμεσα στα δύο. Πιο συγκεκριμένα το δικό μας σύστημα δείχνει 2 βαθμούς κελσίου περισσότερους, 16% υγρασία λιγότερο και 5 hPa περισσότερο ατμοσφαιρική πίεση. Αυτές οι αποκλίσεις οφείλονται α) στην ποιότητα του αισθητήρα BME συγκριτικά με τον αντίστοιχο του Geodes, β) στο ξύλινο κουτί που περιβάλλει το κύκλωμα και εξιδανικεύει ελαφρώς τις συνθήκες μέσα αλλά και γ) στην έλλειψη calibration του BME στο σωστό επίπεδο θάλασσας. Παρόλα αυτά είναι αξιοσημείωτη η ευστοχία των μετρήσεων του ανέμου αλλά και η πρόβλεψη του καιρού.

## 10.ΒΙΒΛΙΟΓΡΑΦΙΑ

1. <https://www.arduino.cc/en/Reference/> - Το documentation της πλατφόρμας ανάπτυξης Arduino
2. <https://www.mathworks.com/help/thingspeak/getting-started-with-thingspeak.html> - Εγχειρίδιο χρήσης της πλατφόρμας ThingSpeak
3. <http://docs.blynk.cc/> - Το documentation της πλατφόρμας blynk
4. [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf) - Το documentation του esp8266
5. [https://www.cs.cmu.edu/~coke/history\\_long.txt](https://www.cs.cmu.edu/~coke/history_long.txt) - Η πρώτη IoT κατασκευή
6. [https://el.wikipedia.org/wiki/%CE%97%CE%BB%CE%B9%CE%B1%CE%BA%CE%AE\\_%CE%B5%CE%BD%CE%AD%CF%81%CE%B3%CE%B5%CE%B9%CE%B1](https://el.wikipedia.org/wiki/%CE%97%CE%BB%CE%B9%CE%B1%CE%BA%CE%AE_%CE%B5%CE%BD%CE%AD%CF%81%CE%B3%CE%B5%CE%B9%CE%B1) – Ηλιακή Ενέργεια
7. <https://el.wikipedia.org/wiki/%CE%A6%CF%89%CF%84%CE%BF%CE%B2%CE%BF%CE%BB%CF%84%CE%B1%CF%8A%CE%BA%CE%AC> – Φωτοβολταϊκά
8. <https://www.bbc.com/bitesize/guides/z3bbb9q/revision/1> - Η επίδραση του ανθρώπου στην κλιματική αλλαγή
9. Παναγιώτης Παπάζογλου M.Sc.,Ph.D. , Σπύρος Πολυχρόνης Λιώνεις M.Sc., «Ανάπτυξη εφαρμογών με το Arduino», Εκδόσεις Τζιόλα
10. [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer?fbclid=IwAR0YJpEQewkA4Fx39eX9JRdeVRB7\\_ORFhbWiOpgTHrpbAFq92RWyhlTHShs](https://en.wikipedia.org/wiki/Representational_state_transfer?fbclid=IwAR0YJpEQewkA4Fx39eX9JRdeVRB7_ORFhbWiOpgTHrpbAFq92RWyhlTHShs) – RESTful API